ECONOMIC GAIN-AWARE ROUTING PROTOCOLS FOR DEVICE-TO-DEVICE CONTENT DISSEMINATION

By

Faezeh Hajiaghajani Memar

A THESIS

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Electrical Engineering-Doctor of Philosophy

2017

ABSTRACT

ECONOMIC GAIN-AWARE ROUTING PROTOCOLS FOR DEVICE-TO-DEVICE CONTENT DISSEMINATION

By

Faezeh Hajiaghajani Memar

The objective of this dissertation is to investigate Device-to-Device content dissemination protocols for maximizing the economic gain of dissemination for given combinations of commercial and network parameters. We pose this as a gain-aware multicast Delay Tolerant Network (DTN) routing problem in a Social Wireless Network (SWN) setting. Commercial parameters in this framework include the revenue out of selling a product, discount, rebate, content expiry time, and a content-specific redemption function. These parameters are inputs provided by a content generator. Network parameters are decided based on a Social Wireless Network's characteristics such as consumers' interest profiles and their social bindings summarized in their mobility profiles. We explore two solution approaches. First, we investigate stochastic design methods for gain-aware DTN protocol design including prediction-based dissemination mechanisms. We develop a *predictive gain utility* which is used for deciding relative importance of an immediate peer in comparison with potential future peers. We propose two versions of the utility; the first one, Predictive Gain Utility Routing-Individual (PGUR-I) uses node-specific information, and the second one, Predictive Gain Utility Routing-Aggregated (PGUR-A) implements a more scalable version by probabilistically aggregating node interaction information across consumers with similar consumption interests. In the second approach, we explore learning mechanisms including evolutionary learning and reinforcement learning for synthetizing communication protocols. We explore evolutionary learning, specifically, evolving state machines for protocol synthesis. A specific network protocol is coded as a genotype and its resulting state

machine behavior manifests in the form of the corresponding phenotype, leading to specific performance. Using the developed framework, we explore the evolutionary design of Medium Access Control (MAC) protocols including ALOHA, S-ALOHA and Carrier Sensing Multiple Access (CSMA). Finally, we leverage Reinforcement Learning (RL), which enables nodes to gather local information and make efficient routing decisions in runtime.

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisor, Dr. Subir Biswas for his endless support, encouragement and guidance throughout my work. It was through his mentorship that I learned how to be curious, optimistic and enthusiastic, how to think independently, own a research problem and find hope in failure.

I would also like to thank my committee Dr. Charles Ofria, Dr. Jian Ren and Dr. Kalyanmoy Deb for their time and guidance.

My thanks also go to past and current members of our lab: Debasmit Banerjee, William Tomlinson, Stephan Lorenz, Bo Dong, Qiong Huo, Saptarshi Das, Ian Shi, Feng Dezhi, Rui Wang, Salim Muhamed, Henry Griffith and Brandon Harrington for their collaboration, support and for many memorable moments. I am deeply grateful for my East Lansing friends who have been my family throughout these years. I would like to thank my brother, Foad, as his presence in my life has been a motivation for me in ways he may not realize. Last but not least, I would like to give thanks to my parents for making sure that I feel their love and support for the last 1819 days, despite being 6342 miles away.

TABLE OF CONTENTS

LIST OF	TABLES	ix
LIST OF	FIGURES	Х
LIST OF	ALGORITHMS	xiv
1	Introduction	1
1.1	Social Wireless Networks	
1.2	Device-to-Device (D2D) Content Dissemination in SWN	2
1.3	Problem Statement	
1.3.1	Consumption Interest and Coupon Redemption	
1.3.2	Consumer Privacy	
1.3.3	Role of a Content Server	
1.3.4	Incentive Mechanisms	9
1.4	Current Solutions for DTN Routing	
1.5	Dissertation Objectives	
1.6	Scope of Dissertation	
2	Related Work	
2.1	Unicast	
2.1.1	Community-based Routing	
2.1.2	Community-independent Routing	
2.2	Multicast	
2.3	Interest-aware and Incentive-based Mechanisms	
2.4	Communication Protocol Evolution	
2.5	Reinforcement Learning based Approaches	
2.6	Summary	
3	Gain-aware Unicast Dissemination Protocol (GDP)	
3.1	Introduction	
3.2	Network Model	
3.3	Problem Formulation	
3.4	Gain Upperbound	
3.5	Gain-aware Dissemination Protocol	
3.5.1	State Variables	
3.5.2	Updating State Variables	
3.5.3	Forwarding Logic	
3.6	Performance Characterization	
3.6.1	Experiments with Clique-forming Synthetic Mobility	
3.6.1.1	Protocol State Warm up	
3.6.1.2	Steady State Contact Statistics	
3.6.1.3	Protocol Self-tunability	
3.6.1.4	Evolution of Gain and Latency	

3.6.1.5	Delay, Forwarding Cost, and Gain	
3.6.1.6	Impacts of Forwarding Cost	50
3.6.1.7	Impacts of Value Duration	
3.6.1.8	Joint Impacts of Forwarding Cost and Value Duration	
3.6.1.9	Effects of Varying Content Generation Time	
3.6.2	Experiments with Trace Based Mobility	
3.6.2.1	Effects of Different Content Sources	
3.6.2.2	Effects of Varying Content Generation Time	
3.6.3	Gain-positive Delivery Ratio	
3.7	Summary and Conclusion	
4	Economy Driven Multicast Content Dissemination	64
4.1	Introduction	
4.2	Economic Gain of Dissemination	
4.2.1	Cost, Revenue, and Gain	
4.2.2	Consumer Interest and Coupon Redemption	
4.2.3	Consumer Privacy and the Role of Centralized Server	
4.3	Problem Definition.	
4.4	Economy Driven Content Dissemination	
4.5	Compared Protocols	
4.5.1	Interest-Aware Dissemination (IAD)	
4.5.2	Social-aware Opportunistic Routing Protocol (SCORP)	
4.5.3	Random Forwarding	
4.6	Experimental Details.	
4.6.1	Interest and Redemption Function	
4.6.2	Target Exposure Index (TEI)	
4.6.3	Degree of Penetration (<i>DoP</i>)	
4.7	Dissemination Performance	
4.7.1	Social Wireless Network with Clique Formation	
4.7.2	Second Mobility Model	
4.8	Summary and Conclusion	87
5	Multicast Routing using Predictive Gain Utility	
5.1	Introduction	
5.2	System Model	
5.3	Problem Formulation	
5.4	Proposed Protocols	
5.4.1	Predictive Gain Utility Routing -Individual (PGUR-I)	
5.4.2	Predictive Gain Utility Routing -Aggregated (PGUR-A)	
5.5	Compared Algorithms.	
5.6	Dissemination Performance	
5.6.1	Synthetic Mobility	
5.6.1.1	Monolithic Mobility	
5.6.1.2	Community-based Mobility	
5.6.2	Trace-based Mobility	
5.7	Summary and Conclusion	
	-	

6	MAC Protocol Design Using Evolvable State Machines	107
6.1	Introduction	107
6.2	Protocol State Machine	108
6.3	Evolution Platform	111
6.4	Results	113
6.4.1	Partial EDBF (EDBF: 1, 2, 3)	113
6.4.1.1	Evolvable Probability: PWW P = $0 \rightarrow P = 1$ (EDBF: 1)	113
6.4.1.2	Evolvable Probabilities: $PTWT = 1 \rightarrow T = 0$ and PLW (EDBF: 2)	115
6.4.1.3	Evolvable Probabilities: PWL and PLW (EDBF: 2)	118
6.4.1.4	Evolvable Probabilities: $PTLT = 1 \rightarrow T = 1$, $PTWT = 1 \rightarrow T = 1$, and	
	PLW (EDBF: 3)	119
6.4.2	Full EDBF (EDBF: 6)	120
6.4.3	Protocol Adaptation	120
6.4.3.1	Adaptation with Changing Traffic Load	120
6.4.3.2	Adaptation with Changing Network Size	122
6.5	Summary and Conclusion	123
7	MAC Protocol Evolution with Time Sensing and Channel Sensing Modalities	124
7.1	Introduction	124
7.2	Evolving Protocol State Machines	125
7.2.1	Machine with Time Sensing Modality	125
7.2.2	Machine with Channel Sensing Modality	128
7.3	Evolution Architecture	130
7.3.1	Gene Architecture	130
7.3.2	Fitness Formulation	130
7.3.3	Reproduction and Evolution	131
7.4	Results	132
7.4.1	Evolution with Global Time Sensing Enabled	132
7.4.1.1	Limited Evolutionary Degree of Behavioral Freedom (i.e.,:1,2)	132
7.4.1.2	Full Evolutionary Degree of Behavioral Freedom (EDBF: 8)	135
7.4.1.3	Protocol Adaptation	136
7.4.2	Evolution with Channel Sensing Enabled	138
7.4.2.1	Limited Evolutionary Degree of Behavioral Freedom (i.e., 1,2)	138
7.4.2.2	Protocol Adaptation with Maximum EBDF (i.e., 8)	140
7.5	Summary and Conclusion	143
8	Q-learning based Gain-aware Routing	144
8.1	Introduction	144
8.2	System Model	145
8.3	Problem Formulation	146
8.4	Q-learning Model	146
8.5	Q-learning based Gain-aware Routing (QGR)	149
8.6	Compared Protocols	153
8.7	Results	155
8.7.1	Experimental Details	155
8.7.2	Community-based Synthetic Mobility	156

8.7.2.1	Impact of Rebate Cost (cf)	156
8.7.2.2	Impacts of Content Expiry Duration (τ) and the Number of Content Copie	es (T)
		159
8.7.3	Trace-based Mobility	160
8.8	Summary and Conclusion	162
9	Scalable Q-learning based Gain-aware Routing	164
9.1	Introduction	164
9.2	Scalable Q-learning based Routing (SQGR)	165
9.2.1	Gain Quanta	165
9.2.2	Q-learning Model	165
9.3	Scalable Q-learning based Gain-aware Routing	167
9.4	Results	171
9.4.1	Experimental Details	171
9.4.2	Community-based Synthetic Mobility	171
9.4.3	Trace-based Mobility	175
9.4.4	Working Day Mobility	177
9.5	Summary and Conclusion	179
10	Future Work	180
10.1	Introduction	180
10.2	Evolution of DTN Routing Protocols	182
10.2.1	Parameter Evolution	182
10.2.2	Protocol Evolution	183
10.3	Neural Network Driven Q-learning	184
10.4	Gain Estimation using Time Series Prediction	185
10.5	Redemption Functions	186
10.6	Managing Selfishness	187
BIBLIO	GRAPHY	189

LIST OF TABLES

Table 3-1: Notations and corresponding definitions used in GDP	36
Table 4-1: Notations and corresponding definitions in EDCD	71
Table 6-1: Independent transition probabilities	111
Table 7-1: Independent transition probabilities for both state machines	128
Table 7-2: Possible transitions from <i>W</i> to <i>T</i> in the machine in Figure 7-2	130

LIST OF FIGURES

Figure 1-1: Example redemption functions	7
Figure 1-2: Scope of the dissertation	13
Figure 3-1: Example user interaction and DTN routing scenario	
Figure 3-2: RGAP for finding performance upper bound	35
Figure 3-3: Experimental network topology with six social cliques	39
Figure 3-4: Warm up time based on inter-contact time evolution	41
Figure 3-5: Example contact statistics of a C4 node	42
Figure 3-6: Gain comparison between GDP, PROPHET, and RGAP	44
Figure 3-7: Comparative delivery delays and gain evolution	46
Figure 3-8: Delay vs. Forward count with cost: 0.01 and VD: 20K	47
Figure 3-9: Delay vs. Gain with cost: 0.01 and VD: 20K	50
Figure 3-10: Forward count vs. Gain: cost 0.01 and VD 20K	50
Figure 3-11: Delay vs. number of forwards for GDP	51
Figure 3-12: Delay vs. gain for GDP	52
Figure 3-13: Number of forwards vs. gain for GDP	52
Figure 3-14: Delay vs. number of forwards with	53
Figure 3-15: Delay vs. gain with forwarding cost: 0.01	53
Figure 3-16: Number of forwards vs. gain with forwarding cost: 0.01	53
Figure 3-17: Number of forwards vs. gain with forwarding cost: 0.01	55
Figure 3-18: Performance for content generation at 16000s	55
Figure 3-19: Performance for content generation at 17000s	56

Figure 3-20: Warm up time for the INFOCOM '06 mobility trace	57
Figure 3-21: VD vs. Gain for INFOCOM '06 trace; Source node N5;(a): <i>Cost 0.05</i> and (c): Cost 0.14	5, (b): Cost <i>0.1</i>
Figure 3-22: VD vs. Gain for INFOCOM '06; Source node N1; (a): Cost 0.05, (b) (c): Cost 0.14	: Cost 0.1 and
Figure 3-23: Performance for content generation at <i>82000s</i> ; Infocom '06 traces; So Cost <i>0.05</i>	urce node N5; 61
Figure 3-24: Gain-positive Delivery Ratio (GPDR) for a) GDP, b) Epidemic, c) PROPHET	SNW, and d) 62
Figure 4-1: Example redemption function	
Figure 4-2: Example user interaction and routing	
Figure 4-3: Example Gaussian redemption function	75
Fig. 4-4: Network with three social cliques (<i>TEI</i> : 21.5)	76
Fig. 4-5: Multicast dissemination with different Depth of Penetration (<i>DoP</i>)	77
Figure 4-6: Mobility for higher target consumer exposure (<i>TEI: 37.5</i>)	
Fig. 4-7: Gain in scenarios of high <i>TEI</i>	
Figure 4-8: Total rebate (forwarding cost) per destination	
Figure 4-9: Expected gain from one-hop deliveries	
Figure 4-10: Five clique network: (a): <i>TEI:27.8</i> , (b) <i>TEI:32.1</i>	
Figure 4-11: Expected gain distribution from one-hop deliveries	
Figure 4-12: Maximum gain and the time to reach maximum	
Figure 5-1: Example user interaction and routing	
Figure 5-2: Example Gaussian redemption function	
Figure 5-3: Evolution of gain for different expiry time; $T=10$	
Figure 5-4: Dissemination behavior of PGUR protocols	102

Figure 5-5: Final gain for all combinations of T and τ for CMCI mobility scenario 103
Figure 5-6: Gain dynamics for different expiry time; $T=15$
Figure 6-1: (a) Two-state machine for Pure-ALOHA MAC; (b) Network-wide load G vs. Throughput S (in Erlang) for pure-ALOHA phenotype
Figure 6-2: An evolvable generalized FSM with embedded pure-ALOHA logic 109
Figure 6-3: System structure for the proposed evolutionary framework
Figure 6-4: (a) Evolution of <i>PWW</i> $P = 0 \rightarrow P = 1$ and (b) the corresponding throughput 114
Figure 6-5: Trajectory of effective load <i>G</i> for different applied load <i>G</i>
Figure 6-6: Evolution of pair $PTWT = 1 \rightarrow T = 0$ and PWL , and the corresponding throughput 117
Figure 6-7: Evolution of pair PWLand PLW for different network-wide loads 118
Figure 6-8: Evolution of $PTLT = 1 \rightarrow T = 1$, $PTWT = 1 \rightarrow T = 1$ and PLW for different load conditions
Figure 6-9: Evolution of all six independent transition probabilities for full EDBF 121
Figure 6-10: Adaptation of probabilities with runtime changes of network load 122
Figure 6-11: Adaptation of probabilities with runtime changes of network size
Figure 7-1: An evolvable generalized FSM with time sensing capability
Figure 7-2: An evolvable generalized FSM with channel sensing capability 129
Figure 7-3: Genotype formation and the system components
Figure 7-4: Fitness dynamics before and after time sensing is added
Figure 7-5: Genotype dynamics before and after time sensing is added
Figure 7-6: Evolution of all probabilities in different loading conditions
Figure 7-7: Throughput with runtime parameter changes
Figure 7-8: Adaptation of probabilities with runtime parameter changes
Figure 7-9: Genotype dynamics before/after channel sensing is added

Figure 7-10: Throughput with runtime parameter changes
Figure 7-11: Adaptation of probabilities with runtime changes
Figure 8-1: Abstraction of states and actions in the Q-learning based routing
Figure 8-2: Community-based synthetic mobility 156
Figure 8-3: Evolution of gain for forwarding cost cf: (a) 0.002, (b) 0.02, (c) 0.1 and (d) 0.2 158
Figure 8-4:Evolution of Q-values
Figure 8-5: Impacts of content expiry duration (τ) and number of copies (T) on gain 160
Figure 8-6: Dynamics of a) gain and b) time to reach maximum gain 162
Figure 9-1: Probabilistic state machine modeling for Q-Learning
Figure 9-2: Community-based mobility 172
Figure 9-3: Evolution of gain over time for expiry duration τ : (a) half a day, (b) 1.5, (c) 3 and (d) 5.5. days
Figure 9-4: Final gain values for (a) T=15 and (b) T=25 174
Figure 9-5: Q-tables' content at each node for (a) Qii \rightarrow 0 and (b) Qii \rightarrow 1.2 175
Figure 9-6: Maximum gain and time to reach that for expiry duration: (a) half a day and (b) 2.5 days
Figure 9-7: Q-table size per node
Figure 9-8: Gain evolution for working day mobility with $\tau = 4$ days
Figure 10-1: Example general state machine consisting of a content dissemination protocol 184
Figure 10-2: Time series of gain observed by a node- <i>i</i>

LIST OF ALGORITHMS

Algorithm 3-1: The key components of GDP protocol	
Algorithm 4-1: The key components of the EDCD algorithm	73
Algorithm 5-1: The key components of the PGUR-I algorithm	94
Algorithm 5-2: The key components of the PGUR-A algorithm	96
Algorithm 8-1: Key components of the QGR algorithm	
Algorithm 9-1: Summary of the SQGR algorithm	

1 Introduction

1.1 Social Wireless Networks

Similarity breeds connection [1]. This principle (i.e., *homophily*) structures social ties of every type, including friendship, collegiality, co-membership, and other types of relationships. Such pairwise similarities are the building blocks of the concept of Social Wireless Networks (SWNs). SWNs are formed by ad hoc wireless links among data-enabled wireless devices when people carrying such devices regularly gather in certain places. SWNs are classified under Delay Tolerant Networks (DTNs) in the sense that they lack continuous network connectivity. According to the *homophily* principle, human mobility is partially shaped by their social relationships [2]. Reciprocally, it is shown [3] that the similarity between movement trajectories of two persons can be a strong indication of their tie in a physical social network. This binding between social interactions and mobility similarities is reflected in SWNs. Examples include the SWN formed by students in an academic department within a university campus. In a smaller scale, students residing in a specific department and having a common interest in a specific type of food, form an SWN. Other SWN examples include networks formed in work places, malls, airports, train stations and other public places.

Interaction patterns in an SWN can be indexed with social metrics such as pairwise contact duration, pairwise contact frequency, betweenness centrality [4], and degree centrality [4]. These metrics summarize the mobility profile of a person in relation to other entities in the network. This additional information layer about mobility and social interactions of nodes in a network leads to the emergence of a new field of Socially Aware Networking (SAN) [5]. By exploiting social properties of nodes, the abstraction of SAN can be used for better networking support to innovative

applications and services, such as Device-to-Device content dissemination in Social Wireless Networks.

1.2 Device-to-Device (D2D) Content Dissemination in SWN

Due to human mobility and the lack of complete coverage by WiFi access points, an SWN can be susceptible to intermittent disconnections to the Internet. This can result in partitions of devices that can communicate with each other using ad hoc routing protocols, but do not have Internet connectivity. This lack of connectivity may affect server-based content dissemination inside an SWN. Intermittent Internet connectivity is one of the factors which necessitates device-to-device content dissemination approaches in such SWNs. Device-to-Device (D2D) content dissemination can be implemented through any of most recent technologies including Bluetooth, Bluetooth Low Energy (BLE), Bluetooth Service Discovery Protocol.

While there are many applications of multicast D2D dissemination in healthcare [6], emergency handling [7], and vehicular networks [8], in this dissertation we specifically focus on a commercial content dissemination application. It is about delivering commercial content such as coupons to a target population of customers. Consider an example scenario in which an ethnic restaurant in a mall sends out an electronic coupon (through Bluetooth) to the mobile device of a customer who is known to be interested in that ethnic food. The customer, who happens to be a University student, forwards the coupon to other students in her social DTN communities in the University campus via mobile-to-mobile links. It is desired that the coupon gets disseminated to students who are most likely to redeem them so that the resulting commercial gain from the dissemination is maximized.

Even in the presence of Internet connectivity, device-to-device content dissemination can be of interest for the following reasons:

1) <u>Scalability and privacy</u>: reliance on a-priori knowledge about specific destination node identifiers is not feasible in practical coupon dissemination applications in which the coupons need to be delivered to nodes based on their dynamic interest levels and redemption chances instead of their static identities. Therefore, in a server-based approach, a server needs to keep track of every node's interest level toward every product/subject. This, not only exposes users' preferences to an external server, but also imposes a heavy control load on the server. In a D2D approach, however, nodes share their preferences (interest profile) only with certain peers upon an interaction, and not with a centralized server, thus, partially protecting privacy and obtaining scalability.

2) <u>Targeted dissemination</u>: D2D dissemination can promote coupon distribution within the geographical proximity of the coupon generator. Consider the earlier mentioned example scenario in which an ethnic restaurant in a mall sends out a coupon to the mobile device of a customer who is known to be interested in that ethnic food. The customer, who happens to be a student in a University in proximity of the mall, forwards the coupon to other students in her social communities in the University campus via mobile-to-mobile links. Since the coupon holder's peers are in the same general geographical area, they can similarly access the restaurant (coupon generator). That is, the D2D dissemination takes care of delivering to users which are in spatial proximity of the mall, and thus, may be more likely to redeem coupons. Also, this dissemination happens without any exposure of nodes' geographical locations to a centralized content server.

There are a few other common characteristics that the coupon holder may share with her/his student peers, other than the characteristic of commuting in each others' proximity. For example, most of such peer are likely to belong to the same social and financial class. As a result, it is also more likely that the coupon holders' peers share similar interest levels. In a centralized approach,

the server can not easily identify such detailed preferences for large consumer population, thus leading to scalability concerns.

3) <u>Viral marketing</u>: D2D content dissemination implements viral marketing by leveraging inter-personal interaction information to direct the coupons to most fit consumers. Viral marketing is essentially a consumer-to-consumer process and can not be implemented in a centralized manner. It leverages the power of individuals to influence peers through various online means including social networks, emails, instant messaging, and online chat. Electronic viral marketing facilitates asynchronous round-the-clock connectivity with customers, and also considerably expands the scope and scale of influence compared to face-to-face contacts [9]. Depending on an application's requirements regarding privacy preservation, a coupon forwarder may or may not expose his/her identity. In either case, D2D dissemination is still more effective [10] than server-based advertising methods because 1) people trust their acquaintances more than ads coming from a coupon provider, and 2) a user receives targeted coupons based on his/her interests which are most probably similar to his/her acquaintances' interests [1] and choices.

The main design goal for a D2D commercial content dissemination algorithm is about maximizing a predefined commercial gain for a Content Provider (CP). In other words, a carrier of content/coupon identifies most fit individual Content Consumers (CCs), based on its pattern of social interaction (e.g., contact duration, centrality, etc.) with target consumers. In the following section, the requirements and design challenges of D2D commercial content dissemination approaches are discussed in detail.

1.3 Problem Statement

In traditional server-based approaches [11], coupons are sent to consumers directly by Content Providers (CPs). The economic value of a disseminated coupon is determined based on statistically estimated redemption rates and the known costs of coupon generation and distribution. The overall goal is to maximize the content provider's commercial gain, which is defined as the generated revenue out of potential redemptions minus all involved costs of preparation and distribution.

In D2D distribution, the primary cost comes from incentivizing mobile nodes for taking part in D2D dissemination process. D2D coupon forwarding costs bandwidth, storage, CPU and, more importantly, battery power which can deter people from taking part in dissemination. Thus, the objective here is to develop coupon dissemination mechanisms that can maximize a coupon's economic gain in the presence of costs/rebates for D2D forwarding incentives. This can be posed as a Delay Tolerant Network (DTN) multicast routing problem. Most DTN multicast routing approaches in literature attempt to individually optimize performance indices such as message delay, forwarding cost, and storage. None define composite cost factors by combining multiple such indices into a single *gain* index. Due to the lack of gain-*awareness*, these protocols cannot control the economic gain of dissemination, as defined in this context.

1.3.1 Consumption Interest and Coupon Redemption

<u>Consumer Interest</u>: Consumers' interest level towards an item (e.g., product, brand, etc) changes across time and across individuals. Such interest level is mostly dynamic. For example, a consumer may develop interest for a brand after multiple satisfactory purchases from that brand. In another example, a consumer may lose his/her interest in a product after reading negative reviews from other purchasers who used the product (word-of-mouth effect [12]). Modeling and embedding such interest level into the D2D dissemination is necessary in order to have an elaborate system model. Interest level for a consumer can be set through many possible means including: 1) an explicit input by the consumer herself, 2) computed based on the number of purchases made

for a product through the consumer's mobile application, or 3) locally extracted from the metadata collected from searching and browsing activities on the consumer's device.

<u>Redemption Probability</u>: Redemption probability for a consumer for a coupon, represents her/his likelihood of redeeming a coupon within a defined coupon expiry time. The determination of redemption probabilities can be helpful when planning and monitoring coupon campaigns. Redemption probabilities for a specific type of coupon (i.e., the corresponding product or service) are usually predicted based on past redemption data and actual data from an ongoing coupon campaign [13].

Modeling redemption probabilities in a D2D dissemination framework is another step toward simulating a real coupon campaign. For that, a redemption function can be presented which maps a user's interest level for a specific coupon to a redemption probability.

While interest level is a characteristic of a consumer, redemption probability is a function that depends on how consumers react to a coupon based on their interests in the product or service that the coupon represents. If a coupon is originally designed with the goal of motivating high-interest/loyal consumers, then the redemption function is likely to map higher redemption probabilities to consumers with higher interest levels. This can be modeled as an exponential redemption function with a maximum redemption probability (e.g., the shown in Figure 1-1). Interest level in Figure 1-1 is shown as ρ_i and redemption probability is indicated with $P_r(\rho_i)$, which is a function of the interest level.



Figure 1-1: Example redemption functions

Another example, a Gaussian redemption function with $\mu = 0.75$ is shown in Figure 1-1. This function corresponds to a coupon campaign designed to motivate customers with medium interest level. Such model can be justified based on findings [14] which imply that coupon delivery to highly loyal (interested) customers may kill customer loyalty. That is because sending coupons to customers who already buy specific products (possibly at regular price) communicates that the product they are buying has a lower value than what they pay normally. In that case, the sentiment of exclusivity and lowering the bridge are lost, because they are already customers [14]. Thus, light to moderately loyal customers, as opposed to most loyal ones with higher interest levels, may maintain higher redemption rates.

To mention another example, customers based on the stage of their buying cycles can be categorized in different classes, namely, active, at risk, and lost [15]. These customers' interest levels fall into different ranges of high, moderate, and low. The definitions and boundaries of such ranges of interest levels can be maintained by the CP's server. From a marketing point of view, each class of customers need to be treated with different strategies, thus different coupon campaign models are designed. For example, a win-back coupon [15] is designed to target at risk customers, who maintain moderate interest levels rather than being highly or marginally interested.

The interest level for a coupon type for a user is reflected in his/her *interest profile*. Redemption probability is then determined based on the interest level and specific coupon type. The combination of the mobility (Section 1.1) and interest profiles of users in an SWN, when shared within peers, provides information based on which D2D coupon dissemination protocols can be designed. This information is not made accessible to a server (in a server-based dissemination approach) due to privacy concerns explained in the next section.

1.3.2 Consumer Privacy

From a privacy standpoint, both interest (Section 1.3.1) and mobility (Section 1.1) profiles of a consumer are preferred to be stored locally within the mobile device, and should not be exposed to a remote server. Under this privacy model, consumer devices are allowed to share their interest profiles with other devices selectively upon meeting, but are not allowed to share with a centralized entity. The latter provides partial user privacy. Towards the end of this dissertation document (in Chapter 9), we propose a dissemination mechanism which does not require nodes to store peer specific interest and mobility profile on their devices. This information is still shared upon pairwise meetings for a one-time use for required state variable computations on peer's device, however, this information is deleted afterwards and is not stored on peer's device. This is the highest level of privacy considered across methods proposed throughout this dissertation.

1.3.3 Role of a Content Server

While the dissemination process can be done in a fully Device-to-Device manner in order to preserve consumers' privacy (Section 1.3.2), a centralized server may be present with the following roles: 1) consumer authentication, 2) maximum dissemination enforcement, and 3) other long term control and management tasks. This is of course feasible if user devices can reach the

server through vertical links (i.e., 3G/4G) for such control and management operations. Note that even when a vertical link to a server is available, D2D dissemination is desirable for the reasons outlined in Section 1.2.

For horizontal (i.e., Device-to-Device) links between devices, different communication technologies such as Bluetooth, WiFi-direct, and Bluetooth SDP can be adapted.

1.3.4 Incentive Mechanisms

As mentioned earlier, D2D coupon forwarding costs bandwidth, storage, CPU and, more importantly, battery power which can deter people from taking part in dissemination. Incentive mechanisms are needed to encourage users to participate in sharing resources. Here, few basic incentive mechanisms are discussed:

1) Uniform dispensing [16]: Considering a total reward of R_c , every node in the forwarding chain (with size *l*) receives equal amount of reward R_c/l . Forwarding chain refers to the intermediate nodes who participate in forwarding a copy of a content from a source to a destination. This approach leads to an undesirable situation in which users may refuse to forward a coupon further in order to keep the length of the forwarding chain smaller, thus, maximize their own reward.

2) Fixed and equal dispensing [16] :in this method, every node on the forwarding chain receives a similar amount of reward irrespective of the chain's length.

3) Geometric dispensing: intuitively, users' contribution along a forwarding chain is uneven. Geometric dispensing is based on the assumption that the closer a user is to the initial coupon holder in the chain, the greater her contribution is. With Geometric dispensing, the reward distribution is a geometric series with a constant ratio p. Uniform dispensing is a special case of geometric dispensing, when p=1. Also, p=0 results in a single-level dispensing mechanism in which the initial coupon holder receives reward for every forwarding, but the rest of the nodes in the chain do not receive any reward if they forward further. Few other incentive mechanisms that are proposed in literature, which will be discussed in the next chapter (Related Work).

Along with every incentive mechanism, security concerns need to be addressed. One of the possible threats is when a node either from inside or outside of a forwarding chain hacks into the incentive mechanism and manipulates the rewards for itself and other nodes to collect higher reward. To avoid such issues, different encryption methods are applied [16, 17]. Furthermore, an incentive mechanism may require the trajectory of a coupon to be stored and exposed to the coupon provider for redeeming rewards eventually. That itself discloses users' social interactions and trajectories as it shows the social interactions which took place in the process of delivering a coupon to one or multiple receivers.

1.4 Current Solutions for DTN Routing

The problem of D2D coupon dissemination in SWNs can be posed as a multicast Delay Tolerant Routing problem. There exists a rich body of literature on both Unicast and Multicast DTN routing solutions. Among these methods, only those which use a specific type of multicast, namely, *dissemination*, are fit for targeted coupon distribution problem in this dissertation. Dissemination, does not rely on a-priori knowledge about specific destination node identifiers, which is not always available in practical coupon dissemination applications in which the coupons are delivered to nodes based on their dynamic interest levels and redemption chances instead of static identities.

Under the broad class of D2D dissemination algorithms, there are examples which apply social-aware dissemination [18], embed incentive mechanisms in their proposed framework (e.g., [19]), and/or rely on utility functions [20] designed towards performance indices such as delay and

forwarding cost. Social-aware dissemination approaches leverage the information regarding existing social ties among users in an SWN to identify and disseminate a content to receivers interested in the content. Incentive-based methods are suited for D2D dissemination framework in terms of providing incentive methods to encourage nodes' participation in dissemination process. Utility-aware approaches apply a defined utility as the routing criterion. This is similar to the general approach in our research here where we use a defined routing utility in order to maximize the eventual economic gain.

However, most existing multicast approaches are not applicable to marketing applications, since they do not: 1) explicitly define and employ a composite economic gain for the content generator, and 2) incorporate a model considering interest profile of nodes toward a content/product. The approaches in this dissertation attempt to address both the problems.

1.5 Dissertation Objectives

The objective is to design a D2D dissemination algorithm which for a given combination of commercial and network parameters, maximizes the total economic gain within the lowest possible dissemination delay. Commercial parameters include the revenue out of selling a product, discount cost, rebate cost, number of available coupons, coupon expiry time, and a redemption function. These parameters are inputs provided by a coupon generator. Network parameters are given based on the SWN's characteristics such as consumers' interest profiles and their social bindings demonstrated in their mobility profiles. This work formulates the gain of dissemination in perspective of a commercial coupon generator and proposes few Unicast and Multicast gain-aware routing algorithms which maximize a defined economic gain.

We explore two solution approaches. First, we use traditional protocol design methods for gain-aware DTN protocol design. In the second approach, we explore learning mechanisms for

11

communication protocol design with a focus on routing layer protocol design, specifically, gainaware DTN protocols for coupon dissemination.

1.6 Scope of Dissertation

Chapter 2 is a survey of Unicast and Multicast DTN routing strategies. We explain why such approaches are not practical in target marketing applications. Furthermore, we highlight the differences and advantages of our proposed gain-aware dissemination framework compared to most recent routing schemes in the literature.

In Chapter 3, we propose a Gain-aware Unicast Dissemination Protocol (GDP) which routes a content to a receiver with the goal of maximizing the defined economic gain for the content provider. Here, gain of dissemination is defined as a combination of a time-based value function and cost of delivery.

In Chapter 4, we formulate the gain in a more practical elaborate manner. We define the value of a coupon delivery as a function of redemption probability of the coupon receiver, rather than time of the delivery. Then, we define gain as a combination of the defined value and cost of dissemination. We then propose an Economy Driven Content Dissemination (EDCD) routing protocol that controls D2D content dissemination with an end goal of maximizing the defined economic gain for a coupon-generator.

In chapter 5, using the same gain formulation as in chapter 4, we introduce a Predictive Gain Utility Routing framework consisting of two routing protocols. Using gain prediction, this framework provides more structured and scalable routing approaches compared to EDCD (chapter 4). In chapter 6 and chapter 7, we present the proposed framework of evolutionary-based learning for protocol synthesis using probabilistic evolving state machines. We apply this framework to MAC layer protocols.

In chapter 8 and chapter 9, we propose reinforcement learning, specifically Q-learning based gain-aware dissemination mechanisms for D2D coupon dissemination. The proposed online learning framework is expected to be robust in diverse and dynamic mobility environments.

In chapter 10, we present the summary and future work with a discussion on how the presented mechanisms in this dissertation may open new directions in the area of commercial content dissemination in SWNs. Figure 1-2 demonstrates a pictorial representation of the presented work.



Figure 1-2: Scope of the dissertation

2 Related Work

This section reviews existing DTN routing protocols that use social interactions information for route computation in Social Wireless Networks. Several Unicast and Multicast approaches are first reviewed. Then, few approaches which are specifically fit for Device-to-Device (D2D) coupon dissemination applications are presented. These approaches embed concepts such as consumer interest, redemption probability, and forwarding incentive mechanisms. Finally, a class of existing work which use learning based techniques (evolutionary and reinforcement-based) to develop network routing protocols are reviewed.

2.1 Unicast

Unicast social-aware routing approaches are more extensively investigated in the literature compared to Multicast approaches. Few Unicast mechanisms can be extended to multicast by running the unicast logic for multiple destinations simultaneously. Those are referred to as unicast-based multicast routing [21].

2.1.1 Community-based Routing

Community-based routing algorithms rely on extracting information regarding nodes' affiliation to communities. These methods can be classified [22] into two categories, namely, *self-reported*, and *detected*. *Self-reported* social bindings are not always practical, or even accurate due to the dynamic nature of users' interests and affiliation. In *detected routing*, social ties are inferred runtime based on nodes' interaction statistics. Many of them attempt to extract community structures prior to routing. LABEL [23] uses nodes' pairwise inter-contact time values to detect co-memberships to a community. Then, each node is labeled with its affiliation to a community.

A packet is forwarded to a destination directly or it is forwarded to a node which has a label similar to the destination's label.

Bubblerap [24] detects inter-node social relationship such as existing community structures or level of centrality of each node. Node centrality is a measurement of a node's topological importance, which is used to identify the key nodes that can bridge between different network segments and partitions [25]. Following a hierarchical pattern, a message is first forwarded toward the community in which the destination resides, and consequently, using intra-community interactions it is delivered to the destination.

[26] proposes a friendship based routing in which each node forms a temporal friendship community. Friendship with a peer is estimated based on the frequency, longevity, and regularity of pairwise meetings. [26] demonstrates that the delivery decisions based on such communities can improve both delivery ratio and delivery cost. Diverse Routing (DR) [27] is a tunable protocol designed to cope with nodes deviating from their habitual activities. Habitual activities, reflected in community movement, helps in reducing average communication delay and improves routing efficiency. When nodes deviate from their habitual activities, affecting the routing performance, DR ensures that at least one of the deviant nodes can receive a copy of the packet. This is done by DR's runtime way of social clustering and scattering copies of the packet into these clusters. DR supports Multicast as well as Unicast.

All mentioned community-based routing methods rely on the *homophily* [1] principle which implies that nodes with similar interests, belonging to the same community, have higher chances of meeting each other. Therefore, the main effort in this class of methods is to detect communities and route the packet to the destination through nodes which belong to the same community as that

of the destination. Dynamic topology of SWNs and frequent changes in nodes' affiliation may affect the performance of community detection and the subsequent routing process.

2.1.2 Community-independent Routing

Community-independent routing approaches build routing algorithms based on *utilities* without relying on information about nodes' affiliation to any communities. In social-aware routing mechanisms, *utility* is computed based on nodes' inter-personal relationships. Content carriers with higher utility are more fit to perform the content dissemination task.

SimBetTS [28], for example, defines utility as a weighted combination of three social utilities including similarity, betweenness and tie-strength. Two nodes' similarity level is a function of the number of common peers they are expected to meet in future. Betweenness for a node is calculated based on the number of nodes in the network that are indirectly connected through that node. Tie-strength between two nodes is calculated based on their meeting frequency, closeness, and freshness of meetings.

PROPHET family of protocols [29], use history of inter-node encounters and transitivity for computing a *delivery predictability* metric based on which the forwarding decisions are made. *Delivery predictability* of a node captures the frequency with which the node meets the destination directly or transitively through other nodes. When a node carrying a content meets another node, the content is forwarded to the latter only if it has higher delivery predictability (to the destination) compared to the node that is currently holding the content. *PROPHET* implicitly attempts to balance between delay and forward-count in a best-effort manner. However, according to [30], protocols such as *PROPHET* often face a fundamental challenge in choosing the right time scale for delivery predictability estimation which is consistent with the underlying mobility. This causes certain performance loss. Authors in [30] derive mechanisms to dynamically determine the time

scale of *PROPHET*'s delivery predictability based on mobility, and show that when such time scale matches with mobility, the routing performance indeed improves.

MaxProp [31] uses prioritized packet transmissions and packet drops (when buffers are low on storage) based on information collected via system-wide acknowledgements about path failure likelihood and hop-lists, which show previous intermediate recipients. Fresh packets, which are packets with hop count less than a threshold value, are given transmission priority. Extra copies of delivered packets are removed from involved nodes' buffers as an acknowledgment is broadcast in the network.

Both MaxProp and PROPHET use human mobility patterns to predict the best forwarders for accomplishing a higher delivery ratio and lower forwarding cost.

Encounter Based Routing (EBR) [32] aims to maximize delivery ratio while minimizing the number of replicas and delivery delay. The basic idea is that nodes that frequently meet other nodes are better carriers than those who infrequently meet other nodes. Each node maintains an *Encounter Value (EV)* which represents the node's past rate of encounters with other nodes. *EV* is formulated as an exponentially moving weighted average. When two nodes meet, the number of replicas of the content that are exchanged is proportional to the *Evs* of the nodes. The authors in [32] show that this mechanism amounts to a quota-based system in which the total number of copies of a content does not depend on the number of network nodes; instead it depends on a preset quota. It also shows that the encounter based forwarding implicitly minimizes both delivery delay and number of copies of a message.

The protocol *Selectively Making pRogress Towards delivery (SMART)* [33] defines *companions of the destination,* which are nodes that frequently meet the destination. In the spray

phase of *SMART*, a node forwards the packet only to the companions of destination. In its second phase, companions of destination forward the node to other companions until the packet is delivered to the destination. *SMART* exploits nodes' interaction information when forwarding messages. Thus it does not assume a node knows other nodes' mobility patterns. The protocol attempts to minimize both delay and forwarding cost by forwarding packets to companions of the destination. None of the protocols, EBR and SMART employ any explicit mechanism for gain maximization as this thesis sets out to accomplish.

Another category of routing protocols assume presence of *regularity* in human mobility. For example, Nazir et al., [34] assumes that people follow similar mobility patterns daily (i.e., Monday to Friday). It is designed specifically to reduce end-to-end delay and increase delivery performance for time-critical applications.

Community-independent methods are light-weight to implement compared to communitybased methods since they do not require a community detection scheme to run in parallel to the routing framework. However, a warmup period is required for the nodes to build their mobility profiles including the information regarding their social ties.

2.2 Multicast

There is less coverage on social-aware multicast DTN routing in the literature [22]. In [21], relay selection for multicast is formulated as a unified knapsack problem by extracting node centrality and community features, based on which it builds a social weight metric. Each node maintains a path to each multicast destination with the maximum social weight. This approach reduces the number of used relays, thus addresses the forwarding cost issue.

Ning and et. al. [35] propose an incentive based forwarding scheme which rewards the last hop relay node. Every node computes its Effective Interest Contact Probability (ECIP) for each data category, which represents the probability that this node contacted a node interested in the corresponding data category either directly or indirectly. When two nodes meet, they exchange data messages based on computed ECIPs such that they maximize their own expected credit. This approach is restricted in the sense that a node without any messages is unable to receive any data item from other nodes it encountered, unless the data item is of its interest.

The publish/subscribe Multi-Receiver Incentive Based Dissemination (MuRIS) [36] develops a hop-count based multicast utility to achieve minimum-hop multicast dissemination. During the warmup period or after each idle period, a publisher broadcasts probe messages labeled as a particular data category that this publisher will publish. Every subscriber, upon receiving this probe message, sends a receipt message containing the path information carried by the corresponding probe message to all relay nodes on the recorded path.

Many existing multicast schemes (e.g., MuRIS [36]) face scalability issues as packets need to contain the address of every node on the path from a publisher to a subscriber.

A category of flooding-based methods are designed based on Epidemic routing [37]. One such example, [38], analytically determines how a service provider can allocate its bandwidth optimally to make the content at users as "fresh" as possible. It defines an optimization problem around a global fairness objective (namely, maximizing the aggregate utility over all users) and solves it applying a gradient descent. [38] also specifies a condition under which the system is highly scalable.

Other than the scalability issue mentioned for Multicast approaches, most Unicast and Multicast approaches are not fit to coupon dissemination applications because they: a) lack the abstraction of economic gain for content generator; b) they do not incorporate a model of interest profile, and c) they rely on a-priori knowledge about specific destination node identifiers. The

latter is not feasible in practical coupon dissemination applications in which the coupons are delivered to nodes based on their dynamic interest levels and redemption chances instead of the static identities.

In the following section, we introduce existing frameworks which are more fit to coupon dissemination applications in the sense that they focus on embedding consumers' interest profile and incentive mechanisms as parts of their system model and solution framework.

2.3 Interest-aware and Incentive-based Mechanisms

For a content dissemination method to be applicable for commercial applications, consumers' interest profiles (e.g., interest type, interest level) should be incorporated into the content dissemination framework. In other words, nodes are identified with interest types/levels. Also, an effective incentive mechanism should compel nodes to contribute to message forwarding in such a way that the overall cost (e.g., delay, number of hops) is minimized.

Interest-Based (IB) routing [39] uses an m-dimensional interest profile. Similarities between interests are measured using the cosine similarity metric. Forwarding is done to the relay nodes, which have similar interest profile(s) to that of the destination. Here, a destination must be known a-priori.

In the social-aware interest-cast mechanism SCORP [20], nodes are assigned binary interest levels (i.e., θ or I) toward different items. A utility of *social weight* is used for capturing a node's contact durations with the nodes interested in a type of content. Nodes with higher *social weight* are considered better forwarders. SCORP improves delivery rate and delivery cost. The binary interest level used in SCORP's model does not match practical business models with non-binary consumer's interests for a coupon type.

 P^3 -coupon [17] proposes a probabilistic coupon dissemination method focusing on promptness and privacy. Users are not identified with interest profiles. Thus, this approach does not attempt to route a coupon to destination(s) with specific interest. Instead, it assumes all users are potentially interested in owning a coupon. A coupon receiver becomes an owner of the coupon based on a defined probability. Each intermediate node on the forwarding chain receives some portion of the total associated reward (designated by the coupon generator) based on its rank in the forwarding chain. P^3 -coupon uses Bluetooth Service Discovery Protocol (SDP) toolkit for Unicast coupon exchange which does not require pairing, thus, provides prompt coupon exchange.

MobiCent [40] is an incentive compatible scheme proposed for replication based DTN routing protocols. It works on top of existing routing protocols to ensure that selfish actions do not result in larger rewards. Using Multiplicative Decreasing Award (MDR), MobiCent provides an incentive compatible payment mechanism to cater to clients that want to minimize either payment or data delivery delay. The protocol is also capable of identifying and reacting to DTN related attacks such as edge insertion attack and edge hiding attack. MobiCent treats the routing protocol as a black-box and is independent of the specific algorithm used. Authors choose Epidemic [37] as the underlying routing algorithm to evaluate MobiCent's performance. MobiCent requires a trusted third party to verify and manage payment transactions. Another method RELICS (innetwork ReaLization of Incentives to Combat Selfishness) [41] introduces *in-network-realization* of incentives, in which it does not pay nodes virtually, but rewards them in a physically realizable way. RELICS assigns a rank to each node based on node's reciprocity of service. That is, a node obtains a higher rank if it forwards messages originated from others more frequently. Message forwarding is then prioritized based on the message generator's rank. RELICS, not a routing

protocol itself, can be mounted on top of any routing protocol to facilitate energy- and incentiveaware message forwarding.

COUPONS [19] introduces a multilevel incentive mechanism for coupon dissemination in mobile networks. The incentive mechanism rewards intermediate nodes in a forwarding chain equally or based on a geometric series (See Section **1.3.4**). The applied dissemination mechanism is not social-aware. However, based on feedbacks from the network, including acknowledgements and node density, COUPONS controls the rate of coupon duplication and forwarding. COUPONS improves delivery ratio similar to P^3 -coupon. These methods are fit to coupon dissemination applications in the sense that they address privacy issues regarding D2D dissemination, as well as encouraging nodes' cooperation (with embedded incentive mechanisms). But they do not formulate nodes' interest levels and redemption probability.

SocialCast [42] defines a system model embedding nodes' interest profiles. It then defines a combined node utility based on 1) a node's colocation with another node sharing the same interest type, and 2) change of degree for a node. SocialCast uses Kalman Filter [42] to predict nodes' utility. The higher a node's utility is, a better content carrier it is. SocialCast is shown to improve delivery ratio and latency compared to a non-prediction based method.

The system model introduced in [16] is closer to coupon dissemination business models as it considers consumers' redemption probabilities. Its forwarding logic assumes that each node knows its neighbors' redemption probabilities and topological degrees. It defines a social-aware incentive mechanism such that a node's gain (acquired incentive) is a function of the redemption probability of the nodes to which it forwards the coupon. Therefore, users implicitly maximize advertisers' interest by maximizing their own interest. This method is tested on a synthesized contact network
with a known redemption probability distribution, rather than a wireless social network with node mobility.

Social-Aware Networking (SANE) [43] explicitly considers user interest information. It uses an interest profile, and similarities between interests are measured using the cosine similarity metric. Forwarding is done to the relay nodes, which cosine similarity of their interest profile(s) to that of the destination's is larger than a relevance threshold. The parameter relevance threshold needs to be adjusted experimentally. SANE attempts to take efficient forwarding decisions using utility prediction. However, it does not use a structured learning framework to perform this task. Next section addresses the application of learning in protocol evolution on different network layers.

2.4 Communication Protocol Evolution

Communication network protocols are designed based primarily on heuristics and past experience of the designer. While this works well for simple networks with homogeneous and prior known operating conditions, often this approach does not scale well for heterogeneous and dynamically changing networks. This motivates a field of network protocol design using evolutionary mechanisms that can adapt with heterogeneity and changing operating conditions, thus avoiding redesigns with changing conditions.

Communication protocol evolution is a tool that can be used by a collection of network entities to synthesize a set of rules as protocol components for a given network and loading conditions. *Parameter evolution*, as a rudimentary form of protocol evolution, allows the evolution of one or multiple parameters of an already-designed protocol logic.

An example of parameter evolution can be found in [3] where Epidemic Routing [3] is used as the target protocol for which *forwarding probability* is selected as the evolvable parameter. The

23

authors use Genetic Algorithms (GA) to evolve the parameter *forwarding probability* for maximizing fitness. While being useful for logic optimization, parameter evolution itself does not serve the purpose of logic/protocol synthesis.

The anycast evolutionary-based routing proposed in [44] defines a chromosome as a set of randomly selected possible paths. Using Genetic Algorithms (GA), it evolves more fit set of routes (e.g., with lower delivery delay). Simulation results show that the proposed GA-based method can reduce the average delay compared to the shortest path algorithm. This approach however, faces challenges when used in highly dynamic networks (such as SWNs) where paths between sources and destination(s) change frequently.

[45] finds a routing solution for flying Unmanned Aerial Vehicles (UAVs) with mutual sense and communication capability. The proposed fuzzy-based ant colony routing strategy computes local heuristic information based on *crowd density* of peers around a source node and *relative velocity* between a source and its peers. Combining the locally acquired information and ant pheromone, a routing strategy is exploited which improves delivery probability and average latency compared to few existing methods, including Epidemic [37] and Direct Delivery [46].

Genetic Programming (GP) is widely applied for program evolution [4]. GP contributes to the evolution of a set of instructions (i.e., implementing algorithms) as genes, rather than a set of parameters as done in [3]. [6] is an example of using GP for protocol evolution in which the evolution of IEEE802.11 contention window behavior is explored. It was reported that the types of evolved MAC contention window behaviors are less complex than the well-known exponential back-off. The methodology [6] does not apply to the whole MAC functionality. Instead, it aims to optimize MAC protocol performance through evolving the contention window behavior.

Very few studies investigate use of evolutionary techniques in developing routing protocols for DTNs and specifically for SWNs. One reason may be that SWNs usually involve extreme dynamic human mobility patterns, while evolutionary methods generally require a convergence period to adapt to environment changes. Feasibility of evolving content dissemination protocols in SWNs in an online manner can be considered as an open issue.

2.5 Reinforcement Learning based Approaches

Reinforcement learning (RL) has been shown to address routing challenges caused by dynamicity of wireless networks [47]. Using RL, nodes attempt to learn about a defined quality metric (e.g., buffer occupancy, distance to destination, etc.) using information they collect locally from their encounters. The goal is to maximize a reward which is associated with the defined quality metric.

A Q-learning based routing, namely, Q-routing, was first introduced in [48]. Simple experiments on a 36-nodes network irregularly connected, proved Q-routing to be superior compared to a non-adaptive shortest path based routing algorithm in terms of adaptation and maintaining efficient routing in presence of network dynamicity, such as varying network load.

Delay Tolerant Reinforcement-Based (DTRB) [49] is a flooding-based routing solution for 802.11 wireless networks. It utilizes Multi-Agent Reinforcement Learning techniques to learn about routes in the network and forward/replicate messages that produce the best reward. Reward, in DTRB's framework is associated with nodes' distance to destination(s). It is shown that DTRB improves delivery ratio especially in dense population areas.

Adaptive Reinforcement-Based Routing (ARBR) for DTNs [50] uses a Collaborative Reinforcement Learning (CRL) technique for enabling a group of nodes to make cooperative forwarding decisions that can increase the delivery ratio and reduce the number of transmissions

and message delivery time. The quality metric in this learning framework is associated with nodes mobility statistics, congestion, and buffer occupancy.

Both DTRB and ARBR [15] are not interest-aware protocols and their quality metrics are different than the defined gain used in the proposed approaches in this dissertation.

2.6 Summary

Most DTN Multicast and Unicast routing approaches in literature attempt to optimize individual performance indices such as message delay [37], forwarding cost, and storage. None define composite cost factors by combining multiple such indices into a single *gain* index. None of the existing methods are applicable to commercial dissemination applications, since they do not formulate the notion of *economic gain of dissemination* for a coupon provider. Throughout this proposal, we introduce a *Gain Utility* to control D2D coupon dissemination with an end goal of maximizing the economic gain for a coupon generator.

Few existing approaches are not applicable to marketing purposes since they do not model consumers' interest and redemption chance for specific types of content. Among those which define consumers' interest (e.g., SCORP [20]), binary consumer interest levels (i.e., 0 or 1) are defined. According to existing business models [11], consumers' interest level toward an item is neither binary, nor static.

Most routing protocols rely on a-priori knowledge about specific destination node identifiers (e.g., [39]). This is not feasible in practical coupon dissemination applications in which the coupons are delivered to nodes based on their dynamic interest levels and redemption chances instead of the static identities.

Methods which require packets to record the whole forwarding chain between a source and a destination (e.g., [17]) are less scalable. Furthermore, from a privacy preservation point of view,

26

storing the forwarding chain in a coupon exposes information about consumers' social interactions to a centralized server. To preserve consumer's privacy, a consumer's interest profile is preferred to be stored locally within his/her mobile device. In the proposed model, a consumer device is allowed to share consumer's interest profile with other devices selectively upon meeting, but is not allowed to share such information with a centralized entity.

Among learning based approaches, D2D content dissemination protocol design using evolutionary frameworks is considered as an open issue. Communication protocol design using evolutionary mechanisms provides adaptability with heterogeneity and changing operating conditions. Throughout this work, we present first steps toward evolving network communication protocols (e.g., for Medium Access Control layer).

Another group of learning based approaches, namely, reinforcement learning based ones, are not applicable to marketing applications, since they do not: 1) explicitly define and employ a composite economic gain for the content generator, or/and 2) incorporate a model considering interest profile of nodes toward a content/product. Part of this dissertation attempts to address both the limitations by embedding a learning mechanism with the goal of maximizing the defined economic gain.

3 Gain-aware Unicast Dissemination Protocol (GDP)

3.1 Introduction

Routing in Delay Tolerant Networks [51] addresses the technical issues in delivering single or multiple content in heterogeneous networks that may lack continuous network connectivity. When no end-to-end path is present, intermediate nodes take custody of the data being transferred whenever opportunity arises. To choose the most appropriate content forwarders, majority of routing protocols attempt to explicitly minimize one of the DTN routing indices such as message delay [37], forwarding cost [46], or storage [31]. Those protocols attempt to explicitly minimize any one of the above three indices at a time.

From an operational standpoint, it is often necessary to strike a balance between multiple of those indices instead of targeting the best performance for a single index while neglecting the others. For instance, a routing protocol that offers excellent delay performance may require an unacceptably large number of forwarding (e.g. flooding), thus causing impractical amount of energy burden on the intermediate forwarding nodes. Similar practical problems also exist for the protocols that attempt to minimize forwarding-count or storage without considering the primary application requirement, namely, delay. Therefore, a desired DTN routing protocol should be able to strike an operational balance between the different objectives.

As a first step towards this goal, we develop a composite delay-forwarding cost index, and design a DTN routing protocol around that index. Many practical mobile applications would benefit from DTN routing based on such a composite cost index that combines the effects of delay, forwarding cost, and sometimes storage.

There are many applications of DTNs ranging across education, healthcare, government, interplanetary networks [52, 53], vehicular networks [54] and marketing services [19]. As a

marketing strategy, consider a coupon distributing application which distributes time-constrained coupons through the wireless network using users' mobile devices. Let the *value* of a coupon be the amount of discount its recipient gets when she or he redeems the coupon at the store. And the economic gain of the store from disseminating the coupon is defined as the generated value upon delivery minus the forwarding costs. Therefore, for a *value* function that linearly decreases with time, it is desirable to deliver the coupon as early as possible with the minimum possible forwarding cost, i.e., through the fewest number of intermediate hops. In other words, the goal would be to find the right combination of delivery delay and number of forwards so that the economic gain of dissemination is maximized.

In self-organizing unmanaged networks nodes cannot be expected to forward messages without considering their own energy-cost of forwarding. This may require the content distributor to pay a *rebate* (i.e., contributing to the forwarding cost) to each user whenever he/she forwards the content. The concept of *rebate* is particularly important in self-organizing social settings in which there is no central authority that can enforce nodes to store and forward content without incentives. In addition to the coupon example above, the concepts of *value*, *rebate*, *and economic gain* apply to a number of other contents such as advertisements, event notifications, etc.

In this chapter, we propose an economic gain-aware DTN routing protocol that automatically adapts itself to different combinations of value functions and forwarding costs without requiring any explicit parameter tuning. Specific contributions are as follows. First, we design and parameterize a composite *gain* function that combines the effects of delay (via *value*) and forwarding-cost (via *rebate*) during unicast DTN routing. Second, we develop a unicast routing mechanism that yields the best possible *gain* for a given network and mobility pattern. Third, we develop a unicast DTN routing protocol *Gain-aware Dissemination Protocol* (GDP) that utilizes the time-dependent *gain* function as a forwarding decision metric in both direct and transitive manners, to route content from a mobile source to a mobile destination. Finally, using DTN simulator ONE [55], the proposed protocol is extensively characterized along with few other existing unicast DTN routing protocols.

The proposed mechanism assumes full node cooperation. In other words, it does not address issues raised in the presence of non-cooperative and colluding selfish nodes, and attacks such as Sybil [56] and whitewashing attacks [57].

3.2 Network Model

DTN Networks can be formed in public places such as in a University where students carrying mobile devices can form an ad-hoc system. Due to periodic schedules of classes, lunch hours, and other social events, students in a University often form communities which are groups of students with similar mobility patterns. Students taking same set of courses or students belonging to same dormitory are examples of community formation.

Consider a student X whose mobile device is in possession of an electronic coupon from a fast food restaurant in the campus. The coupon was delivered to the device when student X visited the restaurant. Student X can now distribute the coupon (single hop or multi-hop) to students of dormitory D_y through X's classmates who reside in dormitory D_y . In this section, we design a unicast DTN routing protocol GDP which can maximize the *gain* of the coupon delivery from source student X to a pre-determined destination student Y.

Consider the DTN in Figure 3-1, in which the vertices represent mobile nodes, the edges represent the occurrence of contacts between node-pairs, and the weights on the edges indicate the average inter-contact time between the nodes. There are three unicast paths from source *S* to destination *D*, namely *S-M-N-O-P-D*, *S-A-B-D* and *S-X-D*. The delivery latencies from *S* to *D*

across these paths follow the pattern: $S-M-N-O-P-D \ll S-A-B-D \ll S-X-D$. This is because the nodes in the path S-M-N-O-P-D meet each other frequently, whereas the nodes in the path S-X-D meet each other very infrequently. When the *value* obtained by delivering the content to D does not depend on the delivery time and the forwarding cost (i.e., the *rebate* given to the forwarding nodes) is high, it is preferable for S to hold on to the content and deliver it to D along the path S-X-D so that the number of forwards is minimized. On the other hand, if the value of the content falls rapidly with time and the forwarding cost is low, it is preferable for S to forward along the path S-M-N-O-P-D so that the delivery delay is minimized.



Figure 3-1: Example user interaction and DTN routing scenario

3.3 Problem Formulation

<u>Value</u>: The value of content is a monotonically non-increasing function of time. For example, the value of an electronic discount coupon delivered over a phone-to-phone DTN network is expected to be the maximum in the beginning, and zero after its expiry time, which is termed as the value duration. The shape of the function (e.g., liner, exponential, step etc.) is set by the content originator. Value can be abstracted for other time-critical electronic content including short-cycle advertisements, warning/alert messages, and peer-to-peer auction services. Very similar value abstraction can be applied to a short-cycle advertisement being distributed by a merchant such as a department store. The objective is to deliver an advertisement to a group of interested destinations such as students in a campus. A price-aware algorithm should ideally manage delivering the

content to an interested person within the defined expiration period. Earlier delivery induces a higher value. The value function in this example would be a step function with an initial amplitude that becomes zero at the time of expiry. Another embodiment of value would be to distribute time-bound content within a peer-to-peer social network. Consider an example in which a person twits a content which should be peer-to-peer distributed among a group of interested receivers within a certain period of time that is determined by the lifetime of the content (e.g., the lifetime of a photo in Snapchat; http://www.snapchat.com/). The value function of such a content in this application is determined by the lifetime of the content set by its originator.

A simple linearly declining *value* function can be written as:

$$V(t) = undefined, \quad t < t_0$$

$$= V_0 - \alpha(t - t_0), \quad t_0 \le t$$
(3-1)

where t_0 is the time of content injection, V_0 is the initial value, α is the value gradient, and $V(t) \ge 0$ for all *t*.

Cost: Forwarding *cost* at time *t* can be expressed as:

$$C_F(t) = C_R \cdot N_F(t) \tag{3-2}$$

where N_F (t) is the number of forwarding till time t, and C_R is a fixed cost of each forwarding, which is the rebate or reward given to the forwarding users. Practical formulation of C_R will depend on specific target application. For a coupon application, C_R can represent a credit, possibly a fraction of the coupon value [19], which is given to each forwarding node as an incentive for energy-expensive content forwarding. Other formulations of C_R are possible for specific applications.

Gain: Combining *value* and *cost*, the gain can be written as:

$$G(t) = V(t) - C_F(t)$$
(3-3)

Gain can be negative. Meaning the overall benefit of a content delivery can be negative when an expired content is delivered or when the overall forwarding cost is higher than the obtained value upon delivery. Protocols which are unaware of the notion of gain are in danger of delivering the packet with a negative gain, i.e., a worthless delivery.

<u>Routing Problem</u>: For given value function with specified V_0 and α , cost factor C_R , and network mobility, the objective is to design a unicast DTN routing algorithm that maximizes the gain G(t) at content delivery.

3.4 Gain Upperbound

To investigate the gain performance of the experimented DTN routing protocols (including that of GDP), it is essential to know the gain upper-bound with given network mobility and content value profiles. To this end, we developed a strategy to find the gain upper bound for when all system parameters including node mobility is known *a prioi*. In other words, we attempt to find the maximum gain a packet can achieve in a certain scenario, so that we can compare the achieved upper-bound with the gain that other protocols, including GDP, can obtain. A full flooding approach can be used for: a) finding all possible routes, b) computing the gain via such routes, and c) choosing the one with maximum gain. This cannot be implemented (even in simulation) because of non-polynomial complexity of storage of content that are stamped with all possible routes during flooding. A scalable Routing with Gain Aware Pruning (*RGAP*) for finding the performance upper bound is devised as follows. Note that the role of RGAP is much like the Oracle based DTN routing protocols [46] that are able to provide the best-performing forwarding when the node mobility is known *a priori*.

In RGAP, An intermediate node 'i' maintains a variable l_i which represents the Minimum

Loop-free Forwarding Count (LFC) among the paths taken by all copies of the content received by node 'i'. The concept is illustrated in Figure 3-2. In the left, since there is no forwarding loop, l_i in this case is 3. In the middle, the number of forwarding is larger than the loop-free forwarding count – which is 2. The right one shows that additional looping does not change l_i . Up to a given time, if node I received three copies via those forwarding scenarios, then n_i at that time will be 2, which is the minimum of the three.

Before receiving the first copy, l_i is set to a large value at each node *i*. Each copy maintains a field n_{LFC} , representing the LFC encountered by that copy so far. Before forwarding its copy, node *I* updates the n_{LFC} field as $n_{LFC} = l_i + 1$. When a node *j* receives a copy, if for that copy $n_{LFC} < l_j$, then l_j is set to n_{LFC} , and the copy is subsequently flooded. If $n_{LFC} \ge l_j$, then the received copy is discarded. The destination maintains a variable G_{max} which at a given time represents the maximum of the gains obtained from the individual received copies so far. Upon receiving a copy, the destination computes (using V(t) and C_R) a temporary gain G_{rx} . If $G_{rx} > G_{max}$, then G_{max} is replaced by G_{rx} .

<u>Proof of Gain Bound</u>: The objective of RGAP is to prune out routes (from all possible end-toend routes as a result of flooding) which are guaranteed to not yield higher gain than the current maximum gain at the destination. The pruned routes are the ones that have both higher delay and number of forwards than the route corresponding to the current maximum gain at the destination. When an intermediate node j receives a copy with $n_{LFC} \ge l_j$, the path followed by that copy can be pruned because it results in both higher delay and forwarding counts, and hence, lower potential gain at the destination. This ensures that for given mobility and value/cost profiles, within the time till value becomes zero (i.e., value duration), G_{max} of RGAP monotonically reaches the maximum possible gain at destination.



Figure 3-2: RGAP for finding performance upper bound

3.5 Gain-aware Dissemination Protocol

Gain-aware Dissemination Protocol (GDP) is a heuristic based approach and driven by the key idea of forwarding content through a path with *low inter-node contact intervals*, so that the expected gain at the destination is maximized. GDP is a history based protocol that observes locality in node interaction patterns and develops utilities based on that for gain-aware forwarding.

3.5.1 State Variables

Each node *I* maintains two variables: 1) n_i : the expected number of hops to destination on an estimated route which yields the maximum gain, and 2) Φ_i : the expected delivery latency for the same route. Nodes interchange n_i and Φ_i via periodic *Hello* messages. When the node *I* meets a destination node *j*, it initializes Φ_i to be $\Delta_{i,j}$ which indicates the mean inter-contact time between nodes *I* and *j*. It also initializes the quantity n_i to 1. When node *I* meets any node *j* (i.e., node *j* is a destination or non-destination), both nodes update the quantities $t_{i,j}$ and $\Delta_{i,j}$, which are the meeting time and the mean inter-contact time between nodes *I* and *j* respectively.

Notation	Definition		
n_i	Expected number of hops to destination on an		
-	estimated route which yields the maximum		
	gain		
Φ_{i}	Expected delivery latency for a route to		
	destination on an estimated route which yields		
	the maximum gain		
t _{i,j}	Meeting time of nodes <i>I</i> and <i>j</i>		
$\Delta_{i,j}$	Mean inter-contact time between nodes I and j		
Ui	Expected gain by delivering a content		
	generated by node <i>i</i>		
$V(\Phi_i)$	Generated value after a delivery latency Φ_i		
C_R	Compensation given for each forwarding along		
	the route to destination		
t_o	Time of content generation and injection in the		
	network		
g _i	Expected gain by forwarding the content to		
2)	node <i>j</i> , when node <i>I</i> meets an intermediate node		
	<i>j</i> at time t_c ($t_o < t_c$)		

Table 3-1. Notations	and correst	onding def	initions use	ed in GDP
		Jonania avi		

The latter adaptively captures any locality in inter-contact pattern. Table 3-1 shows the notations used in this section and their corresponding definitions.

3.5.2 Updating State Variables

The expected gain by delivering a content generated by node I can be written as

$$U_i = V(\Phi_i) - n_i * c_R \tag{3-4}$$

where $V(\Phi_i)$ is the generated value after a delivery latency Φ_i , and c_R is the compensation given for each forwarding along the route to destination. Let us now consider a node *I* that possesses a content and it meets node *j*. If the latter is the destination, then n_i and Φ_i must be updated only if $U_i < V(\Delta_{i,j}) - c_R$. If that is true, then node *I* updates n_i as 1 and Φ_i as $\Delta_{i,j}$. If node *j* is not the destination, node *I* checks to see if it is profitable to use node *j* as an intermediate carrier to the destination. If yes, then n_i is updated such that it considers the one additional intermediate hop between node *I* and destination through node *j*. Also, Φ_i is updated to take into account both the average inter-contact time between nodes *I* and *j*, and the expected delivery latency from node *j* to the destination. Thus, if $U_i < V(\Delta_{i,j} + \Phi_j) - (1 + n_j) * c_R$, update n_i as $(1 + n_j)$ and Φ_i as $(\Delta_{i,j} + \Phi_j)$.

3.5.3 Forwarding Logic

When a node in possession of content meets the destination, it makes a delivery only if the remaining *value* for the content is larger than the forwarding compensation C_R . Using the constructs of n_i and Φ_i , a node chooses the best forwarding entity from its social peers who may or may not currently be in direct contact. Consider node *I* which is in possession of a content that was generated and injected in the network at time t_o . When node *I* meets an intermediate node *j* at time t_c ($t_o < t_c$), the expected gain by forwarding the content to node *j* can be written as:

$$g_j = V((t_c + \Phi_j) - t_o) - n_j * c_R$$
(3-5)

The expected gain by holding on to the content by node *I* is g_i . Every time node *I* meets an intermediate node *j*, in addition to computing g_i and g_j , it also computes the expected gain by forwarding the content through nodes that it met in recent past, but not currently (at time t_c) in direct contact. For such a node *k* which was last met by node *I* at time $t_{i,k}$, if $(t_{i,k} + \Delta_{i,k} > t_c)$, meaning it is not yet time for the next expected meeting, the expected gain by forwarding the content to node *k* can be written as:

$$g_k = V(t_{i,k} + \Delta_{i,k} + \Phi_k - t_o) - n_k * c_R$$
(3-6)

However, if the time for the next meeting has passed, that is if $(t_{i,k} + \Delta_{i,k} < t_c)$, the expected gain can be computed as

$$g_k = V(t_c + \Phi_k - t_o) - n_k * c_R$$
(3-7)

After computing g_i , g_j ($j \in all$ direct neighbors), and g_k ($k \in all$ recent past neighbors), node I first chooses the node with the highest g value for all j and k. If the node with that highest g (i.e. g_{max}) corresponds to a past neighbor, then no forwarding is done. If it is a current neighbor, then a forwarding to the corresponding neighbor is performed only if $g_{max} > g_i + C_R$. All the key concepts of GDP protocol are summarized in Algorithm 3-1.

Updating state variables:	Forwarding logic for node- <i>i</i> at time t_c :		
Node- <i>i</i> maintains Φ_i and n_i and	Compute expected gain for current neighbor-j		
measures $\Delta_{i,j}$;	as:		
When node- <i>i</i> meets node- <i>j</i> :	$g_j = V\left(\left(t_c + \Phi_j\right) - t_o\right) - n_j * c_R;$		
if (node-j is destination & $U_i < U_i < U_i$	Compute expected gain for past neighbor-k as:		
$V(\Delta_{i,j}) - c_R)$	$if \left(t_{i,k} + \Delta_{i,k} > t_c \right)$		
$n_i = 1; \varphi_i = \Delta_{i,j};$	$g_k = V(t_{i,k} + \Delta_{i,k} + \Phi_k - t_o) - n_k *$		
if $(U_{1} < V(A_{1} + \Phi_{1}) - (1 + n_{1}) *$	<i>c_R</i> ;		
$(\mathbf{U}_{i,j} + \mathbf{U}_{j,j})$	else		
r_{R}	$g_k = V(t_c + \Phi_k - t_o) - n_k * c_R;$		
$n_i = (1 + n_j); \ \Psi_i = (\Delta_{i,j} + \Psi_j);$	$g_{max} = \max \{g_i, g_j \ (j \ \epsilon \ all \ direct \ neighbors),$		
	g_k (k ϵ all recent past neighbors)};		
	if $(g_{max}$ belongs to a current neighbor &&		
	$g_{max} > g_i + C_R$)		
	Forward content to the neighbor with g_{max} ;		

Algorithm 3-1: The key components of GDP protocol

Note that a flooding/spray based version of GDP was not considered for the following reason. The basic premise of GDP is to be able to design a utility based forwarding in which the utility is explicitly designed based on the core objective of gain maximization. All the system parameters and their algebraic handling in Section 3.5 are based on this premise. Introducing spraying, which is a form of naïve flooding, goes against this core effort towards gain-awareness and introduces a sense of blindness in the forwarding effort. GDP is intended to avoid this by making the protocol as utility driven as possible.

3.6 Performance Characterization

Using the ONE [55] simulator, we compare GDP with Direct Delivery [46], PROPHET with estimation [30], Epidemic [37], Spray and Wait [58], and the RGAP (i.e., the performance upper bound) routing approaches. Protocol performance was evaluated using a community based synthetic mobility model and a trace based model [59] in which mobility traces were collected for participants in the conference INFOCOM '06.



Figure 3-3: Experimental network topology with six social cliques

3.6.1 Experiments with Clique-forming Synthetic Mobility

A *101*-node network, was structured in five *20*-node and one *1*-node social cliques or communities. The cliques are formed due to node mobility around six pre-positioned Point of Interests or POIs. Movement of each node is stochastically programmed by an inter-waypoint probability transition matrix. As shown in Figure 3-3, the cliques C1, C2, and C3, with *20* nodes each, are designed to be isolated in that the members of an isolated clique interact among each other, but do not move out to meet nodes from other cliques. The members of the other two *20*-node cliques C4 and C5 are migratory between C1-C2 and C2-C3 respectively. Finally, the single node in C6 migrates between cliques C1 and C3. An example realization of this model would be

the student interactions in a campus where the POIs represent class rooms, food court, athletic centers, etc., and the cliques represent students from specific academic units, dormitories, etc.

All results presented for this model corresponds to content generated at a node in C1 and is destined to a node in C3. With the above clique configuration, content needs be routed either through a path through migratory cliques C4 and C5, or through a path through the only node in the migratory clique C6.

We used a constant forwarding cost (i.e. C_R) and a linearly decreasing value function with the initial value set to 1. The maximum possible value of C_R is also set to 1. Main experimental variables are C_R and *Value Duration* (*VD*), which is the time for the value function to reach zero. The chosen range for C_R and *VD* are 0.02 through 0.18, and 1000sec through 20000sec respectively. Low VDs indicate faster reduction of content's value with time. The reported gain is averaged over ten different source-destination pairs with the source in clique C1 and destination in clique C3.

3.6.1.1 Protocol State Warm up

Before content is injected in the network the simulation is run for a warm up period during which the protocol state variables, as introduced in Section 3.5.1, are allowed to stabilize. Figure 3-4 shows the evolution of protocol state variable $\Delta_{i,j}$ (i.e., the expected inter-contact time) for a number of arbitrarily chosen node pairs in a representative simulation run.



Figure 3-4: Warm up time based on inter-contact time evolution

In this particular case, the warm up period is chosen to be 15,000s by which time the state variable $\Delta_{i,j}$ for all node-pairs were observed to have stabilized. Content injection and routing operation was initiated at this point. The warm up period of 15,000s was experimentally justified after observing the stabilization of $\Delta_{i,j}$ for a large number of node pairs (i.e., in this case 11). In our experiments, the inter-contact time was found to be in the range of 1s to 7000s, representing the most and least frequently interacting node pairs respectively.



Figure 3-5: Example contact statistics of a C4 node

3.6.1.2 Steady State Contact Statistics

Figure 3-5 depicts example node interaction statistics as CDFs of inter-contact time (ICT) between a node from the migratory clique C4 and all other network nodes in Figure 3-5. In Figure 3-5:a, the clusters of CDF lines indicate formation of ICT localities due to interaction affinities that GDP can leverage for gain-maximizing dissemination. Figure 3-5:b and Figure 3-5:c show the PDF of ICTs between a node in C_4 and nodes in C_1 and C_6 respectively. Results in Figure 3-5 clearly indicate that the C4 node's social interactions with nodes from C1 and C2 are stronger than with those from C5 and C6. This implies that to route content to a node in social clique C4, nodes from C1 and C2 would serve as better next hops compared to those from C5 and C6.

3.6.1.3 Protocol Self-tunability

Results in this subsection aim to establish the ability of the proposed GDP in achieving gain closer to the RGAP upper bound (see Section 3.4) compared to PROPHET. It also establishes that unlike PROPHET, GDP can self-tune its operation with varying value and cost parameters.

Figure 3-6:a shows the gain upper-bound computed using RGAP algorithm. The theoretical gain upper-bound is $1-C_R$, which happens when content is directly delivered immediately upon generation. Higher gains are achievable for lower C_R and for higher VD. Figure 3-6:b reports the performance of PROPHET when its parameters (i.e., initialization, transitivity, and aging constants) are tuned for each { C_R , VD} combination to ensure the best achievable gain. It performs well only for low C_R and high VD since for high C_R and low VD, often the content remains undelivered, indicated by zero gain.

Figure 3-6:c shows how PROPHET performs when it is tuned for a specific scenario (i.e., $C_R = 0.04$, VD = 4000 sec). While performing well in the vicinity of the tuned-for { C_R , VD} combination, it shows poor gain, often negative, at other regions of the { C_R , VD} space. Figure 3-6:b,c indicate the need for parameter 43enability depending on mobility, value, and cost profiles. Since mobility information is not always known a priori, parameter tuning may not be feasible for PROPHET and other parameter-configurable DTN protocols. Note that while observing gain across different protocols, more emphasis should be given on relative gains as opposed to on the absolute gains.



Figure 3-6:Gain comparison between GDP, PROPHET, and RGAP The absolute gains depend directly on the ratio of the initial content value V(0) to the forwarding cost C_R . That ratio also determines the amount of negative gains.

Figure 3-6:d reports performance of the proposed GDP protocol. Observe that the protocol is able to obtain gains that are close to that of the RGAP-indicated upper bound, except for few scenarios when the forwarding cost is too high and the content value reaches zero too quickly. It is worth mentioning that the high-gain performance of GDP was obtained without parameter tuning for different { C_R ,VD} combinations, and also for different mobility scenarios. This is mainly due to the fact that the protocol adapts itself with mobility by observing the inter-contact dynamics $\Delta_{i,j}$, and with { C_R ,VD} combinations by making forwarding decisions based on the gain-aware utility function as proposed earlier.

3.6.1.4 Evolution of Gain and Latency

Figure 3-7:a,b show comparative gain and latency for the proposed GDP and scenario-tuned $\{C_R = 0.04, VD = 4,000 \text{ sec}\}$ PROPHET for a representative run. As shown in Figure 3-7:b, since the route taken by gain-agnostic PROPHET remains unchanged, its delivery delay remains constant for all C_R . GDP, on the other hand, changes route based on the forwarding cost. Initially for low C_R , it forwards content through a low-delay (and high forward-count) path, but as C_R increases, the route is shifted towards a higher delay (and low forward-count) path. Figure 3-7:a shows that irrespective of the path shifting, GDP always maintains high gain, which is what it is designed for. Figure 3-7:a also shows the corresponding performance of RGAP, which is very close to GDP.

Figure 3-7:c shows gain evolution for PROPHET and GDP with time. Generally, gain reduces by C_R upon each forward, and finally rises upon delivery. PROPHET in this case delivered earlier than GDP, but due to more aggressive forwarding, its eventual gain is smaller.

To summarize, the results in the above two subsections indicate GDP's self-tuning ability to perform gain-aware forwarding and to obtain gain values that are close to the practical upperbound observed from the RGAP mechanism.

3.6.1.5 Delay, Forwarding Cost, and Gain

Having observed the overall gain trend of GDP in the previous two subsections, in this and a few following subsections we characterize the protocol in further details. In addition to PROPHET and RGAP, we analyze Epidemic Routing [37] and Spray and Wait (SNW) [58] in this characterization phase. For each routing protocol under consideration, we ran *100* experimental scenarios, each with a different seed for enabling different mobility and node interaction scenarios.



Figure 3-7: Comparative delivery delays and gain evolution

Figure 3-8 shows a scatter-plot for delay versus forwarding count for different protocols when the forwarding cost is set to 0.01 and the value duration (*VD*) is set to 20,000s. Each point in the graph for a given protocol corresponds to a simulation run with a specific seed (i.e., mobility/node interaction scenario). The numbers in parentheses indicate average Delay (*D*), Number of forwards (*F*) and Gain (*G*) across all 100 scenarios.

Note that the cluster for RGAP in Figure 3-8 occupies the region corresponding to the lowest delay and fewest forwarding, thus ensuring the best gain performance. The cluster for Epidemic Routing [37], however, indicates that while the protocol is consistently able to deliver content very early, it incurs high forwarding expenses. The cluster for Spray and Wait (SNW) [58] incurs relatively lower forwarding cost while its delivery delay is higher and spread over a wider range.

The protocol PROPHET [30] has a similar spread in delay, but its forwarding cost is lower than SNW, mainly due to its usage of an adaptive utility of *delivery predictability*. The cluster for GDP also shows a large delay spread, but its forwarding cost is strictly bounded mainly because GDP delivers a packet only if the combination of delay and forwarding cost generates a positive gain. Thus, even when the delay is large, content is delivered only if the forwarding cost is limited enough to maintain a positive gain. In this manner, GDP is able to maintain a forwarding cost bound (i.e., *4*) which is similar to that of RGAP. The intersection of GDP and RGAP clusters indicates how GDP can achieve the same performance as that of RGAP's in a number of node interaction scenarios. Note that the delay-forwarding behavior of the protocols presented in Figure 3-8 were observed to be valid for a wider range of cost and value durations (e.g., cost: *0.02, 0.1*, etc., and *VD*: 5K, 40K, etc.).



Figure 3-8: Delay vs. Forward count with cost: 0.01 and VD: 20K

Note that the initial number of copies for the SNW protocol was set to 723, which turned out to be the minimum number of initial copies needed for content delivery. The reasons for the initial number of copies to be that large were as follows. The adopted community based model contains special node dynamics involving bridge nodes. For example, nodes in community C4

(refer to Figure 3-3) are bridges which receive packet from the source in community C1 and then relay it to any of the nodes in community C2. We observed that in such scenarios, it is essential for the bridge nodes to maintain enough copies of packets, so that they can relay the packet to the next bridge node after sharing the packet with their own neighbors in a binary spraying manner. Note that, a binary version of Spray and Wait is used with the community based model.

Let us now analyze an example which happens in our community based scenario. The static source in community C1 starts forwarding with community 723 copies and shares its copies with other 19 members of community C1 using a binary Spray [58]. Afterwards, most community C1 nodes already hold less than 90 copies of the packet. A member of community C4 visits a node in community C1 (with 90 copies) and takes half of its copies, which is 45. This node reaches the wait phase after forwarding the packet to 6 of community C4 members. Note that a node in its wait phase switched to a direct delivery behavior. Thus, it can be seen how quickly a bridge node from community C4 enters the wait phase and loses its ability to relay the packet to community C2 which contains the bridge nodes that connect community C1 and C4 members in the mentioned scenario, community C4 nodes have a small chance of holding enough number of copies to share with community C2 members. Note that community C2 members. Note that copies with community C5 members later, since community C5 members are the only ones (except the single node in community C6) which meet destination directly and can deliver the packet.

The higher forwarding cost of SNW comparing to PROPHET in Figure 3-8 is explained by the large number of forwards that the spraying phase in SNW incurs due to the large required number of initial copies (i.e. *723*).

48

Figure 3-9 shows delivery delay vs. gain when the forwarding cost and value durations are set to *0.01* and *20,000s* respectively. Observe how Epidemic Routing produces a point cluster with very low delay but sometimes with negative gain. This is because Epidemic delivers content as early as possible in a forwarding cost-insensitive manner, thus, maintaining a bound on the delay – but often with very low gain due to high forwarding cost. For all other protocols, the gain generally decreases with increase in delay, thus reducing the value of content at delivery time. The clusters of PROPHET and SNW are more spread out compared to GDP and RGAP. That is because GDP bounds the number of forwards (Figure 3-8) while attempting to reach the maximum gain.

Figure 3-10 shows the general trend of decreasing gain with increasing forwarding costs. It depicts high forwarding costs for both SNW and Epidemic even when their gains are low. PROPHET, with its *delivery predictability* abstraction, provides slightly better gain, even though its forwarding cost is quite significant. GDP, however, with its gain-aware utility formulation is able to provide a much better gain cluster with extremely low forwarding cost. Observe the proximity and overlapping of the GDP cluster with that of RGAP which provides the gain upper-bound. From the results in this subsection it is apparent that GDP is able to provide the best gain performance when compared to the other protocols, except the upper-bound provided by RGAP.



Figure 3-9: Delay vs. Gain with cost: 0.01 and VD: 20K



Figure 3-10: Forward count vs. Gain: cost 0.01 and VD 20K

3.6.1.6 Impacts of Forwarding Cost

Figure 3-11 reports the (delay, forwards) scatterplots for GDP with two different forwarding costs 0.01 and 0.02, when the value duration is set to 20,000s. A notable observation in Figure 3-11 is that the point clusters for the two cost scenarios overlap significantly, thus pointing to the fact that the protocol is able to adjust its routing behavior in order to maintain an optimal number of gain-maximizing forwards based on the underlying forwarding expenditures. The non-overlapping points in the figure belong to different routes taken toward the destination.



Figure 3-11: Delay vs. number of forwards for GDP

Figure 3-12 and Figure 3-13 corroborate the above observation via the finding that different forwarding costs minimally affect the obtained economic gain. This is accomplished through self-tuning of the forwarding aggressiveness. In both figures, the gain and its spread for GDP with cost 0.01 is slightly larger than that with cost 0.02.

3.6.1.7 Impacts of Value Duration

Figure 3-14 depicts delay vs. number of forwards for three different value durations (*VD*) of 5000s, 20,000s and 40,000s (i.e., 1.3hr, 5.5hr and 11hr) respectively. Observe that with the lowest *VD* of 5000s, GDP offers earliest delivery, but with very high forwarding cost. This is because the protocol attempts quicker delivery via aggressive forwarding before the value of the content reduces too much. As the *VD* increases (i.e., 20000s and 40000s), the protocol becomes less aggressive, thus allowing relatively higher delivery latency and lesser forwarding costs. For the delay-gain scatter plots in Figure 3-15, observe that the gain decreases with a steeper slope as the value duration decreases. This is consistent with the observation from Figure 3-14, validating that

for content with short-lasting value, quick deliveries are essential in order to preserve the potential gain.



Figure 3-13: Number of forwards vs. gain for GDP

From the forward-gain scatterplots in Figure 3-16, it can be observed that the points for majority of the scenarios with VD = 40000s are scattered around the minimum number of forwards, i.e., 4. The reason is that with very large value duration, in an ideal gain-aware routing

protocol, nodes would hold and carry the content for direct delivery. Therefore, higher value duration justifies a lower forwarding cost.





cost: 0.01



Figure 3-16: Number of forwards vs. gain with forwarding cost: 0.01

3.6.1.8 Joint Impacts of Forwarding Cost and Value Duration

Figure 3-17 reports value duration vs. gain for a range of forwarding costs for all the evaluated protocols. Generally, with extended value duration since the value of a content delivery is higher, the gain is also higher for all the protocols. A comparison between the graphs in Figure 3-17:a through Figure 3-17:c reveals that the performance gap between Epidemic and GDP increases as the cost increases from 0.01 through 0.1. Same can be also observed while comparing the gains

achieved by GDP and PROPHET. The reason is that a more disciplined decision should be taken while forwarding when the cost of forwarding is higher. In other words, it is more reasonable to restrict liberal forwarding (as done in Epidemic) when the cost is higher. GDP, on the other hand, chooses a more conservative forwarding approach in such situation, which leads to a higher gain. We have also simulated a Direct Delivery [46] approach which was not able to deliver content due to very low probability of direct contact between the source and destination in this and many other practical mobility scenarios. Both Epidemic and Direct Delivery are naïve in that they are value un-aware and do not consider any routing utilities. They are used only for the purpose of illustrating the typical patterns in the gain dynamics. Overall, Figure 3-17 confirms that GDP is able to provide better economic gain compared to all other simulated protocols in the presence of this model based mobility scenarios.

3.6.1.9 Effects of Varying Content Generation Time

For all the presented results so far, the content was generated at a specific time *15000s*, which is after the network warm up period. In order to validate the performance consistency over different generation times, which can potentially change content trajectory, results are presented for a different content generation time in this section.

Figure 3-18 and Figure 3-19 depict value duration vs. gain when content is generated at *16000s* and *17000s* respectively with the clique-forming synthetic mobility. Similar to all previous results, in both scenarios, the same increasing trend of gain for increasing value duration can be observed. As depicted, for low *VD*s GDP choses no content delivery over a delivery with negative gain. But for higher *VD*s, GDP's gain is closest to that of RGAP compared to the other protocols.



Figure 3-17: Number of forwards vs. gain with forwarding cost: 0.01



Figure 3-18: Performance for content generation at 16000s



Figure 3-19: Performance for content generation at 17000s

3.6.2 Experiments with Trace Based Mobility

As the second mobility model, we have used human interaction traces collected [59] from INFOCOM '06 conference. The trace contains data from 97 devices. The first 20 devices (i.e., iMotes) are static and the rest were carried around by a group of users for all four conference days.

As done with the model based synthetic mobility in Section 3.6.1.1, the system waits for a warm-up period before content is injected. According to the evolution of $\Delta_{i,j}$ shown in Figure 3-20, *75000s* is used as the content injection point for the following experiments. Note that unlike in the synthetic mobility scenario in Section 3.6.1.1, the source and destinations nodes in this case do directly meet in some instances, thus making Direct Delivery feasible.



Figure 3-20: Warm up time for the INFOCOM '06 mobility trace

Figure 3-21:a through Figure 3-21:c show GDP's gain with varying value durations when the forwarding costs are set *at 0.05, 0.1* and *0.14* respectively. As expected, in all scenarios the gain increases with increasing value duration. The gain difference between GDP and Direct Delivery decreases slightly with higher cost and higher value durations. This is because it is desirable for a protocol to withhold the content and perform a direct delivery when both the value duration and forwarding cost are high. GDP adapts itself to such behavior, which results in a similar gain for Direct Delivery and GDP under the mentioned situation.

It should also be observed that with higher forwarding costs, the gain differences between GDP and all other protocols (i.e., Epidemic, SNW, and PROPHET) increase. In fact, GDP is the only protocol which retains a positive gain for almost all value durations with costs as large as 0.1 and 0.14. This is in contrast to all non-GDP protocols that can maintain positive gains only with forwarding costs smaller than 0.05. That is why the minimum cost value is chosen to be 0.05 for these experiments. For the clique-forming synthetic mobility results in Section 3.6, this threshold value was 0.01 (see Figure 3-17). It is noteworthy that for very small forwarding costs and value

durations (e.g., cost: 0.05 and VD: 1000 in Figure 3-21:a) GDP can be outperformed by the less conservative protocols such as Epidemic and PROPHET.



Figure 3-21: VD vs. Gain for INFOCOM '06 trace; Source node N5;(a): *Cost 0.05*, (b): Cost 0.1 and (c): Cost 0.14

3.6.2.1 Effects of Different Content Sources

All our previous results were shown for the distribution of a content that was generated at a specific node. Figure 3-22 shows the comparative protocol performance when content is generated at a different node chosen from the INFOCOM '06 mobility trace. Note that there are no reported results for Direct Delivery in Figure 3-22 since the protocol was not able to deliver content for any of the value duration and cost combinations that we experimented with. This indicates that the source and destination never met.
There are no reported results for the protocol spray and wait (SNW) either. This is because there were not enough contacts for a successful delivery during both the spray and wait phases. These results, along with similar experiments we ran for a broader range of costs (e.g., 0.3, 0.4) demonstrate the consistent performance advantage of GDP (i.e., close to the RGAP performance) with higher value durations for arbitrary content generators.

3.6.2.2 Effects of Varying Content Generation Time

To evaluate the effects of varying content generation time on GDP performance, we generated the packet at a later point than *75000s*. Figure 3-23 depicts value duration vs. gain when content is generated at *82000s* with INFOCOM '06 mobility. Even with change of content trajectory, GDP still holds the maximum gain. Similar to all previous results, increasing trend of gain is visible due to longer value durations.

Overall, from all experimental results (i.e., with different mobility models, forwarding costs, value durations, source nodes, and generation times) it was found that approximately for *87.1%* of the experiments, GDP achieved a gain that is closest to the upper bound found by the RGAP protocol. Direct Delivery and PROPHET win in *10%* and *2.5%* of the experiments respectively. These happen mainly when both the value duration and the forwarding costs are extremely low. These clearly show the dominance of GDP from the standpoint of gain as formulated in this work.



Figure 3-22: VD vs. Gain for INFOCOM '06; Source node N1; (a): Cost 0.05, (b): Cost 0.1 and (c): Cost 0.14

3.6.3 Gain-positive Delivery Ratio

In the present context of gain, successful deliveries are defined as those only with positive gains. The index *Gain-Positive Delivery Ratio (GPDR)* is defined as the fraction of all content that are delivered with positive gains. Note that a protocol may have high delivery ratio in the conventional sense, but it may suffer from poor GPDR if its routing strategies are not gain-aware. Epidemic and Direct Delivery, for example, can maintain a 100% delivery ratio if the notion of gain is not involved. Yet, both of them can perform poorly from a GPDR standpoint. Epidemic in one extreme may use frequent number of forwards, until the forwarding cost exceeds its value. Direct Delivery in the other extreme, may postpone content delivery until its value reaches zero.



Figure 3-23: Performance for content generation at *82000s*; Infocom '06 traces; Source node N5; Cost *0.05*

Figure 3-24 shows the GPDR computed over *100* experiments with clique-forming synthetic mobility for the protocols GDP, Epidemic, SNW, and PROPHET. Combinations of three C_R (forwarding cost) values: *0.01*, *0.02* and *0.1*, and three value durations: *5,000*, *20,000* and *40,000* are used.

Figure 3-24 clearly shows the dominance of GDP in achieving the highest gain-positive delivery ratio, especially when the forwarding cost is high. The other three protocols maintain a lower than GDP, but reasonable gain-positive delivery ratio when the cost is *0.01*. However, their GPDR degrade when forwarding becomes more expensive. These results justify GDP's forwarding logic from the delivery ratio perspective.



Figure 3-24: Gain-positive Delivery Ratio (GPDR) for a) GDP, b) Epidemic, c) SNW, and d) PROPHET

3.7 Summary and Conclusion

Design details and performance of a new economic gain based DTN routing protocol was presented in this chapter. Economic gain from disseminating content is defined as the generated value upon delivery minus the forwarding costs. It was shown that by adaptively balancing between content dissemination latency and number of forwarding, the proposed protocol GDP is able to achieve higher gain compared to a number of existing DTN protocols with partially similar objectives. Using Experimental results from the simulator ONE [55], it was shown that for certain community based mobility scenarios, GDP can provide near-benchmark gain (i.e., close to the gain upper-bound achieved by RGAP) while protocols such as PROPHET suffer from negative gains when their protocol parameters are not explicitly tuned for target mobility scenarios. It was also shown that in certain scenarios GDP can suffer from worse delay or forwarding cost, but manages

to maintain better gain which is a composite index obtained via combing the former two. As shown in Section 3.6.1 for the community based model, in all the performance measures including delayvs-forward-count, delay-vs-gain, and forward-count-vs-gain, GDP performs closest to the benchmark RGAP compared to all other experimented protocols. Similar trends were also observed for the trace based mobility model obtained from INFOCOM '06 traces [59]. It is noteworthy that such performance advantages of GDP were maintained with varying forwarding cost and value dynamics over time.

One future work direction on this topic can be securing GDP against non-cooperative users, and under various threats such as Sybil [56] and whitewashing attacks [57].

In the following chapter, we extend the concept of gain-aware forwarding with multi-cast dissemination.

4 Economy Driven Multicast Content Dissemination

4.1 Introduction

In this chapter, we present a DTN multicast based content dissemination framework with economic underpinnings. Commercial gain from disseminating a content is defined as the generated *revenue* upon delivery minus the *forwarding costs* incurred for stimulating cooperation among the forwarding nodes by compensating for their resource/energy usage. We introduce a novel concept of *Economy Driven Content Dissemination (EDCD)* using which a *Content Generator (CG)* is able to maximize its economic gain by disseminating content among interested *Content Consumers (CCs)*, and that is after compensating the *Content Forwarders (CFs)*. The general approach in the existing DTN protocols [18, 21] is to use routing utilities that minimize latency, forwarding-count, storage, and in some instances – combinations of the three. Due to the lack of gain-*awareness*, these protocols cannot control the economic gain of dissemination.

EDCD, on the other hand, controls content dissemination based on a composite index, termed as economic gain, which combines *revenue from delivery* and *forwarding cost* from disseminating commercial content such as coupons. Specific contributions of this chapter are as follows. First, we define a consumer interest based dissemination gain abstraction within a practical commercial setting. Second, we propose a new gain-aware multicast Device-to-Device dissemination protocol (i.e., *EDCD*) and its usage within the developed network and commercial model. Finally we evaluate *EDCD* and few other competing protocols using simulation to evaluate their relative dissemination performance.

4.2 Economic Gain of Dissemination

As mentioned in Section 1.11.2, interaction patterns in an SWN can be indexed with social metrics such as contact duration and contact frequency. This information about the underlying social interactions can be leveraged to deliver content such as a coupon. The objective is to deliver coupons through multicast D2D routes so that the economic *gain* of dissemination is maximized. Note that, the terms content and coupon are used interchangeably throughout this dissertation.

4.2.1 Cost, Revenue, and Gain

<u>*Cost*</u>: The cost to a coupon-distributing merchant includes: 1) the product discount cost c_d , which is the discount given to a customer upon *redeeming* a coupon, and 2) the rebate c_f , which should be given to nodes which relay one or multiple coupon(s). This can also be perceived as the cost of forwarding.

<u>*Revenue*</u>: For a coupon-distributing merchant, this quantity represents the net revenue from redeeming copies of a coupon by one or multiple recipients. Net revenue is computed as the earned revenue c_p minus the discount c_d . The net revenue of a coupon redemption by consumer-*I* is then defined as:

$$V_i = P_r(\rho_i)(c_p - c_d) \tag{4-1}$$

 $P_r(\rho_i)$ represents the probability of redeeming a coupon by consumer-*i*. This quantity depends on the consumer's interest level ρ_i (on a coupon), and is elaborated in the next subsection.

<u>*Gain*</u>: Combining *revenue* and *cost*, the expected gain achieved through a coupon redemption by consumer-*I* can be written as:

$$G_i = V_i - c_f \tag{4-2}$$

The total network wide gain G^T can be expressed as:

$$G^T = \sum_{i=1}^{n_d} V_i - n \times c_f \tag{4-3}$$

where coupons are delivered to n_d consumers through n forwards. The maximum number of delivery of a coupon is generally restricted by its generator. That is, $n_d \leq T$, where T is that distribution upper bound set by the coupon generator.

4.2.2 Consumer Interest and Coupon Redemption

Each consumer-*I* maintains an interest level ρ_i ($0 \le \rho_i \le 1$) towards a specific content type. Such interest level for a consumer, can be set through many possible means as described in Section 1.3.1. To preserve privacy, in this model, consumer devices are allowed to share their interest profiles with other devices upon meeting.

<u>Coupon Redemption</u>: Redemption probability for a consumer, represents his/her likelihood of redeeming a coupon within a defined coupon expiry time (See Section 1.3.1). One such example with a maximum redemption probability of 0.5 is shown in Figure 4-1. The figure shows redemption probability graph that decreases exponentially with increase in interest level. A corresponding *Gain* function is also shown in the figure. As can be seen in the figure, for a given redemption function, earned revenue c_p , and discount c_d , the gain G_i can be positive only for a specific range of interest level.

A *desirable target consumer* is defined as a consumer with high enough redemption probability such that the induced net revenue from a coupon-redeemed purchase is higher than the forwarding cost (rebate) for the coupon. Such a delivery induces a positive economic gain G_i . Figure 4-1 shows the range of interest levels for an example definition of *desirable target consumers*.



Figure 4-1: Example redemption function

The presented redemption model is general in that it can accommodate other redemption functions including the ones in which the redemption probability increases with higher interest level. It can also accommodate functions in which the redemption probability maximizes for interest levels that are not too low or not too high. As shown in [14], coupon delivery to highly loyal (interested) customers may kill customer loyalty. Thus, light to moderately loyal customers, as opposed to most loyal ones with higher interest levels, may maintain higher redemption rates. Another example of appropriate target population is the new customers with smaller interest levels. Reports show that new customers spend 67% less with repeat purchases compared to loyal (highly interested) customers [60]. In both examples, the objective is to deliver coupons to the target user class with such optimal interest levels so that the cumulative network wide redemption rate can be maximized. The abstraction used for representing redemption probability $P_r(\rho_i)$ can be used for capturing both the scenarios outlined above.

4.2.3 Consumer Privacy and the Role of Centralized Server

Even though the dissemination process presented in this chapter is fully Device-to-Device, we assume the presence of a centralized server with primary roles as explained in Section 1.3.3.

<u>Privacy preservation</u>: In the presented model it is ensured that the data related to consumer interests and consumer-to-consumer interaction is always stored at the consumers' devices. Such privacy-sensitive data is never exported to the centralized server. This opens up the possibility of economic gain-aware coupon dissemination without a centralized server having to know about consumer mobility and their interests in different content types. This is an attractive aspect of the architecture.

Dissemination Termination: It is possible for the server to take an active role in terminating dissemination in order to control the network-wide economic gain. As shown in Section 4.7, the network wide economic gain generated by coupon dissemination can start declining after it reaches a maximum value. In such situations, the server can terminate dissemination at an appropriate time to ensure near-maximum network-wide gain. Before every coupon delivery, the delivering node checks with the server about the current network wide cumulative gain. When the server observes a declining trend, it puts a stop on the dissemination of that specific coupon. This can result in near-maximum final gain from a dissemination process.

Note that such server-assisted termination does not violate the privacy feature as outlined above. Meaning, the centralized termination can be performed without the server having any access to the consumer interest and interaction/mobility data.

4.3 **Problem Definition**

For given earned revenue c_p , discount c_d , rebate c_f , distribution upper bound T, redemption function $P_r(\rho_i)$, and a social wireless network of consumers with interest profiles ρ_i , the objective is to design a multicast coupon dissemination mechanism that maximizes the total gain G^T within the lowest possible dissemination delay. Consider the example delay tolerant network in Figure 4-2. Each consumer is represented as N-*I* ($P_r(\rho_i)$), where *I* is the node-ID and $P_r(\rho_i)$ indicates its redemption probability computed based on a pre-assigned redemption function applied to its interest level ρ_i . For example, the consumer N-0 (0.8) represents node-0, which is likely to redeem a coupon (of the type in question) with a probability of 80%.



Figure 4-2: Example user interaction and routing

Consider a scenario in which N- θ (θ .8) possesses at least five copies of a coupon that needs to be disseminated such that the overall gain from its dissemination is maximized. There are two possible paths: though N-I (θ .I) and N-2 (θ .8). Given the social interaction pattern it is obvious that dissemination through N-I (θ .I) can provide more revenue even though N-I itself does not contribute too much to the gain because of its relatively low redemption probability. Node N-2(θ .8), on the other hand, while being able to contribute to the revenue significantly, does have relatively worse dissemination potential compared to N-I (θ .I). The gain performance also depends on the forwarding cost c_f . The objective of the protocol is to be able to choose the right forwarding path so that the overall gain is maximized.

4.4 Economy Driven Content Dissemination

Economy Driven Content Dissemination (EDCD) relies on the knowledge of: 1) nodes' interest profiles (i.e., ρ_i), and 2) their underlying social interaction patterns. They key idea is that a node-*I*, carrying *T* copies of a content, first evaluates any of its peer *j*'s quality as a consumer and second, as a potential forwarder. Based on that evaluation it may transfer: 1) a copy of the coupon for *j*'s consumption, and 2) certain number of copies to *j* for further forwarding.

<u>State Variables</u>: Any node-*I* maintains the following variables:

 $f_{i,j}$: *i*'s frequency of meeting node-*j* over a pre-defined time window of τ (e.g., a week)

 δ_i : expected cumulative gain for node-*I* for delivering the content to its one-hop neighbors encountered during time window τ . It can be expressed as:

$$\delta_i = (\sum_{k=1}^{N_i} [(c_p - c_d) P_r(\rho_k) (f_{i,k}) - c_f \times 1])$$
(4-4)

where N_i is the set of nodes interacted by node-*I* during the time window τ . This includes only the interacted nodes that have not already received the content in question. A larger δ_i implies that the nodes that *I* directly interacts with have higher redemption probability, thus yielding higher expected gain.

When *I* meets *j*, it updates $f_{i,j}$ and δ_j . It also receives *j*'s expected gain of delivery to all its non-*I* one-hop peers in the time window τ , which *j* computes based on Equation 4-4.

Forwarding Logic: When a node-*I* with *T* copies of a coupon meets node-*j*, it first evaluates the possibility of *j* being a potential consumer of the coupon based on its redemption probability. Meaning, based on the delivery probability $P_r(\rho_j)/P_r^{max}$, *I* forwards a consumption-copy to *j* and sets $T_{i,j}^c = 1$. With probability $1 - P_r(\rho_j)/P_r^{max}$, $T_{i,j}^c$ is set to zero and no consumption-copy is forwarded to *j*. Normalization by P_r^{max} is done for linearly mapping the delivery probability $P_r(\rho_j)/P_r^{max}$ within the range [0,1]. The redemption probability varies within the range [0, P_r^{max}]. Table 4-1 summarizes all the state variables used in this section.

Notation	Definition				
f _{i,j}	Frequency of meetings between nodes I and j in a preset time window τ				
δ_i	Expected cumulative gain from all <i>i</i> 's one-hop delivery in time τ				
Cp	Net price of the product (earned revenue)				
C _d	Product discount cost				
C _f	Rebate given to nodes which relay the coupon (forwarding cost)				
$T_{i,j}^c$	Number of copies (out of <i>T</i> copies) forwarded for <i>j</i> 's consumption from <i>I</i> to j ($T_{i,j}^c \in [0,1]$)				
$T_{i,j}^f$	Number of copies (out of $[T-T_{i,j}^c]$ copies) forwarded for dissemination through <i>j</i> from <i>I</i> to <i>j</i>				
$P_r(\rho_i)$	Redemption probability of node <i>I</i> with interest level ρ_i				
V_i	Net revenue of a coupon redemption by consumer- I				
G _i	Merchant's expected gain through a coupon redemption by node i				
G^T	Network-wide total expected gain G^T				

Table 4-1: Notations and corresponding definitions in EDCD

After the decision about delivering a consumption-copy to node-*j* is made, if additional copies are left, *I* evaluates *j*'s suitability as a forwarder. This forwarding evaluation is done based on δ_j . First, node-*I* computes the quantity $\delta_j^{norm} = \delta_j / \delta_i^{avg}$, where δ_i^{avg} is the average of all δ_k ($k \in N_i$, the set of all nodes that *I* directly interacted with during the predefined time window τ). A larger δ_j^{norm} shows that *j* is a better forwarder compared to the other nodes encountered by node*i*. Therefore, a larger percentage of the remaining $[T - T_{i,j}^c]$ copies of the coupon should be forwarded to node-*j* for dissemination. A positive δ_j^{norm} is then mapped to an integer $T_{i,j}^f$, which indicates the number of copies to be forwarded by *I* to *j* for dissemination. The value of δ_i^{norm} belongs to the range $\left[\frac{\delta_i^{min}}{\delta_i^{avg}}, \frac{\delta_i^{max}}{\delta_i^{avg}}\right]$, where δ_i^{min} and δ_i^{max} are the minimum and maximum of δ_k respectively $(k \in N_i)$. To map δ_j^{norm} to $T_{i,j}^f$, which is an integer in the range $[0, T - T_{i,j}^c]$, the min-max normalization from [61] is applied. Such linear transformation preserves the relationship between data values from the old range (i.e., $\left[\frac{\delta_i^{min}}{\delta_i^{avg}}, \frac{\delta_i^{max}}{\delta_i^{avg}}\right]$) to the new range $[0, T - T_{i,j}^c]$.

If $\delta_j^{norm} \leq 0$ or if node-*j* already possesses copies of the coupon, then node-I does not need to transfer any copies to node-*j* for forwarding purposes. Meaning the quantity $T_{i,j}^f$ is set to zero. The key components of the described EDCD algorithm are summarized in Algorithm 4-1.

In the above logic it is assumed that a node-*j* always reports correct values of $P_r(\rho_j)$ and δ_j to its peers. This may not be true when consumers act selfishly. Such selfishness is considered outside the scope of this research.

4.5 Compared Protocols

4.5.1 Interest-Aware Dissemination (IAD)

This is a degenerate version of EDCD in which no node-interaction information is used. Content such as coupon is delivered based only on the redemption probability. Similar to EDCD, when a node-*I* carrying *T* copies of a coupon meets node-*j*, $T_{i,j}^c$ is set to *I* with probability $P_r(\rho_j)/P_r^{max}$. However, the number of forwarding copies transferred from node-*I* to node-*j* is determined randomly without using the interaction information $f_{i,j}$, as utilized in the social-aware EDCD. The number of forwarding copies $T_{i,j}^f$ is chosen randomly from a range of $[0, T - T_{i,j}^c]$. To summarize, IAD is expected to show the dissemination performance of when nodes do not register any social interaction information, but are ready to share their own interest level with interacted nodes.

1: When node-*I* meets node-j: *I* updates $f_{i,j}$ and δ_j for node-*j* in its records; 2: **Forwarding Logic:** 3: For node-*I* carrying *T* copies of a coupon; $T_{i,j}^{f} = 0;$ 4: $T_{i,i}^{c} = 0;$ 5: 6: //set number of copies for *j*'s consumption 7: $T_{i,i}^c = 1$ with probability $P_r(\rho_i)$; 8: if $(T - T_{i,j}^c > 0)$ then 9: if (*j* has no copies for dissemination) then //set number of copies to forward to *j* for 10· dissemination $T_{i,j}^{f} = \left| \frac{\frac{\delta_{j}}{\delta_{i}^{avg}} - \frac{\delta_{i}^{min}}{\delta_{i}^{avg}} \times (T - T_{i,j}^{c})}{\frac{\delta_{i}^{max}}{\delta_{i}^{avg}} - \frac{\delta_{i}^{min}}{\delta_{i}^{avg}}} \right| ;$ 11: 12: end if 13: end if 14: Forward $T_{i,j}^c$ and $T_{i,j}^f$ number of copies to *j*, for consumption and dissemination respectively;

Algorithm 4-1: The key components of the EDCD algorithm

4.5.2 Social-aware Opportunistic Routing Protocol (SCORP)

This interest aware protocol [20] works with binary interest levels (i.e., 0 or I). It uses a *social weight*, computed using a node's interest profile and social interaction patterns. Node-*a* computes the quantity $TCTI(a, x)_i$, which is its *Total Connected Time to Interest* (*TCTI*) with nodes with interest-*x* using a daily sample-*i*. The average $ATCTI(a, x)_{i,i}$ is computed for day-*j* as:

$$ATCTI(a, x)_{j,i} = \frac{TCTI(a, x)_{j,i} + (j-1)ATCTI(a, x)_{(j-1),i}}{j}$$
(4-5)

Content is transferred from node-*I* to node-*j*, when node-*j* has consumption interest, or, its social weight (computed based on its *ATCTI*) is higher than that of *i*. We have incorporated the following changes in order to adapt the baseline SCORP [20] with binary interests to the present framework of multi-level interests. The key idea is to let the node interest level ρ_i vary in the range

0 to 1, but to restrict the redemption probability $P_r(\rho_j)$ to binary values of either 0 or 1. $P_r(\rho_j)$ is set to 1 for $\rho_1 < \rho_j < \rho_2$, and zero for other values. The thresholds ρ_1 and ρ_2 are chosen such that the resulting gain from a single delivery is positive, that is, it can offset the forwarding cost c_f . We use a Gaussian redemption function, and a threshold pair ($\rho_1 = 0.63$, $\rho_2 = 0.88$) chosen based on the above strategy.

4.5.3 Random Forwarding

This resembles Epidemic [37] DTN routing which is oblivious to both interest and interaction profiles. When a node-*I* with *T* copies of a coupon meets node-*j*, it forwards $T_{i,j}^f$ copies to node-*j* with a predefined forwarding probability p_f . The number of forwarded copies $T_{i,j}^f$ is chosen randomly between *I* and *T*. The forwarding probability affects dissemination gain dynamics in terms of the maximum gain and the time to reach that maximum value. The scenario with $p_f = 1$ represents Epidemic DTN forwarding. In following set of results we show results for the $p_f = 0.5$ scenario.

4.6 Experimental Details

4.6.1 Interest and Redemption Function

We use the DTN simulator ONE [55] for evaluating EDCD and other protocols with two community based mobility models. The models differ in terms of: 1) number of nodes, 2) community structure, and 3) interest level distribution. For all the experiments we use a redemption function in which the redemption probability maximizes for interest levels that are not too low or not too high. This is motivated by the observation [14] that while improving purchase probability for moderately loyal (interested) customers, coupon delivery may often kill product loyalty for highly loyal customers.

To capture this effect, a Gaussian model for the redemption function is used. Figure 4-3 shows the applied Gaussian redemption function with mean $\mu = 0.75$ and maximum redemption probability $P_r^{max} = 0.5$. The corresponding gain curve in Figure 4-3 is when the forwarding cost c_f , earned revenue c_p , and discount c_d are set to 0.2, 5 and 2 respectively. These values are also used for all subsequent experiments.



Figure 4-3: Example Gaussian redemption function

4.6.2 Target Exposure Index (TEI)

TEI indicates how the desirable target consumers, as defined in Section 4.2.2, are visible to the rest of the nodes in a network. *TEI* is computed as the expected maximum number of contacts with target consumers and is expressed as a per node metric. It is a network property in terms of the mobility and social interaction patterns. Higher *TEI* is expected to yield better gain of dissemination. It should be noted that *TEI* is computed only to characterize the mobility model dynamics and its impacts on dissemination performance for different protocols. There are no protocol syntaxes that rely on this parameter. More about *TEI* computation is presented in the performance section.

4.6.3 Degree of Penetration (DoP)

This is a protocol parameter, which is defined as the maximum depth of a multicast dissemination tree. The root of the tree is the node, which initially receives all T copies of a content from its generator such as a coupon merchant. When the *DoP* is set to one, the source is not allowed to forward any forwarding copies to any of its directly interacted peers. It is allowed to distribute only consumption copies to *T*-*1* peers. In other words, those peers cannot forward the copies, thus maintaining the depth of the tree to be *1*. When the *DoP* is set to 2, only one level of re-forwarding is allowed. Larger *DoPs* allow deeper trees to form, thus, more thorough search can take place in order to find nodes with high redemption probabilities.

4.7 Dissemination Performance

4.7.1 Social Wireless Network with Clique Formation

As shown in Figure 4-4, a *100*-node network is structured in three clique-based communities C1, C2 and C3.



Fig. 4-4 Network with three social cliques (TEI: 21.5)

The numbers in parentheses show the population of each community. The inter-clique arrows indicate the flow of inter-community mobility. For example, *10* members of C2 migrate between C2 and C1, and another *10* members of C2 migrate between C2 and C3. Nodes inside each community meet more frequently than nodes from different communities. Based on the strategy described in Section *4.2.2, desirable target consumers* are designated as those with interest levels

in an approximate range of 0.63 to 0.88 (see Figure 4-3). While such desirable target consumers are deployed within the communities C1 and C3, nodes with lower interest levels (i.e., < 0.6) are inserted in the community C2.

An example real-world representation of this model would be the daily student interactions in a University campus. Students who are more interested in a specific product interact more often compared to those who do not share similar interests. For example, a community of *desirable target consumers* as shown in Figure 4-4, can consist of students who share a common interest in a type of food.



Fig. 4-5: Multicast dissemination with different Depth of Penetration (*DoP*)
Figure 4-5 shows evolution of the average cumulative gain from dissemination for all four
evaluated protocols under different *DoP* scenarios. For all cases, the mobility shown in Figure 44 is used which results in a *Target Exposure Index (TEI*) of approximately 21.5.

TEI indicates how visible the *desirable target consumers* are to all other consumers in the network. The cumulative gains for all the protocols in Figure 4-5 under all *DoP* scenarios have a generally increasing trend before they start falling after reaching a maximum. In line with the method outlined in Section 4.2.3, dissemination is terminated when the moving average (i.e., computed over a half a day window) of the cumulative gain falls by at least x/t % of the attained maximum. The quantity t is the elapsed time since the coupon generation, and x is an empirically chosen parameter that controls termination. For all presented results, the value of x is chosen to be 0.24, which provided consistent performance across all the evaluated protocols.

The results show that among all the evaluated protocols, the proposed EDCD offers the largest maximum gain across all the *DoP* scenarios. Also, the maximum attainable gain for EDCD first increases with increasing *DoP*s (i.e., from *I* to *3*), but then it falls for larger *DoP*s (i.e., larger than *3*). With DoP < 3, the depth of the multicast dissemination tree is not sufficient to reach many desirable target consumers residing in C1 and C3. This explains the need for a sufficiently large *DoP* setting for maximizing the gain. Too large *DoP*s, on the other hand, incur the cost of many unnecessary forwards, which bring the gain down. This is evident from the graphs in Figure 4-5.

The gain of the protocol IAD is worse than that of EDCD because its forwarding component is not social interaction-aware. Also, IAD's performance does not change much with larger penetration levels. The reason is that IAD forwards one consumption-copy along with random number of forwarding-copies to nodes based on their interest level and the resulting redemption probability. For any preset *DoP* level, IAD does not allow the non-consumer nodes (i.e., the migrating nodes between C1 and C2 in this case), to receive any copies for consumption/forwarding. This restricts its ability to attain high economic gain. The protocol SCORP, however, achieves a higher gain as *DoP* increases. This is because it leverages more intermediate nodes to relay the content to farther destinations. For smaller *DoPs*, the extra cost of forwarding is a hindrance to the growth of network-wide gain. Although for large *DoPs*, the large cumulative revenue achieved across a deeper multicast tree offsets the forwarding cost. Observe that EDCD takes longer than all other protocols to reach its maximum gain because all other protocols rely on some form of randomization for the forwarding decisions. EDCD, on the other hand, relies on a specific criterion, which often cause longer forwarding delays, but results in better quality forwards.

Note in Figure 4-5, that the random dissemination reaches its maximum gain earlier than other protocols because of its random forwarding criterion in choosing coupon consumers, which are also the forwarders. Random choices are inherently faster than using a criterion, as done by the other protocols. This fast dissemination, however, comes at a steep price of lower maximum gain compared to the non-random protocols.

<u>Increased Exposure to the Desirable Consumers</u>: In this scenario we increase the network wide visibility of the nodes with higher redemption rates. This is done by imposing more frequent contacts between the members of community C2 and the target consumers of community C3.



Figure 4-6: Mobility for higher target consumer exposure (*TEI*: 37.5) As shown in Figure 4-6, in addition to C2's migrating nodes to C3 and C1, now the members of C3 also occasionally migrate to community C2. These modifications caused an increase in *TEI* to a value of approximately 37.5. Figure 4-7 shows the evolution of gain for different approaches with DoP = 2 and DoP = 5. Observe that EDCD achieves the highest gain compared to the other approaches for both the DoPs. The maximum gain, however, is achieved at a lower DoP (i.e., 2), compared to the cases in the low-*TEI* scenario (DoP: 3) in Figure 4-5:b. The reason is that more exposure of target consumers in this high-*TEI* case allows the nodes with high coupon redemption probability to be reached through a shallower multicast tree.



Fig. 4-7: Gain in scenarios of high TEI

Also, with this higher target consumer exposure, generally it takes shorter to reach the maximum gain. For example, it takes around *11.6* days for EDCD to reach its maximum gain (i.e., *33.7*) across both the *DoP*s (i.e., Figure 4-7:a), while, it takes around *16.6* days to reach its maximum gain of *9.63* when there is less exposure to target consumers (Figure 4-5:b). Higher exposure to target consumers also implies that there are more relay nodes between content carrier(s) in C1 and the part of target consumer population, which reside in C3. According to the low-*TEI* scenario in Figure 4-4, the 10 migrating nodes moving between C2 and C3 are mainly responsible for relaying the content to C3. In the high-*TEI* topology in Figure 4-6, however, all low-interest members in C2 can act as relay nodes in addition to those *10* migrating nodes. The maximum attainable gain of the social interaction unaware protocol IAD does not significantly depend on the target

consumer exposure level. This is because IAD does not forward coupons to non-consumer nodes ignoring any effects of enhanced target consumer exposure in high-*TEI* scenarios. With large *DoPs* (i.e., *5*), the Random dissemination achieves a higher gain (i.e., *6.4*) in the current high-*TEI* scenario when compared to its maximum gain of around *2.6* in the low-*TEI* scenario as shown in Figure 4-5:c. This is because more nodes with high redemption probability are in reach of the carrier nodes in the high-*TEI* scenario.

SCORP's gain increases with an increase in *DoP* as well, because the increase of reachability to the desired targets provided by a large *DoP* (i.e., *5*) compensates for the additional rebate cost incurred by random deliveries. Figure 4-8 shows the forwarding costs in terms of the average cumulative rebate given to intermediate nodes as forwarding incentive. IAD and the Random dissemination incur very similar rebate/forwarding costs in both the *TEI* scenarios. The reason is that the Random method blindly forwards a random number of copies irrespective of the quality of the receivers and level of exposure to target consumers.



Figure 4-8: Total rebate (forwarding cost) per destination

For IAD, even with increased contact frequencies between the members of C2 and C3, content cannot be effectively relayed further to where C2 and C3 meet. This limitation stems from IAD's inability to use the social interaction information for forwarding decisions.

The protocol SCORP incurs slightly higher rebate cost in the high-*TEI* scenario especially for the larger *DoP* (i.e., *5*). This is mainly due to more exposure of target consumers to potential relay nodes from community C2.

The maximum amount of rebate-increase with higher *TEI* happens for the protocol EDCD. It almost doubled across the shown scenarios. The reason is very similar to that for SCORP. Meaning, more members from community C2 maintain high δ_i ($i \in C2$) values (refer to Table 4-1) due to their higher exposure to the members of C3 in the high-*TEI* scenario.

To obtain further insight, we plot the value of the expected gain δ_i in Figure 4-9. It is plotted with time for the average values computed from: 1) nodes in communities C1, C2, and C3 (i.e., denoted as $\overline{\delta_k}$, k = 1,2,3), 2) the migrating nodes between C1 and C2 ($\overline{\delta_{C1:C2}}$), and 3) the migrating nodes between C2 and C3 ($\overline{\delta_{C2:C3}}$).

Figure 4-9:a shows various $\overline{\delta_k}$ values for the lower *TEI* scenario. Note that $\overline{\delta_k}$ can be negative when the expected generated revenue is smaller than the cost of forwarding. Observe that the





quantity $\overline{\delta_{C1:C2}}$ initially rises due to the meetings between the migrating nodes and the members of C1 with high redemption probabilities. As soon as the majority of the desirable target consumers in C1 receive the coupon, $\overline{\delta_{C1:C2}}$ falls and reaches negative values. The more frequently a node meets peers with high redemption probabilities, a higher $\overline{\delta_k}$ is expected. Observe that $\overline{\delta_{C3}}$ is the largest among all $\overline{\delta_k}$ due to more frequent intra-community contacts in C3 among the nodes holding high redemption probabilities.

Figure 4-9:b (the high-*TEI* scenario) shows that $\overline{\delta_{C2}}$ maintains the largest $\overline{\delta_k}$ values since the members of C2 frequently meet nodes with high redemption from communities C1 and C3. A similar *declining* trend is visible for all $\overline{\delta_k}$. This is due to the decrease in number of potential coupon receivers as the dissemination progresses. In other words, as the number of potential coupon redeemers goes down, δ_j reduces. The reason that $\overline{\delta_{C3}}$ and $\overline{\delta_{C2:C3}}$ do not fall in the low-*TEI* scenario in Figure 4-9:a is that a portion of the population in C3 with high redemption *probability* do not receive the coupon within the shown time duration. The fall for $\overline{\delta_{C3}}$ and $\overline{\delta_{C2:C3}}$ happens much earlier in the high-*TEI* scenario in Figure 4-9:b. This is due to faster dissemination in the presence of greater exposure to nodes with higher redemption chances.

Different $\overline{\delta_k}$ dynamics shown in Figure 4-9 across the two *TEI* scenarios explain the performance gap discussed for the two scenarios in Section 4.7.1. The main reason for higher gain achievement in the high-*TEI* scenario is the relatively higher $\overline{\delta_{C2}}$, which implies that the densely populated community C2 provides high-quality relay nodes, even though the nodes in C2 themselves have low redemption probability. Higher $\overline{\delta_{C1:C2}}$ values also show that more coupons can be relayed from C1 to C2 through these intermediate nodes.

To summarize, the presented results demonstrate how EDCD outperforms other evaluated approaches across most of the depth of penetration (DoP) scenarios. To verify the generality of this claim, the protocols are evaluated for another mobility scenario as presented in the following subsection 4.7.2.

4.7.2 Second Mobility Model

In this model 150 nodes are deployed across five clusters.



Figure 4-10: Five clique network: (a): *TEI:27.8*, (b) *TEI:32.1* Figure 4-10:a and Figure 4-10:b show the network topologies with target exposure indices

(TEIs) of 27.8 and 32.1 respectively.

The numbers in parentheses show the community population and the arrows show the flow of migrating nodes between communities. Higher cross-community migrations in Figure 4-10:b result in higher *TEI* compared to the scenario in Figure 4-10:a. Before looking into the overall dissemination performance, in Figure 4-11 we plot the average δ_j for nodes from each community for the low-TEI scenario.



Figure 4-11: Expected gain distribution from one-hop deliveries

Observe how the δ_j values for nodes from the same community form distinct clusters, indicating the similarities in their exposure to desirable target consumers. The δ_j values also show wide variations across different communities. For example, community C2 has the smallest δ_j since its nodes are generally isolated with less contact with target consumers. The situation is similar for the nodes in C1. The communities C3, C4, and C5 show higher δ_j because either their nodes themselves are highly populated with nodes of with high redemption rates, or there are enough nodes that migrate across communities containing desirable target consumers.

Figure 4-12 shows the dissemination performance for both *TEI* scenarios for all the evaluated protocols, namely, EDCD, SCORP, IAD, and Random Dissemination with forwarding probability (p_f) of 0.5. In Figure 4-11:a and Figure 4-11:b, for each protocol there are multiple points, each corresponding to a specific Depth of Penetration (*DoP*).

Observe that for most DoPs in Figure 4-12:a, EDCD generally achieves the maximum gain across all the *DoP*s. The protocol SCORP achieves the second-highest gain across DoPs in the low-*TEI* scenario. Both these protocols rely on some form of social weight based utility for dissemination, which explains their performance advantages compared to IAD and Random dissemination. Protocols that rely on some form of randomness in choosing a relay or number of copies to relay (e.g., Random, SCORP, and IAD) generally reach their maximum gain sooner compared to EDCD. Since IAD only allows forwards to nodes with high-redemption probability, it can take longer to meet such nodes, especially with low exposure to desirable target consumers. That is why in few scenarios IAD takes longer than other protocols to reach its maximum gain.

Carriers hold on to the coupons until consumers are met, and in this case only consumers from C1 and C4 are accessible. In the high-*TEI* scenario (Figure 4-12:b), EDCD maintains its general performance advantage over the other protocols. IAD here does better than SCORP and EDCD in few *DoP* scenarios.



Figure 4-12: Maximum gain and the time to reach maximum This is because the high exposure of target consumers compensates for IAD's limitation of not having a social-aware forwarding policy. Better exposure to the desirable nodes also explains IAD's performance improvement compared to the low-*TEI* scenarios. More specifically, the coupon carriers in C1 in this case meet consumers in C5 directly or through other members of C5 with high δ_j ($j \in C5$) values. This enables IAD to perform good quality forwarding without having to use a social interaction-aware utility. Like in the low-*TEI* case, the random method in high-*TEI* case also reaches its maximum gain in short time duration due to the ready access to higher exposure target consumers.

Generally speaking, all the protocols attain their maximum gain sooner when the *TEI* is high (i.e., in Figure 4-12:b) due to better exposure to the desirable target consumers.

The results demonstrate that the Economy Driven Content Dissemination protocol (EDCD) is able to achieve higher economic gain compared to few other competing protocols under different mobility and *DoP* scenarios. Note that the presented version of EDCD does not self-select the optimal DoP for maximizing gain. The value of DoP needs to be *a priori* dimensioned through simulation for given network parameters. Designing self-adjusting DoP within EDCD is currently being investigated.

4.8 Summary and Conclusion

An economy-driven framework for D2D content dissemination was proposed and evaluated in this chapter. The primary new concept is an *economic gain* of dissemination, which is defined as a composite routing metric. It is constructed by combining the *revenue from delivering a coupon* and the *forwarding cost* of dissemination. Using experimental results from the simulator ONE [55], it was shown that the proposed EDCD can achieve better economic gain compared to a number of competing DTN routing protocols under different mobility scenarios and Depths of Penetration (*DoP*). Experiments also demonstrate that the maximum achievable gain is generally higher in scenarios in which nodes with high redemption rates have higher network wide exposure. Future work on this topic can be: a) methods for self-adjusting *DoPs*, b) learning based framework that can auto-learn mobility patterns of the desirable target consumers for high-gain dissemination, and c) mechanisms for handling consumer selfishness.

EDCD suffers from certain amount of performance loss due to the fact that the disseminating node does not use prediction about the consumption interests of its potential future contacts. In the next chapter, we address that limitation by the way of developing a *predictive gain utility* which is used for deciding relative importance of an immediate peer in comparison with potential future peers.

5 Multicast Routing using Predictive Gain Utility

5.1 Introduction

In previous chapter, we developed the notion of rebate and economic gain of dissemination. The proposed multicast dissemination protocol EDCD (Chapter 4) relies on the consumption interests of nodes that a disseminating node is in direct contact with. While it was shown to perform better than I protocols such as Direct Delivery and Epidemic [37], it suffers from certain amount of performance loss due to the fact that the disseminating node does not use prediction about the consumption interests of its potential future contacts.

The presented work in this chapter addresses that limitation by the way of developing a *predictive gain utility* which is used for deciding relative importance of an immediate peer in comparison with potential future peers. We propose two versions of the utility; the first one, Predictive Gain Utility Routing Individual (PGUR-I) uses node-specific information, and the second one, Predictive Gain Utility Routing Aggregated (PGUR-A) implements a more scalable version by aggregating node interaction information across consumers with similar consumption interests.

The specific contributions in this chapter are as follows. First, we introduce the gain-aware DTN multicast routing utility (similar to Chapter 4) that is designed around the notion of *consumption interest* and *coupon redemption probability*. We propose two versions of a utility based protocol; the first one uses node-specific information, and the second one implements a more scalable version by aggregating node interaction information across consumers with similar consumption interests. Simulation experiments for functional validation and performance evaluation with respect to a benchmark and a comparative protocol are then performed. Finally, sensitivity of the proposed protocol is evaluated for a number of social and commercially

meaningful parameters namely, network mobility, consumption interest patterns, a coupon's expiry time, and the number of distributed copies.

5.2 System Model

Using the same *cost* and *revenue* (Section 4.2.1) formulation as defined in Chapter 4, we compute the expected gain achieved through a coupon redemption by consumer-*I* as:

$$G_i = V_i - c_f \tag{5-1}$$

The total network wide gain G^T can be expressed as:

$$G^T = \sum_{i=1}^{n_d} V_i - n \times c_f \tag{5-2}$$

where coupons are delivered to n_d consumers through n forwards within a pre-specified coupon *expiry time* (τ). The maximum number of deliveries of a coupon and the coupon expiry time are generally restricted by its generator. Also, $n_d \leq T$, where T is that distribution upper bound of a coupon, which is set by the coupon's generator. In the proposed methods in this chapter, only one-hop deliveries are allowed, thus n always equals to n_d .

Similar to the proposed consumers' interest and redemption probability model in Section 4.2.2, each consumer-*I* maintains an interest level ρ_i ($0 \le \rho_i \le 1$) towards a specific coupon type. Redemption function $P_r(\rho_i)$, represents the probability of redeeming a coupon by consumer-*i*. Note again that in the presented model, to preserve consumers' privacy, it is assumed that the data related to consumer interests and consumer-to-consumer interaction is always stored at the consumers' devices. It is never exported to a centralized server.

5.3 **Problem Formulation**

For a given combination of earned revenue c_p , discount c_d , rebate c_f , distribution upper bound T, expiry time τ , redemption function $P_r(\rho_i)$, and a social wireless network of consumers with

interest profiles ρ_i , the objective is to design a multicast coupon dissemination mechanism that maximizes the total gain G^T within the lowest possible dissemination delay.

5.4 **Proposed Protocols**

The proposed Predictive Gain Utility Routing (PGUR) accomplishes the above objective by adjusting its delivery strategy based on 1) the available number of copies of a coupon with the coupon carrier, and 2) the time till the coupon's expiry. The concept is explained using the following example of a four-node network shown in Figure 5-1.



Figure 5-1: Example user interaction and routing

Each consumer in the figure is represented as N_i ($P_r(\rho_i)$), where *I* is the node-ID and $P_r(\rho_i)$ indicates its redemption probability computed based on a pre-assigned redemption function applied to its interest level ρ_i . For example, the consumer N_0 (0.8) represents node-0, which is likely to redeem a coupon (of the type in question) with a probability of 80%. The time by an arrow indicates the time of contact between two nodes. For example, the quantity t_1 corresponds to N_0 's contact time with N_1 . Across the scenarios, source node N_0 with two copies of the coupon (i.e., T = 2) meets the other three nodes such that $t_1 < t_2 < t_3$, $t_4 < t_5 < \tau$, where τ is the expiry time of the coupon in question.

In scenario-*a*, since both N_2 and N_3 have higher redemption probabilities than N_1 , it makes more sense for N_0 to wait and deliver the two copies of the coupon to those two later nodes. In scenario-*b*, based on the redemption probabilities, N_0 should give one copy to N_1 and the other copy to N_2 . This strategy, however, can be implemented when the entire interaction dynamics is known *a priori*, as in oracle-based routings.

In the absence of the knowledge about future interactions, upon meeting N_1 , N_0 can compute the probability of meeting nodes with higher redemption probabilities before the coupon's expiry time. A delivery decision to N_1 can be made based on such expected overall gain. More specifically, if N_0 is expected to meet K number of nodes with redemption probabilities higher than that of N_1 's, and if K is larger than the current number of copies T, then no delivery to N_1 is made. Otherwise, one out of T copies is given to N_1 . The proposed Predictive Gain Utility Routing (PGUR) accomplishes such logic by computing expected gains based on past interaction locality information. It does so in the presence of the constraints on number of available coupon copies and the remaining time to coupon expiry.

5.4.1 Predictive Gain Utility Routing –Individual (PGUR-I)

Predictive Gain Utility Routing –Individual (PGUR-I) relies on the knowledge of: 1) nodes' interest profiles (i.e., ρ_i), and 2) their underlying social interaction patterns. A coupon-carrying node uses this information to evaluate how a delivery to a peer contributes to the overall gain in the coupon's expiry time.

State Variables: Any node-I maintains the following variables with respect to another node-j.

 $\lambda_{i,j}$: *i*'s rate of meeting node-*j* over a pre-defined time window (e.g., a week). This parameter is updated every time the two nodes meet.

 ρ_j : node-*j*'s interest level toward the merchandize corresponding to the coupon. Node-*I* computes node-*j*'s redemption probability $P_r(\rho_j)$ based on the known redemption function corresponding to the coupon.

<u>Forwarding Logic</u>: Consider a situation in which a node-*I*, carrying *T* copies of a coupon, meets another node-*j* that does not already have a copy of the coupon. Node-*I* first uses Eqn. 2 to compute the gain G_i , which node-*j* can achieve through redeeming the coupon.

It also computes the expected gain of delivery to nodes it may visit during the remaining time $(\tau - t_c)$ to coupon expiry, where t_c denotes the current time. Using the recorded frequency of meeting with each node k that I met so far, I computes the probability of meeting k at least once, during $\tau - t_c$ as:

$$p_k^i(\tau - t_c) = 1 - e^{-\lambda_{i,k}(\tau - t_c)}$$
(5-3)

This assumes that the inter-contact duration between node pairs follow exponential distribution. The closer the above probability gets to the coupon expiry time, the lower is the chance of meeting node k. The higher the contact frequency is between node-I and node-k, the more probable it is for I to meet k during $\tau - t_c$. The expected gain from a possible delivery from node-I to node-k can be computed by node-I as:

$$E_k^i(\tau - t_c) = p_k^i(\tau - t_c) \times G_k \tag{5-4}$$

 G_k is the expected gain achieved by a coupon redemption by consumer-*k* as defined in Eqn. 5-1. It is computed based on ρ_k and the corresponding redemption probability $P_r(\rho_k)$.

After node-*I* computes $E_k^i(\tau - t_c)$ for each node-*k* from its past interaction list, it identifies the number of such nodes for which the expected gain of delivery is larger than the gain G_j , which can be achieved by delivering a copy of the coupon to an immediate contact *j*. Those nodes form a set $S = \{ \bigcup k \mid \forall k : E_k^i(\tau - t_c) > G_j \}$, which is the set of target receivers (formed from the past interaction list), whose members individually maintains a higher expected gain compared to the gain generated by delivering to the immediate contact *j*. If $|S| \ge T$, it means that node-*I* is expected to meet more nodes (than it has coupon copies) with expected gain higher than that of the immediate contact node-*j*. In this situation, node-*I* holds onto all its copies and does not make any delivery to node-*j*. If |S| < T, a copy of the coupon is delivered to the immediate contact node-*j*, since the expected number of future contacts with higher gain in this case is less than the available number of copies *T*. Until *T* becomes zero, this process is repeated every time node-*I* meets another node, which does not already have a copy of the coupon. The key components of the described PGUR-I algorithm are summarized in Algorithm 5-1.

This protocol is termed as PGUR-I (or *Individual*) since a node-*I* is required to maintain the state variables about other nodes on a per-node or individual basis. This may give rise to scalability problems, which are addressed in PGUR-A (or *Aggregated*) as presented next.

5.4.2 Predictive Gain Utility Routing – Aggregated (PGUR-A)

Information aggregation in PGUR-A is done by maintaining state variables for node groups in specific *gain quanta* as opposed to for individual nodes.

<u>Gain Quantum</u>: The entire gain range is divided in *L* equal gain quanta. Gain of delivery, as defined in Eqn. 5-2, varies in the range of $[G_{min}, G_{max}]$, where $G_{min} = V_{min} - c_f$ and $G_{max} = V_{max} - c_f$.

1:	When	node-I	meets	node-	j:

2: *I* updates $\lambda_{i,j}$ and ρ_j for node-*j* in its records;

Forwarding Logic:

- 3: $S = \emptyset$; //Set of nodes *k* with expected gain larger than G_i
- 4: For node-*I* carrying *T* copies of a coupon:
- 5: For every node-*k* met by *I* before:
- 6: //compute the expected meeting probability
- 7: $p_k^i(\tau t_c) = 1 e^{-\lambda_{i,k}(\tau t_c)}$
- 8: //compute the expected gain
- 9: $E_k^i(\tau t_c) = p_k^i(\tau t_c) \times G_k$
- 10: if $(E_k^i(\tau t_c) > G_i)$ then
- 11: add the node-*k* to the set *S*;
- 12: **if** (|S| < T) **then**
- 13: Deliver one copy of the coupon to node-*j*;

Algorithm 5-1: The key components of the PGUR-I algorithm

 V_{min} and V_{max} are the minimum and maximum possible net revenues for a delivery. Minimum and maximum achievable revenues can be computed using Eqn. 4-1, as follows:

$$V_{min} = P_r^{min}(c_p - c_d) \tag{5-5}$$

$$V_{max} = P_r^{max}(c_p - c_d) \tag{5-6}$$

where, P_r^{min} and P_r^{max} are the minimum and maximum possible redemption rates for a coupon.

Each gain quantum *l* corresponds to gain values in the range of $\left[\left(G_{min} + \frac{G_{max} - G_{min}}{L}\right) \times (l-1), \left(G_{min} + \frac{G_{max} - G_{min}}{L}\right) \times (l)\right]$. The quantum 1 is represented by the gain ceiling in that quantum (\hat{G}_l) , which is $\left(G_{min} + \frac{G_{max} - G_{min}}{L}\right) \times (l)$. For example, if gain varies in range [0,1] and $L = 10, \hat{G}_l = 0.5$ corresponds to a set of nodes with gain range [0.4,0.5].

Depending on its interest level in a coupon category and the corresponding redemption probability, each node belongs to a specific *gain quantum*. Nodes are logically grouped based on
the quantum they belong to. The number of quanta (L) should be large enough to provide sufficient granularity, so that \hat{G}_l is a fair estimation of individual gains for members in a *quantum*.

State Variables: Each node-I maintains the following variables:

 n_l : No. of nodes in *gain quantum l* that are met by node-*I* so far.

 $\hat{\lambda}_l^i$: *i*'s rate of meeting with nodes in the *gain quantum l* over a pre-defined time window (e.g., a week).

Forwarding Logic: when node-*I*, carrying *T* copies of a coupon, meets node-*j*, which does not possess a copy, the former computes G_j based on Eqn. 5-1 and the interest value shared by node-*j*. Based on *j*'s *gain quantum*, *I* also updates the quantities $\hat{\lambda}_l^i$ and n_l at this stage.

For each *gain quantum l* with $\hat{G}_l \ge G_j$, node-*I* computes the following probability of meeting at least one node in *quantum l* during the remaining time to expiry for the coupon in interest.

$$\hat{p}_{l}^{i}(\tau - t_{c}) = 1 - e^{-\hat{\lambda}_{l}^{i}(\tau - t_{c})}$$
(5-7)

Node-*I* then computes the aggregated expected gain of delivery to nodes in *gain quantum l*, where $\hat{G}_l \ge G_i$:

$$\hat{E}_l^i(\tau - t_c) = \hat{p}_l^i(\tau - t_c) \times \hat{G}_l$$
(5-8)

A cumulative prospective recipient count $\hat{k}_j^i(\tau - t_c)$ is then computed by adding the quantum member count n_l over all the quanta for which the quantity $\hat{E}_l^i(\tau - t_c)$ is larger than G_j .

$$\hat{k}_{j}^{i}(\tau - t_{c}) = \sum_{l=l_{1}}^{l_{2}} n_{l}$$
(5-9)

The quantities l_1 and l_2 correspond to the smallest and largest gain quantum which have higher aggregated expected gain compared to G_j . The quantity $\hat{k}_j^i(\tau - t_c)$ corresponds to the number of nodes (that node-*I* is expected to meet during the remaining time to expiry) that can individually produce larger expected gain compared to G_j . Delivery to this set of nodes is therefore supposed to lead to a higher overall gain.

1: W	hen node- <i>I</i> meets node-j:
2:	<i>I</i> updates $\hat{\lambda}_l^i$ and n_l for node- <i>j</i> in its records;
Fe	orwarding Logic:
3: Fo	r node- <i>I</i> carrying <i>T</i> copies of a coupon;
4:	For each gain quantum <i>l</i> with $\hat{G}_l \ge G_j$:
5:	//compute the expected meeting
	probability
6:	$\hat{p}_l^i(\tau - t_c) = 1 - e^{-\hat{\lambda}_l^i(\tau - t_c)};$
7:	//compute the aggregated expected gain
8:	$\hat{E}_l^i(\tau - t_c) = \hat{p}_l^i(\tau - t_c) \times \hat{G}_l$
9:	if $(\hat{E}_l^i(\tau - t_c) > G_i)$ then
10:	//add the number of nodes in gain
	quantum <i>l</i>
11:	$\hat{k}_{l}^{i}(\tau - t_{c}) += n_{l};$
12:	if $(\hat{k}_j^i(\tau - t_c) < T)$ then
13:	Deliver one copy of the coupon to node- <i>j</i> ;
Algorithm	5-2: The key components of the PGUR-A algorithm

If $\hat{k}_{l}^{i}(\tau - t_{c}) \geq T$, it means that node-*I* is expected to meet more nodes (than it has coupon copies) with expected gain higher than that of the immediate contact node-*j*. In this situation, node-*I* holds on to all its copies and does not make any delivery to node-*j*. Otherwise, a copy of the coupon is delivered to the immediate contact node-*j*, since the expected number of future contacts with higher gain in this case is less than the available number of copies *T*. Until *T* becomes zero, this process is repeated every time node-*I* meets another node, which does not already have a copy of the coupon. The key components of the described PGUR-I algorithm are summarized in Algorithm 5-2.

5.5 Compared Algorithms

<u>Benchmark</u>: This approach is used for finding the gain upper bound when all system parameters including node mobility/interactions are known a-priori. The first-time interactions

between a coupon carrier I and every node-j that it meets till the coupon's expiry time are tracked. The top T interactions in terms of gain from the entire interaction set are chosen for coupon dissemination from node-I to each such node-j. The cumulative gain (G_j) is then computed by adding up the individual gains obtained across all T dissemination. This cumulative value does indicate the upper bound of achievable gain for the given mobility/interaction and interest profiles.

<u>Social-aware Opportunistic Routing Protocol (SCORP)</u>: In order to make SCORP fit the existing framework, we modify this protocol slightly different compared to the modified version used in previous chapter. The interest aware protocol, SCORP [20], works with binary interest levels (i.e., θ or I). It uses a social weight, computed using a node's interest profile and social interaction patterns. Node-a computes the quantity TCTI(a, x)_i, which is its Total Connected Time to Interest (TCTI) with nodes with interest-x using a daily sample-i. The average ATCTI(a, x)_{j,i} is computed for day-j as:

$$ATCTI(a, x)_{j,i} = \frac{TCTI(a, x)_{j,i} + (j-1)ATCTI(a, x)_{(j-1),i}}{j}$$
(5-10)

Content is transferred from node-*I* to node-*j*, when node-*j* has consumption interest, or, its social weight (computed based on its *ATCTI*) is higher than that of *i*. We have incorporated the following changes in order to adapt the baseline SCORP with binary interests to the present framework of multi-level interests. The idea is to let the node interest level ρ_i vary in the range 0to *I*, but to restrict the redemption probability $P_r(\rho_j)$ to binary values of 0 or *I*. In that case, $P_r(\rho_j)$ is set to *I* for all nodes, which induce a positive gain, and it is set to zero otherwise.

This approach does not differentiate between interest levels, which are associated with higher redemption probabilities, and those, which are associated with, lower redemption probabilities. Therefore, it underestimates the achievable gain. To prevent that, we modify SCORP's logic such that the delivery to an interested node-j takes place based on peer's redemption probability normalized between 0 and 1.

5.6 Dissemination Performance

We use the DTN simulator ONE [55] for evaluating PGUR-A, PGUR-I and other target protocols under three different community based synthetic mobility models and one real mobility trace (i.e., from INFOCOM '06 [59]). The main differences across the models are in their community structure and interest level distributions. For all the experiments we use the Gaussian redemption function as explained in Section 4.6.1. Figure 5-2 shows the applied function with mean $\mu = 0.75$ and maximum redemption probability $P_r^{max} = 0.5$. The corresponding gain curve in Figure 5-2 is when the forwarding cost c_f , earned revenue c_p , and discount c_d are set to 0.2, 5 and 2 respectively. These values are also used for all subsequent experiments.



Figure 5-2: Example Gaussian redemption function

5.6.1 Synthetic Mobility

Synthetic mobility follows map-based movements in a network of 10 waypoints. Nodes move between waypoints with assigned transition probabilities. For all experiments,

network size is set to 100 nodes. Each experiment is repeated for around 32 source nodes with highest redemption probability.

5.6.1.1 Monolithic Mobility

Each of the *100* nodes travels from any waypoint to any randomly chosen waypoint with equal probability. Interest levels for nodes are randomly assigned within the range $\rho_i \in [0.62, 0.89]$ following a uniform distribution.



Figure 5-3: Evolution of gain for different expiry time; $T = 10^{-10}$

Figure 5-3 shows evolution of the network-wide cumulative gain over time, beginning from the content generation time on day *16*, which is when the network simulation warms up in terms

of stabilizing the interaction profile. Experiments with ten initial coupon copies (i.e., T = 10) were carried out for expiry time τ ranging from 12 hours all the way up to 10 days.

The benchmark mechanism (i.e., BM), as explained in Section 5.5, shows the maximum achievable gain for all scenarios in Figure 5-3. All the protocols, including benchmark, show an increasing trend for gain over time, as more coupons are delivered.

Figure 5-3 shows that even though the proposed PGUR protocols have different timedynamics, in most scenarios they deliver near-BM final cumulative gain performance. PGUR-A achieves a higher gain compared to PGUR-I in all cases except for the very short expiry time. The reason is as following.

Imagine a node-*I* carrying a copy of a coupon meets node-*j*. In case of an immediate delivery, the induced gain is G_j . The protocol PGUR-A first computes the expected gain of delivery to nodes belonging to gain quantum (see Eqn. 5-8) higher than G_j . This expected gain is a function of the probability of meeting *any* of the qualified nodes in the mentioned quantum. PGUR-I, on the other hand, computes the expected gain of delivery to future contacts based on the probability of meeting *any* node with gain larger than G_j (see Eqn. 5-4). Since, the probability of meeting *any* node belonging to a gain quantum is greater than the probability of meeting *each individual* node with gain of delivery to future contacts is usually higher for PGUR-A compared to that for PGUR-I. Therefore, PGUR-A is more conservative regarding immediate deliveries. This allows PGUR-A to achieve a higher gain by waiting for more qualified consumers with higher gain and not dispensing resources (i.e., coupon copies) rapidly. That is why PGUR-A achieves a higher gain when the expiry time is longer than half-a-day. For short expiry times (*12* hours), however, PGUR-I achieves a higher gain, since generous dissemination to immediate contacts is more efficient under tight time constraints.

Figure 5-3 shows that the other evaluated protocol SCORP performs worse than both PGUR-A and PGUR-I in all the experimented scenarios. The main reason for such performance deficiency is that while dispensing coupons, SCORP does not predict the chances of meeting nodes with higher gain before the coupon expiry time. In contrast, with both PGUR-I and PGUR-A, a carrier often avoids immediate delivery according to the prediction it does about possibility of meeting more qualified receiver nodes. This lack of visibility particularly affects SCORP's gain when the expiry time is long (e.g., 5 or 10 days). It is also the reason why SCORP finishes dissemination in a shorter time compared to PGUR in all scenarios.

Generally, SCORP achieves a higher gain with increase in expiry time initially, because in longer time duration, a larger fraction of the sample space is explored and nodes with higher expected gain are found. Though, for $\tau = 10$ days, a slight decrease in gain is observed. That is because with SCORP a coupon carrier may deliver one or multiple copies to another node, which already has the content. Such deliveries do not generate revenue since no consumptions happen. Those deliveries, however, incur the forwarding cost in the form of rebates. With longer expiry time, more of such deliveries happen, thus resulting in loss of cumulative gain G^{T} .

Figure 5-4 depicts the behavior of PGUR protocols in terms of how they perform immediate deliveries and how they hold on to coupon copies with the anticipation of meeting good prospective recipients. These results correspond to the dissemination pattern for one of the coupon carriers shown in Figure 5-3:d, where T=10 copies and $\tau = 10$ days. The red lines show the change in available number of copies at the coupon carrier over time. *T* starts at *10* and it reaches *0* showing that all copies are disseminated eventually. Upon interacting with node-*j*, node-*I* computes the number of nodes (with individual gains higher than G_i) that node-*I* is expected to meet before the

coupon's expiry time. This quantity, which corresponds to the size of set S in PGUR-I and refers to the value $\hat{k}_i^i(\tau - t_c)$ in PGUR-A's logic, is shown as the blue circles in Figure 5-4.



Figure 5-4: Dissemination behavior of PGUR protocols

The points above the red line show the instances of meetings for which the number of better future consumers (compared to *j*) exceeds the number of available copies. Therefore, no delivery to *j* happens. All *10* points below the red line correspond to instances, where, the number of available copies are more than the number of nodes with potentially higher gain compared to G_j . That is when a delivery takes place. After the last delivery, the number of copies reaches *0*.

With a fixed expiry time in both cases, PGUR-A finishes disseminating all copies on day 25 (Figure 5-4:b), while for PGUR-I dissemination is done by day 19 (Figure 5-4:a). This indicates that PGUR-A is less likely to deliver to immediate contacts. It rather holds on to the content for a longer duration. This phenomenon, usually leads to a higher eventual gain.

5.6.1.2 Community-based Mobility

We experimented with a *Community-based Mobility with Community-based Interest* (*CMCI*) model. Nodes are divided into 4 communities, each with 25 nodes. Members within a community are made to follow the same stochastic mobility by enforcing the same inter-waypoint

transition probabilities as discussed in Section 5.6.1. This ensures higher intra-community meeting frequency compared to inter-community on a node-pair basis. Consumption interest is assigned as follows. The entire redemption probability spectrum is divided into *4* equal segments, each specific to a community. Then the interest levels for nodes in a community are uniformly randomly assigned such that the corresponding redemption probabilities fall into the community-specific segment. This way, the members of one community are similarly interested in receiving coupons for a product.



Figure 5-5: Final gain for all combinations of T and τ for CMCI mobility scenario Experiments with CMCI are performed for a large combination of T and τ . The observed trends of gain evolution over time for all experimented protocols are very similar to those in the Monolithic scenario (i.e., Figure 5-3). The final gain values for all protocols are summarized in

Figure 5-5. Each point corresponds to the final gain achieved by a protocol for a combination of initial number of coupon copies *T*, and an expiry time τ .

Observe that under CMCI, the final gain provided by the benchmark (BM) protocol slightly increases compared to that under the monolithic model, as reported in Figure 5-3.

This is because a coupon carrier's most frequently contacted peers are nodes from the same community with similarly high interest levels and the resulting redemption probabilities. The same observation and explanation also apply to all non-BM protocols.

For all protocols, the final gain increases with higher number of coupon copies. Also, as the expiry time extends, the protocols are able to achieve larger gains, since the chances of meeting nodes with higher redemption probabilities grow. Generally, with longer expiry time, the performance gap between SCORP and the PGUR protocols increases for all *T*'s. The reason is that in a shorter expiry time it is more urgent to deliver coupons to immediate contacts rather than holding on to the coupon to meet better consumers in future. Hence, SCORP's rapid delivery logic is closer to the Benchmark's strategy. For longer contact duration, such strategy hurts SCORP's performance compared to those of benchmark and the other protocols that waits for perspective better receivers.

5.6.2 Trace-based Mobility

All the dissemination protocols were also tested for a human interaction trace collected [59] from INFOCOM '06 conference. The trace contains data from 97 devices including few static ones and those, which were carried around by a group of users for all four conference days. All experiments in this section are run for at least *15* source nodes.

Figure 5-6 shows the maximum achievable gain (Figure 5-6:a) and the time to reach that (Figure 5-6:b) as a function of the coupon expiry time. Observe that the overall comparative

performance across the protocols, as observed for the two previous synthetic mobility scenarios, also applies for this trace based case. As before, the performance of the proposed PGUR protocols lay in between those for benchmark and the SCORP mechanism.



Figure 5-6: Gain dynamics for different expiry time; T=15

Overall, results from synthetic mobility and the trace-based mobility verify the usefulness of PGUR-A in achieving near-benchmark gain various combinations of number of coupon copies and their expiry times. Only for short expiry time, PGUR-A's performance is slightly worse than PGUR-I's. However, considering the scalability advantages of PGUR-A, this small gain compromise can be acceptable. The larger the expiry time is and the more the number of initially available coupon copies, the higher economic gain is provided by the PGUR protocols.

5.7 Summary and Conclusion

We developed two economic gain-aware DTN multicast routing protocols that maximize the economic gain of disseminating coupons. Economic gain was defined as a composite index, which combines *revenue from delivery* and *forwarding cost* from dissemination. Using experimental results from the simulator ONE [55], it was shown that the proposed protocols achieve near-Benchmark economic gain, which is better compared to a competing DTN routing protocol.

Sensitivity of the protocols was evaluated for different network mobility, consumption interest patterns, a coupon's expiry time, and the number of distributed copies.

Future work includes developing: a) learning-based framework for high-gain dissemination, b) a routing framework which uses information regarding nodes' transitive interactions, and c) mechanisms for handling consumer selfishness.

6 MAC Protocol Design Using Evolvable State Machines

6.1 Introduction

Communication protocols can be modeled [62] as collaborative state machines that interact through inter-node message passing. Designing a protocol requires constructing an optimized state machine that usually maintains a balance between multiple performance indices such as communication delay and throughput. Protocol design is considered to be an optimization problem for which rarely there is a closed form solution. Currently, designing network protocols are mostly based on heuristics, which relies on past experience of the designer in known network conditions. While heuristics can work for relatively static networks, a protocol's performance can and do degrade when the network conditions change over time. To cope with such changes, network designers often need to redesign protocols periodically with changing network conditions.

One alternative approach is to leverage learning mechanisms which allow network protocols adapt to the dynamicity of the environment with minimum intervention from the designer. This section explores evolutionary learning, specifically, evolving state machines for protocol synthesis. A specific network protocol is coded as a genotype and its resulting state machine behavior manifests in the form of the corresponding phenotype, leading to specific performance. The genotype or state machine is allowed to evolve under the pressure of desired performance fitness. In this chapter, we explore the evolutionary design of ALOHA [63], a simple yet widely used family of MAC protocols for which the theoretical upperbounds are well known.

Figure 6-1 shows a two-state machine for the Pure-ALOHA logic along with the protocol's attainable throughput for different network-wide load G (Figure 6-1:b). The state machine shows when a node has a packet to send, it simply sends the packet, and subsequently gets back to the wait-for-packet state. Part of the investigation in this chapter will be to explore when the basic

building blocks for the ALOHA state machine are provided to an evolutionary substrate, if and how the desired state machine emerges as a result of an evolution process.



Figure 6-1: (a) Two-state machine for Pure-ALOHA MAC; (b) Network-wide load *G* vs. Throughput *S* (in Erlang) for pure-ALOHA phenotype

In the next chapter, it will be also explored if progressively more complex protocols such as slotted-ALOHA and Carrier Sense Multiple Access (CSMA) can emerge out of an evolution process when time slotting and channel sensing capabilities are added. Specific contributions of the chapter are as follows. First, a probabilistic state transition model is developed for genotype coding of protocol state machines. Second, a genotype to behavioral phenotype modeling mechanism is developed for evaluating the fitness of a protocol genotype for a given network and traffic load. Third, a Java based evolution substrate is developed in which state machines can evolve with user-specified evolution operators such as mutation. Finally, results are presented in the form of evolution and performance trajectories in the presence of varying network and traffic conditions.

6.2 **Protocol State Machine**

We define a generalized state machine (see Figure 6-2), which is modeled after a typical learning process that a baby goes through for making a meaningful conversation. Such learning involves:

a) identifying appropriate time slots to start talking in a competitive environment with multiple speakers, b) ending a sentence in a graceful manner rather than interrupting herself or getting distracted in the middle of a sentence, and c) waiting or distracting herself for subconsciously adjusting the speaking rate in order to minimize interruptions to others. The state machine in Figure 6-2 provides a general framework that embeds those learning components and more. The machine is intentionally kept general with high degrees of behavioral freedom. The hope is that in the presence of such behavioral freedom, unpredictable speaking protocols will emerge under right environmental conditions. In general, speaking a sentence is mapped as sending a fixed size packet. Three states, namely, *Wait* (*W*), *Transmit* (*T*) and *Loitering* (*L*) are defined. *Wait* is a state in which a node remains silent while waiting to receive the next packet to be transmitted from the upper network layer.



Figure 6-2: An evolvable generalized FSM with embedded pure-ALOHA logic

A transmission takes place in the state *Transmit*. The transition from W to T is the only legitimate transition that can initiate a packet transmission. *Loitering* models the distraction phase as outlined in the baby's learning scenario. This state does not actually induce any explicit fitness reward,

though it ensures additional behavioral freedom. Lingering in any of the states T or L can delay possible new transmissions. The labels on the arrows in Figure 6-2 indicate the transition probabilities, and the text within the circle by a transition describes the actions associated with it. Additionally, the machine uses two flags P (Packet flag) and T (Transmission flag). When set, they indicate generation of a packet and an ongoing transmission respectively. It should be noted that while a specific set of transition probabilities in this machine implements the known pure ALOHA protocol, it is general in that it contains many possible execution sequences implementing other working and non-working protocols.

There are six independent transition probabilities as follows. The probability $1 - P_{WW}^{P=0 \rightarrow P=1}$ refers to the probability of a transition from W to T when a packet is generated. In pure-ALOHA, This probability equals to I, representing a node greedily attempting to access the channel as soon as a packet is generated. P_{WL} is the probability of moving to the *Loitering* state, while P_{LW} models the return path from L to W. Arbitrary movements between the states W and L based on the probabilities P_{WL} and P_{LW} control random loitering that will prove to be useful for self-adjusting the effective load from a node. In pure-ALOHA logic, a node returns to state W as soon as a previous transmission is over.

However, in the generalized model, a node can decide between returning to state W with probability $P_{TW}^{T=1 \rightarrow T=0}$ and moving to state L instead, with a probability $1 - P_{TL}^{T=1 \rightarrow T=0}$. Note that a redundant transition to the state L delays possible packet transmissions. Furthermore, a node is allowed to abandon its transmission and move to state W or state L with probabilities $P_{TW}^{T=1 \rightarrow T=1}$ and $P_{TL}^{T=1 \rightarrow T=1}$ respectively. Table 6-1 summarizes the independent transition probabilities and values which would reduce the state machine to that of pure-ALOHA logic in Figure 6-1:a.

$P_{WW}^{P=0 \rightarrow P=1}$:	W to W, when packet is generated, θ
<i>P_{WL}</i> :	W to L without any external event, θ
<i>PLW</i> :	L to W without any external event, θ
$P_{TW}^{T=1 \rightarrow T=0}$:	T to W when an ongoing transmission is over, 1
$P_{TW}^{T=1 \rightarrow T=1}$:	<i>T</i> to <i>W</i> while a packet transmission is in progress, <i>0</i>
$P_{TL}^{T=1 \rightarrow T=1}$:	<i>T</i> to <i>L</i> while a packet transmission is in progress, <i>0</i>

Table 6-1: Independent transition probabilities

6.3 Evolution Platform

The evolution platform (see Figure 6-3) consists of two main components, namely, a Java based evolution fabric and a C based protocol evaluation module. A gene embodies the collection of all the independent transition probabilities. The evolution fabric is responsible for the evolutionary tasks including generation of the initial gene population, sorting fitness values, a tournament selection, mutation, and reproduction [64]. A mapping function is used for mapping each gene to a protocol behavior phenotype code, which is evaluated within the C based evaluation module. The protocol embedded in network nodes runs and induces a network-wide fitness reward. The protocol performance is then fed back into the evolution fabric in the form of gene-fitness.



Figure 6-3: System structure for the proposed evolutionary framework

<u>Fitness</u>: Network-wide throughput is used as the fitness indicator. The hypothesis here is that by the way of effective genotype evolution of the state machine in Figure 6-2, nodes will be able to adjust to the optimum loading conditions in order to achieve the best possible throughput (i.e., of ALOHA) as shown in Figure 6-1:b.

Evolution: Evolution starts with an initial population (with size p) that includes genes consisting of independent transition probabilities from Table 6-1 with randomly assigned values. Note that the extreme probability values (i.e., θ and I) can lead to state pruning and inclusion, which can induce drastic changes in node behavior. For example, a gene which has both probabilities P_{WL} and $P_{TL}^{T=1\rightarrow T=1}$ set to zero, prunes the loitering state L from the corresponding phenotype protocol.

Reproduction is implemented using *tournament selection* [64], which is used in Genetic Algorithm for providing selection pressure (i.e., network throughput) by holding a tournament among a set of selected members, which is 2m% of p in this case. After the tournament selection, m% of p members are selected. Subsequently, the selected m% of the p genes goes through a mutation process to produce a new generation. It was observed that by choosing m around 20%, sufficient genetic diversity can be maintained across generations. The mutation operator of the Breeder Genetic Algorithm proposed in [65] is used. A quantity *Value_i* is mutated as:

$$Value_{i}^{mutated} = Value_{i} + s_{i} \cdot r_{i} \cdot a_{i} , i \in \{1, 2, ..., n\}$$
 (6-1)

where s_i is a random variable $\in \{-1,1\}$, and r_i is defined as $r \times domain_i$. The quantity r is the mutation range with typical values within $[0.1,10^{-6}]$ and $domain_i$ is the domain of the variables. The quality a_i is defined as $2^{-u.k}$ where $u \in [0,1]$ and k is the mutation precision with typical values in range $\{4,5,...,20\}$. Parameter k is associated with the mutation step-size, which is the proximity of a mutated value to its parent individual. Mutation steps are limited to the range $[r,r.2^{-k}]$. Modifying parameters in Eqn. (6-1) provides different search strategies with different step-sizes. Mutating top (most-fit) members of a population is supposed to maintain diversity among members of the new generation and avoid settling in local optimums.

Each member of the new generation models a specific protocol behavior, which gets evaluated within the *C* based evaluation module in Figure 6-3. The evolution process including sorting fitness values, tournament selection, mutation based reproduction, and population replacement are then performed on members of the new generation which consists of m% of p number of children and the p - m% of p members from the previous generation.

6.4 Results

We define an Evolutionary Degree of Behavioral Freedom (*EDBF*) as the number of transition probabilities (in Figure 6-2) that are allowed to evolve in an experiment. In scenarios with smaller *EDBF*s, fewer evolvable probabilities limit the phenotype search space. The gene structure consists of six independent probabilities mentioned in Section 6.2. The population size in each generation is set to 100 in all experiments. Evolution is run for 300 generations with *m* (see Section 0) set to 20%.

6.4.1 Partial EDBF (EDBF: 1, 2, 3)

6.4.1.1 Evolvable Probability: $P_{WW}^{P=0 \rightarrow P=1}$ (EDBF: 1)

 $P_{WW}^{P=0 \rightarrow P=1}$, the probability of remaining in state *W* when a new packet is generated, is the only evolvable probability in these experiments. Figure 6-4:a shows the evolution of average $P_{WW}^{P=0 \rightarrow P=1}$ over 100 members of each generation for scenarios with different network-wide applied load (*G*). Only 50 generations out of 300 are shown since probabilities stabilize by that point.



Figure 6-4: (a) Evolution of $P_{WW}^{P=0 \rightarrow P=1}$ and (b) the corresponding throughput For a given node-specific load g (i.e., G/N), any non-zero $P_{WW}^{P=0 \rightarrow P=1}$ reduces the nodes' effective individual load \hat{g} following the expression $\hat{g}=g(1-P_{WW}^{P=0 \rightarrow P=1})$. In general, $P_{WW}^{P=0 \rightarrow P=1}$ reduces a node's transmission persistence. The network-wide effective load can be expressed as $\hat{G} = G\left(1-P_{WW}^{P=0 \rightarrow P=1}\right)$. For any applied load G, by the way of evolving the probability $P_{WW}^{P=0 \rightarrow P=1}$, the nodes are able to adjust the network-wide effective load \hat{G} to its optimal value 0.5 at which the network-wide throughput is maximized to pure-ALOHA's throughput of 18.4%. This load-reduction effect can be evidenced in the evolution trajectories of $P_{WW}^{P=0 \rightarrow P=1}$ and throughput (when G=0.8 and G=1) in Figure 6-4:a and Figure 6-4:b. The expected steady state value of $P_{WW}^{P=0 \rightarrow P=1}$ is:

$$P_{WW}^{P=0\to P=1} = 1 - \hat{G} / G \tag{6-2}$$

For example, when G is 0.8, $P_{WW}^{P=0\to P=1}$ is expected to evolve to 0.37 (i.e., $1 - \frac{0.5}{G}$), which is evidenced in Figure 6-4:a. When G is set to less than the optimal value 0.5, since no reduction in load is needed, $P_{WW}^{P=0\to P=1}$ evolves to a near-zero value. This helps maintaining the effective load \hat{G} at the same level of the applied load *G*. Figure 6-4:b shows that throughput trajectory with evolving $P_{WW}^{P=0 \rightarrow P=1}$ for the G = 0.3 case settles at the theoretical throughout of 0.16 ($S_{ALOHA} = Ge^{-2G}$). When the applied load *G* is equal or higher than the optimal value 0.5, the throughput is able to reach the theoretical maximum of 18.4%.

Figure 6-5 depicts the trajectory of the effective load \hat{G} averaged over each generation. For all $G \ge 0.5$ scenarios, \hat{G} is adjusted to 0.5 by the way of evolving the transition probability $P_{WW}^{P=0\to P=1}$. However, \hat{G} does not need to be adjusted when the applied load G is less than the optimal value 0.5. In such cases (e.g., G = 0.3), $P_{WW}^{P=0\to P=1}$ evolves to near-zero values such that \hat{G} converges to G.

To summarize, the above results indicate that the generalized state machine in Figure 6-2 can evolve towards the ALOHA state machine in Figure 6-1:a, when genes containing $P_{WW}^{P=0\rightarrow P=1}$ are allowed to evolve under network-wide throughput as the fitness pressure.



Figure 6-5: Trajectory of effective load \hat{G} for different applied load G 6.4.1.2 Evolvable Probabilities: $P_{TW}^{T=1 \rightarrow T=0}$ and P_{LW} (EDBF: 2)

Following a packet transmission, a node returns to state W or L with probabilities $P_{TW}^{T=1 \rightarrow T=0}$ or $1 - P_{TW}^{T=1 \rightarrow T=0}$ respectively. A node moves out of state L with probability P_{LW} . In this set of

experiments, we allow both these probabilities to evolve. Figure 6-6:a through Figure 6-6:d depict the joint evolution trajectory of the pair $P_{TW}^{T=1 \rightarrow T=0}$ and P_{LW} for applied loads G = 0.3, 0.5, 0.8 and 1.0. Each point in these graphs represents the pair $\{P_{TW}^{T=1 \rightarrow T=0}, P_{LW}\}$ and the corresponding throughput. In the beginning, the probabilities change at a fast rate, causing the inter-generation probabilities to be well separated. With advancing generations, the probabilities reach nearconvergence values, forming distinct clusters. For all $G \ge 0.5$, such clusters are formed around the throughput 0.18, indicating evolved ALOHA behavior. For the scenarios with applied load $G \leq 0.5, P_{TW}^{T=1 \rightarrow T=0}$ evolves towards *l*, which is the corresponding transition probability for pure-ALOHA. This can be observed in Figure 6-6:a and Figure 6-6:b. Any non-unity $P_{TW}^{T=1 \rightarrow T=0}$ causes the effective load \hat{G} to be smaller than load G. For $G \le 0.5$, since such load reduction can drag down the throughput, the nodes evolve the $P_{TW}^{T=1 \rightarrow T=0}$ part of their genes towards *l* in order to maintain the throughput at its maximum possible values. P_{LW} increases initially slightly to allow the needed exits from state L. It then evolves to values between 0.6 and 0.8, which can maintain the maximum possible throughput. Figure 6-6:c and Figure 6-6:d depict scenarios with G > 0.5. The initial fall in $P_{TW}^{T=1 \rightarrow T=0}$ helps reducing \hat{G} towards the optimal value 0.5. The fall for the G =1.0 case in Figure 6-6:d is sharper than that for G = 0.8 in Figure 6-6:c. This is because with higher load, a larger reduction is needed and a smaller $P_{TW}^{T=1 \rightarrow T=0}$ provides that reduction.



Figure 6-6: Evolution of pair P^{T=1→T=0}_{TW} and P_{WL}, and the corresponding throughput After the initial fall, P^{T=1→T=0}_{TW} increases and stabilizes at values in the range between 0.7 and 0.8 which, along with the corresponding P_{LW} values, maximizes the throughput around 18%.
The effective load Ĝ in this EDBF= 2 scenario follows a pattern that is very similar to what is shown in Figure 6-5.

Overall, these results indicate that even when the gene structure constitutes two probabilities $P_{TW}^{T=1\rightarrow T=0}$ and P_{LW} (i.e., EDBF = 2), the generalized state machine in Figure 6-2 can evolve towards the pure-ALOHA behavior as depicted in Figure 6-1:a.

6.4.1.3 Evolvable Probabilities: P_{WL} and P_{LW} (EDBF: 2)

Figure 6-7:a through Figure 6-7:d demonstrate the trajectory of the probabilities P_{WL} and P_{LW} over 300 generations. For small *G* values 0.3 and 0.5, P_{WL} approaches zero, indicating that the nodes rarely digress from the desirable packet transmission flow.



Figure 6-7: Evolution of pair P_{WL} and P_{LW} for different network-wide loads In other words, due to almost zero entries into the *L* state, the system becomes insensitive to the probability P_{LW} . To maintain a balanced effective load \hat{G} , P_{LW} slightly increases (Figure 6-7:b) as a response to a slight decrease in P_{WL} . Due to traffic overload (i.e., higher *G*), P_{WL} increases in Figure 6-7:c as well as in Figure 6-7:d. Higher probability of entering to state *L* imposes longer intervals between packet transmissions; thus a decrease in \hat{G} .

6.4.1.4 Evolvable Probabilities: $P_{TL}^{T=1 \rightarrow T=1}$, $P_{TW}^{T=1 \rightarrow T=1}$, and P_{LW} (EDBF: 3)

Figure 6-8:a through Figure 6-8:d show the trajectory of the triple $P_{TL}^{T=1 \rightarrow T=1}$, $P_{TW}^{T=1 \rightarrow T=1}$ and P_{LW} for 300 generations. The sparse points show the start of the evolution. As probabilities evolve and converge to optimum values, dense clusters form.



Figure 6-8: Evolution of $P_{TL}^{T=1\rightarrow T=1}$, $P_{TW}^{T=1\rightarrow T=1}$ and P_{LW} for different load conditions The transmission abandoning probabilities (i.e., $P_{TL}^{T=1\rightarrow T=1}$ and $P_{TW}^{T=1\rightarrow T=1}$) approach zero when *G* is small, i.e., 0.3, 0.5. In fact, smaller abandoning probabilities enforce full use of sending potential, which is required for small *G* values. However, for larger *G*, 0.8 in Figure 6-8:c and 1.0 in Figure 6-8:d, the system learns to leverage larger $P_{TL}^{T=1\rightarrow T=1}$ and $P_{TW}^{T=1\rightarrow T=1}$ to abandon redundant transmissions to adjust the effective load \hat{G} to 0.5. P_{LW} , however, declines for larger *Gs*.

The reason is that P_{LW} facilitates returns from L to W and aids subsequent packet transmissions. Thus, lower values of P_{LW} , acting as a \hat{G} -reducer, emerge as network traffic increases.

6.4.2 Full *EDBF* (*EDBF*: 6)

Full *EDBF* is accomplished by letting all six transition probabilities to evolve. Figure 6-9 depicts the evolution of all six probabilities for loads *G* set to 0.3, 0.5, 0.8 and 1.0. Observe that P_{WL} increases for larger *G* (i.e., 0.8 and 1.0). The reason is that lager P_{WL} values cause more frequent loitering (*L*), which delays packet transmissions, thus causing reduction in effective load \hat{G} . Large $P_{WW}^{P=0 \rightarrow P=1}$ can also be seen in large *G* scenarios.

For all the full-*EDBF* scenarios a network is able to adjust its effective load \hat{G} to 0.5, leading to the target network throughput of 0.18. These full-*EDBF* results demonstrate that the generalized state machine in Figure 6-2 is evolvable towards pure-ALOHA even when all its independent transition probabilities are allowed to evolve.

6.4.3 Protocol Adaptation

6.4.3.1 Adaptation with Changing Traffic Load

In a 100-node network, load G is runtime altered in four instances. Starting at 0.3, G is changed every 50 generations to values 0.8, 1.0, 0.5 and 0.8. Each stage is run for 50 generations. The results in Figure 6-10 correspond to an *EDBF-2* case in which the independent probabilities $P_{TW}^{T=1\rightarrow T=0}$ and P_{LW} are set to evolve.

When G = 0.3, both probabilities converge to large values (i.e., 0.8 through 1) in order to use the full transmission capacity via expedited packet transmissions. As a result, after approximately 13 generations, the throughput reaches the maximum value for pure-ALOHA, which is 0.16 (i.e., for G = 0.3). As G is changed from 0.3 to 0.8 after 50 generations, $P_{TW}^{T=1 \rightarrow T=0}$ drops in order to decrease the effective load \hat{G} (i.e., towards 0.5) by the way of limiting the number of transitions to state W following a transmission. P_{LW} reduces as well to enforce the same effect. Observe how by adapting those probabilities, the system is able to reach back to the pure-ALOHA throughput after it had dropped immediately due to the load transition from 0.3 to 0.8.



Figure 6-9: Evolution of all six independent transition probabilities for full EDBF The throughout stabilizes back to the desired value roughly after 20 generations since the load transition. When *G* changes from 0.8 to 1.0, a similar drop in probabilities can be observed. When *G* reduces to 0.5 from 1.0, the probabilities increase again to facilitate more packet transmissions and finally, a fall is observed as load increases in the final transition from 0.5 to 0.8.

As marked in the figure, the throughput changes and settles near the pure-ALOHA throughput of *0.18* after each of the transitions. Overall, along all the load change points, the system is able to adapt transition probabilities in order to reach and maintain the desired pure-ALOHA throughput.



Figure 6-10: Adaptation of probabilities with runtime changes of network load6.4.3.2 Adaptation with Changing Network Size

The probability P_{LW} (in Figure 6-11) evolves commensurately to adapt the state machine towards the maximum throughput of 18%. As N transitions to 200, the overall load is increased.

As an attempt to bring \hat{G} down, P_{WL} and P_{LW} further evolve for maintaining the maximum throughput. When N reduces to 100, there is a need for increasing P_{LW} . It is needed to achieve a higher packet transmission rate which the less-crowded network can accommodate while maintaining the effective load \hat{G} at 0.5. Maximum P_{LW} and minimum P_{WL} values are observed after the next network size transitions to 25. Finally, when the number of nodes is increased back to 200, the probabilities evolve and converge to similar values as in the second stage in order to reduce the effective load to the optimal level. To summarize, like in the variable loading case, the system is able to adapt transition probabilities in order to reach the desired ALOHA throughput. These adaptation runs were carried out for all combinations of partial *EDBF* (Section 6.4.1) as well as full *EDBF* (Section 6.4.2) scenarios, in which similar adaptabilities were observed.



Figure 6-11: Adaptation of probabilities with runtime changes of network size

6.5 Summary and Conclusion

An evolutionary based learning framework was proposed in this chapter for MAC layer protocol synthesis. A generalized probabilistic state machine, a subset of which can mimic the behavior of pure-ALOHA MAC, was defined. Through simulations it was verified that a desired state machine with near-benchmark (pure-ALOHA) performance can evolve from that generalized state machine. This was investigated with static and adaptive implementations when both limited and full behavioral freedom were granted.

In the next chapter, we add sensory capabilities such as time slotting and carrier sensing, and investigate if more complex protocols such as slotted-ALOHA and Carrier Sense Multiple Access (CSMA) can be evolved progressively using the same framework. We then propose a different learning approach, namely, reinforcement based learning to the problem of communication protocol design, specifically, the multicast content dissemination in Social Wireless Networks.

7 MAC Protocol Evolution with Time Sensing and Channel Sensing Modalities

7.1 Introduction

Building on the research in previous chapter on ALOHA protocol evolution, we extend the core concepts for the next natural steps towards Slotted ALOHA and CSMA logic. Pure-ALOHA logic can be formulated with a two-state machine including a wait-for-packet and a transmit state. If global time sensing ability with fixed packet sized slot granularity is added to Pure-ALOHA's logic, the behavior of Slotted-ALOHA (i.e., S-ALOHA) emerges which doubles the protocol's attainable throughput (i.e., from 0.18 to 0.36). To model S-ALOHA's behavior, ALOHA's state machine can be equipped with an extra state indicating a node buffering a packet until the beginning of the next designated slot. Adding the channel sensing modality to ALOHA's state machine results in a three-state CSMA machine in which a node keeps listening to the channel in "wait for free medium" state, and transmits the packet as soon as it finds the channel to be idle. Increasing throughput under various loading conditions demonstrates how adding different node capabilities can transform the Pure-ALOHA logic to more efficient and complex protocols such as S-ALOHA or CSMA. In previous chapter (Chapter 6), it was verified that protocol with nearbenchmark pure-ALOHA performance can emerge from that generalized state machine as a result of the evolution process. In this chapter, we explore if and how more complex protocols such as S-ALOHA and CSMA can emerge when time slotting and channel sensing capabilities are added to the network nodes.

The contributions of this chapter are as follows. First, building on the preliminary concepts developed in previous chapter, the proposed evolutionary protocol synthesis concepts are applied for networks of nodes with global time or carrier sensing capabilities. Second, the mechanism's feasibility is evaluated with varying degrees of evolutionary behavioral freedom. Finally, protocol

adaptations are experimented with when node capabilities, network size, and loading conditions are changed runtime, leading to dynamic protocol synthesis. The objective of this specific chapter is to study the feasibility of Genetic Algorithm (GA) based design approach for simple protocols with known performance bounds.

7.2 Evolving Protocol State Machines

Two state machines, one with global time access (i.e., slotting) and the other with carrier sensing modality are defined.



Figure 7-1: An evolvable generalized FSM with time sensing capability

7.2.1 Machine with Time Sensing Modality

A generalized state machine that can produce arbitrary protocol behavior is shown in Figure 7-1. The machine is modeled after a possible process that a baby goes through to learn to talk at designated intervals rather than talking in between intervals: a) identifying designated intervals and talking in the beginning of such intervals in a competitive environment with multiple speakers,

b) ending a sentence in a graceful manner rather than interrupting herself or getting distracted, andc) intentionally waiting or unintentionally distracting herself for adjusting the talking rate in order to minimize interruptions to others.

The state machine in Figure 7-1 is intentionally kept general with high degrees of behavioral freedom. The hypothesis here is that when the state machine is evolved under the right fitness pressure (e.g., network throughput), optimal communication protocols will emerge for given network and loading conditions. Since the nodes are allowed to sense global timing (i.e., time slotting) and the S-ALOHA logic is a subset of this machine, the evolved solutions are expected to have performance close to S-ALOHA. The state machine consists of four states, namely, *Wait* (W), Wait-With-Packet (WWP), Transmit (T), and Loitering (L). In Wait, a node silently waits to receive the next packet to be transmitted. A transmission takes place in the state *Transmit*. A node can initiate a packet transmission by transitioning from W to T, or by going through W-WWP-T which models deferring transmission to the beginning of the coming slot. In the latter, the node transitions to the *WWP* state while buffering the packet and waiting for the appropriate time slot for transmission. Loitering models the distraction phase as outlined in the baby's learning scenario. This state is different from intentional *Wait* in that transition from any other state to *Loitering* is arbitrary and without any explicit strategy. As a result, it does not induce any explicit fitness reward. Lingering in any of the states T or L can delay possible new transmissions. The labels on the arrows indicate the transition probabilities. The notation P_{XY} indicates the transition probability from State-*X* to State-*Y*.

The machine uses two flags P (Packet flag) and T (Transmission flag). When set, they indicate generation of a packet and an ongoing transmission respectively. Texts within rectangles describe the action(s) associated with each state transition. While a specific set of transition probabilities in

this machine implements the known S-ALOHA protocol, it is general in that it contains many possible execution sequences implementing other working and non-working access protocols. The state machine in Figure 7-1 contains 8 independent transition probabilities listed in Table 7-1 (i.e., parts a and b). P_{WL} is the probability of moving from W to L; P_{LW} models the probability of the return transition. Transitioning between the states W and L based on those two probabilities control random loitering which proved to be useful towards self-adjusting traffic load for network-wide throughput maximization. According to the state machine, after a transmission is over, a node can decide between returning to state W with probability $P_{TW}^{T=1\to T=0}$ or moving to state L with probability $1-P_{TL}^{T=1\to T=0}$. Note that a redundant transition to the state L delays possible future packet transmissions. Furthermore, a node is allowed to abandon its transmission and move to state W or state L with probabilities $P_{TW}^{T=1\to T=1}$ and $P_{TL}^{T=1\to T=1}$ respectively.

To accommodate global time sensing, 3 additional probabilities are included as follows. P_{SA} , namely, the probability of agreeing with slot status, refers to probability of deferring a packet transmission to the beginning of the next slot. Note that, even after a deferral, packet transmission is not guaranteed. It depends on the probability $1 - P_{WW}$, which refers to the probability of a transition to state T when a packet is available. This transition takes place either through W, or through WWP depending on the P_{SA} value. When $P_{SA} = 0$, the state machine becomes insensitive to global time sensing (i.e., time slotting). In that case, the best evolvable solutions for the generalized state machine is expected to be the pure-ALOHA behavior [63]. P_{WWPL} is the probability of arbitrary transitions from WWP to L during which the node loses the chance to transmit the packet on the scheduled time slot and drops the packet as entering the L state. Note that at the end of each row in Table 7-1, the probability value for which the state machine reduces to that of the most likely solution S-ALOHA is mentioned.

7.2.2 Machine with Channel Sensing Modality

The next functional augmentation is to develop a state machine that incorporates channel sensing abilities. When mapped to the model for a baby's learning process to talk (Section 7.2.1), this machine enables the additional option for not starting to speak when someone else is speaking. The new hypothesis here is that when such a state machine (i.e., in Figure 7-2) is allowed to evolve under the right fitness pressure, the evolved solutions are expected to have performance close to that of CSMA. Two new transition probabilities, namely, P_{CS} and P_{ACS} , are added. While P_{CS} refers to the probability of channel sensing, P_{ACS} indicates the probability of a node respecting the status of the channel when it is actually sensed. In other words, CSMA logic can be produced when both P_{CS} and P_{ACS} are set to 1 in a reduced state machine mimicking CSMA logic.

(a) Common transition probabilities:		
P_{WL} : From W to L without any external event, θ		
P_{LW} : From L to W without any external event, θ		
$P_{TW}^{T=1 \rightarrow T=0}$: From T to W after a transmission is over, 1		
$P_{TW}^{T=1 \rightarrow T=1}$: From T to W during a transmission, 0		
$P_{TL}^{T=1 \rightarrow T=1}$: From T to L during a transmission, 0		
(b) Probabilities to accommodate global time sensing		
P _{WW} : From W to W/WWP, when packet is available, 0		
P_{SA} : Agreement with slot status, postponing packet transmission		
procedure to the beginning of next slot, I		
<i>P</i> _{<i>WWPL</i>} : From <i>WWP</i> to <i>L</i> , without any external event, 0		
I Probabilities to accommodate channel sensing		
$P_{WW}^{P=0 \rightarrow P=1}$: From W to W, when packet is generated, θ		
P _{CS} : Sensing the channel before deciding on transmission, 1		
P_{ACS} : Agreement with channel status; postpone transmission when channel		
is occupied, 1		

Table 7-1: Independent transition probabilities for both state machines

If such CSMA logic is to emerge through evolution, it is expected that a node would first learn to sense the channel by the way of evolving P_{CS} , and then would learn to respect the channel status by evolving P_{ACS} . The quantity $1 - P_{WW}^{P=0 \rightarrow P=1}$ refers to the probability of a transition from W to T when a packet is generated. The rest of the transition probabilities have similar function as described in the generalized state machine in Figure 7-2 for evolving S-ALOHA. These three probabilities are added in Table 7-1.



Figure 7-2: An evolvable generalized FSM with channel sensing capability

All eight independent probabilities for the state machine in Figure 7-2 are included in parts a and c of Table 7-1. At the end of each row, the probability value for which the state machine reduces to that of the most likely solution CSMA is mentioned. Table 7-2 describes how each transition from W to T can possibly happen in the state machine shown in Figure 7-2.

$$\begin{array}{c} 1 - P_{WW}^{P=0 \rightarrow P=1}\left(P_{CS}\right)\left(P_{CF}\right)\left(P_{ACS}\right): \ W \ {\rm to} \ T, \ {\rm packet} \ {\rm is} \ {\rm to} \ {\rm be} \ {\rm transmitted} \ {\rm while \ channel} \ {\rm is} \ {\rm sensed} \\ {\rm as} \ {\rm free} \\ 1 - P_{WW}^{P=0 \rightarrow P=1}(P_{CS})(1-P_{CF})(1-P_{ACS}): \ W \ {\rm to} \ T, \ {\rm packet} \ {\rm is} \ {\rm to} \ {\rm be} \ {\rm transmitted} \ {\rm although} \\ {\rm channel} \ {\rm is} \ {\rm sensed} \ {\rm as} \ {\rm occupied} \\ 1 - P_{WW}^{P=0 \rightarrow P=1}(I-P_{CS}): \ W \ {\rm to} \ T, \ {\rm packet} \ {\rm is} \ {\rm to} \ {\rm be} \ {\rm transmitted} \ {\rm although} \\ {\rm transmitted} \ {\rm transmitted}$$

Table 7-2: Possible transitions from W to T in the machine in Figure 7-2

7.3 Evolution Architecture

7.3.1 Gene Architecture

As shown in Figure 7-3, the system consists of a Java based evolution fabric and a C based protocol evaluation module. A gene embodies the collection of all the independent transition probabilities described in Table 7-1. The upper part of the gene contains the common probabilities from Table 7-1: part a, and the lower part contains either the probabilities for incorporating global time sensing (from Table 7-1: part b) or carrier sensing (from Table 7-1:part c). Note that a gene under this formulation is used for protocol state machine evolution either with time sensing or with carrier sensing, but not both of them together. The evolution fabric is responsible for the evolutionary tasks. A mapping function is used for mapping each gene to a protocol behavior phenotype code, which is evaluated within the C based evaluation module for given target networks. The protocol performance (i.e., throughput) is then fed back into the evolution fabric in the form of gene-fitness values.

7.3.2 Fitness Formulation

Network throughput is chosen as the fitness indicator for the evolution experiments since the evolved solution for the state machines in Figure 7-1 and Figure 7-2 are expected to be close to S-ALOHA and CSMA, and the maximum possible throughput for these protocols are well-known.


Figure 7-3: Genotype formation and the system components

7.3.3 Reproduction and Evolution

The reproduction and evolution processes follow the ones explained in Section 0.

Evolution starts with an initial population (with size p) with randomly assigned probability values to the gene structure shown in Figure 7-3. First, the *tournament selection* process is applied on 2m% of the most-fit individuals within a generation. Through experiments and our prior experience (Chapter 6), it was observed that by choosing m around 20% sufficient genetic diversity can be maintained across generations. The mutation operator of the Breeder Genetic Algorithm proposed in [65] is used. Finally, m% of the least-fit members of the population is replaced. Thus, children (i.e., mutated members in the last step), substitute the m% least-fit members. Each of the new members model a specific protocol behavior, which gets evaluated within the C based evaluation module in Figure 7-3. The evolution process including sorting fitness values, tournament selection, mutation based reproduction, and population replacement are then performed on members of the new generation which consists of m% of p number of children and the p - m% of p members from the previous generation.

7.4 Results

Similar to Section 6.4, Evolutionary Degree of Behavioral Freedom (*EDBF*) is defined as the number of transition probabilities that are allowed to evolve in a given experiment. In scenarios with smaller *EDBF*s, fewer evolvable probabilities limit the phenotype search space. The population size and network size are both set to *100* unless stated otherwise.

7.4.1 Evolution with Global Time Sensing Enabled

7.4.1.1 Limited Evolutionary Degree of Behavioral Freedom (i.e.,:1,2)

Evolution of the state machine from Figure 7-1 in these experiments starts with P_{WW} as the only evolvable probability (i.e., *EDBF* is set to 1). After 50 generations, the value of *EDBF* is bumped up to 2 by including P_{SA} (i.e., enabling slotted time sensing) in the evolvable set. Figure 7-4 shows the throughput evolution for different network wide loading conditions. For the first 50 generations when time slot sensing is not enabled, the converged throughput indicates that the generalized state machine from Figure 7-1 evolves to yield the same throughput of pure-ALOHA protocol. Pure ALOHA's throughput is expressed as $S_{ALOHA} = G e^{-2G}$, which is maximized to 0.18 when G =0.5. As can be seen in Figure 7-4, for G = 0.5, the fitness stabilizes at around 0.18 while for G =0.1, it stabilizes at 0.08 which is correct for pure-ALOHA. An interesting observation is that for G > 0.5 cases, the throughput is still maintained at the maximum value of 0.18.

It turns out that the system evolves to adjust the P_{WW} value such that the nodes drop packets, and the resulting effective network wide load is adjusted down to 0.5 for maintaining the maximum throughput.



Figure 7-4: Fitness dynamics before and after time sensing is added

After 50 generations, when time slotting is enabled (i.e., by making the probability P_{SA} evolvable), the fitness of the system stabilizes at larger values for all *G* values. As expected, the fitness in this case indicates the emergence of S-ALOHA, with throughput $S_{S-ALOHA} = G e^{-G}$. It has the maximum value of 0.36 for G = 1. The graph in Figure 7-4 shows how S-ALOHA performance is maintained for all G < 1, and it is maintained around the maximum value of 0.36, for $G \ge 1$ by the way of adjusting the effective load. It is observed that for all $G \ge 0.5$, the effective load \hat{G} converges at 0.5 before time sensing is enabled after 50 generations. This is because the state machine evolves to Pure-ALOHA for which the fitness/throughput is maximized when the applied load is 0.5. However, for G < 0.5, \hat{G} maintains the same value as G. After time sensing is enabled, the state machine evolves to S-ALOHA for which the maximum fitness happens at the effective load of 1. This explains as to why for all $G \ge 1$, the effective load \hat{G} converges around 1. For G < 1, \hat{G} remains same as G. Figure 7-5 reports the genotype dynamics in terms of the evolvable probabilities P_{WW} and P_{SA} . The evolved state machine in Figure 7-1

indicates how any non-zero P_{WW} causes packet drops and resulting load reduction. For G = 0.1and G = 0.5 cases, before time sensing is enabled (i.e., P_{SA} is made evolvable), P_{WW} evolves to 0.



Figure 7-5: Genotype dynamics before and after time sensing is added

This is because the load is already less than or equal to the optimal load 0.5, thus no load reduction by packet drop is needed. After time sensing is added, P_{SA} evolves to 1, which represents transmission compliance with time-slotting at the emergence of the S-ALOHA logic. The value of P_{WW} remains zero since load values of 0.1 and 0.5 are still less than the optimal load 1 for S-ALOHA. For G = 1, however, P_{WW} initially evolves towards 0.5 which reduces the effective load \hat{G} towards the optimal pure-ALOHA load 0.5. After time slotting is enabled, much like the previous two loading scenarios, P_{SA} evolves to 1 for ensuring full transmission compliance with time slotting. After time sensing is enabled, P_{WW} evolves towards zero since the load G = 1 now becomes equal to the S-ALOHA optimal load 1, thus eliminating the need for any load reduction through packet drops. The dynamics is different for higher loads when G = 2. In this case P_{WW} first evolves to a larger value of 0.75 due to the need for a greater load reduction. After time sensing is enabled, the need for load reduction reduces due to the higher optimal G of S-ALOHA. As a result, P_{WW} reduces and finally converges towards a value of around 0.27. The P_{SA} dynamics remain similar to the lower loading scenarios.

7.4.1.2 Full Evolutionary Degree of Behavioral Freedom (EDBF: 8)

Results in this Section correspond to the evolution of the state machine with enabled time sensing (i.e., in Figure 7-1) when all its eight independent probabilities are allowed to evolve.



Figure 7-6: Evolution of all probabilities in different loading conditions

Experiments show that even when the evolution is completely unguided, meaning with the maximum degree of freedom, the system is still able to evolve towards the emergence of slotted ALOHA logic with its maximum throughput of around *36%*. This further validates the ability of this framework to synthesize optimal MAC protocols when time sensing is enabled.

Figure 7-6 shows the evolution of all eight probabilities for G = 1.0 (Figure 7-6:a) and G = 3.0 (Figure 7-6:b). Observe that under the higher load of 3.0, the probabilities are set to regulate the effective load toward the optimal load 1.0. E.g., $P_{TW}^{T=1\rightarrow T=0}$ and P_{WWPL} decrease to interrupt

the packet transmission process by not returning to W state immediately following a successful transmission in first case, and discarding the packet before the scheduled time slot for transmission is reached in the second case. $P_{WW}^{P=0\rightarrow P=1}$ and P_{WL} grow larger when G = 3.0 to maintain the same effect, that is to regulate the effective load \hat{G} .

7.4.1.3 Protocol Adaptation

One of the primary motivations for evolution based protocol synthesis is to incorporate the ability for self-adaptation when node capabilities and network environment are changed. After establishing the baseline abilities to synthesize protocols using evolution, we investigate their adaptability in this section. Adaptation is investigated with: a) varying traffic load, b) turning global time sensing on and off, and c) various degrees of guidance in terms of EDBF. Figure 7-7 depicts adapted throughput values for a 100-node network for which the MAC protocol is evolved for 300 generations. Global time sensing is enabled only for the second half of the experiment for 150 generations. Six distinct phases, each for 50 generations, are executed. Parameters that change across consecutive phases are underlined in the figure. The corresponding genotype dynamics in terms of up to three independently evolving probabilities (i.e., EDBF = 3) is shown in Figure 7-8. Starting with EDBF-1 in first 50 generations, P_{WW} evolves to around 0.75 to adjust the load G = 2.0 down to the optimal load 0.5, resulting a near-ALOHA throughput of 0.18. During generations 50-100, EDBF changes to 3 as two additional probabilities (i.e., P_{WL} and P_{LW}) are allowed to evolve. In addition to that, G is reduced to 0.5. Since there is no need for further reducing the effective load, P_{WW} converges to near-zero values. The three evolving parameters maintain a balance such that the maximum throughput of 0.18 is achieved in this

second phase. In the third phase, as G increases to 1.0, P_{WL} increases to help reducing the effective load.



Figure 7-7: Throughput with runtime parameter changes



Figure 7-8: Adaptation of probabilities with runtime parameter changes

Starting from generation 150, global time sensing is enabled and EDBF=1, thus allowing P_{WW} to regulate the effective load by converging to values around 0.27. The corresponding throughput in Figure 7-7 converges to 0.36 which proves that evolved solutions mimic the S-ALOHA benchmark's behavior. For the next 50 generations, G is reduced to 0.5 and EDBF=3.

Probabilities evolve such that the effective load is maintained near 0.5, at which the state machine induces a throughput of around 0.30, which is in agreement with the S-ALOHA throughput at that load. In the final 50 generations, with change of G to 1.0, P_{LW} further increases to assist the network in expediting packet transmissions. The resulting throughput is near 36% which is in line with the known maximum throughput for a random access protocol with time sensing enabled (i.e., S-ALOHA). Overall, these adaptation experiments indicate the system's ability to adjust genetic material (i.e., probabilities) such that evolved near-benchmark solutions are maintained under varying loading, time sensing, and behavioral degrees of freedom. The above results demonstrate the feasibility of evolution based approach for designing time-slotted MAC layer protocols under varying network and traffic conditions.

7.4.2 Evolution with Channel Sensing Enabled

Evolution of the state machine from Figure 7-2 is presented in this section. Note that time sensing is disabled for these experiments.

7.4.2.1 Limited Evolutionary Degree of Behavioral Freedom (i.e., 1,2)

In the following set of experiments, we observe evolution of throughput as well as genotype dynamics in the absence and presence of carrier sensing. Using the general state machine in Figure 7-2, we first allow $P_{WW}^{P=0\rightarrow P=1}$ as the only probability to evolve (i.e., EDB=1) in the initial 50 generations; afterwards, P_{CS} is also allowed to evolve, thus enabling carrier sensing. Propagation delay (normalized by packet duration) *a* is set to 0.1. In the first 50 generations, throughput evolves to around 0.18 for $G \ge 0.5$, the generalized machine reduces to that of pure-ALOHA when channel sensing is disabled. For G = 0.1, theoretical ALOHA benchmark throughput is around 0.08 which is the throughput of the evolved solutions. After generation 50, it

is observed that solutions with higher throughput are evolved as the nodes take advantage of the auxiliary modality of carrier sensing. It can be shown that throughput evolves to a near-CSMA (1-persistent) throughput which follows $S_{CSMA} = \frac{G e^{-aG}}{G(1+2a)+e^{-aG}}$ [66]. Figure 7-9 reports the corresponding evolutionary trends of $P_{WW}^{P=0 \rightarrow P=1}$ and P_{CS} during 100 generations when G = 0.5 and G = 3.0.



Figure 7-9: Genotype dynamics before/after channel sensing is added In first 50 generations, $P_{WW}^{P=0 \rightarrow P=1}$ converges to near-zero values for G = 0.5 as load is not greater than the optimal load 0.5 for pure-ALOHA. In case of an over-optimal load (G = 3.0), $P_{WW}^{P=0 \rightarrow P=1}$ converges to around 0.83 to decrease the effective load by the way of blocking a fraction of possible transmissions. After 50 generations, when channel sensing is enabled, P_{CS} evolves to 1 for both the loading scenarios, indicating that nodes learn to sense the channel when making transmission decisions. Upon emergence of CSMA behavior in the second half of the experiment, $P_{WW}^{P=0 \rightarrow P=1}$ acts in order to boost transmissions when load is less than the optimal CSMA load 2.6 (Figure 7-9:*a*). For G = 3.0, $P_{WW}^{P=0 \rightarrow P=1}$ acts in opposite direction to lessen the excessive input load. Similar experiments performed for varying *a* and *N* demonstrated emergence of 1-CSMA throughput solutions as the system learns to use channel sensing.

7.4.2.2 Protocol Adaptation with Maximum EBDF (i.e., 8)

A thorough test of adaptability was performed by changing all system parameters including a) normalized propagation delay a, b) network size N, c) network load G, and d) availability of channel sensing. All eight transition probabilities in Table 7-1 (i.e., parts (a) and (c)) were allowed to evolve during these experiments. Figure 7-10 depicts adapted throughput values in eight distinct phases, each of which was run for 50 generations. Channel sensing is enabled only for the second half of the experiment after 200 generations. Parameters that change across consecutive phases are underlined in the figure. Figure 7-11 shows the corresponding genotype dynamics. Observe in Figure 7-10, when there is no channel sensing, the throughput converges near the ALOHA benchmark of 0.18. In Figure 7-11, the parameter P_{WL} increases due to the excessive input load compared to the optimal load of 0.5 for pure-ALOHA. With change of a to 0.1 in phase 2, throughput is kept at 0.18 and probabilities follow a similar trajectory as in phase 1. That is because pure-ALOHA performance is insensitive to propagation delay ($S_{ALOHA} = G e^{-2G}$). In first 2 phases, both P_{WL} and $P_{TW}^{T=1 \rightarrow T=0}$ increase such that the effective load is adjusted to 0.5. Next, network size increases to 200, while the individual node load (g) remains at 0.02 (similar to node load in second phase). Such increase in network size results in a network wide load of 4.0 which enforces probabilities responsible for abandoning the transmission $(P_{TW}^{T=1 \rightarrow T=1})$ and $P_{TL}^{T=1 \rightarrow T=1}$ slightly increase. $P_{WW}^{P=0 \rightarrow P=1}$ also sharply increases to impose less frequent packet transmissions. Final phase of experiments with no channel sensing is performed by fixing N and a, and reducing the network load to 1.4. Diverging in opposite directions, lower $P_{WW}^{P=0 \rightarrow P=1}$ and higher P_{LW}



Figure 7-10: Throughput with runtime parameter changes



Figure 7-11: Adaptation of probabilities with runtime changes

facilitate more packet transmissions. Despite the initial decrease of throughput in third and fourth phases, system adapts to the new environment such that the throughput climbs to 0.18 again in both cases (Figure 7-10). Starting from phase five, channel sensing is enabled. Similar change of network parameters is repeated for the next four phases. P_{cs} and P_{Acs} which were not subject to

evolution in the first four phases move toward *l* in the fifth phase. They remain at near-*l* values indicating that nodes learn to apply the channel sensing modality and comply with channel status when sending packets. In this manner, a more complex protocol, i.e., CSMA emerges. This is proven in Figure 7-10 where near-benchmark CSMA throughput of 0.23 is achieved in the fifth phase. With changing a to 0.1 in the sixth phase, P_{WL} decreases sharply. The reason is that, under corresponding network parameters, CSMA attains the maximum throughput of 0.51 at G = 2.6. That is why the system attempts to adjust probabilities such that the effective load equals the input load 2.0. Observe how throughput bumps up to around 0.51 in the sixth phase of Figure 7-10. With changing network size to 200 in next phase, the network wide load approaches 4.0, which is larger than the optimal load 2.6. P_{WL} increases while $P_{TW}^{T=1 \rightarrow T=0}$ decreases, both aiming at reducing the effective load. As observed in Figure 7-10, then throughput falls initially, though it is restored to around maximum achievable throughput (0.51) by adjusting the appropriate probabilities. In the last phase, when network-wide load is reduced to 1.4, the system attempts to adjust the effective load to input load G again. Increase of $P_{TW}^{T=1 \rightarrow T=0}$ enforces legitimate returns to state W following successful transmissions. Besides, reduction in P_{WL} and the transmission abandoning probabilities help a) reducing the effective load \hat{G} , and b) maintaining throughput around the maximum value of 0.47. Overall, these set of adaptability experiments show: 1) how a more complex protocol (CSMA) can emerge when the system is equipped with channel sensing, and 2) how the evolutionary fabric is able to adapt to optimal solutions under dynamic network conditions. To summarize, the results in this section demonstrate the feasibility of using an evolution based approach for designing carrier-sensing MAC protocols.

7.5 Summary and Conclusion

Building on preliminary concepts developed in previous chapter (Chapter 6), in this chapter we explored the feasibility of evolving multi-access MAC protocols in presence of global time and channel sensing capabilities. Starting from a generalized state machine, it was shown that under appropriate genotype definition and fitness pressure it is indeed feasible to evolve ALOHA, Slotted ALOHA and CSMA protocols under a wide variety of network and loading conditions.

Future work on this topic includes evolving protocol state machines with: 1) both carrier and time sensing simultaneously enabled, 2) replacing full network connectivity by arbitrary mesh connections, and 3) heterogeneous loading conditions in different parts of a network. Designs with delay and other non-throughput fitness factors can also be explored.

The ultimate goal of this research is to build on the presented feasibility study, and enable evolutionary protocol synthesis for more complex network protocols and traffic conditions including variable size packets.

In the next chapter, we present a different form of learning based protocol design approach, namely, reinforcement learning to tackle the problem of communication protocol design. Reinforcement learning is concerned with decision makings with the goal of maximizing some notion of cumulative reward. In that sense, reinforcement based learning can closely be fit the problem of commercial content dissemination in SWNs.

8 Q-learning based Gain-aware Routing

8.1 Introduction

This chapter explores a reinforcement learning approach for synthetizing routing protocols in DTNs. As mentioned in Section 1.1, SWNs, as special type of Delay Tolerant Networks (DTNs), often reflect inter-personal social ties, interactions, and similarities in content and commodity consumption patterns. We leverage this information for D2D commercial content (e.g., coupon) dissemination with the goal of maximizing a predefined commercial gain for a Content Provider (CP).

The presented history-based multicast dissemination [67] in Chapter 4 suffered from certain amount of performance loss due to the fact that the disseminating node does not use prediction about the consumption interests of its potential future contacts. Predictive Gain Utility Routing Individual (PGUR-I) [68] introduced in Chapter 5 addressed that limitation by the way of developing a *predictive gain utility*, however, this method does not leverage transitive deliveries, which as opposed to direct delivery, refer to multi-hop content delivery. Such a delivery is done through intermediate (transitive) node encounters. PGUR-I is not scalable to mobility scenarios in which content can only be delivered to target destinations through transitive interactions. To address such routing challenges caused by high dynamicity of SWNs, in this chapter, we leverage the concept of Reinforcement Learning (RL) [47], which enables nodes to gather local information and make online efficient routing decisions. We design a Q-learning [47] based Gain-aware Routing (QGR) protocol, which aims at maximizing the economic gain for a content provider.

To address the stochastic nature of the presented routing problem, including nodes' stochastic meeting patterns, we apply the standard Q-learning framework [47] in an stochastic version. In this framework, nodes learn the estimated gain associated with each probabilistic

forwarding action and select the actions, which lead to higher expected reward or economic gain of dissemination. This online learning component in QGR is expected to be robust in diverse and dynamic mobility environments.

The specific contributions of this chapter are as follows. First, we briefly review the gain utility [68] that is designed around the notion of *consumption interest* and *coupon redemption probability* (Section 4.2.1). Then, we map this DTN routing utility as a stochastic value function within the Q-learning framework for developing a Q-learning based Gain-aware Routing (QGR) protocol, which attempts to maximize a defined expected economic gain of dissemination. Finally, we perform extensive simulation based experiments for functional validation and performance evaluation of the presented QGR protocol in comparison to two existing protocols. The developed protocol is evaluated for a number of social and commercially meaningful parameters, namely, network mobility, consumption interest patterns, content expiry time, forwarding (rebate) cost and the number of content copies.

8.2 System Model

Using the same *cost* and *revenue* (Section 4.2.1) formulation as defined in Chapter 4, we compute the expected gain achieved through a coupon redemption by consumer-*I* as:

$$G_i = V_i - c_f \tag{8-1}$$

The total network wide gain G^T can be expressed as:

$$G^T = \sum_{i=1}^{n_d} V_i - n \times c_f \tag{8-2}$$

where coupons are delivered to n_d consumers through n forwards within a pre-specified coupon *expiry time* (τ). The maximum number of deliveries of a coupon and the coupon expiry time are generally restricted by its generator. Also, $n_d \leq T$, where T is that distribution upper bound of a coupon, which is set by the coupon's generator. In the presented methods in this chapter, only one-hop deliveries are allowed, thus n always equals to n_d .

Similar to the presented consumers' interest and redemption probability model in Section 4.2.2, each consumer-*I* maintains an interest level ρ_i ($0 \le \rho_i \le 1$) towards a specific coupon type. Redemption function $P_r(\rho_i)$, represents the probability of redeeming a coupon by consumer-*i*. Note again that in the presented model, to partially preserve consumers' privacy, it is assumed that the data related to consumer interests and consumer-to-consumer interaction is always stored at the consumers' devices. It is never exported to a centralized server.

8.3 **Problem Formulation**

For a given combination of earned revenue c_p , discount c_d , rebate c_f , distribution upper bound T, expiry time τ , redemption function $P_r(\rho_i)$, and a social wireless network of consumers with interest profiles ρ_i , the objective is to design a learning based multicast coupon dissemination mechanism that maximizes the total gain G^T within the lowest possible dissemination delay.

8.4 Q-learning Model

Reinforcement Learning (RL) has been shown to address the routing challenges caused by dynamicity of wireless networks [47]. Q-learning [47], is a popular RL approach in which an agent (i.e., a node) is modeled as a three-tuple entity consisting of *{States (S), Actions (A), Reward I}* [47]. To address the stochastic nature of the routing, we propose a stochastic Q-learning based routing framework in which the set of states and actions (i.e., state transitions) are represented in the form of a probabilistic state machine. This state machine is formed when an agent meets a peer node at any time instant *t*.

a) State, Action and Reward

<u>States (S)</u>: An agent observes decision making factors in the local environment in the form of states [47]. An agent observes a state $s_t^i \in S$ at a time instant *t*. Here, the state space *S* consists of the set of all nodes in the network.



Figure 8-1: Abstraction of states and actions in the Q-learning based routing

Figure 8-1 shows an example of the abstraction of the mapped probabilistic state machine including the states observed by each agent *I* and *j* at time *t*. The figure shows an interaction between a content source node-*I*, and a node-*j* at *t*. This interaction is shown with a solid arrow line from node-*I* to node-*j*. The agent node-*I* is in a state of meeting node-*j*. Such state is deterministic in the sense that the interaction actually happens at time *t*. If node-*j*'s interest level (ρ_j) indicates that it is a target consumer, then the external state (j) observed by node-*I* is "a target consumer". We model a set of probabilistic states observed by an agent which refer to node's prospective interactions in future.

In Figure 8-1, a number of probabilistic states observed by node-*I* are shown as nodes M_i , $i \in [1, l]$. Similarly, the probabilistic states observed by node-*j* are shown as *j*'s prospective future encounters with nodes N_i , $i \in [1, k]$. These probabilistic interactions (transitions) are shown with dotted lines. Note that, node-*I* indirectly observes states N_i , $i \in [1, k]$ through node-*j*. In other words, node-*j*'s encounters are observed by node-*I* through transitivity. Deterministic and probabilistic transitions in perspective of node-*I* are stated on the right side of the figure. In following sections, we explain how the probability associated with each probabilistic transition is computed. Node-*I*, maintains the local information about its own observed states. To take an efficient routing action, node-*I* also collects (from node-*j*) the information regarding the probabilistic states, which it can reach via a two-hop transitivity through node-*j*.

<u>Actions (A)</u>: An agent chooses an action a_t^i (at time t) among its available set of actions (A). For agent node-I carrying T number of copies in Figure 8-1, an example action can be the transmission of a copy to a node N_1 through node-j. Another possible action for node-I is to transmit a copy for node-j's consumption. Or, node-I can decide to keep a copy to deliver to a prospective peer M_1 .

<u>Reward I:</u> Each action a_t^i is associated with a reward r_{t+1}^i . The reward can represent any performance metric. To design a gain-aware routing framework, we define the reward associated with an action of forwarding a copy to a peer node-*j* as the expected gain out of that delivery (G_j). Reward may refer to an *immediate reward* such as the gain yielded after delivering a consumption copy to an immediate contact node-*j* (at time t+1). It can also refer to the *discounted reward* [47] which is the expected accumulated reward to be received at time $\hat{t} > t$. Here, an example of such discounted reward is the expected gain obtained through the transitive deliveries that happen through node-*j* in future. The goal of an agent is to maximize its total reward, which is the gain through an immediate delivery and the gain obtained through transitive deliveries.

Assuming that all nodes contribute to such goal in an unselfish manner, maximizing the individual gain results in maximizing the total (network-wide) gain.

b) State-Action Functions (Q-values)

An state-action function, $Q_t^i(s_t^i, a_t^i)$ estimates the long term reward that can be obtained by agent node-*I* if it performs an action $a_t^i \in A$, while in state $s_t^i \in S$. An agent maintains a table containing Q-values for all possible state-action pair. The size of the table depends on the number of probabilistic states a node observes. That is the number of unique peers the node met so far, thus have future possibility to meet again.

From this point onwards, we express the iterative updates of the Q-values without explicitly indicating the iteration time steps 't'. This notation has already been used in Figure 8-1. For example, $Q_i^{i \to M1}$ refers to the Q-value for the possible action of transmitting a copy to the prospective encounter M_1 by i. Every time two nodes (M_1 and i) meet, they update their Q-values.

8.5 Q-learning based Gain-aware Routing (QGR)

Q-learning based Gain-aware Routing (QGR) protocol relies on the knowledge of: 1) nodes' interest profiles (i.e., ρ_i), and 2) their underlying social interaction patterns. A coupon-carrying node uses this information to compute and evaluate Q-values which reflect how a delivery to a peer contributes to the overall gain in the coupon's expiry time.

<u>State Variables</u>: Any node-*I* maintains the following variables with respect to another node-*j*. $\lambda_{i,j}$: *i*'s rate of meeting node-*j* over a pre-defined time window (e.g., a week). This parameter is updated every time the two nodes meet.

 ρ_j : node-*j*'s interest level toward the merchandize corresponding to the coupon. Node-*I* computes node-*j*'s redemption probability $P_r(\rho_j)$ based on the known redemption function corresponding to the coupon.

Q-values: Q-values are initialized at *0*. When a node-*I* meets a node-*j*, the Q-value for the possible action of transmitting a copy to *j* is computed using the following equation. This value is updated even if node-*I* does not possess any copies of the coupon at the time of the interaction.

$$Q_i^{i \to j} = (1 - \alpha)Q_i^{i \to j} + \alpha \max\{G_j, \gamma \widehat{Q_j^{max}}\}$$
(8-3)

 α ($0 \le \alpha \le 1$) is the learning rate. A very high α value may cause random fluctuations in Q-values. With α close to θ , learning may not happen and the agent only relies on its old Q-value.

The right side of the equation calculates the maximum of the two parameters G_j and γQ_j^{max} . G_j is the immediate reward corresponding to the action of forwarding a consumption copy to node *j*. G_j equals the gain of delivery to node-*j* and is computed based on Eqn. 8-2. Note that, G_j is an indication of how good a consumer node-*j* is.

 Q_j^{max} is the discounted (i.e., non-immediate) reward and refers to the maximum attainable expected reward from delivery *through* node-*j*. This quantity reflects how suitable node-*j* is as a forwarder. In other words, it estimates the expected gain out of a transitive delivery through node*j*. $\overline{Q_j^{max}}$ is computed based on the following equation:

$$\widehat{Q_j^{max}} = \max_{all \ k \in K} \{ p_k^j (\tau - t_c) \times Q_j^{j \to k} \}$$
(8-4)

,where, *K* is the set of nodes that node-*j* has seen so far and has the possibility to meet again. $p_k^j(\tau - t_c)$ is defined as that probability of *j* meeting *k* at least once, during the remaining time to coupon expiry $(\tau - t_c)$. t_c denotes the current time. Assuming that the inter-contact duration between node pairs follows exponential distribution, $p_k^j(\tau - t_c)$ is computed as follows:

$$p_k^j(\tau - t_c) = 1 - e^{-\lambda_{j,k}(\tau - t_c)}$$
(8-5)

The closer it gets to the coupon expiry time τ , the lower is the chance of meeting node k. The higher the contact frequency, $\lambda_{j,k}$, is between node-j and node-k, the more probable it is for j to meet k during $\tau - t_c$. The quantity $p_k^j(\tau - t_c)$ indicates the probability of obtaining reward $Q_j^{j \to k}$, which is the expected reward associated with the action of content delivery from node-j to its prospective future peer node-k. Computing the maximum reward over the rewards expected from transitive deliveries to all nodes $k \in K$, reflects the highest achievable expected gain out of a transitive delivery through node-j. Finally, the parameter γ ($0 \le \gamma \le 1$) in Eqn. 8-3 is the discount factor which determines agent node-*i*'s level of preference for relying on the long-term expected reward $\widehat{Q_j^{max}}$. If $\gamma = 0$, the immediate reward, which is the gain out of delivery to node-*j*, is the only factor affecting $Q_i^{i \to j}$. In that case, the expected gain out of transitive deliveries through node*j* is ignored.

Using the expression max $\{G_j, \gamma Q_j^{max}\}$ in Eqn. 8-3, node-*I* evaluates the quality of node-*j* as a consumer, or as a prospective forwarder. Then, $Q_i^{i \to j}$, the total expected reward out of content delivery to node-*j*, is updated.

Note that a proof of convergence for the Q-values is not provided in this paper, and it remains as a major future work. According to [69], for a stochastic framework such as the presented one, the learning algorithm was shown to converge under certain conditions on the learning rate.

<u>Forwarding Logic</u>: When a node-*I* carrying *T* number of copies meets node-*j*, if node-*j* has not consumed a copy of the content yet, *I* computes the gain G_j , which node-*j* can achieve through redeeming the coupon (Eqn. 8-2). If node-*j* has already consumed a copy, then $G_j = 0$, since *j* is not a potential consumer anymore.

Node-*I*, then receives the product $p_k^j(\tau - t_c)Q_j^{j \to k}$, for all $k \in K$ stored at node-*j*. *K* is the set of peers that *j* has seen so far, thus, may encounter again in future. The Q-value $Q_j^{j \to k}$ indicates the reward (i.e., gain) from a transitive delivery to node-*k* via node-*j*. Multiplying the probability of pairwise meeting $p_k^j(\tau - t_c)$ by the Q-value yields the *expected* reward (i.e., gain) of a delivery through *j*. Node-*I* also maintains $Q_i^{i \to l}$ values, where $l \in L$, and *L* is the set of peers that node-*I* has encountered so far. At this point, Node-*I* sorts the following values in a descending order in a list of expected Q-values, namely, SQ: 1) G_j , 2) $p_l^i(\tau - t_c)Q_l^{i \to l} - c_f$, ($\forall l \in L$), and 3) $p_k^j(\tau - t_c)$

 $t_c)Q_j^{j \to k} - 2 \times c_{f,i} (\forall k \in K)$. The quantity $p_l^i(\tau - t_c)$, the probability of meeting between node-*I* and its peer node-*l*, when multiplied by $Q_l^{i \to l}$, indicates the expected reward for a delivery to node-*l* by *i*. Such delivery may be a direct delivery of a copy to node-*l* for its consumption, or it may be a copy for a transitive delivery through node-*l*. That depends on which of the two elements out of G_l and γQ_l^{max} were superior when $Q_i^{i \to l}$ has been computed using Eqn. 8-3.

In the sorted set SQ, one unit of forwarding cost c_f is subtracted from each $Q_i^{i \to l}$ value to consider the rebate given to node-I upon a delivery to node-I. Two units of forwarding cost are subtracted from $Q_j^{j \to k}$ values to compensate for the cost of delivery from node-I to node-j and from node-j to node-k. If there is a common peer that node-j and node-I both may encounter, that is, if for an entry $Q_j^{j \to k}$, there exists a $Q_i^{i \to l}$ (in SQ) such that l = k, then node-I evaluates the expected reward for each action, If delivery through node-j results in a higher expected reward compared to a delivery to node-k by I itself, then the entry corresponding to $Q_i^{i \to l}$ is replaced with $p_k^j(\tau - t_c)Q_i^{i \to k} - 2 \times c_f$. Finally, node-I assigns copies to node-j based on the top T entries of the sorted expected Q-values list. If, G_j is among the top T entries, a copy is transmitted to j for consumption. For each Q-value entry that belongs to a potential transitive delivery to a node-k via node- $j(Q_j^{j \to k})$, a copy out of T copies is assigned to node-j. For each Q-value entry associated with node-i's encounters ($Q_i^{i \to l}$), a copy remains with node-i. Algorithm 8-1 depicts the key components of the QGR algorithm.

1: When node-*i* meets node-*j*: *i* updates $\lambda_{i,j}$, ρ_j and $Q_i^{i \rightarrow j}$ for node-*j* in its records; 2: **Forwarding Logic:** 3: L: set of past encounters of node-i; 4: *K*: set of past encounters of node-*j*; 5: $Q = \emptyset$; //Set of estimated Q-values $Q_i^{i \to \forall l \in L}$, $Q_j^{j \to \forall k \in K}$ and G_i 6: For node-*i* carrying *T* copies of a coupon: $G_i = P_r(\rho_i)(c_p - c_d) - c_f;$ 7: $Q = G_i \cup \{p_l^i(\tau - t_c)Q_i^{i \to l \in L} - c_f\} \cup$ 8: $\{p_k^j(\tau - t_c)Q_j^{j \to k \in K} - 2 \times c_f\}$ FOR EACH $k \in K$ 9: // If both nodes can potentially deliver to k IF EXIST $Q_i^{i \to l} \in Q$ such that l = k THEN 10: IF $(p_k^j(\tau - t_c)Q_i^{i \to k} - 2 \times c_f >$ 11: $p_k^i(\tau - t_c)Q_i^{i \to k} - c_f) \text{ THEN}$ Delete $p_k^i(\tau - t_c)Q_i^{i \to k} - c_f \text{ from } Q;$ 12: 13: **ENDIF** 14: **ENDIF** 15: **END FOR** //sort Q in descending order 16: SQ = sort(Q);17: IF $(SQ[0..T] \text{ contains } G_i)$ THEN Deliver one copy for consumption to node-*j*; 18: 19: T = T - 1;20: ENDIF 21: pq = 0; // peer quota, number of copies to be //forwarded through node-j **FOR EACH** $(Q_i^{j \to k}, \forall k \in K \text{ in } SQ[0..T])$ 22: pq += 1;23: 24: **END FOR** 25: IF (pq > 0) THEN Deliver *pq* number of copies to node-*j*; 26:

Algorithm 8-1: Key components of the QGR algorithm

8.6 Compared Protocols

<u>Social-Aware Networking (SANE)</u>: SANE [43] is designed based on the observation that individuals with similar interests tend to meet more often. This characteristic is preserved in the example SWNs (mobility scenarios) used for evaluation in this chapter. SANE models each

individual's interest profile as an *m*-dimensional interest vector. We implement the interest-cast SANE-SW (Spray and Wait), in which a content carrier *I* forwards half of its *T* copies to a peer *j* if the cosine similarity metric [43] between the relevance of the message *M*, denoted as *R*(*M*), and the interest profile of $j(\rho_j)$ is higher than a given *relaying threshold* ρ . Cosine similarity between *m*-dimensional vectors *A* and *B* is defined as $\theta(A, B) = A.B/||A||||B||$. We have incorporated the following changes in order to adapt the baseline SANE-SW with *m*-dimensional (*m*>1) interest profiles to the present framework in which nodes can have a one-dimensional interest profile (*m*=1).

We change our one-dimensional interest profile ρ_i for each node-*I* to a two-dimensional interest profile vector of $\langle \rho_i, z \rangle$ in which *z* is assumed to be a fixed interest level to another type of content. This enables SANE to work under the experiment settings of this paper. It also does not affect the performance of any of the protocols. This way, the cosine similarity between two interest profiles reflects the similarity interest level (ρ_i), because the second element *z* is the same for all profiles.

The relevance of the message R(M) is in the same form as nodes' m-dimensional interest profiles and is set to (ρ_i^{max}, z) . ρ_i^{max} is the interest level with the highest redemption probability $(P_r(\rho_i))$ according to the redemption function. R(M) explains the target's desirable set of interest levels.

To set the *relaying threshold* ρ , the possible range of cosine similarities for different pair of nodes with various interest levels are first computed using z=0.5. ρ should be a value inside the range of possible cosine similarities. We test different ρ values and choose the one which yields the best results (highest gain) for SANE under the tested mobility scenarios. <u>PGUR-I</u>: Predictive Gain Utility Routing Individual (PGUR-I) [68] as described in Chapter 5 uses node-specific interest and interaction information to evaluate how a delivery to a peer contributes to the overall gain in the coupon's expiry duration. This protocol does not consider transitive delivery through multi-hops. However, using gain prediction, it provides a strategy to choose a forwarding strategy with the goal of maximizing the defined gain. When a content carrier node-*I* meets a node-*j*, it computes the expected gain of delivery (Eqn. 8-2) to nodes it may visit during the remaining time to coupon expiry ($\tau - t_c$). If node-*I* is expected to meet more number of nodes (than it has coupon copies) with expected gain higher than that of the immediate contact node-*j*, it holds onto all its copies. Otherwise, a copy of the coupon is delivered to the immediate contact node-*j*.

8.7 Results

8.7.1 Experimental Details

We use the DTN simulator ONE [55] for evaluating QGR, SANE and PGUR-I under a community based synthetic mobility model and a real mobility trace (i.e., from INFOCOM '06). The models are different in their community structure and interest level distributions. For all the experiments we use a redemption function in which the redemption probability maximizes for interest levels that are not too low or not too high. This is motivated by the observation [14] that while improving purchase probability for moderately loyal (interested) customers, coupon delivery may often kill product loyalty for highly loyal customers (Section 1.3.1). To capture this effect, a Gaussian model for the redemption function (Fig. 4-3) is used with mean $\mu = 0.75$ and $P_r^{max} = 0.5$. The relevance of the message R(M) for SANE is set to <0.75, z=0.5>, because 0.75 is the interest level of the target consumer with maximum redemption probability. The Q-learning

parameters learning rate (α) and discount factor (γ) are set at 0.5 and 0.9 respectively. After experimenting with a range of values, these values were found to be able to provide fast learning convergence.

8.7.2 Community-based Synthetic Mobility

Synthetic mobility follows map-based movements in a network of *10* waypoints [55]. Nodes move between waypoints with assigned transition probabilities. *90* nodes are divided into three communities as shown in Figure8-2. Numbers in parentheses show community's population. Arrows show the flow of migrant nodes between communities. Members within a community hold higher intra-community meeting frequency compared to inter-community meeting frequencies.



Figure 8-2: Community-based synthetic mobility

Members of each community also tend to have more similarity in interest levels toward a specific content compared to the similarity of interest level between members of two different communities. We assign random interest levels to members of each community such that $P_r(\rho_i), (i \in C_1) < P_r(\rho_i), (i \in C_2) < P_r(\rho_i), (i \in C_3)$. Each experiment is repeated for around 30 source nodes and the average performance is reported.

8.7.2.1 Impact of Rebate Cost (c_f)

Figure 8-3 shows the evolution of gain over time for different forwarding cost (rebate) c_f . In all experiments, content is generated on day 27, which is long after when the network simulation warms up in terms of stabilizing the node-interaction profiles. Experiments with 15 initial coupon copies (i.e., T = 15) were carried out for expiry time $\tau = 8$ days.

All the protocols show an increasing trend for gain over time, as more coupons are delivered. QGR shows the maximum achievable gain for all the scenarios in Figure 8-3. SANE has the second best gain mostly because it leverages multi-hop transitive deliveries despite PGUR-I, which only performs direct delivery. PGUR-I has the lowest performance since the source nodes, especially those located in community C_1 , can not directly reach target nodes with maximum redemption probabilities in community C_3 . QGR achieves the maximum gain since it manages to distribute copies among nodes according to their consumption and forwarding capabilities. SANE attempts to achieve a similar goal by finding consumers which are most similar to the target consumer (i.e., node with the highest redemption probability). However, SANE uses a binary distribution of copies which involves certain amount of randomness that causes the gain to suffer. As the rebate cost increases from 0.002 (Figure 8-3:a) to 0.2 (Figure 8-3:d), gain for all protocols decreases.

The reason is that QGR's more node-specific strategy of content dissemination results in distribution of the copies in breadth rather than forwarding bundles of copies to nodes in a binary spray and wait [43] manner. With cost increase, QGR's more spread distribution of copies results in a larger rebate cost and therefore, a lower gain. QGR also starts and finishes content delivery with some delay compared to SANE and PGUR-I.

That is because the protocol predicts meetings with nodes with a higher redemption probability. Source nodes reserve copies for such high-gain yielding deliveries, deferring dissemination. To better understand the working dynamics of the QGR protocol, the evolution of Q-values $Q_i^{i \rightarrow j}$ for $\tau = 8$ days (Figure 8-3:d) is shown in Figure 8-4. Two pair of nodes are randomly chosen from each community (see Figure8-2). For each node-*j*, the moving average over each 50 samples of $Q_i^{i \rightarrow j}$'s seen by any arbitrary node-*I*, is measured. The average Q-values are shown from day 2, after an initial warm-up period. Initially, the only non-zero element in Eqn. 8-3 is G_j . Thus, Q-values reflect αG_j . As Q-values build up, they show the quality of node-*j* as a consumer and as a forwarder, depending on the result of max $\{G_j, \gamma Q_j^{max}\}$ in Eqn. 8-3. The highest Q-values belong to nodes from community C_3 , because the nodes in C_3 maintain the highest redemption probabilities. These nodes can also be high-quality forwarders due to their intra-community encounters with similarly interested peers.



Figure 8-3: Evolution of gain for forwarding cost c_f : (a) 0.002, (b) 0.02, (c) 0.1 and (d) 0.2 While nodes from community C_2 hold medium redemption probabilities, those from C_1 maintain the lowest redemption probabilities (and gain), hence, have the lowest Q-values.

The monotonically increasing Q-values for each node-*j* from community C_3 stabilize around G_j . This shows that these nodes are high-quality consumers and G_j is so large that the result of max $\{G_j, \gamma Q_j^{max}\}$ in Eqn. 8-3 is mostly G_j .

Q-values for nodes *j* from C_1 and C_2 initially increase to values larger than their own gain G_j . This shows that these nodes can reach peers which are better consumers than themselves. Since nodes from C_2 visit target consumers of C_3 with high redemption probabilities, their maximum Qvalues are higher than those for nodes from C_1 . A fall in $Q_i^{i \to j}$ (*i.e.*, $j \in C_2$ and C_1) is observed for the following reason. Closer to the content expiry time, the probability of meeting any node-*k* by a node-*j*, $(p_k^j(\tau - t_c))$ becomes very small. Thus, $\overline{Q_j^{max}}$ (i.e., in Eqn. 8-4), which reflects the capability of *j* for transitive delivery to the target consumers reduces. That leads to the domination of G_j over $\gamma \overline{Q_j^{max}}$ in the Q-value computation formula in Eqn. 8-3.





The Q-values converge to G_j eventually. The rising and falling trend of the Q-values for the members of C_1 and C_2 show that these nodes are more qualified forwarders than consumers.

8.7.2.2 Impacts of Content Expiry Duration (τ) and the Number of Content Copies (T)

Figure 8-5 shows the impacts of content expiry duration (τ) and the number of copies (*T*) on gain when $c_f = 0.02$. Total gain of delivery for all protocols increases as more number of content copies are available. Stretching τ duration has similar effect.

For fewer number of copies (e.g., T = 5) and for short expiry duration (e.g., $\tau = 0.5 \ day$), the difference of gain between the protocols is relatively small. That's because the protocols have less number of choices to make and less chance to use their prediction/learning ability to make efficient routing decisions. For example, for a small number of copies (*T*), allowing fewer intermediate forwarding and eliminating the rebate cost is a more economical solution to maintain a higher gain. This is basically how the non-transitive protocol PGUR-I works. The closer performance of QGR to PGUR-I for limited *T* and τ indicates QGR's ability to adjust its forwarding strategy to such application level parameters.



Figure 8-5: Impacts of content expiry duration (τ) and number of copies (T) on gain
Note that an extra analysis on the metric "*number of forwards*" involved in a content delivery
is not performed because the composite metric "gain" already contains this cost in the form of
rebate cost. This is the main cost since the other cost which is due to transferring pairwise interest
and interaction profiles (upon nodes' meetings) is similarly imposed in all protocols.

8.7.3 Trace-based Mobility

The dissemination protocols were also tested for a human interaction trace collected [59] from INFOCOM '06 conference. The trace contains data from devices which were carried around by

users for all four conference days. All experiments in this section are run for *30* random source nodes out the 97 nodes.

Figure 8-6:a and Figure 8-6:b show the dynamics of gain and time to reach the maximum gain respectively. Forwarding (rebate) cost c_f is fixed at 0.002. Each cluster in the figures corresponds to a scenario with a specific number of content copies *T*. Each of the *12* points in each cluster corresponds to a combination of protocol and expiry duration: 0.5, 1, 1.5 and 2.5 days. Figure 8-6:a shows that QGR achieves the maximum gain compared to other protocols for most combinations of *T* and τ . As expected, larger *T* results in higher gain. Also, in each cluster in Figure 8-6, the points toward the right side generally have higher gain as they correspond to larger τ . Similar to the synthetic mobility results (in Figure 8-3), the gain difference between the protocols is smaller under smaller number of copies such as *T*=5. The protocol SANE has the second highest gain in most cases. This is because it utilizes transitive deliveries, using which it is possible to better explore the environment in comparison to PGUR-I which is direct delivery based.

As shown in Figure 8-6:b, the observed higher gain for QGR comes at the price of longer time to reach the maximum gain. In most cases, PGUR-I reaches its maximum gain at a later point of time compared to SANE. The reason lies in PGUR-I's direct delivery nature in which the source node holds on to copies to deliver to the best consumers using predictions. SANE, on the other hand, reaches target consumers in a shorter time delay, leveraging its transitive deliveries. As shown in all the clusters in Figure 8-6, it naturally takes longer for all protocols to deliver all copies when the number of copies is larger. Experiments with other combinations of T, τ and rebate cost (c_f) for this trace-based mobility have shown similar gain dynamics as what is seen for the synthetic mobility.



Figure 8-6: Dynamics of a) gain and b) time to reach maximum gain Overall, results from both synthetic and trace-based mobility verify that the designed QGR protocol is able to achieve higher gain by striking a better balance between the forwarding cost and revenue for most combinations of commercial and network parameters.

8.8 Summary and Conclusion

In this chapter, we developed a Q-learning based Gain-aware DTN multicast Routing (QGR) protocol which is specifically designed for commercial content types such as coupons. In comparison with the history-based and prediction-based dissemination methods in Chapters 4 and 5, the online learning component in QGR is expected to be robust in diverse and dynamic mobility environments. The protocol's objective is to maximize the economic gain of dissemination which

is defined as a composite index that combines *revenue from delivery* and *forwarding cost* from dissemination. In this framework, a node learns the suitability of its encountered peers both as consumers as well as intermediate forwarders by estimating Q-learning's state-action functions. Experimental results from the simulator ONE [55] showed that the presented protocol achieves higher economic gain compared to two existing dissemination methods in most combinations of network (mobility) and commercial/environment parameters.

Future work on this topic can include: a) studying the effects of learning framework parameters (e.g., learning rate (α)), b) providing a proof of convergence for action-state functions and c) developing mechanisms for handling consumer selfishness.

In next chapter, we extend this protocol to a Scalable Q-learning based Gain-aware Routing protocol (SQGR) which provides higher scalability and maintains better user privacy.

9 Scalable Q-learning based Gain-aware Routing

9.1 Introduction

In previous chapter [70], we showed that Q-learning is able to address the dynamicity and stochasticity associated with routing in SWNs. This chapter presents a Scalable Q-learning based Gain-aware Routing (SQGR) which extends QGR (Chapter 8) [70] focusing on system scalability and privacy preservation. Using an Scalable Q-learning model, unlike QGR [70], we avoid storing consumer specific interaction and interest profiles on devices, thus, providing better privacy.

In this chapter, we formulate the gain of dissemination as a long-term reward in a Q-learning based Gain-aware routing approach. The goal is to take a series of forwarding actions which result in delivery to the highest gain-yielding target customers.

The specific contributions of this chapter are as follows. We first introduce a definition of the economic gain of dissemination as used in our prior work (Chapter 4). That is a function of forwarding cost (rebate) and revenue. We then present the Scalable Q-learning based Gain-aware Routing (SQGR) algorithm. Finally, we present simulation results, which verify SQGR's functionality and performance in comparison to few existing protocols.

The system model including the gain formulation used in this chapter follows the same model presented in Chapter 8. We also tackle the same problem of maximizing total economic gain G^T for the content generator, except that we focus on embedding an additional level of scalability and privacy.



Figure 9-1: Probabilistic state machine modeling for Q-Learning

9.2 Scalable Q-learning based Routing (SQGR)

9.2.1 Gain Quanta

<u>Scalability and Privacy</u>: We divide the possible gain range $[G_{min}, G_{max}]$ to *b* number of *gain quanta*. Maximum and minimum possible gain can be computed by Eqn. 8-2. Width of each quantum, s_b is $(G_{max} - G_{min})/b$. We refer to each gain quantum with its lower bound. n_q is the number of nodes falling into each gain quantum q.

9.2.2 Q-learning Model

Following the Q-learning [47] model, each agent (i.e., a node) is modeled as a three-tuple entity consisting of *{States (S), Actions (A), Reward I}* [47]. A probabilistic state machine, as shown in Figure 9-1, formulates states along with stochastic transitions (i.e., actions) when two nodes (N_i and N_j) meet at time *t*.

<u>States (S)</u>: At each point of time t, an agent may observe a set of deterministic and probabilistic states. The state space S consists of a set containing all nodes in the network. In Figure 9-1, agent N_i observes the deterministic state N_j , as long as the connection between the two nodes is held.

Agent N_i observes a set of probabilistic states shown with dashed circles. Each circle consists of a group of nodes whose gain falls in a specific gain quantum. Probabilistic states may or may not be observed *in future* $\hat{t} > t$. The agent N_i observes N_j 's prospective states transitively (e.g., nodes $R_1 \dots R_{n_{q+s_b}}$ with gains in $[q, q + s_b)$).

Actions (A): At each point of time *t*, an agent chooses an action among the set of actions (A). Considering the stochastic nature of the routing problem, A consists of both deterministic and probabilistic actions shown as the transitions among states in Figure 9-1. For example, agent N_i , assuming that it holds T copies of content, can choose the action of forwarding a copy to node N_j . This is a deterministic action. Agent N_i can choose to hold certain number of copies to transmit to a node belonging to the group of nodes with gain in quantum \dot{q} or to transmit (through transitivity) to a prospective peer of node N_j (e.g., R_2). The latter actions are probabilistic and are shown as probabilistic transitions. Each such transition is associated with a transition probability. For example, the transition probability to the state of node N_i meeting any node from quantum qthrough node N_j is a function of $\lambda_{j,q}$, the meeting frequency between N_j and members of quantum q.

<u>Reward I:</u> When two nodes meet, an action $a \in A$ is associated with a reward $r \in R$. Set of rewards R consists of *immediate* and *discounted rewards* [47]. In the presented model, reward associated with each action is defined as the resulting expected economic gain of delivery in perspective of the content provider. Deterministic actions, such as a delivery from node N_i to N_j (Figure 9-1), are associated with immediate reward, which is G_j . Transitive deliveries which take place at $\hat{t} > t$, are associated with discounted rewards.

<u>*Q*-values</u>: At each point of time t, a state-action function, namely, a Q-value, estimates the immediate/discounted reward associated with an action $a \in A$ considering a current state $s \in S$.
For example, the Q-value $Q_i^{i \to \dot{q}}$ indicates the action of forwarding one or multiple copies of a coupon to any node-*o* with gain within the gain quantum \dot{q} (i.e., $G_o \in [\dot{q}, \dot{q}+s_b)$). The subscript of the term shows the owner of the Q-value, while the superscript indicates the action which is associated with this Q-value. For an immediate contact such as node N_j , in Figure 9-1, G_j denotes the value associated with a direct delivery to node-*j*. All Q-values are stored at each node's *Q-table* (Figure 9-1).

9.3 Scalable Q-learning based Gain-aware Routing

The proposed SQGR protocol relies on the knowledge of: 1) nodes' interest profiles (i.e., ρ_i), and 2) their underlying social interaction patterns. Unlike in Q-learning based Gain-aware Routing (QGR) [70], these state variables are not stored per-node. They are instead aggregated over a gain quantum. A coupon-carrying node uses this information to compute and evaluate Q-values which reflect how a delivery to a peer with a certain interest level, contributes to the overall gain in the coupon's expiry time.

<u>State Variables</u>: Any node-*I* maintains *and* updates the following variables everytime it meets a node-*j*.

 $\lambda_{i,q}$: *i*'s rate of meeting with any node *j* in the gain quantum *q* where $G_j \in [q,q+s_b]$ over a predefined time window (e.g., a week).

 n_q : the *number* of nodes in gain quantum q which are visited by node-*I* so far. This is increased by one at every instance of meeting between node-*I* and a node with gain in quantum q.

Q-values: Q-values are initialized at 0. When a node-*I* meets a node-*j*, the Q-value for the possible *action* of transmitting a copy to a node belonging to the gain quantum q ($G_j \in [q,q+s_b]$] is computed using the following equation.

$$Q_i^{i \to q} = (1 - \alpha)Q_i^{i \to q} + \alpha \max\{G_j, \gamma \widehat{Q_j^{max}}\}$$
(9-1)

 α ($0 \le \alpha \le 1$) is a standard Q-learning parameter, namely, the learning rate. A very high α value may cause random fluctuations in Q-values. Very small α values indicate more reliance on old Q-values.

A higher G_j indicates a higher redemption probability for node-*j*, thus shows the quality of node-*j* as a consumer. Node-*j* can also be a forwarder. Such quality of *j* is reflected in $\widehat{Q_j^{max}}$, the discounted (i.e., non-immediate) reward. Q_j^{max} is the maximum attainable expected reward from delivery *through* node-*j*:

$$\widehat{Q_j^{max}} = \max_{all \ q \in Q} \{ p_q^j (\tau - t_c) \times Q_j^{j \to q} \}$$
(9-2)

,where, Q is the set of gain quanta that node-*j*'s previously visited peers belonged to. $p_q^j(\tau - t_c)$ is defined as the probability of *j* meeting any node with gain within the gain quantum *q* at least once, during the remaining time to coupon expiry $(\tau - t_c)$. t_c denotes the current time. Assuming that the inter-contact duration between node pairs follows exponential distribution, $p_q^j(\tau - t_c)$ is computed as follows:

$$p_q^{j}(\tau - t_c) = 1 - e^{-\lambda_{j,q}(\tau - t_c)}$$
(9-3)

The quantity $p_q^j(\tau - t_c)$ indicates the probability of obtaining reward $Q_j^{j \to q}$ associated with a coupon delivery from node-*j* to a prospective future peer with gain within quantum *q*.

The Q-learning parameter γ ($0 \le \gamma \le 1$) in Eqn. 9-1 is the discount factor which determines agent node-*i*'s level of preference for relying on the long-term expected reward $\widehat{Q_j^{max}}$ compared to immediate reward G_j . For a $\gamma = 1$, immediate delivery is discouraged. Using the expression max $\{G_j, \gamma Q_j^{max}\}$ in Eqn. 9-1, node-*I* evaluates the quality of node-*j* as a consumer, or as a prospective forwarder in a group of nodes which gain fall into the same gain quantum.

Note that, for SQGR, the maximum size of a Q-table on a consumer's device reduces to *b*, the total number of gain quanta, which is smaller in comparison with *N-1* (network size), the maximum size of Q-table in QGR [70]. Lower Q-table size translates to less usage of storage on mobile devices.

<u>Forwarding Logic</u>: When a node-*I* carrying *T* number of copies meets node-*j*, if node-*j* has not consumed a copy of the content yet, *I* computes the gain G_j , which node-*j* can achieve through redeeming the coupon (Eqn. 8-1).

Node-*j*, then shares the product $p_q^j(\tau - t_c)Q_j^{j \to q}$, for all gain quanta $q \in Q$ with node-*i*. Q is the set of gain quanta which node-*j*'s previous encountered peers belonged to. *J* may encounter one of these nodes at least once in time remaining to coupon expiry with the probability $p_q^j(\tau - t_c)$. The Q-value $Q_j^{j \to q}$ indicates the reward (i.e., gain) from a transitive delivery to any node with gain in range $[q,q+s_b]$ via node-*j*. Therefore, $p_q^j(\tau - t_c)Q_j^{j \to q}$ indicates the *expected* reward (i.e., gain) of a delivery through *j*. Node-*I* also maintains $Q_i^{i \to \dot{q}}$ values in its Q-table, where $\dot{q} \in \dot{Q}$. Here, \dot{Q} is the set of gain quanta that node-*I*'s previously encountered peers belong to. Node-*I* then sorts the following values in a descending order in a list of expected Q-values, namely, $q_s: 1 | G_j, 2 | p_q^i(\tau - t_c)Q_i^{i \to \dot{q}} - c_f$, $(\forall \dot{q} \in \dot{Q})$, and $3 | p_q^j(\tau - t_c)Q_j^{j \to q} - 2 \times c_f$, $(\forall q \in Q)$.

1: Upon meeting between node- <i>i</i> and node- <i>j</i> with G_j :
2: <i>i</i> updates state variables: $\lambda_{i,q}$, n_q and $Q_i^{i \to q}$;
Forwarding Logic:
3: $\overline{Q, Q}$: set of gain quanta belonging to nodes visited
by <i>i,j</i> ;
4: For node- <i>i</i> carrying <i>T</i> copies of a coupon:
5: $Q_e = p_{\dot{q}}^i (\tau - t_c) Q_i^{i \to \forall \dot{q} \in \dot{Q}} - c_f, p_q^j (\tau - t_c) Q_j^{j \to \forall q \in Q} - 2 \times c$
and G _j ;
6: IF (<i>i</i> and <i>j</i> maintain estimated Q-values for
delivery to a gain quantum $q' = q$) THEN
7: $Q_e = Q_e - \min \{ p_{\dot{q}}^i (\tau - t_c) Q_i^{i \to \forall \dot{q} \in Q} - $
$c_f, p_q^j(\tau - t_c)Q_j^{j o \forall q \in Q} - 2 \times c_f\};$
8: $Q_s = \text{sort}(Q_e)$; //Sort in descending order
9: FOR EACH (entry e in Q_s)
10: IF $(e = G_j)$ THEN
11: IF $(T \ge 0)$ THEN
12: Deliver a copy for consumption to
node-j;
13: ELSE
14: $T = \min\{0, T - n_q\};$
15: END IF
16: END FOR
Algorithm 0.1, Summany of the SOCD algorithm

Algorithm 9-1: Summary of the SQGR algorithm

The one unit of forwarding cost c_f is subtracted from each $Q_i^{i \to \hat{q}}$ value is to consider the rebate given to node-I as it cooperates in a transitive delivery. Two units of forwarding cost are subtracted from $Q_i^{j \to q}$ values to compensate for the cost of delivery from node-*I* to node-*j* and from node-*j* to another node with gain belonging to quantum q. For common gain quanta that node-*i* and node-*I* both hold in their Q-tables, the highest expected Q-value is stored.

If G_i is the highest value in the sorted list, a copy is forwarded to node-*j*. Otherwise, for each expected Q-value $Q_i^{i \to \hat{q}}$ or $Q_j^{j \to q}$ higher than G_j , n_q or $n_{\hat{q}}$ copies (out of *T* available copies at nodei) are held at I for transitive deliveries. This means that there are transitive deliveries which result in a higher expected gain compared to a delivery to the immediate contact node-j. Note that $n_{\dot{q}}$

and n_q are the number of N_i and $N'_j s$ encountered peers which belong to the gain quanta \dot{q} and q respectively. Eventually, if any copies are left with *I*, it is forwarded to node-*j* for consumption. This forwarding algorithm is depicted in Algorithm 9-1.

9.4 Results

We have compared the designed SQGR algorithm with Social-Aware Networking (SANE) [7] and Direct Delivery (DD) mechanisms. In SANE, a content carrier *I* forwards half of its *T* copies to a peer *j* if the cosine similarity metric between the relevance of the message *M*, denoted as R(M), and the interest profile of $j(\rho_j)$ is higher than a given *relaying threshold* ρ . In DD, a node forwards a copy of a coupon to a peer that does not possess the coupon at their meeting time.

9.4.1 Experimental Details

We use the DTN simulator ONE [55] to evaluate performance of SQGR, SANE, DD and QGR [70]. We run experiments for three mobility scenarios. We use a Gaussian redemption function (Figure 5-2) with $\mu = 0.75$ and maximum redemption probability $\rho_r^{max} = 0.5$. The relevance of the message for SANE is set such that it targets consumers who have the closest similarity of interest to the maximum target interest level, 0.75. Learning rate (α) and discount factor (γ) are set to 0.5 and 0.2 respectively. These values are set experimentally after proving to provide fast learning coverage. The impact of different learning parameters remains for the future work. Earned revenue c_p and discount cost c_d are set to 5 and 2 respectively.

9.4.2 Community-based Synthetic Mobility

The synthetic mobility consists of 90 nodes forming 3 distinct communities (Figure 9-2). The arrows show the flow of nodes migrating between communities and the numbers inside each

community shows the size of the community. Intra-community interest levels are more similar compared to those across communities. Also, $G_i(i \in C_1) < G_i(i \in C_2) < G_i(i \in C_3)$.



Figure 9-2: Community-based mobility

Evolution of Gain: Figure 9-3 shows the evolution of gain over time during the lifetime of a coupon. The coupon is generated after a warmup period, when Q-values are settled network-wide. The gain is computed as the average over *30* experiments, each using a random source node. Forwarding cost is set as *0.002*.

Observe that the gain has an increasing trend for all protocols as more number of coupons are distributed over time. SQGR achieves a higher gain compared to other protocols for all expiry durations. With an increase in expiry duration from half a day (Figure 9-3:a) to 5.5 days (Figure 9-3:d), gain for all protocols except for DD, increases due to more exploration duration. For DD, the source and the receivers of the coupon deliver all T=15 coupons to the first 15 nodes which they encounter. A longer expiry duration for the DD case does not necessarily change the set of encountered nodes, thus, the gain does not increase. SQGR performs better than QGR in this case because SQGR takes the forwarding decisions based on comparison of the immediate gain of delivery G_j with expected Q-values of transitive deliveries to *any* node from a gain quantum higher than the quantum G_j belongs to. The probability of meeting such node is relatively higher than meeting any *one* specific node with gain higher than G_j . The latter is the forwarding criteria used in QGR. For this reason, SQGR finds immediate delivery less urgent and instead waits to meet better consumers in future.



Figure 9-3: Evolution of gain over time for expiry duration τ : (a) half a day, (b) 1.5, (c) 3 and (d) 5.5. days

The final gain values when forwarding cost is set higher at $c_f = 0.2$ are shown in Figure 9-4. Figure 9-4:a and Figure 9-4:b show gain results for number of coupons T = 15 and T = 25 respectively. Similar to the observation in Figure 9-3, all protocols have an increasing trend in gain with longer coupon expiry duration.

For T=15 (Figure 9-4:a), final gain values for all protocols are lower compared to the equivalent scenarios with the smaller forwarding cost of 0.002 in Figure 9-3:a to Figure 9-3:d. Higher gain values in general is observed when T=25. SQGR mostly achieves a higher gain compared to QGR for the reasons mentioned earlier, that is, maintaining copies for longer duration and eventually delivering to better consumers.



Figure 9-4: Final gain values for (a) T=15 and (b) T=25

To gain more insight into SQGR's performance, we show Figure9-5. Figure9-5 shows Q-values $Q_i^{i\to 0}$ (Fig 9-5:a) and $Q_i^{i\to 1.2}$ (Fig 9-5:b) at every update for every node in the network. Fig 9-4:a shows the quality of nodes for delivering to any consumer whose gain belongs to the gain quantum 0. This corresponds to the smallest possible gain values in the current experiment. The high concentration of points in cluster of Q-values observed by members of C_1 in Fig 9-5:a indicates that these nodes meet a higher population of nodes with gain within quantum 0.

This is due to the high contact frequencies of the members of this community with one another. The higher Q-values shown in this cluster indicate that nodes with IDs in range [0,29] are good candidates for delivering coupons to nodes maintaining gain within quantum 0. The lower values of $Q_i^{i\to 0}$ recorded for nodes belonging to C_3 shows that these nodes are less capable of delivering copies to nodes maintaining small gain values. This is because members of C_3 only encounter nodes with relatively small gain values when they migrate to C_2 .

Absence of Q-values $Q_i^{i \to 1.2}$ for members of C_1 in Fig 9-5:b shows that these nodes do not directly encounter nodes which hold gain in range [1.2, 1.3], that is, nodes belonging to C_3 . Note that, this is the highest possible gain range for nodes under the tested mobility scenario. Unlike members of

 C_1 , nodes in C_2 migrate to community C_3 and update Q-values $Q_i^{i \to 1.2}$ whenever they visit any peer residing in C_3 . That's the reason why relatively high $Q_i^{i \to 1.2}$ can be seen for nodes with IDs in range 0-29.. Similar trend of Q-values is observed for the members of C_3 due to the frequent intracommunity encounters.

In summary, Figure9-5 shows how nodes differ in quality of delivery (direct or transitive) to low (Figure9-5:a) and high quality (Figure9-5:b) consumers. SQGR dissemination mechanism works based on identifying such qualities and choosing the most suitable nodes for delivering content to high quality consumers.



Figure 9-5: Q-tables' content at each node for (a) $Q_i^{i \to 0}$ and (b) $Q_i^{i \to 1.2}$

9.4.3 Trace-based Mobility

The dissemination protocols were tested on INFOCOM mobility trace [59], which consists of human social interaction data for *97* devices, during *4* days of the INFOCOM '06 conference. Each node is assigned a random interest level.

<u>Gain Vs. Time to reach maximum gain:</u> Figure 9-6 shows the maximum gain and the time to reach the maximum gain for all protocols for short expiry duration of half a day (Figure 9-6:a) and longer expiry duration of 2.5 days (Figure 9-6:b). The forwarding cost is fixed at *0.2*. Similar

to the results for the community based mobility (Section 9.4.2), SQGR mostly provides a higher gain value in both expiry duration scenarios. This is because, SQGR relies on the probability of meeting *any* higher quality consumer in future. This results in a longer time for SQGR to reach its maximum gain. DD has the fastest delivery at the cost of lower gain. SANE and QGR almost do similarly in terms of gain and time to reach that. As expected, the total gain of delivery and time to reach that increases for all protocols, if longer expiry duration is allowed (Figure 9-6:b).

Scalability: One of the primary value propositions of SQGR, in comparison with QGR, is higher scalability and better privacy preservation by not storing consumer specific personal interaction and interest profile on peers' devices. As mentioned in Section 9.2, the higher scalability of SQGR is a result of smaller Q-tables stored at consumers' mobile devices. To demonstrate this, we furnish Figure9-7, which shows the average Q-table size for *10* randomly chosen nodes recorded at the time of the coupon expiry. The figure shows that QGR requires nodes to store more number of Q-values, for all expiry durations between half a day and 2.5 days. For QGR, the maximum number of Q-values stored in a node's Q-table is *N-1*, where *N* is the network size. That is equal to the maximum number of peers a node may visit before a coupon expires. In this case, N= 97. For SQGR, the maximum number of Q-values equals the number of gain quanta (*b*), since Q-values are computed and stored per gain quantum. Here, the gain varies in [0,1.3], which results in 13 gain quanta, each of size 0.1.



Figure 9-6: Maximum gain and time to reach that for expiry duration: (a) half a day and (b) 2.5 days

9.4.4 Working Day Mobility

Here, we combine the community-based movement with reasonable human daily routines [55]. 90 nodes are divided in three communities residing in their home areas in 3 distinct areas on the map of the city of Helsinki [55].



Figure 9-7: Q-table size per node

Members of each community are similarly interested in a coupon content. Nodes wake up in the morning, go to work, go shopping or do similar activities in the evening and eventually return home and sleep. Experiments were run for two forwarding cost values of 0.002 and 0.2 and τ between *half a day* and 4 days. Similar results to what was observed for the community based and trace based mobility scenarios were seen. Those will not be shown here to avoid redundancy. One interesting example of gain evolution for a $\tau = 4$ days is depicted in Figure9-8. The dashed lines show the start of a new working day. It can be seen that for all protocols, gain starts to increase in initial hours of a working day. That is when nodes start visiting each other. Gain stabilizes as soon as nodes go back to their homes and sleep.



Figure 9-8: Gain evolution for working day mobility with $\tau = 4$ days

Overall, results from the three tested mobility scenarios demonstrate that SQGR is mostly able to achieve the highest gain among other protocols under various combinations of forwarding cost c_f , number of coupons *T*, and expiry duration τ . We observed that SQGR does not obtain a noticeably higher gain compared to QGR and SANE for some combinations of forwarding cost c_f and very short expiry duration. However, due to its higher scalability and providing a better consumer privacy, SQGR appears to be more suited to the task content dissemination with the goal of maximizing gain among other tested protocols.

9.5 Summary and Conclusion

We presented a Scalable Q-learning based Gain-aware Routing (SQGR) protocol for disseminating commercial content in Social Wireless Networks. We defined an economic gain, which is a composite metric that combines the revenue from delivering and the costs associated with that delivery for a vendor. Content forwarding and the associated gain are then mapped to the state-action and reward for the Q-Learning framework. The key benefit of SQGR is that it does not store any per-node consumption interest or interaction information across nodes, thus providing both scalability and privacy. The analysis and performance evaluation performed using the ONE simulator demonstrates the superiority of SQGR in terms of gain performance and scalability in comparison to three existing protocols. The future work on this topic includes: 1) studying the effects of consumer (node) selfishness, and 2) automatically fine-tuning the Q-learning parameters (e.g., discount factor and learning rate).

10 Future Work

10.1 Introduction

There are many applications of multicast Device-to-Device (D2D) content dissemination in healthcare [6], emergency handling [7], and vehicular networks [8]. In this dissertation we specifically focused on a commercial content dissemination application which is about delivering commercial content such as coupons to a target population of customers. The main design goal for a D2D commercial content dissemination algorithm is about maximizing a predefined commercial gain for a Content Provider (CP). In other words, a carrier of content/coupon identifies most fit individual Content Consumers (CCs) based on its pattern of social interaction (e.g., contact duration, centrality, etc.) with target consumers.

The objective of this dissertation has been to develop coupon dissemination mechanisms that can maximize a coupon's economic gain in the presence of costs/rebates for D2D forwarding incentives. This can be posed as a Delay Tolerant Network (DTN) multicast routing problem.

In chapter 3, we developed a Unicast Gain-aware Dissemination Protocol (GDP) with the goal of maximizing the defined economic gain for the content provider. Gain of dissemination was defined as a combination of a time-based value function and cost of delivery. In chapter 4, we extended this work to multicast content dissemination introducing an Economy Driven Content Dissemination (EDCD) protocol. We applied a more elaborate gain formulation which is a combination of a redemption probability based value function and cost of delivery. In chapter 5, using the same gain formulation, we presented a Predictive Gain Utility based Routing framework consisting of two routing protocols. With gain prediction, this framework provides multicast gain-aware routing in more structured and scalable routing approaches compared to EDCD.

So far, the designed routing mechanisms were designed primarily based on past experience of the designer with no embedded learning mechanism. This approach may not scale well for heterogeneous and dynamically changing networks and with newly emerged gain formulation. This motivates the field of network protocol design using learning mechanisms that can adapt with heterogeneity and changing operating conditions, thus avoiding redesigns with changing conditions.

In this direction, we presented a first step toward evolving network communication protocols, specifically for Medium Access Control layer protocols (Chapter 6 and Chapter 7). A generalized probabilistic state machine was defined initially. It was shown that under appropriate genotype definition and fitness pressure, it is indeed feasible to evolve ALOHA, Slotted ALOHA and CSMA protocols under a wide variety of network and loading conditions.

In Chapter 8, we leveraged reinforcement learning to tackle the problem of communication protocol design. Reinforcement based learning can closely fit the problem of commercial content dissemination in SWNs as it is concerned with decision makings with the goal of maximizing some notion of cumulative reward. We developed Q-learning based Gain-aware DTN multicast Routing (QGR) protocol which is specifically designed for commercial content types such as coupons. In comparison with the history-based and prediction-based dissemination methods in Chapters 4 and 5, the online learning component in QGR is expected to be robust in diverse and dynamic mobility environments.

In Chapter 9, extending QGR, we developed Scalable Q-learning based Gain-aware Routing (SQGR) protocol. The key benefit of SQGR in comparison to QGR is that it does not store any per-node consumption interest or interaction information across nodes, thus providing both scalability and privacy.

Building on the developed research in this dissertation, in this chapter, we outline different directions of future work and the involved challenges.

10.2 Evolution of DTN Routing Protocols

10.2.1 Parameter Evolution

<u>1) Evolving Depth of Penetration (*DoP*):</u> It was shown in Chapter 4 that the presented EDCD can achieve better economic gain compared to a number of competing DTN routing protocols under different mobility scenarios and Depths of Penetrations (*DoPs*). The balance between the obtained value and cost usually leads to gain maximization at one specific *optimal DoP* for any given mobility scenario. A possible future work direction on this topic is to use Genetic Algorithms (GA) for self-adjusting the *DoP* parameter such that the optimal value of *DoP* can be set online. Optimal *DoP* may change according to the mobility scenario, nodes' interest profiles and type of the coupon (with different expiry time). An evolutionary approach may manage to adjust the *DoP* value runtime according to such changes.

<u>2) Delivery pattern:</u> A new direction could be that without sharing any interest level among consumers, a device learns about the average interest level of peers in its surrounding and adjusts its delivery rates accordingly. Let us assume that the probability of sending a coupon for consumption (p_{sfc}) exponentially increases as time remaining to coupon expiry decreases. For example, $p_{sfc} = \frac{1}{e^{x(\tau-t_c)}}$, where $\tau - t_c$ indicates the time remaining to coupon expiry. The slope of the curve, x, if smaller, causes more number of deliveries in the early stages after coupon generation. Larger x values impose more number of deliveries around the expiry time. A possible future work in this direction could be using GA to evolve this parameter in order to fine tune the pattern of delivery according to network conditions.

The fitness function in both examples of parameter evolution can be the "effective" gain out of deliveries that a node does. Challenges which need to be addressed here include: notifying a node regarding its fitness which is its contribution to the total gain without a server's involvement.

10.2.2 Protocol Evolution

Based on the evolutionary framework presented in Chapter 6 and Chapter 7, it can be proposed to design and evolve probabilistic state machines which implement D2D content dissemination protocols. One such example is shown in Figure 10-1. States of the machine include: 1) Wait (W): which shows the state in which a node holds at least one copy of a coupon and is waiting to meet a peer, 2) Send (S): represents sending one copy of a coupon to a peer for consumption and 3) Terminate (T): represents termination of the content transmission procedure between two nodes. The quantity p_k is a binary input parameter. If $p_k = 1$, it indicates that the coupon carrier has more than enough number of copies to deliver to prospective nodes with expected gain larger than an immediate contact's gain (before the coupon expiry time). $p_k p_{WS}$ represents a valid transmission, while $(1 - p_k) p_{WS}$ indicates that a copy is delivered to an immediate neighbor when there are not enough number of copies to be delivered to more qualified contacts in future. $p_{SW} = 1$ indicates a rogue behavior which allows a source node to return to the Wait state and attempt to send more number of consumption copies for the same peer, instead of terminating the transmission process. This state machines is general in that it contains both valid and rogue scenarios for disseminating a content. The objective is to investigate if and how the transition probabilities evolve towards generating different forwarding protocols.



Figure 10-1: Example general state machine consisting of a content dissemination protocol

The fitness criterion in this case would be the induced gain by disseminating coupons using a routing protocol that each set of transition probabilities (state machine) represent. It is challenging to design a more general state machine by granting higher degree of behavioral freedom to it. Consider when inputs to the state machine are 1) expected gain induced by an immediate contact, and 2) the expected gain of delivery through prospective contacts. In that case, the state machine should manage to evolve the right set of calculations and comparisons on the inputs in order to use them in making a valid delivery decision. With more complexity, this approach searches through a wider sample space. That may lead to finding better solutions compared to when the state machine has less degree of behavioral freedom as shown in Figure10-1. Thus, one future work on this topic includes investigating the feasibility of evolving D2D content dissemination protocols using evolving probabilistic state machines with different evolutionary degrees of behavioral freedom.

10.3 Neural Network Driven Q-learning

Neural Networks (NNs) can be used for approximating Q-functions, which are Q-values [48, 71] in a learning formulation. A possible future work direction on the designed reinforcement

learning based methods, QGR (Chapter 8) and SQGR (Chapter 9) is to estimate the Q-values (Section 8.1.2) using a Neural Network instead of using the state-action functions. Using this approach, the input of the NN would be states and actions. The output is the estimated Q-value for the corresponding state-action pair. In this process, the need for a Q-table look up is eliminated. Based on the specific Q-learning formulation presented in this dissertation and the analysis presented in Section 9.4, for QGR, the maximum number of Q-values stored in a node's Q-table is N-1, where N is the network size. That is equal to the maximum number of peers a node may visit before a coupon expires. For SQGR, this number reduces to the number of pre-defined gain quanta. Training an NN for estimating Q-values instead of looking up the Q-tables is faster and more scalable especially for large Q-tables.

Using NNs may result in a more generalized state-action function estimator. In the alternative approach, every time a state-action is observed, the corresponding Q-value is computed as a function of the previous Q-value based on a pre-defined state-action function.

10.4 Gain Estimation using Time Series Prediction

In the prediction based method presented in Chapter 5, we assumed that the inter-contact durations between node pairs follow exponential distribution. Based on this assumption, we developed two Predictive Gain Utility Routing (PGUR) methods in which forwarding logic is built upon predicting expected gains that are based on nodes' past interaction locality information.

A future work direction on this topic is to relax the assumption that nodes' inter-contact durations follow a specific distribution. Instead, use prediction tools such as Recurrent Neural Networks, specifically Long Short Term Memory (LSTM), to predict [72] the gain values observed by each node. Based on such predicted values, a node can leverage a wider perspective in order to decide whether it needs to hold on to a content copy or forward it when a peer is met.

Note that, the gain values observed by a node is a time series. Thus, the problem can be posed as time series prediction. To accomplish this, nodes interactions should inhibit some sort of temporal locality, which is the case in SWNs.

One example of such gain time series can be seen in Figure10-2. The synthetic series shows the gain values corresponding to the peers met by an arbitrary node-*I* over *10* days. This corresponds to a simulation done with *100* nodes following a Working Day Movement (WDM) mobility model [55].



Figure 10-2: Time series of gain observed by a node-i

Consider a benchmark method in which an oracle is aware of the future gain observations by every node in the network. In that case, the problem is defined as if the oracle can take the best forwarding decision for each node such that the overall gain is maximized.

10.5 Redemption Functions

The redemption function framework introduced in 1.3.1 is generalized enough to capture varying interest level to redemption probability mapping. A coupon generator can manually design such a mapping as a redemption function. The goal of a D2D dissemination protocol is to maximize the economic gain for a given redemption function. A future work direction on this topic includes performing experiments with a wide range of sample redemption functions to

accommodate for different marketing strategies and analyzing the impacts of varying marketing psychology on the economic gain performance.

10.6 Handling Selfishness

A selfish user is one who deviates from the dictated content dissemination strategy with the goal of earning more rebate. Such deviation results in forwarding decisions which result in a lower network-wide gain for the content provider. This happens if a node arbitrarily forwards content as an intermediate node and is rewarded with rebates, while increasing the cost of forwarding for the content provider. The other way a node can deviate from the designed dissemination strategy is to avoid performing some forwarding decisions selected by the embedded dissemination algorithm. In that case, it stops itself and the peers which were supposed to receive the content copy, from contributing to the gain maximization process.

A future direction on this work is to analyze the impacts of selfishness on gain performance. To accomplish that, parameters such as number of selfish nodes, degree of node selfishness can be defined. One possible way to define node selfishness is to have a selfish node do x number of arbitrary forwards. A full degree of node selfishness under such definition is to make a selfish node forward a single copy to every peer it meets. Such forwarding, without considering a peer's potentials as a 1) consumer and 2) a forwarder, increases the forwarding cost and may result in a negative gain.

The next step would be providing solutions including 1) detecting selfish nodes and 2) combating selfishness through embedding appropriate strategies in the dissemination protocol. One approach to infer selfishness for the dissemination algorithm on a node's application is to detect a decreasing trend of individual gain as a warning and either report this to the content provider or run an embedded strategy to combat the node's selfish behavior. One simple approach

187

to combat node's selfishness is to de-authorize the node's mobile device for participating in the dissemination process. Combatting selfishness in incentive-based content dissemination in SWNs and DTNs is extensively discussed in the literature [5, 73]. Similar methods can be adapted to the commercial content dissemination framework presented in this dissertation.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," in *Annual Review of Sociology*, 2001.
- [2] J. Goldenberg and M. Levy, "Distance is not dead: Social interaction and geographical distance in the internet era," in *Arxiv*, *abs/0906*, 2009.
- [3] E. Cho, S. Myers, J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *KDD '11*, NY, 2011.
- [4] D. Koschützki, et. al., "Centrality Indices," *Network Analysis: Methodological Foundations*, p. 16–61.
- [5] F. Xia, L. Liu, J. Li, J. Ma and A. V. Vasilakos, "Socially Aware Networking: A Survey," *IEEE Systems Journal*, vol. 9, pp. 904-921, 2015.
- [6] S. Olariu, K. Maly, E.C. Foudriat, S. M. Yamany and T. Luckenbach,, " A Dependable Architecture for Telemedicine in Support of Disaster Relief," in *Dependable Systems*, John Wiley and Sons, 2004.
- [7] M. Asplund, S. Nadjm-Tehrani, and J. Sigholm, "Emerging Information Infrastructures: Cooperation in Disasters," in *Lecture Notes in Computer Science, Critical Information Infrastructure Security*, Springer, 2009, pp. 258-270.
- [8] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay, and C. Cervello-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," IEEE Commun. Surv. Tutorials, 2012.
- [9] M. Subramani and B. Rajagopalan, "Knowledge-sharing and influence in online social networks via viral marketing," *Commun. ACM*, vol. 46, no. 12, pp. 300-307, 2003.
- [10] J. Berger, "Is Word of Mouth Better Than Advertising?," 2015. [Online]. Available: http://jonahberger.com/is-word-of-mouth-better-than-advertising/.
- [11] K. P. S. T. D. Goel, "Determining a value for a coupon". Google Patents 2013.
- [12] M. Trusov, R. Bucklin, K. Pauwels, "Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking," *Journal of Marketing*, vol. 73, pp. 90-102, 2009.

- [13] A. Udd, "Predicting Redemption Probability of Gift Cards," Master of Science Thesis, Royal Institute of Technology, KTH, 2013.
- [14] T. Lehman, "Using coupons to kill customer loyalty," [Online]. Available: https://www.newnorth.com/using-coupons-to-kill-customer-loyalty/.
- [15] M. Boll, "Effective Win-Back Email Campaigns For 3 Stages Of Customers," 2016. [Online]. Available: https://glew.io/effective-win-back-email-campaigns-3-stagescustomers/.
- [16] L. Guo, C. Zhang, H. Yue and Y. Fang, "A privacy-preserving social-assisted mobile content dissemination scheme in DTNs," in *INFOCOM*, Turin, 2013.
- [17] B. Zhang, J. Teng, X. Bai, Z. Yang and D. Xuan, "P3-coupon: A probabilistic system for Prompt and Privacy-preserving electronic coupon distribution," in *Pervasive Computing and Communications (PerCom)*, Seattle, 2011.
- [18] K. Wei; X. Liang; K. Xu, "A Survey of Social-Aware Routing Protocols in Delay Tolerant Networks: Applications, Taxonomy and Design-Related Issues," *Communications Surveys & Tutorials, IEEE*, vol. 16, pp. 556-578, 2014.
- [19] A. Garyfalos and K.C. Almeroth, "Coupons: A Multilevel Incentive Scheme for Information Dissemination in Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 792-804, 2008.
- [20] W. Moreira, P. Mendes, and S. Sargento, "Social-aware Opportunistic Routing Protocol Based on User's Interactions and Interests," in *Proceedings of AdHocNets*, Spain, 2013.
- [21] W. Gao, Q. Li, B. Zhao, G. Cao, "Multicasting in Delay Tolerant Networks: a social network perspective," in *MobiHoc '09*, 2009.
- [22] Y. Zhu; B. Xu; X. Shi; Y. Wang., "A Survey of Social-Based Routing in Delay Tolerant Networks: Positive and Negative Social Effects," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 387-401, 2013.
- [23] P. Hui and J. Crowcroft, "How Small Labels Create Big Improvements," in *PerCom Workshops*, 2007.
- [24] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *ACM MobiHoc*, China, 2008.
- [25] Y. Cao, Z. Sun, "Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 654-677, 2013.

- [26] E. Bulut and B. K. Szymanski, "Friendship based routing in delay tolerant mobile social networks," in *GLOBCOM*, 2010.
- [27] T. Zhou, R. R. Choudhury, and K. Chakrabarty, "Diverse routing: Exploiting social behavior for routing in delay-tolerant networks," in *Proc. CSE*, 2009.
- [28] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, p. 606–621, 2009.
- [29] A. Lindgren, A. Doria, O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proceedings of the Workshop on Service Assurance with Partial and Intermittent Resources*, August, 2004.
- [30] J. Karvo, and J. Ott, "Time scales and delay-tolerant routing protocols," in *In Proc.c of the ACM MobiCom Workshop on Challenged Networks*" 2008, (CHANTS).
- [31] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop:Routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM '06*, Barcelona, Catalunya, Spain, 2006.
- [32] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *INFOCOM*, 2009.
- [33] L.Tang, Q. W. Zheng, J. Liu, X. Y.Hong, "SMART: A Selective Controlled-Flooding Routing for Delay Tolerant Networks," in *Broadband Communications, Networks and Systems, BROADNETS*, 10-14 Sept. 2007.
- [34] F. Nazir, J. Ma, and A. Seneviratne, "Time critical content delivery using predictable patterns in mobile social networks," in *Proc. CSE*, 2009.
- [35] T. Ning, Z. Yang, X. Xie and H. Wu, "Incentive-aware data dissemination in delay-tolerant mobile networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2011.
- [36] Y. Wang; M. Chuah; Y. Chen, "Incentive driven information sharing in delay tolerant mobile networks," in *GLOBECOM*, 2012.
- [37] A. Vahdat, D. Becker, "Epidemic Routing for partially-connected ad hoc networks," Duke Technical Report CS-2000-06, 2000.
- [38] S. Ioannidis, A. Chaintreau and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *INFOCOM*, 2009.

- [39] J. Díaz, A. Marchetti, D. Mitsche, P. Santi, J. Stefa, "Social-Aware Forwarding Improves Routing Performance in Pocket Switched Networks," in *Lecture Notes in Computer Science*, 2011.
- [40] B. B. Chen and M. C. Chan, "MobiCent: A credit-based incentive system for disruption tolerant network," in *INFOCOM*, 2010.
- [41] M. Uddin, B. Godfrey, and T. Abdelzaher, "RELICS: In-network realization of incentives to combat selfishness in DTNs," in *Network Protocols (ICNP)*, 18th IEEE International Conference on, 2010.
- [42] P. Costa, C. Mascolo, M. Musolesi, and G.-P. Picco, "Socially-aware routing for publishsubscribe in delay-tolerant mobile ad hoc networks," *IEEE J. Sel. Areas Commun*, vol. 26, no. 5, p. 748 –760, 2008.
- [43] A. Mei, G. Morabito, P. Santi and J. Stefa, "Social-Aware Stateless Routingin Pocket Switched Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 252-261, 2015.
- [44] C. Zhu and M. Jin, "An anycast routing algorithm based on genetic algorithm," *Wireless Trans. on Comp.*, pp. 113-122, 2009.
- [45] X. Li, L. Ci, B. Cheng, C. Tian, M. Yang, "Ant Colony Based Routing Strategy in UAV Delay Tolerant Networks," 2012.
- [46] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *SIGCOMM '04, ACM, 2004.*
- [47] H. Al-Rawi, M. Ng, K. Yau, "Application of reinforcement learning to routing in distributed wireless networks: a review," *Artif. Intell. Review*, vol. 43, no. 3, pp. 1573-7462, 2015.
- [48] J. Boyan and M. Littman, "Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach," in Advances in Neural Info. Proc. Systems (NIPS), 1994.
- [49] V. Rolla, M. Curado, "A Reinforcement Learning-based Routing for Delay Tolerant Networks," *Eng. Appl. Artif. Intell*, vol. 26, no. 10, pp. 2243--2250, 2013.
- [50] A. Elwhishi, et. al, "ARBR: Adaptive reinforcement-based routing for DTN," in *IEEE 6th International Conf. on Wireless and Mobile Computing, Networking and Comms.*, Niagara Falls, 2010.

- [51] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K., & H. Weiss, "Delaytolerant networking architecture," RFC4838, 2007.
- [52] K. Fall, "A delay-tolerant network architecture for challenged internets," in *In Proceedings* of the conference on Applications, technologies, architectures, and protocols for computer communications, ACM, 2003.
- [53] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, K., V. Cerf, B. Durst, & H. Weiss, "Delaytolerant networking: an approach to interplanetary internet," *Communications Magazine*, *IEEE*, vol. 41(6), pp. 128-136, 2003.
- [54] N. Kumar, & J. Kim, "Probabilistic trust aware data replica placement strategy for online video streaming applications in vehicular delay tolerant networks," in *Mathematical and Computer Modeling*, 2012.
- [55] "The ONE; The Opportunistic Network Environment simulator," [Online]. Available: http://www.netlab.tkk.fi/tutkimus/dtn/theone.
- [56] J. Douceur, "The sybil attack," in IPTPS, 2002.
- [57] E. Friedman and P. Resnick, "The social cost of cheap pseudonyms," *Journal of Economics and Management Strategy*, vol. 10, no. 2, p. 173–199, 1998.
- [58] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05), ACM, 2005.*
- [59] J.Scott, R. Gass, J.Crowcroft, P.Hui, C. Diot, A. Chaintreau, "Cambridge-haggle-imoteinfocom2006-2009-05-29," May, 2009. [Online]. Available: Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006.
- [60] D.Fenn, "10 ways to get more sales from existing customers," [Online]. Available: http://www.inc.com/guides/2010/08/get-more-sales-from-existing-customers.html .
- [61] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
- [62] S.G. Ara'ujo, A.C.P. Pedroza, A.C. Mesquita, "Evolutionary Synthesis of Communication Protocols," in *10th International Conference on Telecommunications (ICT)*, 2003.
- [63] N. Abramson, "The ALOHA System Another Alternative for Computer Communications," in *in Proc. Fall Joint Computer Conference*, 1970.

- [64] L. D. Davis, Handbook Of Genetic Algorithms, Van Nostrand Reinhold, 1991.
- [65] H. Mühlenbein, "The Breeder Genetic Algorithm a provable optimal search algorithm and its application.," in *Colloquium on Applications of Genetic Algorithms, IEE 94/067*, London, 1994.
- [66] L. Kleinrock, F.A. Tobagi, "Packet Switching in Radio Channels: Part I--Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1400-1416, 1975.
- [67] F. Hajiaghajani, S. Biswas, "Device-to-Device Commercial Content Dissemination in Social Wireless Networks," in *The 14th Annual IEEE Consumer Comms. and Networking Conf. (CCNC '17)*, Las Vegas, 2017.
- [68] F. Hajiaghajani, S. Biswas, "Device-to-Device Coupon Distribution using Economic Routing Utilities," in 12th International Conf. on Wireless and Mobile Computing, Networking and Comms. (WiMob '16), New York, 2016.
- [69] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.
- [70] F. Hajiaghajani and S. Biswas, "Learning based Gain-aware Content Dissemination in Delay Tolerant Networks," in 9th International Conference on COMmunication Systems and NETworkS (COMSNETS), India, 2017.
- [71] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS Deep Learning Workshop*, 2013.
- [72] J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," 2016. [Online]. Available: http://machinelearningmastery.com/time-seriesprediction-lstm-recurrent-neural-networks-python-keras/.
- [73] H. Zhou, J. Wu, H. Zhao, S. Tang, C. Chen, and J. Chen, "ncentive driven and freshnessaware content dissemination in selfish opportunistic mobile networks," in *Proc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst. (MASS)*, 2013.
- [74] S.Y. Yang, J.T. Jiang, & P.Z. Chen, "OOPProPHET: A New Routing Method to Integrate the Delivery Predictability of ProPHET-Routing with OOP-Routing in Delay Tolerant Networks," *Journal of Computers*, vol. 8(7), pp. 1656-1663, 2013.
- [75] F. Hajiaghajani, Y. Piolet, M. Taghizadeh, S. Biswas, "Economy Driven Content Dissemination in," *Elsevier Journal of Ad hoc Networks*, vol. 20, pp. 132-149, 2014.

[76] A. K ERÄNEN, and J. OTT., "Increasing reality for DTN protocol simulations," Technical Report, Helsinki University of Technology, Networking Laboratory, July 2007.