

PROTOCOL AND PERFORMANCE OPTIMIZATION FOR HANDLING HETEROGENEOUS
NETWORK DYNAMICS

By

Salim Mohamed

A DISSERTATION

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

Electrical and Computer Engineering - Doctor of Philosophy

2023

ABSTRACT

This study presents advancements in routing strategies for both wired and wireless data communications, organized into three discrete domains. In Part-I, the focus is on IP packet networks; Part-II addresses routing and interference control in cellular communications; and Part-III elucidates collision management techniques for TDMA-based and ad hoc networks.

IP communications: The study examines relay routing in data networks, focusing on the implementation costs of relay solutions. Dynamic relay involves periodic probing to capture routing changes and enhance performance. In contrast, static relay routing utilizes non-periodic probes to measure metrics like latency and identify reliable paths. Network dynamics-induced IP changes are thoroughly explored. The research delves into characteristics like Hop-To-Live (HTL) count, defining the number of hops in a minimum delay relay path. Stable HTL behavior guides stable and low-delay path selection, minimizing probing. The 18,906 paths analysis showcases the viability of IP relay paths that enhance layer-III performance, substantially benefiting TCP bandwidth and transfers. Additionally, a hybrid TCP-UDP protocol is proposed for TCP-based services such as YouTube and VoIP. The suggested estimation-based path selection for this hybrid layer-III relay shows an increased streaming bandwidth across one hundred and forty nodes. This TCP-UDP protocol complements pure TCP services, catering to delay and bandwidth-sensitive applications.

Cellular communications: Emerging cellular networks enable dynamic spectrum access. The FCC's adoption of rules in 2015 facilitates commercialization of the 3550-3700 MHz Citizen Broadband Radio Service (CBRS) band, previously used for radar data communications and now available for LTE dynamic access. To enhance spectrum utilization, dynamic access is proposed. However, the lack of collaboration among cellular carriers poses reliability challenges due to growing interference. A novel architecture is proposed to foster collaboration in multi-Channel Collaborative Multihop and Cellular Networks (CMCNs), featuring suboptimal interference and congestion management. The study evaluates the Minimum Broadcast Delay Forwarding (MBDF) length in slots against the initial and possible minimum schedule. The utilization of conflict graphs transforms MBDF from station granularity to a set of stations, or component granularity. This process facilitates solving the dependency between conflicting components. Results outperform existing delay and message redundancy models. The model has been examined over sets of channels and network parameters.

Ad Hoc communications: TDMA collision management via Q-Learning (QL) has challenges like convergence speed and reward. This work proposes a distributed, minimum frame length, and minimal handshake QL-TDMA protocol. Nodes learn from collision statistics via only NACKs, unlike conventional protocols with ACK-NACK signals. Exploration of slots occurs within the next frame, although accelerating convergence is achievable by considering the next slots in the current frame if time has not elapsed. Effectively, QL-TDMA minimizes collisions across learning episodes, as evidenced by evaluations.

This dissertation is an original intellectual product of the author, Salim Mohamed. The work reported in Chapter 3 is an individual effort, and based on a previous work detailed in Chapter 2. Major parts of the theoretical protocol design in Chapter 4 are individual work presented by the author. The simulation and the problem mapping and transformations is also the author main contribution in this dissertation. The analysis in chapters 4 and 5 have been written and examined without any access to a experimental labs.

TABLE OF CONTENTS

1	Introduction	1
2	The Significance of Layer-3 Traffic Forwarding	6
2.1	Background	6
2.2	Literature Review	6
2.3	Methodology	7
2.4	Contribution	9
2.5	Proposed Relay Forwarding	9
2.6	Discussion and Results	10
2.7	Conclusion	15
3	Hop Characteristics for Relay and Hybrid TCP-UDP Services	17
3.1	Background	17
3.2	Literature Review	17
3.3	Methodology	18
3.4	Contribution	19
3.5	HTL Stability	19
3.6	Dominant Path Prevalence	24
3.7	Node and HTL Symmetry	28
3.8	Dominant Relay Redundancy and Type	29
3.9	HTL and Change Detection	30
3.10	RTT Characteristics	31
3.11	Proposed Relay Performance	33
3.12	Conclusion and Future Work	38
4	Reinforcement Learning Based TDMA Protocol for Ad Hoc Networks	39
4.1	Background	39
4.2	Radio Networks QL-TDMA	41
4.3	Motivation and Contributions	42
4.4	Literature Review	43
4.5	QL Fundamentals	43
4.6	QL-TDMA Protocol	45
4.7	QL-TDMA Evaluation	59
4.8	Conclusion	63
5	Interference Handling for Collaborative Multihop Cellular Networks	65
5.1	Background	65
5.2	Literature Review	67
5.3	Problem Statement and Motivation	68
5.4	Methodology	72
5.5	CMCN System Architecture	73
5.6	Interference and Congestion Handling	73
5.7	P6 Conflict Management	77
5.8	Results and Evaluation	94
5.9	Conclusion and Future Work	101

6 Conclusion and Future Work	103
BIBLIOGRAPHY	105

1 Introduction

The growth of packet networks (wired and wireless) makes measuring, analyzing, and characterizing packet dynamics a challenge. The pace of change and network heterogeneity are causes of such a challenge. Packet delay is a design and performance characteristic of telecommunication networks. This delay specifies latency for a bit of data to cross a network from one communication endpoint to a second one and is typically measured in multiples or fractions of a second [59]. This delay is also sensitive to many network conditions such as link congestion and capacity, and varies according to congestion control schemes. The best-effort IP systems infer packet delay and drop rate, and throughput changes from network resources feedback [59]. Hence, such congestion control schemes should properly decode network dynamics and characteristics to maintain a smooth running of services and optimize quality. The second network dynamic is sending rate, which should be balanced for stable network performance. The low sending rates for video streams for example could introduce a higher latency and delayed low-quality services for end users. High rates, similarly, bring up concerns like congestion and drop rates. Therefore, such a sending rate must be tuned to accommodate hardware speeds for all used network resources along active transmission paths. This is sufficient to ensure no overwhelming of routers or relay queues of wired and wireless systems respectively would occur.

Highly unstable or more time-varying links or channels could lead to more unpredictable dynamics such as throughput or channel capacity when the number of network end users increases or cell-towers deployment is not suitable to handle more users in environments with fading and low gain channels. The fraction of data capacity changes rapidly, and causes lag for end services in cellular networks [59]. The current congestion schemes are behind in accommodating such shifts [59].

End users oftentimes have no privileges to implement such kernel-space congestion control systems over a large scale of public or application-based private networks. This lack of luxury forces users or companies to seek distinct alternatives to optimize services. Relay routing for instance is a user-space approach introduced decades ago to create a virtual layer of physical network resources to re-route data traffics across far and less utilized segments of a network over particular congestion periods. This work proposes a wide characteristics analysis and view of packet dynamics to improve the current best-effort congestion control and routing protocols' quality of service. This work confirms the existence of such beneficial relay paths, over which certain packet dynamics can be optimized easily in contrast to their analogous best-effort physical paths. This relay routing for IP networks has been well-documented over the past years. The implementation cost and characteristics of relay paths have not yet been conclusively identified. Dynamic relay routing depends on periodic probing to catch network dynamics such as delay changes for enhanced performance. The static relay routing, however, is based on non-periodic probes to measure latency and packet

drop over short periods. The aim of a dynamic relay is to stabilize network dynamics for a maximum QoS while a static relay seeks more reliable relay paths over time. There exists research as in [72] focused on understanding IP link dynamics because of routing fluctuations. This thesis is focused on examining characteristics such as the number of hops in a relay path or Hop-To-Live (HTL) of minimum delay relay paths. The introduced HTL is to assist in predicting minimum and stable relay paths while minimizing the probing overhead. By considering delay, we concluded such relay paths are more stable and experience shorter delays. Further, such an HTL is used to quickly predict inexpensive relay paths for changes in network dynamics. From such predicted relay paths, our study proposes a performance analysis when using a hybrid TCP-UDP transport protocol instead of the current pure TCP one for services such as YouTube and VoIP calls. There is a QUIC protocol proposed as a general-purpose transport layer network protocol designed by Google in 2012. This QUIC uses multiplexed UDP streams distinct applications instead of TCP-based HTTP transmissions and loss recovery. Each stream is handled individually when a loss in datagrams occurs between connection ends, for which each lost datagram is re-transmitted over UDP. The hybrid TCP-UDP in our work asks for endpoints of a connection to simultaneously establish 2 separate connections of UDP and TCP. The TCP connection serves as a back-end data recovery layer and is invoked by end applications when a stream suffers from a loss. The UDP connection carries all in-queue streams in a multiplexed manner as with QUIC.

For such a hybrid TCP-UDP scheme we provide an analysis on a network of 140 nodes for an estimation-based and hybrid TCP-UDP streaming performance over specific bandwidth relay paths. Further, we have included an analysis of 18,906 delay traces from a network of 138 hosts to demonstrate a rich existence of IP relay paths that can be leveraged to significantly enhance our examined network routing performance. This work is an attempt to make the case for using layer-3 minimum delay forwarding and demonstrate the performance gain of such a hybrid approach when compared to existing and pure TCP overlay designs. This work is conducted to benefit applications that are sensitive to end-to-end delay and throughput. Here, we also present a specific analysis for the end-to-end delay to enhance TCP performance by increasing throughput and reducing file transfer time via relay paths of default TCP settings. Results show how such minimum delay paths can benefit TCP throughput and minimize transfer time by orders of magnitude. This work can be viewed as an estimation-based layer-3 hybrid transport forwarding analogous to QUIC. The details of HTL characteristics and hybrid TCP-UDP are detailed in Chapter 3.

The wireless sensor ad hoc networks have similar dynamics as in cellular networks such as interference, collision, latency, packet drop, channel capacity, and throughput. The work in Chapter 4 focuses on a dominant challenge in ad hoc networks called data collision. This collision occurs when 2 packets collide at a receiver node, and is a major cause of performance degradation. Time division multiple access (TDMA) scheduling is a well-studied subject for ad hoc networks

for managing collisions. Reinforcement Learning (RL), in particular, Q-Learning (QL) is used to achieve a collision-free TDMA schedule. However, a long scheduling convergence and complex reward functions are challenges for distributed QL-based TDMA scheduling of a minimum frame length.

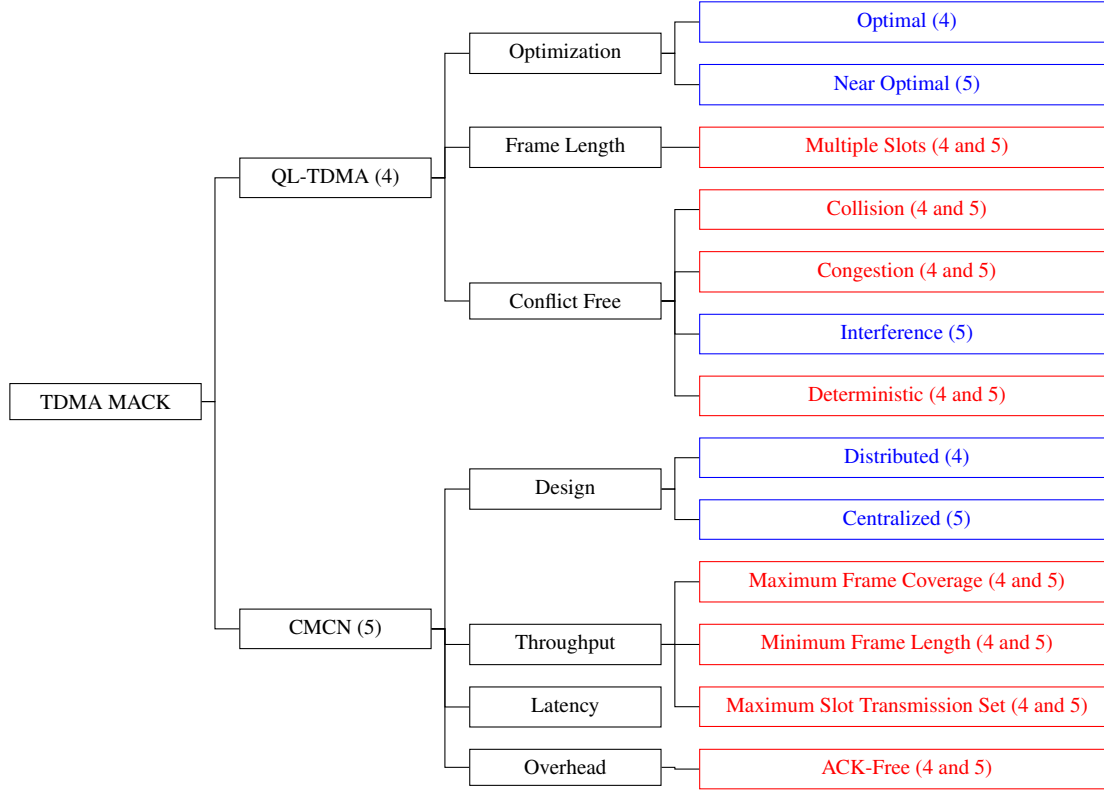


Figure 1.1: Proposed TDMA-Based MAC Protocols in Chapters 4 and 5

The Medium Access Control (MAC) protocol for Wireless Sensor Networks (WSNs) is an application-specific access such as an energy-based or delay-based access protocol. TDMA MACs schedule nodes over non-collided slots and aims for a minimum delay or number of slots. This consequently reduces the energy consumption of nodes. There are distinct classes of MAC protocols as classified into 5 classes: (1) The carrier sensing scheme, in which nodes sense a medium for active transmissions before sending. (2) Collision detection, in which nodes are capable of detecting collisions. (3) Collision avoidance avoids collisions and acts as a traffic controller for nodes. Techniques like prior scheduling of time slots, randomized access, and exponential backing off are examples of such a class. (4) For networks of high loss rates, the collision might remain undetected, and acknowledgments are useful in such a scenario. (5) The backoff policy in general is used to reduce contention by postponed transmissions to reduce collisions. The proposed QL-based TDMA (QL-TDMA) MAC is a collision detection and avoidance protocol based on idle listening without packet acknowledgments. For such an aim, we introduced a new frame structure,

in which only collision detection signals are allowed.

There exist many RL-based TDMA MAC protocols such as RL-MAC in [55], in which a reward function is a composition of minimizing energy consumption and ensuring a maximum throughput. The state in [55] is represented by the number of queued packets ready for sending by the start of each slot. The ϵ -greedy approach in [55] is to balance exploitation and exploration in RL. Their performance shows an efficient energy consumption and a more throughput gain when compared with SMAC as proposed in [110] and TMAC in [27]. This work, in contrast, avoids ϵ -greedy exploration to speed up convergence while relying on non-optimistic Q-tables from the start, and our states are defined as slot IDs instead. The work in [69] implements a similar reward strategy for successful and unsuccessful transmissions. The work in [69], however, allows contention over a single channel and is applied over a small scale of 99 nodes in contrast to our analysis on distinct topologies of a maximum of 1000 nodes.

This work proposes QL-TDMA as a distributed MAC protocol for ad hoc networks. This QL-TDMA is a collision detection and avoidance mechanism based on a new frame abstraction with no successful delivery acknowledgment allowed. This new algorithm is a distributed and slot count-based oriented weighting and cycle detection for wireless sensor networks with a reduced search overhead comparing existing schemes like Rocha–Thattai algorithm in [84]. The proposed QL-based TDMA applies a slot of length 2.5 milliseconds in a minimum frame length for any given topology. The reasons behind such a frame modification when using a minimum required frame are: (1) To achieve smaller communication delay. (2) Then such a minimum frame reduces the scheduling handshakes or acknowledgments, and so a minimum flow burden over such a frame is guaranteed. The literature is rich of conventional TDMA protocols that operate above minimum frame lengths and are designed as ACK/NACK-based schemes. The main aim is to allow each wireless node learns a successful policy that leads to a zero collision frame allocation. This occurs while maintaining no ACK and minimum NACK flow during such a learning process. Each node collects collision statistics from previous frames. This information is utilized to guide a minimum-length TDMA frame discovery. Each newly chosen slot for the next learning frame in our current work context is permitted to be examined only over the next frame and not a currently examined one. This enforced delay in slot access could increase the QL convergence time. However, for a node with a synchronized and just elapsed current sending slot of a current frame of a learning episode and a newly chosen slot, such a node can examine all possible slots between those two slots of the current frame to reduce such a convergence time. This work has shown how wireless nodes could benefit from collided signals to avoid new collisions, and while being in a listening mode can infer 1 hop and 2 hops surrounding collisions. The simulation has shown that wireless sensor nodes can learn a minimum length and collision-free frame within a practical convergence episode count. The discussion in Chapter 4 illustrates our QL-TDMA main concepts and performance

analysis. Figure 1.1 shows the performance and characteristics of QL-TDMA. Parts of Figure 1.1 have been discussed in [17].

For Collaborative Multihop and multichannel Cellular Networks (CMCNs) over shared bands, our work presents a new scheduling MAC to optimize cellular dynamics such as interference, congestion, packet redundancy, delay, and channel capacity. This MAC protocol is a near minimum delay TDMA scheduling and interference management for such collaborative cellular systems. This work presents 5 policy-based scenarios to summarize existing scheduling paradigms. Then, our paradigm provides a Minimum Broadcast Delay Forwarding (MBDF) and approximates polynomial scheduling for cellular providers to manage interference and congestion for wireless links or channels. The simulation shows that our approach outperforms existing models in terms of scheduling delay and message redundancy as discussed in Chapter 5. The confidence in our proposed scheduling has been examined over distinct and simulated sets of the Citizen Broadband Radio Service (CBRS) channels by comparing MBDF height (number of slots) with an initial minimum scheduling demand (minimum spanning tree height without considering conflict). This analysis of cellular dynamics is outlined in Chapter 5. For policy-6, novel complexity mappings and cellular conflict-solving algorithms are mentioned and detailed in Appendices ??, ?? and ?. Figure 1.1 illustrates also performance and characteristics of our proposed CMCN-TDMA.

This dissertation is organized as follows: In Chapter 2, we analyze a new layer-3 data forwarding scheme, in which we find an improvement for end-2-end delay. The examined relay characteristics such as the Hop-To-Live count of minimum delay relay paths and our hybrid TCP-UDP streaming are presented in Chapter 3. The work in Chapter 4 is a QL-TDMA slot allocation framework for distributed sensor networks. The novelty of the proposed QL-TDMA protocol lies in the following contributions. (1) This distributed model does not require any prior topological information between nodes. (2) The TDMA scheduling is converging under the minimum required frame length (number of slots). (3) This work adjusts the commonly used conventional collision handling protocols in such a way that the scheduling overhead between nodes is minimum. To the best of our knowledge, our protocol is novel in literature by using few control messages between nodes when avoiding all existing ACKs. The performance of our protocol is examined using a set of challenging topologies. In Chapter 5, we summarize 6 TDMA policies that summarize existing paradigms. Policy-6 approximates a polynomial and near-optimal TDMA scheduling for MBDF. The conclusion and future work are summarized in Chapter 6.

2 The Significance of Layer-3 Traffic Forwarding

This chapter discusses relay routing and our layer-3 data forwarding scheme. The scheme performance has been evaluated and compared with the state-of-art and best-effort IP performance. Through real-time experiments over Planetlab, we examined the significance of using single-session minimum delay paths to forward layer-3 encapsulated segments. This chapter provides a detailed abstraction of our probing mechanism and explains a new level of analysis for minimum delay relay paths.

2.1 Background

Traditional Internet connections are established via a set of underlying routing protocols at the IP substrate. Relay routing consistently can provide better end-to-end performance such as faster download times, lesser stream re-buffering [56], or higher throughput for end-users. Relay routing achieves this by exploring non-congested paths that are not necessarily discovered by the best-effort Internet routing protocols. This flexibility helps to direct traffic away from congested segments of the network and subsequently improves performance. Relay paths can be chosen based on different parameters including lower delay or drop rate, or higher bandwidth along the path. The main goal of this work is to improve layer-3 TCP performance using minimum delay relay routing.

For a given physical network topology, the following key points have been investigated in our analysis. (1) The existence of relay paths that can provide lower delay compared to the best-effort paths offered by the current Internet protocols. (2) The stability of such paths. (3) The additional resource burden for such relay paths that are generally longer than the best-effort paths. (4) The magnitude of improvement achieved by choosing such relay paths. By analyzing these points, our approach has achieved higher TCP throughput gain and file transfer delay reduction when comparing some physical and relay paths.

2.2 Literature Review

This work can be depicted as an experimental prototype to [74] for content delivery applications. The dominant Internet traffic is served over the connection-oriented transport protocol TCP [40]. However, our approach is not dedicated to TCP applications, but we analyzed its performance using TCP as a possible worst-case scenario for the following reasons. The TCP feedback control mechanism always starts with a small congestion window that impacts the number of packets in flight. Further, the three-way handshaking signals add new additional round trip times to any path latency [40]. Thus, relay paths must remain constant over long periods of time, and can change only when the QoS suffers degradation. In our study, the seven examined relay paths were reused over a period of 5 hours at a rate of 50 times per hour.

Based on the worst-case argument established above, our approach is applicable for live and high-quality video streaming such as Netflix and Voice-over-IP (VoIP) services like Skype. The

VoIP design in [49] can be viewed as an application of our approach. The authors in [49] provide no details about their relay structure that improves call quality by 45% to be close to an oracle-based solution, i.e., all metrics for all paths are known. Akamai’s relay in [92] and the study in [40] are further instances of the use of relay schemes to reduce the content retrieve-time in Data Centers Networks (DCNs), which can be a possible use-case for the relay design outlined in our study. Most of the literature on cloud services uses relay paths as demonstrated in [40], [92], [22], [56] and [54]. However, such studies do not specify their internal relay design, so we establish a comparison with this current study.

The study in [74] proposes a combined bandwidth and delay-aware routing scheme for enhancing the Internet QoS. Their emulated SDN controller takes bandwidth and latency measurements from a separate monitoring entity and pre-calculates appropriate routes for each pair of nodes in advance. This study, in contrast, analyzes an experimental relay topology that can serve high-rate content delivery applications using the single metric of delay calculated using active probing measurements. Interestingly, our design also has achieved higher throughput and delay reductions by a large magnitude over a set of examined relay routes. In the semi-heterogeneous topology discussed in [74], on average the results show that there was 22% of bandwidth increase and $[4 \rightarrow 15]$ RTT reduction.

2.3 Methodology

This section describes the used platform, performed experiments, and our designed procedure for conducting measurements over Planetlab.

2.3.1 Platform

Networking testbeds such as Planetlab and Emulab are distinct. Planetlab, for instance, connects global clusters, from which a user can subscribe to a set of nodes in a slice. The measurements in our work are conducted using 140 nodes distributed across the globe as: North America 63.57%, South America 4.29%, Australia 3%, Asia 17.86%, and Europe 12.86%.

2.3.2 Experiments

Experimentally, Traceroute and Ping were used to conduct 18,906 and 19,460 end-to-end measurements, respectively over a network of $n = 140$ nodes. Traceroute performs short-spaced measurements, i.e., $[5 \rightarrow 10]$ minutes interval. Ping sends a bulk of packets of four distinct sizes of 0.05, 0.1, 0.25 and 0.5 megabytes. These loads were scheduled in the same order 4 times so a result of 16 experiments are collected. Having a diverse measurement interval as suggested in [37] provides more confidence in capturing many possible routing changes.

2.3.3 Probing Procedure

Each probing experiment conducted in our research followed a specific probing paradigm as illustrated in Figure 2.1.

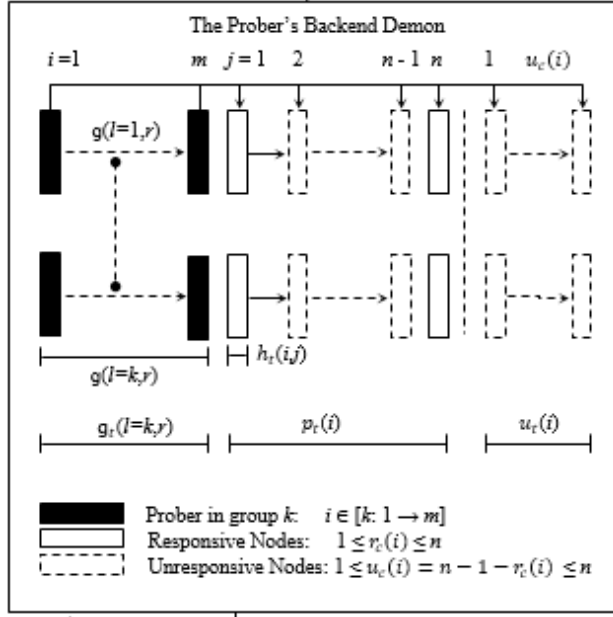


Figure 2.1: Probing Abstract

2.3.3.1 Probing Daemon

The conducted measurements follow the exact probing abstract illustrated in Figure 2.1. Nodes were divided into a number of groups. There is a single measurement at a time performed in each group g_i where $i \in [1 \rightarrow m]$. Each prober runs into two probing cycles: The first, is to probe all $n - 1$ nodes, and the second one is to re-probe unresponsive nodes again. The actual group time is: $\tau_i = \sum_{k=1}^{|g_i|} \psi_i^k + \rho_i^k$. For a group g_i , $\psi_i = \psi_i^1, \psi_i^2, \dots, \psi_i^k$ and $\rho_i = \rho_i^1, \rho_i^2, \dots, \rho_i^k$ are random processes that maps the probing times of the two cycles. Each ψ_i^k or ρ_i^k is a random variable for the times of a prober k . These random variables can be approximated as: $\psi_i^k = \sum_{j=1}^{n-1} \bar{\epsilon}_k$ and similarly $\rho_i^k = \eta \sum_{j=1}^{\theta_i^k} \bar{\epsilon}_k$ where θ_i^k is the number of unresponsive nodes in the first loop. The $\bar{\epsilon}_k$ is the average probing time of prober k of the network. The η is the average count of re-probing a node in our case. This probing scheme is used a server-based synchronization to minimize the effect of the probing conflict that may occur when two daemons or more probe a particular node simultaneously. To reduce such imperfect measurements, we forced all daemons of groups to randomly probe all nodes. This will also minimize active probes to probe each other during a particular instance of time.

2.3.3.2 Probing Success

Furthermore, we defined the probability of success in reducing the influence of probing conflict on measurement accuracy. This probability concerns $n - m$ nodes when no node is coincided of probes of more than one prober of m active ones. The probability of success with no conflict in

probing is approximated as:

$$P(m) = \prod_{i=1}^{m-1} \left(1 - \frac{i}{n-m}\right) \quad (2.1)$$

Here, n and m again are the numbers of nodes and groups respectively. Practically, m represents the number of active probers that can probe the network within a particular time. Therefore, m must be chosen carefully to satisfy a desired success probability. For simplicity in computation when solving for an exact solution for m that satisfies a given demand of success, we can approximate the probability of success in equation 2.1 when $m \ll n$ by:

$$P(m) = \exp\left\{-\frac{m^2}{2(n-m)}\right\} \quad (2.2)$$

Thus, solving for success demand equals 70% leads to $m \approx 10$. For our network of $n = 140$, we found $m \approx 10$. Having 100% of success causes m to converges to one according to $\lim_{m \rightarrow 1} P(m) = 1$ as expected.

2.4 Contribution

The main contributions of this work are as follows: The first, we experimentally demonstrate that in many situations, there exists a rich topology of relay paths that can provide lower Round-Trip-Time (RTT) compared to the best-effort physical paths. The second, we characterize these relay paths in terms of their benefit, temporal stability, and link consumption, i.e., time-to-live (TTL). Finally, we demonstrate that such lower delay relay paths can eventually be leveraged to improve applications layer performance. Here, we demonstrate the latter using File Transfer Protocol (FTP) as a target application while considering transfer time and throughput as performance metrics of the application layer. Packets in a segmented TCP approach suffer from an additional delay at the application stack of every relay node and would require a robust store-forward buffering design in order to minimize the re-transmission times. Furthermore, buffers are required to be coupled with back-pressure schemes to avoid flooding the slowest link at higher rates. This research, in contrast, proposed a slightly different approach to eliminate the previous cost. The implemented IP-layer forwarding preserves all the current properties of the end-to-end TCP protocols except redirecting traffic through other paths that experience the minimum delay among all possible.

2.5 Proposed Relay Forwarding

Physical paths are established via a set of underlying routing protocols that use IP links. In contrast, relay paths use another layer of protocols to completely or partially navigate data away from congested physical links. Long distance ASes might not exchange routing info for a long time. Therefore, for a congested physical end-to-end path better paths could remain undiscovered via IP protocols. Using relay routing allows nodes to be aware of better paths.

2.5.1 Minimum RTT Relay Forwarding

Relay communications are either at the application layer such as application-multicast or the network layer such as IP-multicast. The application layer relay is a multi-session forwarding since each node is re-sending the same data to the next-immediate relay node (source-destination IP addresses are changed per hop). In contrast, layer-3 relay paths allow sources to efficiently send traffic over a single session. This relay scheme is implemented as a single session layer-3 forwarding.

2.6 Discussion and Results

This section summarizes our results and analysis as follows: Section 2.6.1 shows the real-time achieved TCP performance. This gain has been achieved after implementing the proposed relay on seven long distance relay paths that forward 5 megabytes of content between their end nodes. In Section 2.6.2, we provide an abstract characterization for the existence of possible minimum delay relay paths between all available source-destination pairs, and further evaluate the stability of this characterization across the network. The discussion in Section 2.6.3 illustrates the variation of the delay reduction between sources. For relay networks that compete for small end-to-end delay, there is always a trade-off between the desired delay reduction and the burden of utilizing links. In Section 2.6.4, we explore the total link consumption via both the underlying substrate and the obtained relay paths. In Section 2.6.5, however, we show the trade-off between the link consumption and the end-to-end delay. The illustration in Section 2.6.6 is a distribution of the RTT reduction at the node granularity.

2.6.1 Relay and TCP performance

The real-time implementation of the proposed single session and minimum end-to-end delay relay routing was an essential stage in providing some insight into its importance and usability. Following the previous analysis, seven relay pairs have been selected and configured to transfer the 5 megabyte file in between. We argue that although this number is small because of Planetlab IP layer restrictions and the inflexibility using [37], our achieved performance via these paths can be considered as a generalization when using the same design. The comparison in Figure 2.2 shows variation in the TCP throughput using the best-effort IP routing and the proposed relay design for the random set of relay paths with different delay benefits.

Regarding TCP throughput, we notice a considerable improvement for p_1 , an increase of a magnitude for p_2, p_3, p_4 , and p_6 while slightly maintaining the same physical throughput for p_5 and p_7 . The main reasons behind such a leading performance of some paths are as follows: First, since the achieved throughput is inversely proportional to RTT, these minimum delay paths experienced less delay so that for any given TCP congestion window the throughput is maximized. The reason behind maintaining the same physical throughput for p_5 and p_7 is that relaying via those two paths originally provides small delay reductions, and so is the throughput improvement. Figure 2.2

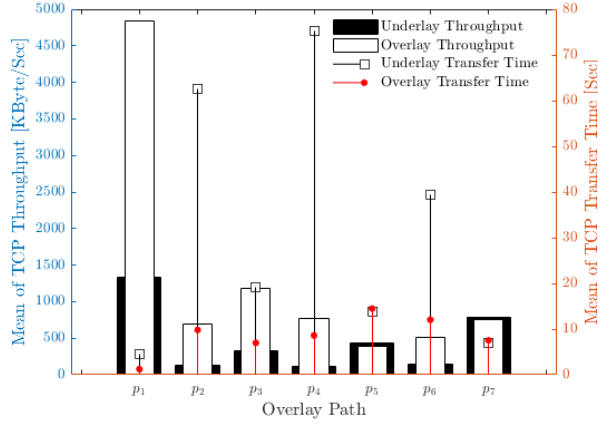


Figure 2.2: TCP Performance

also indicates that the time required to send 5 megabytes of content as a second measure for TCP performance shows similar behavior to the corresponding throughput. The transfer time has been significantly reduced except for p_5 and p_7 , which provide minor delay reductions. This simple design overcomes the multi-segment and the multi-path forwarding by not requiring an extensive packet buffering and reordering at the receiver side or packet encapsulation at the application layer.

2.6.2 RTT Behaviour in Routing

Relay paths are expected to be clustered around smaller RTTs when compared with the physical RTTs that can be seen distributed across the entire delay range. Therefore, the relay delay distribution curve should be clearly shifted to the left of the physical distribution along the entire range, and peak as much as possible near the smallest delay values of the range. In contrast, the two curves are expected to be aligned with each other as the RTT increases. Figure 2.3 confirms such an argument, and show both RTT distributions over the 24 hours measurement period.

By considering the most commonly observed round-trip-times and excluding all delays of the one-hop overlay paths, the plot shows that there is only 2.8% of underlay paths out of 18,906 with 40 milliseconds end-to-end delay. The overlay routing, however, reduces the end-to-end delay to about 25 milliseconds for 2.6% of the paths and raises the number of paths that experience 40 milliseconds to 4% instead. The overlay competing for shorter delays exists up to the 500 ms mark, after which overlay routing starts yielding to underlay paths as their PDF curves are perfectly aligned.

The range between $[10 \rightarrow 500]$ ms is commonly observed and solved by the proposed overlay, as the percent of overlay paths collapses outside this range. In this analysis, we only considered overlay paths whose end-to-end delays are less than 500 ms, and that is why the CDF curves do not approach one. The CDFs also show that for 250 ms delay, 49% and 40% are overlay and underlay

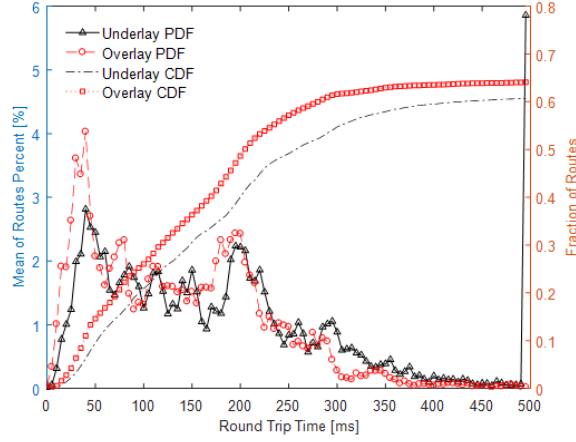


Figure 2.3: Physical v.s. Relay RTT Behaviour

paths, respectively. However, we found that two hop overlay paths can provide distinct delay reduction for 35% of the 18,906 end-to-end connections, i.e., more than the current IP routing can perform.

2.6.3 Magnitude of RTT Benefit

Relay controllers should be able to enforce routing decisions based on metric benefit change. Here, we divide the entire relay design into four delay sets. The set S_1 represents relay paths that reduce the physical end-to-end delay by more than 100 milliseconds. In S_2 paths of delay reduction within the range $[10 \rightarrow 100]$ ms are included.

Failures indicate either one of two different scenarios. The first scenario is that a physical path has no better overlay alternative while the second one is that a physical failure cannot be resolved even by overlay routing. These two scenarios are represented by the sets S_3 and S_4 respectively. Figure 2.4 shows a variation example of this reduction of different nodes.

The overlay RTT benefit is indicated by the degree, to which the overlay distribution is shifted to the left of the underlay. The higher overlay benefit is achieved for nodes where the overlay or underlay distributions are toward the higher RTT values is evident. This range of RTT reduction is further analyzed in 2.6.2. The argument here is that our large scale measurements represent this range reasonably enough for wide area networks because of the heterogeneous resources in Planetlab, and the underlying routing becomes minimum beyond this range. By looking at the relationship between S_2 and S_3 , we can infer the following. For a node a , \bar{a}_{S_i} is the number of minimum delay paths that start from a in S_i . The delay benefits for any node have been found to be proportional with $\bar{a}_{S_2} - \bar{a}_{S_3}$. Thus, as described in Section 2.6.2, the RTT distribution will be more left-shifted. Nodes with small benefits have almost overlapped RTT curves. The analyzed Traceroute experiments have shown identical behavior between the benefit curves. Figure 2.4 only

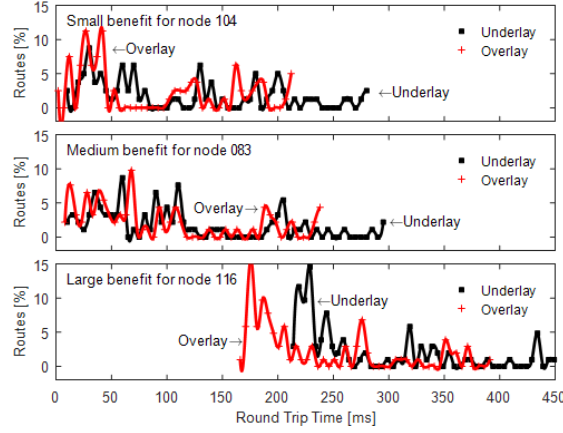


Figure 2.4: Node RTT Benefit

classified 3 nodes in benefit.

2.6.4 Burden of Link Usage

There is always a tradeoff between minimizing the end-to-end delay and the TTL count or the number of physical links used to achieve such an objective. To understand such behavior we analyzed link consumption for both the physical and relay layers. Figure 2.5 has an averaged link consumption for both layers.

The underlay link consumption is approximated by $\mu = 16.5$ and $\sigma = 4.7$ while overlay consumption corresponds to $\mu = 33.7$ and $\sigma = 14$. The μ and σ are the mean and standard deviation of link distributions respectively. From this result, surprisingly, there still exist overlay pairs that communicate under minimized end-to-end delay while maintaining the common range of the underlay link consumption is below 30 links. The peak at 30 links for the link consumption represents both the underlay paths of 30 links and all failed paths.

There is also strong stability in the underlay distribution over 24 hours compared to the over-layer distribution since the link consumption over 24 hours does not deviate much from their means that are represented vertically on the two curves. This indicates that any delay-aware overlay routing can be more greedy but sensitive to delay changes than the traditional IP protocols, and that is because of its global view of the network delay.

2.6.5 RTT and Link Usage

The discussion in Section 2.6.4 defined the amount of overlap between the link usage in both the physical and relay paths. Here, however, we analyze the relationship between the previous link usage and the end-to-end delay reduction. For a given end-to-end path p , we refer by Δ_d^p to the delay difference in RTT between the underlay and the overlay paths $r : i \rightarrow j$. That is $\Delta_d^p = \underline{\text{RTT}}(p) - \overline{\text{RTT}}(p)$. Here, $\underline{\text{RTT}}(p)$ is the physical RTT while $\overline{\text{RTT}}(p)$ is the relay RTT.

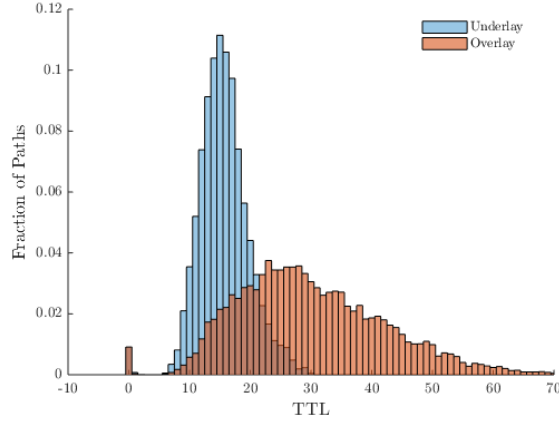


Figure 2.5: Physical v.s. Relay TTL Distributions

Based on this definition, $\Delta_d^p \geq 0$ with equality if the overlay path is just one hop (same as the underlying path). Therefore, when $\Delta_d^p > 0$, that means the overlay path is at least two hops in length: $\Delta_h^p = \overline{\text{HTL}}(p) - \underline{\text{HTL}}(p) \geq 2$ hops. The $\overline{\text{HTL}}(p)$ represents the length in hops of the overlay path while $\underline{\text{HTL}}(p)$ represents the underlay path and is always equal to one. Similarly, at the link granularity, the introduced Δ_l^p on the other hand, can be either positive or negative. When $\Delta_l^p = 0$ it does not necessarily imply that for the given connection its overlay and underlay paths are identical, simply because an overlay path of at least two hops can use in total the same number of links used by the corresponding underlying one-hop path. In this analysis, we only considered the overlay paths that are not in any of the following underlay-overlay combinations: failure-recovery, failure-failure, or recovery-failure, where the last one is related to the underlying paths that have minimum RTT. That is why the cumulative CDF in Figure 2.6a above does not approach one. This result shows a constrained Δ_l^p by a desired delay reduction or Δ_d^p . From Figure 2.6a, we concluded that the $[1 \rightarrow 25]$ ms is the range, on which we can maximize the performance of the overlay routing at the smallest link consumption.

Ideally, as the configuration of the overlay hops consumes less number of hops, the expected link consumption should not be high. The analysis also has looked at the relationship between the three metrics: delay reduction, link, and hop consumption across the entire network. The result in Figure 2.6b shows that throughout the entire measurement period on average, the overlay topology uses two hops of an average link consumption within $[10 \rightarrow 15]$ links in order to achieve an average delay reduction of 30 milliseconds.

2.6.6 Node Mean RTT Reduction

Previously, we discussed the gain in RTT or delay reduction Δ_d^p only at the path p scale. The expectation at the node scale is that the aggregated Δ_d^n for nodes that act as sources will show a

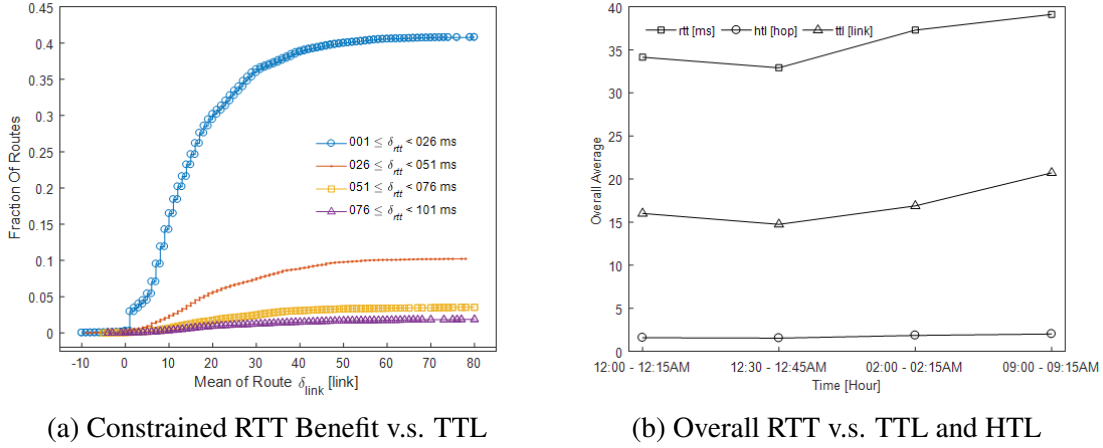


Figure 2.6: RTT v.s. TTL and HTL Usages

peak at the most shared reduction between them. Since we have only considered the overlay set S_2 , this lump will reside somewhere in the range $[1 \rightarrow 100]$. In a network of n nodes, and for a node i in particular, we define the average delay reduction per node as:

$$\Delta_d^n(i) = \frac{1}{n-1} \sum_{i' \neq i} \Delta_d^p \forall p : i \rightarrow i' \quad (2.3)$$

Figure 2.7 analysis shows a complete view of $\Delta_d^n(i)$ of the first Traceroute experiment peaked at 25 milliseconds. This means that the averaged delay reduction range, at which all sources are functioning, is $[22 \rightarrow 27]$. This, of course, does not mean that the overall benefit from implementing an overlay will be only within that range, but on average, this defines a trend of each node.

2.7 Conclusion

This study analyzed the behavior of single-session relay routes that minimize the end-to-end delay for all possible TCP pairs in the network. Our analysis has been derived for the most observed and common range to reduce the delay of $[1 \rightarrow 100]$ ms. The study has confirmed the existence of a considerable number of relay routes that can minimize the physical round-trip times by the mentioned reduction range for the majority of nodes. The advantage of this relay design varies from node to node, based on their topological positions. The study summarizes the trade-off between minimizing the end-to-end delay and the consumption of the network resources by showing the lack of delay reduction using fewer links than the underlying paradigm. In the future, we will investigate the difference in reduction between this minimum delay relay routing and another relay design that will optimize the overall network delay, while being less greedy in consuming network resources.

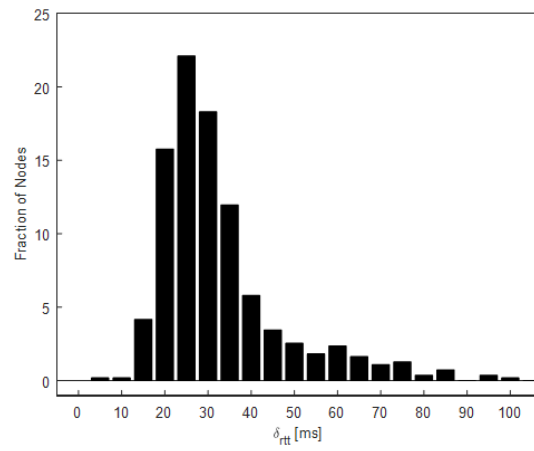


Figure 2.7: RTT v.s. TTL and HTL Usage

3 Hop Characteristics for Relay and Hybrid TCP-UDP Services

Presently, certain relay schemes for IP networks have been well-documented in past years. However, the implementation cost of relay solutions has not yet been conclusively identified. The active relay relies on periodic probing of routing changes for enhanced performance. In contrast, static relay routing uses non-periodic probes to measure latency and packet loss at a lower rate. The ultimate objective of an active relay is to maximize QoS while a static relay seeks reliability in routing over time. There exists considerable research focused on understanding IP link changes due to fluctuations in routing dynamics. This work, in particular, examines characteristics such as the number of hops in a relay path called Hop-To-Live (HTL) of minimum delay relay paths. The work introduces HTL to assist in predicting minimum and stable relay paths while minimizing the probing overhead cost. By considering delay, we find those relay paths are more stable and short. Further, HTL derives inexpensive relay paths for bandwidth changes. This study proposes a performance analysis for using UDP as an alternative to TCP-based services such as YouTube and VoIP applications. This work provides an analysis of streaming performance on a network of hundred-forty nodes using HTL estimation-based relay schemes that offer better relay paths for UDP streams.

3.1 Background

Many performance relay solutions are active while relying on continuous probing for measuring routing changes. In contrast, for the static relay, routing metrics such as latency and packet loss are not continuously measured. The ultimate objective of the static relay is reliability so nodes remain connected. However, the active relay is used to improve QoS periodically. The study in [71] focused on understanding Time-To-Live (TTL) changes. This study, in contrast, examines the influence on the Hop-To-Live (HTL) of the relay substrate as the Round-Trip-Time (RTT) changes. There is a cost concern when using relay paths such as the probing overhead and path configuration speed, at which a relay path is established and ready to forward traffic. In earlier systems, the lack of privilege at the IP layer forced relay networks to be designed at the application layer, such as the application multicast. However, relay costs remain high although recent approaches like SDNs offer such privilege. This work provides an analysis of our HTL estimated-based relay to reduce that relay cost.

3.2 Literature Review

The study in [49] refers to an estimation-based relay scheme for Skype users. The skew in data density mentioned in [49] is due to the lack of measurements (samples) for end-pairs and was replaced by network tomography-based delay estimation. This approach cannot be generalized and replace the probing overhead required for achieving a clear view of network performance. For a tomography-based estimation, our study is a counterexample, in which we found both physical and relay paths are asymmetric in general. Hence, a benefit of relay performance using a tomography-

based scheme might not meet the desired demand for end-users because of lacking delay symmetry. Researchers in [60] have used the same measurements to construct a Layer-3 forwarding scheme for data transfer on small scales. The considered relay paths were selected according to their HTL stability. The difficulty is in performing direct probing for large network studies [51] like [95] to show that it is possible to infer network conditions based on content distribution networks [3] with relaying. The work in [26] is an example of such a scheme. This study is a specific implementation of [103] and [71], in which the authors focused on examining the basic problem of QoS routing for multimedia applications by finding a path that satisfies many performance constraints. In our study we consider a new streaming direction using UDP as the main relay protocol instead of TCP for YouTube and VoIP applications like Skype.

The work in [38] illustrates more stability for physical IPv4 paths than IPv6 ones. This inspired us to further examine the stability of IPv4 relay paths. In [34], authors highlight the importance of new schemes for predicting physical RTTs, which supports our claim for a need for new estimation-based relay schemes. The study in [52] uses relay path stability and symmetry characteristics to overcome the inefficiency of the relay when not considering certain physical links. The introduced stability assists in finding more efficient relay paths. In contrast, we used the HTL count as a stability measure to represent the entire relay path structure in our study. In [71], Paxson examined end-to-end behaviors due to the different directions of physical paths, which often exhibit asymmetries. This author characterized the prevalence and persistence of physical paths. Here, we performed a similar analysis for relay paths. The authors in [41] examined path diversity on relay networks. They used fifty Planetlab nodes to conduct Traceroute measurements. They concluded that the benefit of relay performance is limited by the natural diversity of redundant paths between two end nodes in terms of physical links, routing infrastructure, administrative control, and location. This motivated our characterization of the relay HTL count through prevalence and redundancy. The study in [32] shows that the mean of per-hop delay between parent and child nodes in a relay tree decreases as the level of the host in the relay tree increases. This is because current physical routing is not optimized in terms of delay.

3.3 Methodology

For analysis, we implemented a similar probing procedure as discussed in Section 2.3.3, and included an additional IPerf bandwidth dataset.

3.3.1 Experiment Dataset

Many useful bandwidth datasets such as [49] are private. Public datasets such as CADIA [1] and RIP-NCC [2] have large-scale but demand-based bandwidth traces. Therefore, for meaningful bandwidth estimations on large-scale, we examined a network of hundred-forty nodes. These nodes are universally distributed as categorized in Section 2.3.1. In sum, we analyzed 311,360 Ping traces and 177,144 IPerf bandwidth measurements.

3.3.2 Experiments

Ping performed 19,460 and IPerf conducted 14,762 end-to-end path measurements over a network of hundred-forty nodes. Ping used a bulk of packets of four sizes of five, ten, twenty-five, and fifty percent of megabytes. These packets were used to examine changes in the number of relay hops or HTL. Packets were also scheduled over a constant order four times for sixteen experiments. Having such various measurement intervals as suggested in [71] provides more confidence when monitoring possible routing changes.

3.4 Contribution

The major contribution of our work identifies a short and long-term analysis for minimum RTT paths such as the long-term stability discussed in Section 3.5.1 and HTL alternation sequences detailed in Sections 3.5.2 and 3.5.3. Probabilistic relay attributes such as prevalence are studied in Section 3.6. For analyzing HTL characteristics, we performed 311,360 RTT measurements for all paths in a network of hundred-forty nodes. Through analysis, we found that the HTL prevalence shares a similar behavior to the TTL prevalence studied in [71]. Hence, we conclude that both HTL and RTT are sufficient routing metrics for predicting relay changes and reducing overlay costs. This chapter discusses the performance of two distinct approaches for a hybrid TCP-UDP transport layer as a substitution for only TCP streams and VoIP services. The remainder of Chapter 2 is structured as follows. The HTL stability over twenty-four hours is analyzed in Section 3.5. In Section 3.6, a detailed examination of relay prevalence is introduced. Dominant relay redundancy and type are described in Section 3.8. The proposed HTL detection scheme is briefed in Section 3.9. The hybrid TCP-UDP streaming performance is elucidated in Section 3.11. This analysis is concluded in Section 3.12.

3.5 HTL Stability

Here, we discuss both short and long-term HTL stability attributes. This section also introduces new HTL transition sequences for understanding the nature of HTL changes.

3.5.1 Overall HTL Trend

This section provides a complete identification of HTL changes over the measurement period. The First Ping experiment was considered a baseline because of containing all possible HTL variations for monitoring HTL behavior. Here, we classify eight HTL sets. Each set represents an average stability measure of a subset of 19,460 relay paths for a particular HTL. Meaning each one contains paths whose HTL equals the set's ID. Each path of these sets was randomly traced over ten and fifteen minutes. The horizontal axis of Figure 3.1 indicates the probability of change, by which a set of paths changes over time. For example, $p = 1/15$ indicates a single change over fifteen measurements. Each set of relay paths is described by a combination (p, x) of the probability of switch p and switch type c . This c represents a decrease $-$, same $=$, or increase $+$ in HTL. The subset of paths represented by $(0, +)$ with zero probability switches to longer HTL. The

subset $(0, =)$ indicates a hundred-percent change either to shorter or longer HTL. The $(0, -)$ similarly, refers to an impossible change to a shorter HTL. The change in IP routing dynamics forces such a range of HTL instability for relay routing. Hence, before discussing the actual result in this section, we define an ideal case for a stable HTL of a set of relay paths. The combination $(0, +)$ and $(0, -)$ is maximum at zero while the $(1, =)$ peaks at one. Thus, any analysis of such a nature of change should approximate such an ideal model for concluding by a highly stable HTL set of relay paths.

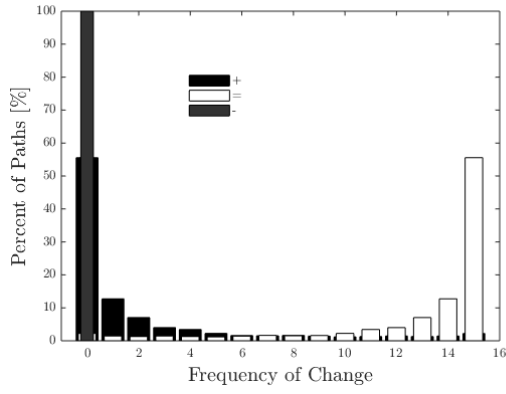
Figure 3.1 as expected shows for all sets, paths follow an exponential decay when switching to higher HTL values. The exponential curve starts to collapse as a set's ID or HTL increases. This means that as HTL increases, relay paths tended not to rise lengths throughout the observation period. For sets of HTL equals two and three hops, we noticed an approximation for the mentioned ideal model with a considerable decrease in all combinations. However, for sets of HTL equals three and four hops, always being at the same HTL has a small chance, and that is why the $(1, =)$ bar decreases and causes a trend of decreasing HTL until peaking at $(1, -)$. The overall analysis of the entire behavior of all HTL sets while focusing on $(p = 0, c)$, is concluded with three important curves.

The first curve refers to the $(0, +)$ combination that starts near fifty percent in the first set and rises toward a maximum at the eighth set. This indicates that as HTL increases, paths tend not to change to longer hops. The second curve for $(0, =)$ starts at zero percent, and with a positive slope also reaches a maximum at the eighth set. That means relay paths never stay at a fixed length as HTL increases. The third curve for $(0, -)$, however, with a negative slope collapsed in the sixth set. This means as HTL between two and six hops increases, paths tend to reduce lengths. The HTL stability analysis is summarized as follows. Each path of four hops or less is dominant in overlay, and either remains at constant HTL or becomes shorter. This tendency to reduce HTL is uniform, in particular, for a path of three hops or less. Predominantly, a range between two and four hops is seen as common for minimum delay overlay paths. Beyond four hops, paths have non-constant HTL values.

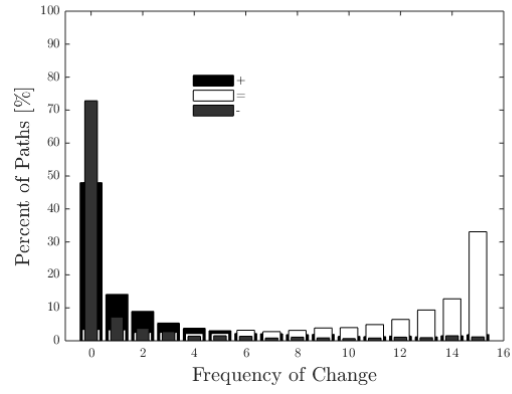
3.5.2 HTL Frequency Sequence

The HTL oscillations occur because of changes in IP dynamics. For an underlay path, we argue that with precise path measurements at random intervals spaced over a sufficient period, observing all available relay paths is possible. This is because the External Border Gateway Protocol (EBGP) only exchanges routing advertisements between adjacent Autonomous Systems (ASes), and as a result, non-neighbor ASes have no bearing on EBGP.

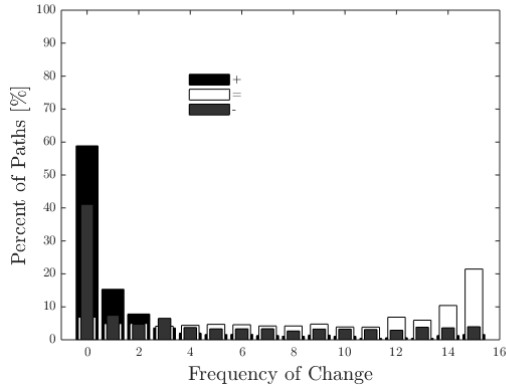
Failures of a physical path can last for two-hundred seconds [71]. During such a period on average, we conducted measurements for some outages and deployed minimum delay relay paths for developing performance similar to [60]. The semi-Markov chain for modeling physical fluctu-



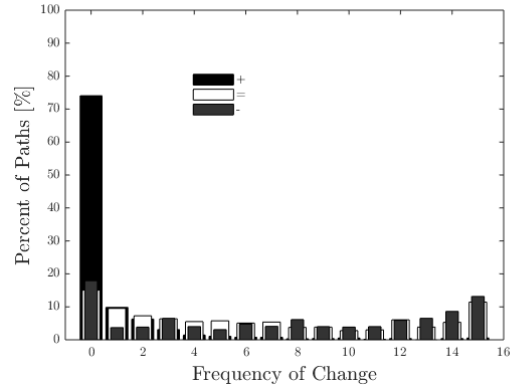
(a) HTL Equals One for 6,884 Paths



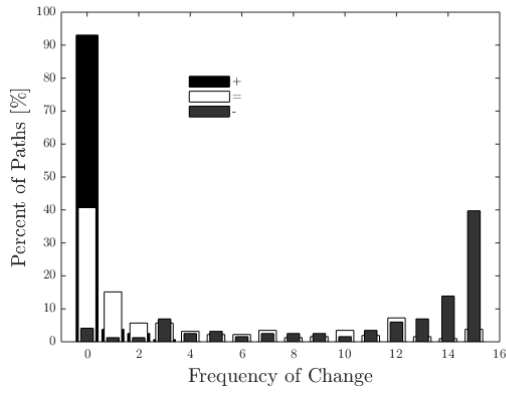
(b) HTL Equals Two for 7,165 Paths



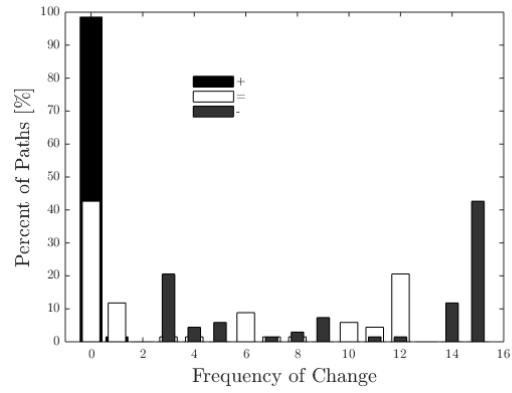
(c) HTL Equals Three for 3,711 Paths



(d) HTL Equals Four for 1,302 Paths



(e) HTL Equals Five for 317 Paths



(f) HTL Equals Six for 68 Paths

Figure 3.1: The Overall HTL Stability

ations makes each state of such chain depend only on a random lifetime drawn independently from a state distribution. Therefore, the steady-state probability of a particular state equals the average

time spent in such a state [71]. Here, each sequence of hop measurements M_h defines states of a particular relay path while each state represents a path of a particular granularity.

The HTL values obtained from a series of measurements M_h can be modeled as a Markov chain. The h here refers to the hop granularity of a relay path. For instance, having $M_h = \{1, 1, 2, 1, 2, 3, 1, 2\}$ of HTL samples where each unique h is an HTL value or state ID. For an interval τ seconds between consecutive measurements, we could perform the following. Possible HTL transition pairs within $(M_h - 1) \times \tau$ are $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 3)$ and $(3, 1)$. For HTL equals one, the corresponding state lifetime is $4 \times (M_h - 1)^{-1}$. The transition $(1, 2)$ occurs with a seventy-five-percent chance. This model updates the Markov chain for a given M_h as more HTL samples come in for a relay path. Further, M_h maps each overlay path onto a transition process composed of possible states of better relay paths. This model reduces the probing overhead by considering changes within a pre-estimated relay sequence.

Furthermore, as we varied the probing rate for conducting a measurement sequence, we were interested in understanding two main concerns. (1) The overall HTL missing rate. (2) The probability of not observing a particular HTL. To address our first concern, we introduced for each physical path a second HTL sequence of the most probable HTL values so that whenever probing is required, only paths with such HTL values will be investigated. This sequence is called the frequency sequence, F_h . Thus, for a physical path, if $M_h = \{3, 3, 5, 2, 4, 5\}$, $F_h = \{3, 5, 2, 4\}$. Finding F_h for every path in the network shows that only around five percent of 19,460 paths with at least one multihop relay path suffer from HTL misses. The interaction between physical and relay substrates causes exactly sixty percent of such five-percent relay paths to suffering from high fluctuations during our measurement period.

During analysis, we found 8,863 paths whose F_h sequences demonstrated multihop paths (relay was always better) and suffered no HTL miss. Furthermore, additional 5,824 paths with at least one minimum delay physical candidate still included in F_h also suffered from no HTL miss. This indicates about seventy-five percent of paths have changed HTL within a stable F_h of our measurement period. By excluding constant physical paths, we found that only 949 paths suffered from HTL miss. This number is small when compared to all 19,640 paths. This percent of paths for each HTL miss is in Figure 3.2.

The Ping bulks were proposed for our measurements to introduce congestion and force path changes. However, regardless of a two percent packet drop on some paths, two percent of such paths out of the total physical paths preserved a constant length of one hop. The geographical location of their ends is a strong reason for such behavior. Larger HTL values were less frequently observed and yet not common for some relay paths.

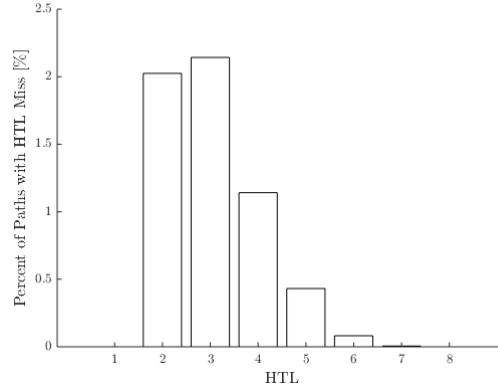


Figure 3.2: Overall HTL Miss

3.5.3 HTL Transition Sequence

The discussion in Section 3.5.2 neither investigates the nature of HTL changes as gradual or random nor how often HTL misses occur. There is no evidence indicating a gradual transition in path symmetry at the node granularity. The new sequence T_h considers timing for identifying HTL misses that occur between consecutive measurements. The M_h is a subsequence of T_h , and as a result, T_h can be generated by placing all in between missing HTL values. For $M_h = \{1, 3, 1, 2, 4, 1\}$ at the HTL granularity, $T_h = \{1, \bar{2}, 3, \underline{2}, 1, 2, \bar{3}, 4, \underline{3}, \underline{2}, 1\}$ is a corresponding transition sequence. The upper bar indicates a miss for switching to a higher HTL, and we used a lower bar for a change to a lower HTL. The HTL of two and three hops is not a miss in M_h but a miss in T_h . This F_h allows catching all missing HTL counts, but neither determines a probable miss-time nor common path fluctuations behind any transition miss.

Nonetheless, T_h indicates that when a particular HTL is a frequent transition miss, such HTL value is unfavorable. Each HTL can show as a miss in T_h while not in F_h . For $M_h = \{3, 1, 3, 3, 3\}$, there is a corresponding $F_h = \{3, 1\}$ and $T_h = \{3, \underline{2}, 1, \bar{2}, 3, 3, 3\}$. Here, we consider T_h as a miss-free sequence even with a miss chance of HTL of two hops equals fifty percent since such HTL is noticed in F_h . However, an $M_h = \{1, 3, 3, 2, 3\}$ has $F_h = \{1, 3, 2\}$ and $T_h = \{1, \bar{2}, 3, 3, 2, 3\}$. This T_h misses the two hops HTL by a twenty-five percent miss chance, and we consider such T_h as a miss sequence because F_h includes two hops. For HTL of two hops, about ninety percent of our relay paths suffer from no miss during path switching. Figure 3.3 shows a CDF breakdown of the transition miss for the common and observed HTL values. From such a plot, we conclude that the nature of HTL switching is a gradual transition that follows the congestion status of physical links.

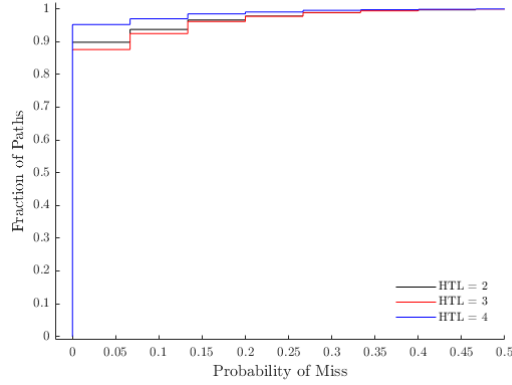


Figure 3.3: HTL Transition Miss Breakdown

3.6 Dominant Path Prevalence

Here, we define dominant prevalence sets for a physical path p with a set of physical measurements $\mathcal{P}_p = \{p_1, p_2, p_2, \dots, p_M\}$ of size M and a corresponding set of minimum delay relay paths $\mathcal{R}_p = \{r_1, r_2, r_1, \dots, r_M\}$. These dominant sets are $\mathcal{D}_p(N)$ for any $N \in \mathbb{N}$. For each physical path p , we define also N_p^* as a maximum number of different dominant sets observed of \mathcal{R}_p to bound $N \leq N_p^*$. Each $\mathcal{D}_p(N)$ contains a subset of \mathcal{R}_p of relay paths with a unique occurrence frequency $\omega_p(N)$. For instance, $\mathcal{D}_p(1)$ contains all prevalent relay paths observed with the highest frequency or $\omega_p(1)$, whereas $\mathcal{D}_p(2)$ contains paths with the second highest frequency or $\omega_p(2)$. The utmost number of dominant sets is $N_p^* = \arg \max_N \sum_N \mathbb{1} \mid \mathcal{D}_p(N) \neq \emptyset$. The maximum possible $N_p^* = M$ occurs when each path of M_h is unique. Furthermore, we describe source prevalence as a more robust indicator of stability than the general HTL stability addressed in Section 2.5. These characteristics of stability are manifested in the overlay structure of p over time. The HTL transition sequence we discussed in Section 3.5.2 is a Markov process whereby a state represents an HTL value. Each relay path of \mathcal{R}_p is also a state of a Markov process. Paxson in [71] states that we can simulate the steady-state probability by observing a state that is equivalent to the duration spent in such a state.

The prevalence is defined by the steady-state probability of the most frequent relay observation of \mathcal{P}_p during the measurement period, as in [71]. Finding the first or second dominant relay path requires careful and frequent analysis of p at the link granularity, or g_k . For our work and similar to [71], we examined such characteristics at each routing granularity such as node, autonomous system, city, HTL, physical RTT, and relay RTT to determine the dominant relay path stability and prediction accuracy of each granularity. Each granularity in the discussion is abbreviated in order as g_n, g_a, g_c, g_h, g_d and g_d^* . Higher level granularity, such as g_n avoids analysis burden at g_k , resulting in quicker estimations for r . Both g_h and g_d have not been examined in related research

for estimating routing changes. Furthermore, g_h and g_d are not location-dependent like g_n , g_a and g_c . For demonstration convenience, we analyzed the conditional probability of each sample of \mathcal{R}_p . For any physical path of an observed dominant relay set, $\mathcal{D}_p(\mathbf{N})$, we define path prevalence as $\rho_p(\mathbf{N}) = \mathbf{M}^{-1}\omega_p(\mathbf{N})$ as a steady-state probability for such a path.

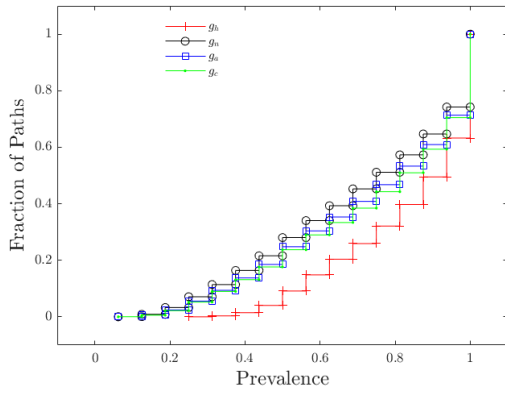
From \mathcal{R}_p , $\mathcal{D}_p(1)$ considers relay paths that show with maximum frequency or $\omega_p(1)$. The size of each dominant set, $\mathcal{D}_p(\mathbf{N})$, represents prevalence redundancy as explained in Section 3.8. Figure 3.4 shows a cumulative distribution of the prevalence of the three dominant sets of all paths in our study at each granularity. For example, at g_n approximately fifty percent (y -axis) was dominated by at least one path with a prevalence of seventy-five percent (x -axis). The result surprisingly is very close to [71], in which forty-nine percent of physical paths have a prevalence of eighty percent. This indicates two important points. (1) These measurements and results are strong enough to be generalized. (2) Having a large-scale measurement allows for capturing a stable view of the relay behavior. This analysis and similar to [49] shows that a thirty-percent of our measurements are stable or long-lived because of exhibiting a prevalence equal to one. For g_c and g_a , we notice a narrow spread in prevalence as occurred in [71] since Planetlab is not diverse enough over such granularities.

In contrast, to [71], Figure 3.4 shows path fluctuations over g_a implies changes over g_c and vice versa for both dominant sets since prevalence at g_c is always strictly below g_a prevalence. More strongly, since g_n curve is strictly above g_a , g_c and g_h with a hundred-percent any change at g_n is reflected at any other granularity. This is because Planetlab's lack of diversity makes it rare to find nodes belonging to distinct ASes and within the same city or vice versa. Having a high prevalence medians of seventy-five percent over g_n and eighty-one percent over g_c and g_a indicate a wide spread in the prevalence distribution. The prevalence over g_h is strictly less than each of the remaining granularities as expected, and so a change in a path over g_n , g_c or g_a will always be noticed as a change in HTL. The HTL for estimating path changes lacks the percentage of error when facing a no-change scenario as discussed in Section 3.9.

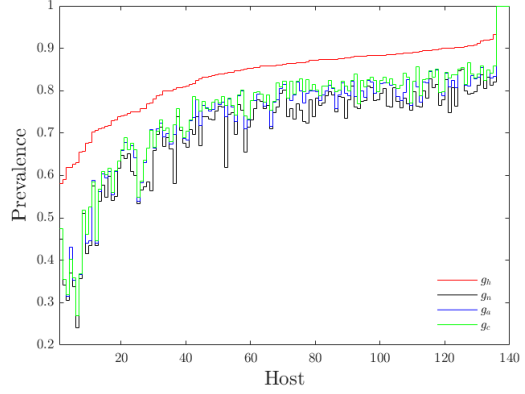
Each node a of a set of one or more hops relay neighbors \mathbf{b} and similarly as in [71] is associated with an overall node prevalence. For each dominant path prevalence $\rho_p(\mathbf{N})$ as mentioned above over a set of minimum delay relay paths from a , a node prevalence is an average of all such paths' prevalence and is calculated as follows.

$$\rho_a(\mathbf{N}) = \frac{1}{b} \sum_{p:a \rightarrow b} \rho_p(\mathbf{N}) \quad \forall b \in \mathbf{b} \quad (3.1)$$

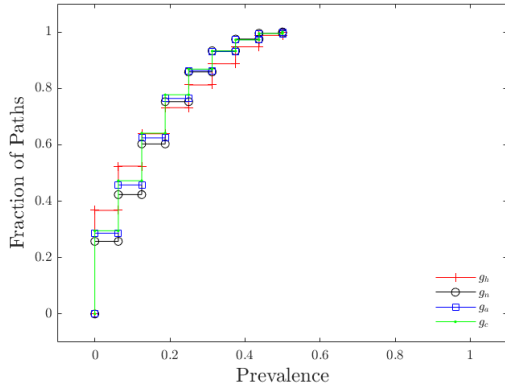
Node prevalence is a measure of relay routing stability for a source node in a set of paths. Higher $\rho_a(\mathbf{N})$ indicates stable dominant relay forwarding via a . Both path and source prevalence



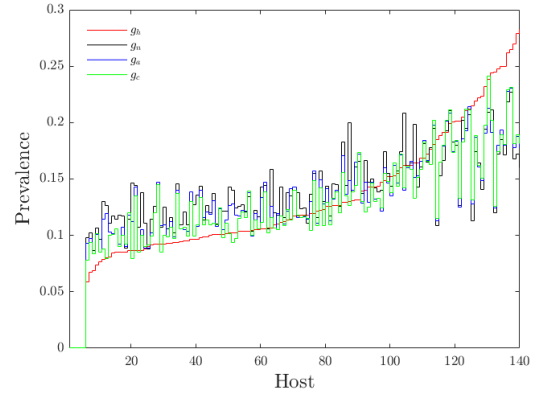
(a) $\omega_p(1)$



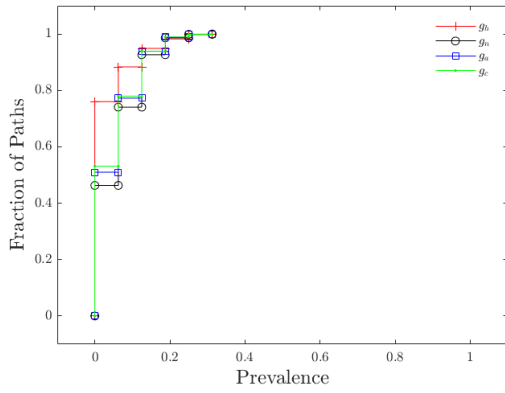
(b) $\rho_u(1)$



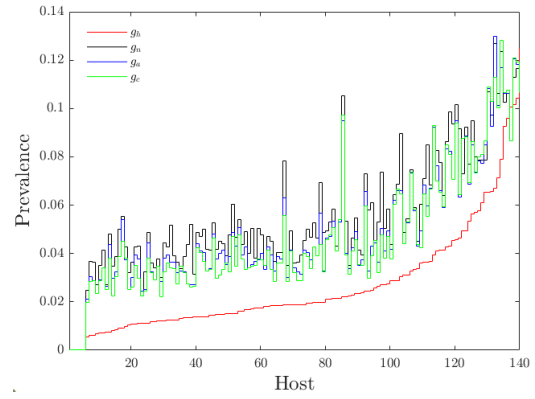
(c) $\omega_p(2)$



(d) $\rho_u(2)$



(e) $\omega_p(3)$



(f) $\rho_u(3)$

Figure 3.4: Path and Host Prevalence

are considered long-term stability characteristics. The source prevalence gives an overall view of a dominant set of source u concerning all remaining nodes. The changes near a particular source will

affect the prevalence of that source alone [71]. The far path fluctuations in a network will affect all sources, not just a particular one [71]. Because physical routing is not optimized in terms of RTT, oftentimes, paths near a source node may follow disjointed links. However, given that our study is an analysis of the minimum relay delay paths, such paths are not expected to be similarly disjointed near their sources, but rather down in the network. Nodes on average have seventy percent of their relay paths stable or long-lived. The prevalence of the second dominant set converges before the first dominant set because a first dominant path shows more than a second dominant one.

Having an underlying path with an extremely small prevalence for example as the minimum value in our study is around six percent indicates a short-lived relay path. No relay path has been shown with a minimum below such percent in our study. Low prevalence generally does not exist in physical routing as [71] confirmed. This is because both intra-domain and inter-domain routing helps to maintain paths with fewer fluctuations. For relay routing, however, the measurement wide view makes such a low prevalence more impossible.

3.6.1 HTL Persistence

For physical routing, persistence is more difficult to analyze than prevalence [71]. This is because a series of measurements for a particular path may not indicate a lack of change [71] over a wide range of time scales from seconds to hours. This section focuses on the relay persistence of the minimum delay paths to understand the implications of the changes that occur within short time scales. For relay paths, we argue that capturing short-lived paths is useless in the benefit to apply relay paths over short scales. Therefore, our analysis only considered changes within a range of ten and twenty minutes. For a physical path p with a set of relay observations \mathcal{R}_p , we refer to such estimated persistence or no-change probability for such path p and a node a and by ν_p and ν_a respectively. Figure 3.5a describes a cumulative distribution of ν_p . There is a similarity between Figure 3.4a and 3.5a. This similarity, small and below ten percent persistence is because of the high prevalence (but not maximum) of the first dominant responsible for all path changes. The maximum persistence for ν_p equals one for relay paths with a single dominant path. Figure 3.5a explains that about fifty percent of relay paths have ν_p below forty percent. Thus, for any relay path p , having such an estimate for ν_p can drive a likelihood model p_X for noticing k no-change transitions within any set \mathcal{R}'_p of M' relay observations. This likelihood is approximated in our work as a Binomial distribution $X \sim \mathbf{bin}(M', \nu_p)$ as in equation 3.2 where M' equals fifteen in our work.

$$p_X(x = k) = \binom{M'}{k} \nu_p^k (1 - \nu_p)^{M' - k} \quad (3.2)$$

The overall network modeled average and maximum view of $p_X(x)$ is in Figure 3.5b. The $\simeq 0.04$ average probability again proves a lack of persistence of relay paths with high dominant prevalence. The $p_X(x)$ is high when $k = 15$ for paths with a single dominant prevalence because

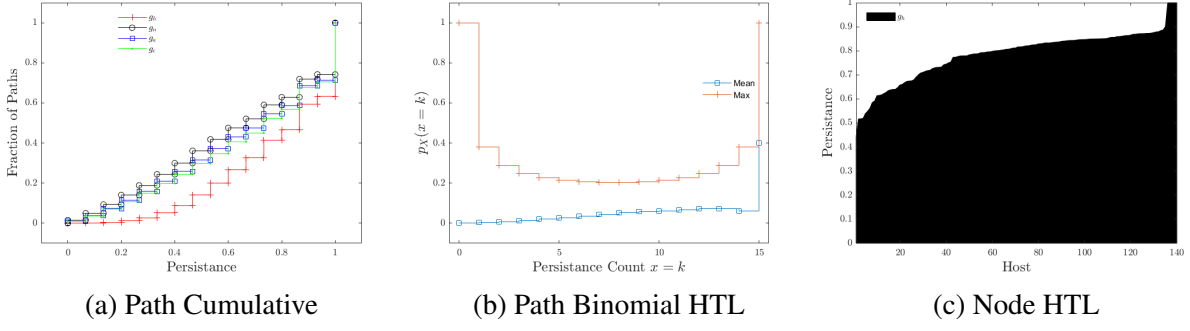


Figure 3.5: Path and Node Persistence

of which path osculations are infrequent. The maximum curve of $p_X(x)$ is concave with only two-hundred and ninety-six paths having zero probability to notice a persistent transition. Five-thousand and thirteen single dominant paths have $p_X(x) = 1$. The average persistence for a node over g_h is seventy-five percent which indicates all hundred-forty nodes use persistent relay paths.

3.7 Node and HTL Symmetry

Routing symmetry is an important characteristic of the Internet. The existence of asymmetric paths highlights the difficulties of providing a consistent topology for the Internet [71]. Path asymmetries increase as we examine fine-grained granularities. For instance, overlay paths are expected to be more asymmetric over g_n than g_c . Here, we define path symmetry κ_p for two relay paths $r_1 := a, \mathbf{a}, b$ and $r_2 := b, \mathbf{b}, a$ by Equation 3.3. Here, \mathbf{a} is a set of intermediate nodes over r_1 and between a and b and similarly for \mathbf{b} .

$$\kappa_p = \mathbf{a} \sqcap \mathbf{b} \times \mathbf{a} \sqcup \mathbf{b}^{-1} \quad (3.3)$$

The symbol \cdot is a set size. Measuring such symmetry requires concurrent measurements in both directions. Instantaneously, using centralized probing control, we synchronized our Pings of four different loads between end nodes. The study in [71] shows forty-nine percent of examined physical paths were asymmetric by at least one different city. The symmetry analysis over g_n shows forty percent of all 9,730 observed pairs of relay paths are completely asymmetric, and only thirteen percent is fully symmetric as Figure 3.6a demonstrates. Despite such high asymmetry of relay paths over g_n , those paths are highly symmetric over g_h as Figure 3.6b presents. For a particular pair of relay paths as defined above, HTL symmetry is specified by an absolute difference in HTL $\mathbf{a} - \mathbf{b}$ between such paths. Figure 3.6b displays eighty-two percent of relay pairs are symmetric over g_h . From analysis, a Laplace distribution with parameters $\mu = 0$ and diversity $\sigma = 0.6$ approximates this symmetry. Thus, $p(\Delta_{\text{HTL}} \mid \mu, \sigma) = \frac{1}{2\sigma} \exp\{-\frac{\Delta_{\text{HTL}} - \mu}{\sigma}\}$ is a probability mass function of Δ_{HTL} .

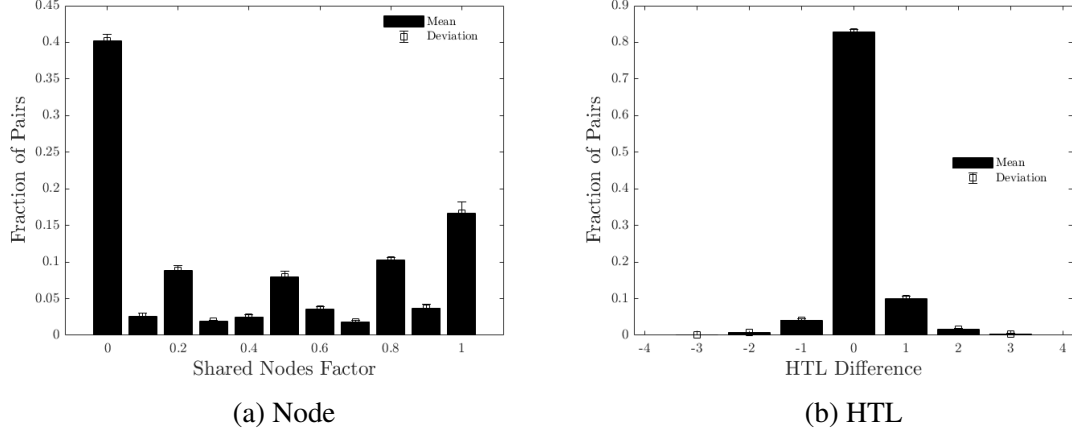


Figure 3.6: Path Node and HTL Symmetry

3.8 Dominant Relay Redundancy and Type

Following Section 3.6, and for reliable relay routing, applications should integrate the multiplicity of dominant paths. This multiplicity represents the size of a dominant set and is denoted as $\phi_p(N) = \mathcal{D}_p(N)$. For example, a physical path p with a set relay observations $\mathcal{R}_p = \{r_1, r_1, r_2, r_2, r_2, r_1, r_3\}$ has two dominant sets with $\phi_p(1) = 2$ and $\phi_p(2) = 1$. Further, of a dominant set $\mathcal{D}_p(N)$, we look for $\pi_p(N)$ or how many paths are relay paths. This metric shows how often relay paths replace p for minimum delay services.

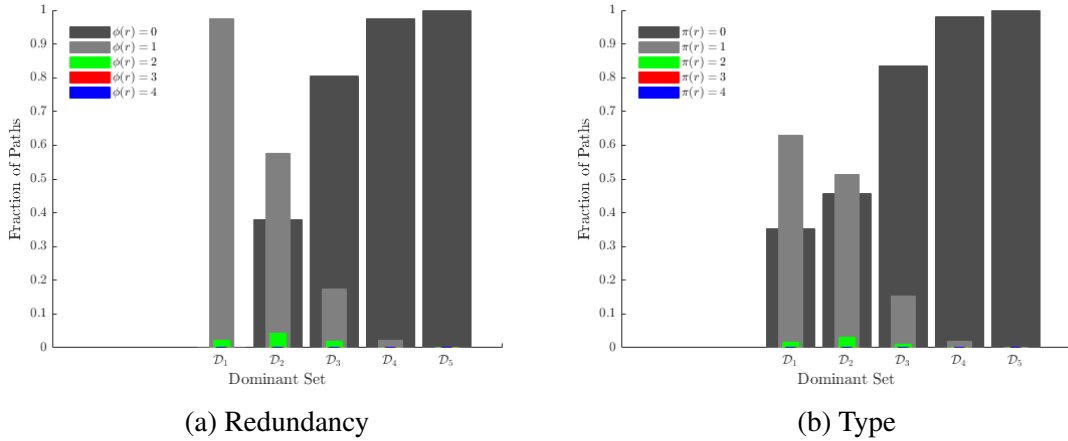


Figure 3.7: Dominant Path Redundancy and Type

For simplicity, we only analyzed dominant redundancy and type at g_h . Figures 3.7 is categorical histograms for $\phi_p(N)$ and $\pi_p(N)$. For $\mathcal{D}_p(1)$ of Figure 3.7a, $\phi_p(1) = 0$ is not zero since any first dominant set is not empty. There is a linear increase in $\phi_p(N) = 0$ as dominant sets tend to be

empty as d increases or prevalence diminishes. The three bars of each d sum up to one, and thus, $\phi_p(N) = 1$ decreases as the path prevalence decreases. Figure 3.5b shows that $\pi_p(N)$ follows a similar pattern to $\phi_p(N)$ but with different rates. This plot shows for $\mathcal{D}_p(1)$ of all examined paths that sixty-one percent of these sets have exactly one relay path while thirty-five percent have no relay path except the physical paths.

3.9 HTL and Change Detection

The discussion in Section 3.6 shows with g_n , nearly fifty percent of 19,460 examined source-destination paths are dominated by a single path with a prevalence equal to seventy-five percent. Hence, given such a high prevalence, our aim is to check whether or not an HTL change is inferred as a relay change with no extra probing overhead. There are many benefits from such a relationship. Detecting relay changes permits nodes to predict performance metrics such as RTT and bandwidth. Therefore, for measurement-based relay routing, recognizing routing changes is important for prospective performance estimations. The current relay headers do not include HTL. This field is as helpful as the TTL for the physical layer. For relay applications, knowing and modifying HTL instead of TTL according to QoS demands is more essential. Having smaller TTL does not ensure a smaller HTL, which means a small processing delay for the relay overhead. Furthermore, for end nodes, a few link changes do not indicate a change in the relay routing as long as such changes still occur between the same relay nodes. Therefore, an additional HTL field should be included in the relay header for end nodes to detect relay changes.

The growth of the Internet might cause routing to exceed the maximum TTL value of two-hundred and fifty-five links. Because of the present existence of relay paths with HTL beyond three hops in our examined network of hundred-forty nodes, we found a widespread consumption distribution of relay links near seventy links. This link usage exceeds the default TTL of sixty-four links [9] and [4]. Hence, relay routing might not improve performance over such a high link consumption. The original IP packet is unmodified except by decrementing TTL while forwarding the packet over a relay path. Therefore, for long relay paths, an initial TTL might become zero before reaching the final destination, and the packet will be dropped. To manage such an issue for the future relay Internet, relay nodes can modify the TTL of the original IP header by allowing such nodes access layer-2 fields. Relay nodes can access the original IP header to increment the TTL count when necessary. However, when a relay node should modify TTL is a challenge. The relay node increments TTL when HTL is not zero yet. Having TTL and HTL makes no need for copying out the original TTL from a selected packet (uniform model) to the outer (relay) IP header before checking and decrementing TTL [98].

The proposed HTL mechanism for detecting relay changes includes HTL in the relay header. Relay nodes can determine as a result whether a change occurs without extra probing. The remainder of this section discusses our evaluation of using HTL to detect relay changes. Table 3.1

summarizes at g_h all associated False Positive F^+ , False Negative F^- , and total Error E when relying on HTL for predicting actual relay changes. The F^- rate can be reported on every scale. For instance, when HTL is two hops for two consecutive measurements and corresponding relay observations $r_1 := a, b, c$ and $r_2 := a, d, c$ although such a change can not be detected via HTL. Thus, HTL does not show an F^+ against g_n since a no HTL change will not be reported as a change. However, at g_a or g_c HTL can report an F^+ when HTL changes but a relay path is still constant at those granularities. The F^+ in Table 3.1 is zero even against g_a and g_c since our Planetlab slice was not dense enough at both granularities.

	F^-	F^+	E
g_n	41	0	15
g_a	35	0	12
g_c	33	0	10

Table 3.1: HTL and Change Detection

3.10 RTT Characteristics

The authors in [61] discussed the use of HTL for detecting path changes. This section, in contrast, discusses how overlay RTT can detect relay changes and new overlay RTT characteristics, such as the relationship between overlay RTT and path symmetry over g_n . The overlay RTT granularity is g_d^* while g_d refers to the underlay RTT. Further, we discuss the behavior of physical RTT prevalence via different RTT statistics.

3.10.1 RTT and Path Change Detection

This subsection shows RTT and changes in relay paths. This analysis shows a symmetrical prevalence since the g_n curve prevalence is in between the g_d^* and g_h curves, as in Figure 3.1. This causes detectable g_n changes by either one. However, using g_d^* introduces a F^+ error compared to HTL. Table 3.2 summarizes the used RTT classification policies that reduce F^- and TE rates by more than half, as in Table 3.3. The \times is when a change occurs at a certain granularity while \times means a no change. The accuracy of such a classifier could improve by imposing extra physical RTT constraints and forming a hybrid classification of HTL and RTT.

	F^-	F^+
g_n		\times
g_d		\times
g_d^*	\times	

Table 3.2: RTT and Change Detection Policy

	F^-	F^+	E
g_n	16	31	25
g_a	13	32	26
g_c	12	33	26

Table 3.3: RTT and Change Detection Error

3.10.2 Physical RTT Prevalence

Many protocols view RTT as an essential performance metric. The TCP for instance uses a minimum RTT estimate to determine the Bandwidth Delay Product (BDP) and set the congestion window size **cwnd**. The maximum RTT, however, is used to decide the re-transmission time for TCP. Moreover, streaming applications rely on average RTT as a performance measure. For relay routing, we need to understand which RTT statistic should be considered for constructing relay paths. The minimum RTT curve has a high prevalence as fifty-six percent of physical paths frequently experience constant minimum RTTs as Figure 3.8a depicts. Hence, a higher node prevalence as in Figure 3.8b occurs as seventy-five percent of the nodes have an average prevalence of eighty percent. This scenario exists when an overall path queue is not busy so that at least one packet of each measurement is served over a minimum RTT. The challenge for physical routing is to maintain constant RTT. The maximum RTT prevalence is smaller because of measurement and cross-traffic and path changes that influence maximum RTTs. Therefore, TCP experiences additional re-transmission times than the **cwnd** when using a maximum-based RTT mechanism for unacknowledged packets. The average RTT prevalence is a close curve to the minimum RTT curve, and accordingly, is more advised than the maximum-based RTT re-transmission. The obvious divergence between such prevalence curves is valuable when using one RTT statistic to assess other statistics.

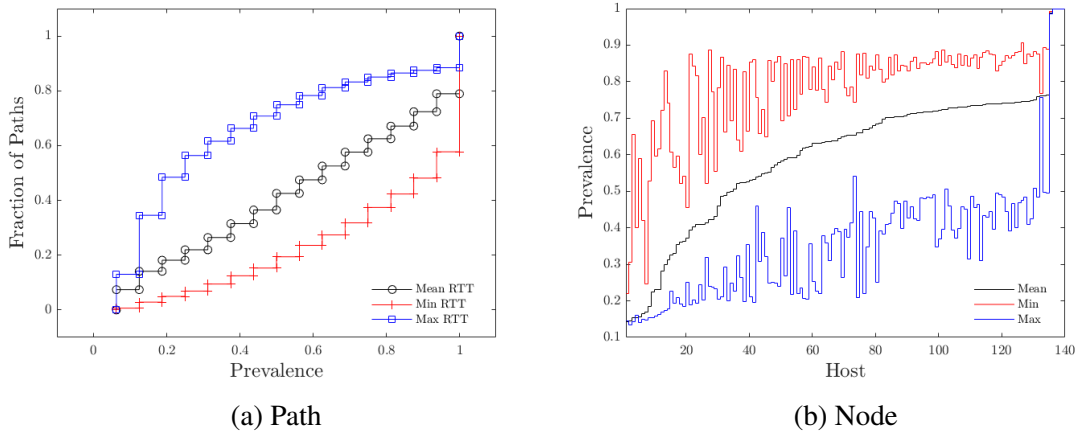


Figure 3.8: Path and Node Statistical First Dominant Prevalence

3.10.3 Path RTT Symmetry

In Section 3.7, we have examined relay path symmetries over g_n and g_h . Here we establish a definition for whether two physical or relay paths are symmetric at g_d or g_d^* respectively. This view is strongly related to our discussion in Section 3.7 as we have concluded relay paths were highly symmetric over g_h but weakly symmetric over g_n . This section examines a delay symmetry

for a node a connected with b neighbors. Node a experiences with each neighbor an absolute difference Δ_{RTT} value between forward and backward directions. The mathematical definition of such a symmetry measure is in Equation 3.4.

$$\kappa_a = \frac{\sum_{\forall b} \mathbb{1} \mid \Delta_{\text{RTT}} \leq \tau}{b} \quad (3.4)$$

The $\mathbb{1}$ is a numeric indicator with one only for a Δ_{RTT} smaller than a delay threshold τ . Larger τ causes a node over g_d to seem more symmetric in delay. Figure 3.9 shows a distribution comparison between the physical and relay RTT symmetries over many τ values. The plot indicates that the overlay is more symmetric as τ peaks. Figure 3.9a shows that nearly fourteen percent of physical and ten percent of relay nodes behave in a complete asymmetric delay manner when τ is zero milliseconds. Further, eighteen percent of the physical and thirty-one percent of relay hosts retain a fifty percent delay symmetry. This measure rises slowly as expected when τ is ten milliseconds. This is because of the high symmetry in relay paths over g_h as discussed in Section 3.7. Therefore, relay paths converge quicker than their counterparts physical paths in RTT symmetry with larger delay symmetry due to a high symmetric HTL behavior.

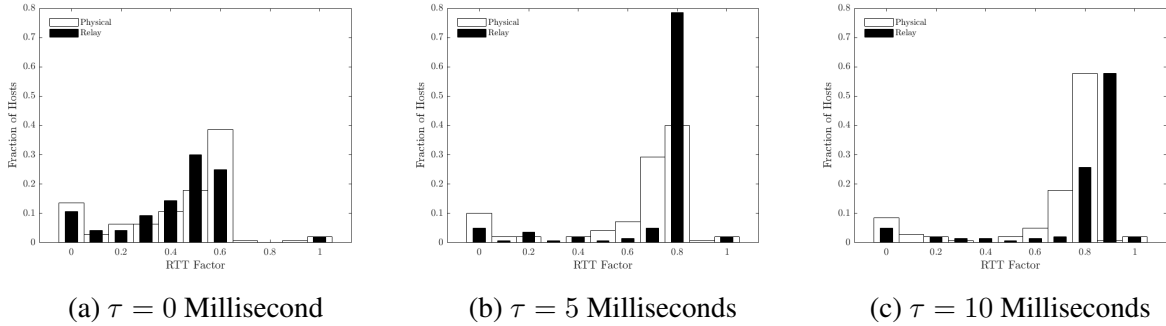


Figure 3.9: Node Physical and Relay RTT Symmetry

3.11 Proposed Relay Performance

Figure 3.10 is an abstract for handling UDP stream requests. Each request consists of addresses of nodes a and b and a bandwidth demand b_p . The controller passes each request to an HTL-based stable path selection process. The measurement database is used by expensive and inexpensive bandwidth schemes. Each scheme returns two sets each of two maximum bandwidth relay paths $p = \langle p_1, p_2 \rangle$ and $p' = \langle p'_1, p'_2 \rangle$ respectively. The choice of a pair $p^* = \langle p_1^*, p_2^* \rangle$ depends on HTL stability and bandwidth. This pair contains a path for a major UDP flow and another one for handling errors and dropped data via TCP. The controller afterward confirms all interfaces for p^* in the network to serve such a request before halting both paths when done.

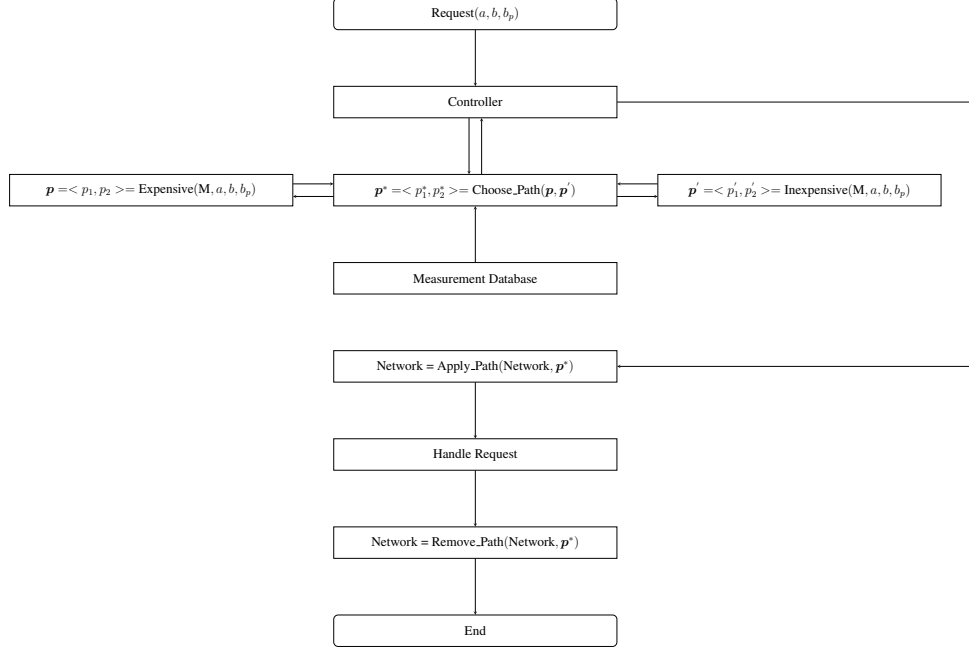


Figure 3.10: Hybrid TCP-UDP Relay Abstract

3.11.1 Bandwidth

Bandwidth is a complex metric to estimate. Many tools such as the listed in [1] are proposed to measure bandwidth. However, none of these tools is robust and scaleable. IPerf is considered a benchmark in many studies. For our work, IPerf assesses all pairs of UDP bandwidth over twelve rates of demand. Here, we assigned a UDP stream for each rate of demand as detailed in Table 3.4. For each desired rate, all IPerf clients render designated traffic for IPerf servers. The higher rates of demand were attempts to reduce the effect of cross-traffic during our measurements by overloading each examined physical path with back-to-back datagrams. Each rate of demand is bounded by a client network hardware. For ensuring consistency in our measurements, IPerf used a default datagram size of a thousand and five-hundred bytes as a Maximum Transmission Unit (MTU) so datagrams are processed as packets. However, few clients with multiple interfaces alter such average datagram size because of hardware constraints.

Load Mbit	4	16	24	40	80	40	40	80	80	80	80	80
Rate Mbps	0.5	4	6	8	10	20	40	80	100	200	400	800

Table 3.4: UDP Traffic and Rate of Demand

For a set C_q^M of M measurements of rates of demand examined over a physical queue of link q (a minimum communication queue between two physical addresses) as in Table 3.4, a maximum transmission rate of an egress interface of q is $c_q^* = \mathbf{arg} \max_m C_q^m \forall m \leq M$. This maximum

rate c_q^* is often called link capacity and an upper bound of q 's available bandwidth $b_q \leq c_q^*$. The bottleneck capacity $c_p^* = \mathbf{arg} \min_{q \in \mathbf{q}} c_q^*$ for a path p of a set \mathbf{q} of links similarly bounds $b_p \leq c_p^*$. Instead of evaluating a bottleneck capacity via an excessive overhead, a few end-to-end bandwidth measurements for a path are sufficient to provide an estimate for b_p . This b_p depends on IPerf accuracy and causes an excessive overhead. IPerf measured UDP bandwidth for all paths while monitoring each resulting packet loss d_p^m .

Many applications accept a limit of one percent packet loss of UDP streams. Here, we estimate our end-to-end bandwidth via a few rates of demand and a packet drop limit τ . For a path p with a set \mathcal{C}_p^M of rates of demand of M samples, we defined $b_p = \mathbf{arg} \max_m \{C_p^m \mid d_p^m \leq \tau \wedge m \leq M\}$. The number of samples for each path in our work is twelve as detailed in Table 3.4. This work introduces b_p as an M -expensive bandwidth estimate because. Figure 3.11 is a cumulative plot of the expensive b_p improvement as τ varies for all 147,62 paths simultaneously. This increase in bandwidth is a difference between the physical and relay b_p values. Each relay value is calculated over conditioned and underlay available bandwidths as mentioned above. The lagging cumulative convergence indicates more paths gain additional bandwidth using such an expensive scheme. The results show few gained relay bandwidths as τ increases. The hundred Mbps bandwidth gain appears as a dominant improvement.

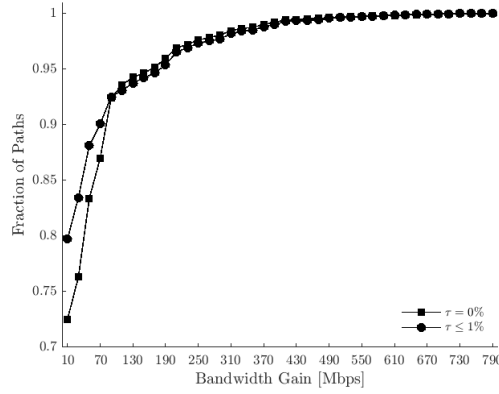


Figure 3.11: Expensive Bandwidth

Finding accurate bandwidth estimates mandates better and more careful probing. However, streaming services like YouTube and VoIP applications such as Skype and Viber might not adopt such schemes. This work compares an inexpensive probing relay scheme based on a single rate of demand to evaluate bandwidth. For a Youtube rate of demand, a centralized controller receives a request from a client with a specific rate of demand and probes the network at such demand for all paths before computing a relay path with a relative available bandwidth. This reduced probing scheme is compared with the results in Figure 3.11. Figure 3.12 shows bandwidth gains when for

a single rate of demand of eight-hundred Mbps.

This inexpensive scheme is a single rate of demand without a need for an exact bandwidth estimate to find better bandwidth relay paths. The expensive scheme, in contrast, uses more rates of demand to obtain more accurate bandwidths for all paths before analyzing for a maximum bandwidth relay path. The inexpensive scheme offers a higher improvement than the expensive one that shows an early cumulative convergence when τ rises. The inexpensive scheme of Figure 3.12a has seventy-two percent of relay paths with bandwidth improvement of eighty Mbps or below. Meanwhile, ninety percent of the expensive relay paths show a similar bandwidth gain. This improvement when τ is equal to or below a two-packet drop percent implies more paths attain more inexpensive bandwidth.

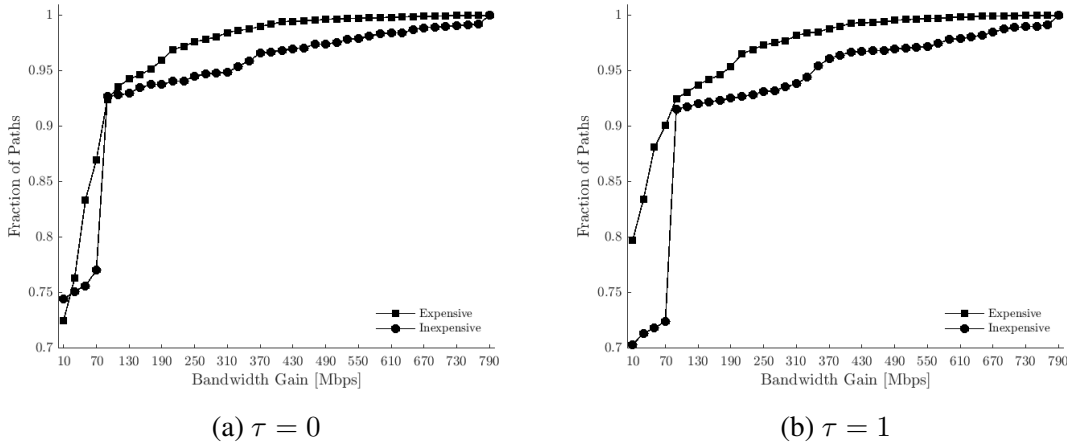


Figure 3.12: Expensive v.s. Inexpensive Bandwidth

3.11.2 Relay Node Usage

The UDP relay bandwidth analysis in our work is based on a single maximum bandwidth path for end clients. Thus, particular nodes would experience a high relay importance when having an increased volume of streaming requests. Here, we analyze the network over a non-overwhelming volume of requests, and the next best relay paths can be considered when such a volume increases. Figure 3.13 is a normalized relay node usage by the total number of relay paths 14,762. This usage refers to the number of relay paths a node is involved in and is also averaged over a set of zero, one, and two percent of packet drop limits. Each bar means a proportional relay node usage and such utilization is not uniform because of focusing on a single maximum bandwidth relay path. Enforcing semi-uniform node usage requires applying the next best bandwidth paths for clients.

3.11.3 Handling Packet Drop in UDP

There is a tradeoff between a sending rate and datagram loss for a UDP stream. End clients discard any duplicate UDP datagrams. Many studies analyzed a Multi-Path (MP) [35, 98, 21]

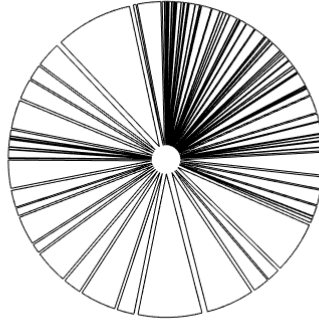


Figure 3.13: Node Usage

and Mutli-Session (MS) TCP performance [11, 67]. The MP-TCP in [35] balances and forwards each traffic over many paths. The Datagram Congestion Control Protocol (DCCP) in RFC-4340 serves as a congestion control mechanism for UDP applications to avoid congested links. The UDP as explained in RFC-4340 uses DCCP as an IP layer protocol. Nonetheless, many solutions for unreliable UDP connections are feasible. First, a hybrid TCP-UDP connection can solve such an issue when by applying TCP as a back-end session for dropped datagrams. The remainder of a stream is served via an independent UDP relay session. The second solution is via a similar approach as with MP-TCP. This Multi-Path UDP (MP-UDP) allows each sender to copy a stream over multiple paths. The receiver as a result has a better chance to recover dropped datagrams. This protocol for example is applicable to copy frames for online games.

For a compromised client-server connection, TCP-UDP searches for a pair of relay paths within an established drop rate given a particular rate of demand. Here, we briefly examine the fraction of relay TCP-UDP paths over which clients correctly receive their corresponding UDP streams within an upper loss bound τ . This means such paths do not invoke TCP sessions for drop and error recovery. Figure 3.14 displays such a fraction vs. the rate of demand. For a client-server relay path, such a path is counted when minimizing the corresponding underlay drop rate below a particular limit τ .

There is an increase that occurs as we raise the rate of demand since the underlay paths forward streams at low sending and high drop rates. Thus, as we increase the underlay rate of demand, packets will experience more loss. The similarity among all drop rates of one percent and above indicates underlay paths experience a **dominant** data drop rate of one percent. This explains the vertical shift of the zero τ curve above all remaining drop curves.

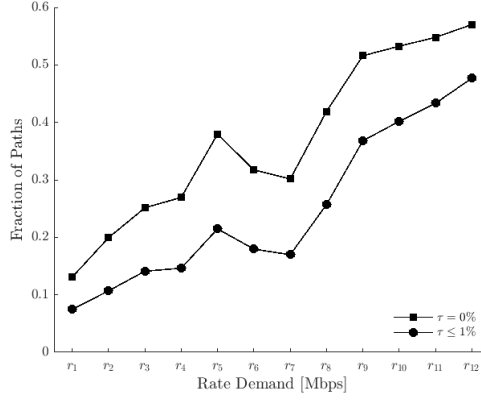


Figure 3.14: Hybrid TCP-UDP Success Probability

3.12 Conclusion and Future Work

This study shows that relay paths are more stable in terms of HTL. The results demonstrate that an estimation-based relay selection can help reduce the search space of better paths and speed up locating better relay paths for streaming services. This work recommends that an estimation-based relay can enhance performance while reducing probing overhead for minimum delay routing. The HTL as a more stable metric can be used to infer relay changes with an error of fifteen percent. For large scale-networks, we find that instead of focusing on exact bandwidth estimates, our inexpensive bandwidth scheme shows that it is highly possible to find other relay paths that can serve a demand rate quickly. Briefly, we introduce an analysis of a hybrid TCP-UDP paradigm to improve YouTube services and VoIP applications.

4 Reinforcement Learning Based TDMA Protocol for Ad Hoc Networks

This chapter presents a Q-Learning-based TDMA slot allocation framework called QL-TDMA for distributed wireless sensor or ad hoc networks. This novel QL-TDMA protocol has the following attributes. (1) The model does not require prior topological information as nodes explore each other while learning. (2) The proposed TDMA scheduling converges over a minimum required number of slots or frame length. (3) The protocol adjusts conventional wireless handshaking communication protocols by a reduced scheduling overhead between nodes until convergence. This protocol is novel in literature by avoiding ACKs between nodes while still inferring collisions. Many degrees, sizes, and topologies, such as linear (three slots), cycle (three and four slots), star, or random, challenges the performance of QL-TDMA. The primary node biases in our design are (1) Each node can listen to other slots while sending only over one dedicated slot. During a sending slot, such a node is deaf, so no message can be received. (2) Each node listens elsewhere for any correct or collided message. (3) NACK is only used to notify about hidden collisions. (4) For one-hop collisions, a node sends with a random offset of its slot and scans the frame for any missing one-hop node in case of a random offset match among one-hop nodes. (5) The two-hops collision NACKs can help to infer a one-hop collision. (6) For a node on slot τ_i with a two-hops collision, any one-hop node on a different slot τ_j will send a NACK to such a node during a specific portion called NACK period. The QL is implemented in a distributed mode as each node works for a convergence policy towards a collision-free frame. Each node has an independent reward and collision punishment mechanism according to node degree and behavior during each learning episode. The learning rate and learning discount factor are adaptive. This work shows that such collision knowledge is precious for any QL-TDMA to converge within a suitable time for a minimum frame length.

4.1 Background

The wireless sensor networks become enablers for many paradigms, including environmental, health monitoring, intrusion detection, inventory tracking, intelligent space management, and collaborative disaster management [80]. There are many protocols to handle collisions in WSNs. The contention-based NACK schemes such as in [73, 42, 14, 90, 94, 109] introduce no-startup overhead in communication, and that allows nodes to immediately communicate with neighbors while doing their best to eliminate collision. In general, any contention-based scheme is a Carrier Sense Multiple Access (CSMA) such as BMAC [73] and SMAC [109] as the most common ones due to simplicity, flexibility, and robustness [81]. BMAC and SMAC require no synchronization or topology information [81]. Furthermore, node mobility is manageable using [73] and [109]. However, BMAC and SMAC are trial-and-error MAC protocols with which collision always exists. For energy-constrained sensor networks, collision is particularly costly when nodes engage in idle listening for an unsuccessful transmission [81]. The RTS (Request To Send) and CTS

(Clear To Send) are commonly used signals among nodes during network operations to avoid collisions. These contention-based MAC schemes handle active networks due to node failures or mobility [80]. However, such schemes are more susceptible to packet loss when performing a packet broadcast [80]. Furthermore, contention-based MAC protocols may suffer performance degradation and high RTS/CTS flows in dense and overloaded networks, which can cause intense power reduction. Further, RTS/CTS signals can degrade BMAC throughput by more than seventy-five percent [81].

The second protocol to address collision in sensor networks is schedule-based, like TDMA. The TDMA scheme is less scalable but efficient (low delay) [80]. This scalability issue exists because of inefficient TDMA designs deviating from the minimum required scheduling signals and frame length. More TDMA drawbacks, as mentioned in [78], are (1) The NP-hard [115] makes computing an efficient no collision TDMA frame requires a centralized scheme, and so is neither scalable nor trivial. (2) TDMA allocation needs accurate synchronization, which causes an extra scheduling overhead. (3) Energy shortage and node failure or mobility are expensive to manage [81, 110]. (4) Interference irregularity in [82] adds a complex analysis when communication and interference ranges differ. Thus, a TDMA schedule based on equal ranges does not necessarily yield a no collision frame [110]. (5) Medium and interference are time-variant characteristics of sensor networks, and no unique TDMA frame serves for an extended period. (6) In low contention scenarios, inefficient TDMA frame underutilizes medium and increases latency compared to CSMA [110]. These downsides of TDMA suggest that a stand-alone TDMA scheme is not practical [110]. From our analysis and similar to [110], we believe any collision frame carries a piece of valuable information to curtail collision in high contention networks. However, since TDMA is for overloaded sensor networks, our QL-TDMA convergence within a minimum frame length and learning overhead is a unique approach to apply. Here, we deviate from [81] and generally all conventional MAC schemes by obtaining minimum length frames within short convergence times and using a minimum scheduling overhead. Throughout our research, such a convergence period refers to the number of QL-TDMA learning episodes to produce a minimum TDMA frame.

However, studies such as [80] and [110] mentioned TDMA benefits. The obtained TDMA collision-free frames arrange transmissions so all nodes can send during private slots without collision. In contrast to contention-based schemes, TDMA schemes can accommodate dense networks of high loads. This work is for such TDMA networks with a minimum number of slots. TDMA-based schemes do not permit instant communications after deployment until no collision exists, in contrast with contention-based schemes. However, such an overhead does not persist for long (only between startup and convergence of scheduling) when a network is in operation, like RTS/CTS control signals in contention-based schemes. Future TDMA schemes could minimize convergence times to overcome contention-based MAC protocols efficiently.

Following a TDMA convergence, nodes can send packets over private slots with no idle listening or overhearing as with contention-based schemes since such modes consume as much energy as a receiving mode [80]. The remaining drawbacks of current TDMA schemes are (1) Non-minimized frame lengths cause higher delays [109] as a noticeable artifact of using more extended frames. (2) Lack of performance when a TDMA scheme operates for active environments [80]. (3) Large-scale sensor networks could consist of millions of nodes [80], and since obtaining a collision-free TDMA schedule is NP-hard [115], such TDMA-based MAC protocols might not scale when missing robust converging schemes. These drawbacks have forced studies to propose centralized topology-dependent TDMA schemes [116] to increase throughput. This increase in throughput comes at the cost of higher delay in under-loaded networks [39]. In contrast, the work in [96] is distributed and offers a TDMA paradigm whose performance depends on network size. The study in [80] proposes a distributed TDMA protocol with a similar abstract as in [116]. The main distributed TDMA is synchronous [80], but with a lower performance when compared to [116]. Topology-independent scheduling, however, is suitable for active topology changes [24]. These algorithms have long convergence cycles while the network size controls the frame length [24].

4.2 Radio Networks QL-TDMA

4.2.1 Radio Packet Networks

Radio packet networks are an ideal multiaccess and slot model example and partition nodes into subsets in which each node v is connected with another subset of nodes [13]. For such a node subset v of size \bar{v} , an ideal partitioned multiaccess model deals with the contention that occurs within v 's contention subset $v = \{v' \mid \bar{p}_{v,v'} \leq 2\}$ of two-hops neighbors that attempt to access the same channel simultaneously. Here, $\bar{p}_{v,v'}$ is a minimum length of a path $v \rightsquigarrow v'$. This model assumes all sent packets have a unique size, and each one requires a single transmission unit of time called a slot. Further, all nodes are synchronized so that each packet starts at an integer time unit and ends before the next one [40]. Packets arrive for transmission at each node of a contention subset according to an independent Poisson process [13]. Let λ be an overall arrival rate of packets for a contention set of nodes, and consequently, $\lambda \times v^{-1}$ is each node arrival rate. Each node is assumed to be only in either a sending or listening mode. Failed nodes are excluded. The size of any contention subset can be infinite $v \rightarrow \infty$, and so each newly arrived packet is for a new node of v [13]. This model applies no buffering and discards any new packet that arrives while another one is in transmission or collides with another packet when received.

4.2.2 QL-TDMA

This work is a unique QL-TDMA over a minimum required frame length and communication overhead. The conventional MAC protocols cannot address all mentioned TDMA drawbacks. This problem is one reason behind introducing our QL-TDMA. Moreover, traditional MAC schemes use

current collision management regulations, and no invested effort exists to modify such rules for enhanced performance. The advances in processor and low-power wireless communications allow deploying of intelligent computation and large-scale sensor networks. The second reason behind introducing QL-TDMA is to examine changes in current collision management rules as discussed in Sections 4.6.3 and 4.6.5. Reinforcement Learning (RL) is a mechanism in which agents or nodes learn from mistakes in a trial-and-error manner towards achieving an aim such as a collision-free schedule. There are many classes of protocols an agent can use. The value-based, policy iteration and off-policy are examples. This research uses a well-known off-policy approach named Q-Learning (QL), for which agents follow no prior policy or model (transitional probability) during each episode. The argument is that TDMA scheduling is much simpler when giving prior policy to improve over episodes. Here, supervised learning is not suitable since knowing a minimum frame allocation in advance is not a reasonable assumption.

4.3 Motivation and Contributions

The latter two reasons guided us to adopt QL as an RL approach to examine whether or not wireless nodes can learn and follow better learning tracks. These tracks are over a minimum state space of size $\tau \times \tau$ where τ is a set of a minimum number of slots. The QL, as an off-policy technique, is commonly used for non-deterministic environments or model-free spaces. In contrast to [13] and [14], agents perform actions based on evolving Q-matrix. For multi-agent settings, RL has been investigated, such as in [14], [15], and [16]. However, applying RL for large scales of distributed systems is a critical challenge when reducing inter-control messages between agents is a concern. There is a shortage of work that opens windows for intelligent sensor networks and paves a road towards making nodes learn from their current and previous mistakes and neighbors' experience (paths for optimal TDMA frame). This work provides a baseline contribution for self-controlled sensor networks. Moreover, our study is motivated by multi-agent decision-making scenarios for distributed, dynamic, and non-dynamic networks. The basic setup of our problem consists of a network of wireless nodes that follow a finite state Markov Decision Process (MDP) in distributed mode. The outcome of our study is to achieve an efficient TDMA protocol with a minimum exchange of collision information to reduce communication overheads in contrast to contention-based signals such as RTS/CTS. The remaining contributions of our work are (1) Experimentally, we have shown that such a QL-TDMA can schedule complex typologies over a minimum frame length. (2) By proposing new collision detection rules, this study has shown nodes' ability to learn minimum length and no collision TDMA frames via a reduced communication overhead. (3) This approach of a learning-based TDMA scheme needs more attention to enhance convergence speed.

The remaining discussion is categorized as follows. The summary of related work on conventional TDMA protocols is in Section 4.4. The discussion in Section 4.6 illustrates our QL-TDMA

protocol. The proposed collision management is explained in Sections 4.6.3, 4.6.5, and 4.6.6. The evaluation of our QL-TDMA protocol is in Section 4.7

4.4 Literature Review

There is notable work on conventional TDMA-based MAC scheduling for conflict-free wireless sensor networks. These protocols vary in their access mechanisms to the band. Random access methods usually suffer from high collision rates [16] and are used when nodes are under-loaded. Random access MAC systems do not require topology or clock information when solving contention among nodes [16]. Therefore, such systems are robust for dynamic sensor networks but impose high communication overhead. The work in [108] mentioned that 802.15.4 low-rate personal wireless area networks could suffer from a packet delivery ratio drop from ninety-five to fifty-five percent when traffic increases from one to ten packets per second. There are two distinct approaches for TDMA scheduling, distributed and centralized. Previous research such as [79] is centralized but needs to be more scalable because of requiring complete topology information. The MAC schemes proposed in [65] and [20] are also centralized and focus on decreasing TDMA frame length.

The cluster-based TDMA schemes such as [101] and [112] are more scalable but suffer from inter-cluster collisions since cluster heads produced by distributed clustering share common nodes. The study in [63] proposed a distributed coloring algorithm within $O(\Delta \log N)$ where Δ is the maximum degree of a network of N nodes. This coloring scheme, however, does not consider two-hops collisions. This NAMA in [85], along with randomized medium access, is a hash-based priority scheduling. The complexity of NAMA is $O(N^2)$ to obtain two-hops neighbors' priority. This scheme, according to [16], is not scalable. The node with NAMA determines a chance to transmit within two-hops neighborhoods. The ZMAC in [81] and [110] behave like CSMA under low contention and TDMA under high contention scenarios. The ZMAC, unlike pure TDMA, is robust to dynamic topology changes and synchronization failures in sensor networks. The ZMAC is similar in objective to QL-TDMA when handling hidden collisions with a small overhead in contrast with CSMA. The channel noise level with one approach of ZMAC is passively estimated by measuring the average number of noise backs-off a sender takes before transmitting a packet without performing active channel sampling. The proposed QL-TDMA uses a minimum NACK overhead for hidden collisions and discovers neighbors while learning similar to many conventional MAC schemes.

4.5 QL Fundamentals

This work sees a sensor network as a multi-anchor or agent environment where nodes interact over a discrete state and action spaces. Figure 4.1 shows how two anchors interact over a state space of size $\tau \times \tau$ slots. Each anchor chooses an action to send over a new slot τ_j while currently sending over slot τ_i . The anchor in blue for example performs action $\tau_2 \rightarrow \tau_1$ marked in blue from

an action matrix $\tilde{\mathbf{a}}$ of size $\tau \times \tau$ as well. The row index for $\tilde{\mathbf{a}}$ is the current slot and the column one is the next slot. Each anchor obtains a positive or negative immediate reward value for each observed new slot.

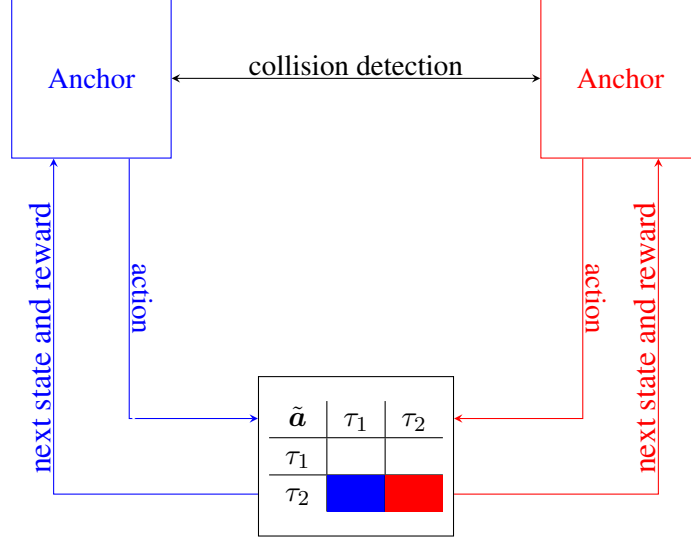


Figure 4.1: QL Paradigm of Two Anchors

The Q-learning is a model-free reinforcement learning algorithm that does not need a model of the learning space. For a Finite Markov Decision Process (FMDP), Q-learning searches for a policy to maximize the expected value of the total reward over all episodes. Each agent or node accesses a set of states or slots $\tau = \{\tau_i \mid i \in \mathbb{R}^+\}$ and a set $\tilde{\mathbf{a}}$ of actions to send over τ_j immediately after τ_i over a next episode for example. The selection of such an action is controlled by equation 4.1.

$$\tilde{\mathbf{a}} \equiv \tau_i \rightsquigarrow \tau_j \mid j = \mathbf{arg} \max_j \left[Q_{\tilde{\mathbf{a}}}(\tau_i) + \sqrt{\frac{2 \times e}{c_{\tilde{\mathbf{a}}_{w-}}} \right] \forall \tau_j \in \tau \quad (4.1)$$

Here, $c_{\tilde{\mathbf{a}}}^{w-}$ is the number of times an action $\tilde{\mathbf{a}} \equiv \tau_i \rightsquigarrow \tau_j$ has caused collision when performed over a window of size w episodes. Equation 4.1 is a composition of Q-value and a regret quantity for each node about an action. For instance, a node would have a small regret for an action that causes many collisions over a learning window. This action reflects a reward or score aiming to maximize an overall expected reward when converged. This maximization for an action $\tilde{\mathbf{a}}$ is achievable by adding a maximum next (a long-term) reward from state τ_j to such an immediate reward $R_{\tilde{\mathbf{a}}}(\tau_i)$. Thus, choosing an action is controlled by the Q-function and a regret value. This Q-function is updated as follows:

$$Q_{\tilde{\mathbf{a}}}^+(\tau_i) = [1 - \alpha_e] \times Q_{\tilde{\mathbf{a}}}^-(\tau_i) + \alpha_e \times [R_{\tilde{\mathbf{a}}}(\tau_i) + \gamma_e \max_{\tilde{\mathbf{a}}} Q_{\tilde{\mathbf{a}}}(\tau_j)] \quad (4.2)$$

The $R_{\tilde{a}}$ is an **immediate** reward gained when a change because of an action \tilde{a} to move from being in state τ_i to τ_j . The learning rate $\alpha_e = \sqrt{\alpha_0 \times [\alpha_0 + c_{\tilde{a}_{w+}}]^{-1}}$ is within $0 \leq \alpha_e \leq 1$. The α_0 is an initial value for each node equals one and is the number of times an action \tilde{a} has caused no collision over a cumulative window w of episodes. This window w in **bold** in our case is not a sliding one as in equation 4.1 and considers each previous episode from the start. The learning episode e is between choosing action \tilde{a} and applying such action while being in state τ_i . This α_e determines an extent by which newly acquired Q-values override previous values. Equation 4.1 shows an $\alpha_e = 0$ causes a node only to exploit prior Q-values while $\alpha_e = 1$ weights more recent updates instead. Fully deterministic spaces converge over constant and high learning rates. This work as a stochastic environment requires a decay in α_e over episodes. The discount factor $\gamma_e = [\gamma_0 \times [\gamma_0 + c_{\tilde{a}_{w+}}]^{-1}]^{\ln \sqrt{e}}$. Here, \tilde{a}_{w+} is a count for no collisions over a window of size w . Each node's initial γ_0 equals 0.7. This γ_e determines the importance of future or next rewards. For $\gamma_e = 1$, nodes give more weight to future rewards and strive for the next and high rewards. The absence of an end state or a no collision frame causes environment statistics such as Q-value and reward to become infinite. Hence, a work in [36] recommends a start by a small discount factor and increases over episodes to achieve a quick convergence. Having $\gamma_e = 0$ causes a node to be short-sighted when only considering current rewards. High or optimistic initial Q-values encourage exploration since equation 4.2 with small α_e would return smaller update values for any chosen action. These optimistic values increase other actions' probability. The work in [89] resets initial Q-values by early reward values and allows immediate learning in case of constant and deterministic rewards. This Reset of Initial Conditions (RIC) model predicts human behavior better than a model with an Arbitrary Initial Condition (AIC) as [89] has indicated.

4.6 QL-TDMA Protocol

This section discusses our protocol details and distinctions. The QL-TDMA proposes a new collision management and frame paradigm for distributed wireless sensor networks. Let us introduce a set of general terminologies defined for sensor networks. **Node identification:** This protocol is scalable enough since nodes are not required to maintain a list of neighbors' IDs. Each node instead stores a count of all discovered neighbors. The RL allows nodes to act while considering neighbors' slots as obstacles to be identified and avoided during learning. The neighboring nodes send NACKs when hearing a one-hop or two-hop collision. Nodes also update $d(a)$ and vacant slots from a passive hearing. **Link asymmetry:** Nodes in sensor networks can act in strange ways. For example, two neighbors during a modified frame of length F^+ of an episode e may not be neighbors in another because of a node or link failure. The work in [66] assumes every node maintains a list of neighborhood IDs and performs several re-transmissions to a neighbor before giving up and removing such a neighbor from a local neighborhood table. This process can be extremely tedious when networks scale up. The second scenario in which nodes can loop in

re-transmission is when waiting for feedback on sent state signals such as **request** and **grant** as mentioned in [82]. The study in [82] is a clear candidate for a deadlock case since neighborhood listing is involved. Link asymmetry occurs when a node sees another one as a neighbor while a reverse connection does not hold. This condition arises when links become unavailable because of changes in channel conditions [82] or a node failure. **Mobility:** In sensor networks, nodes can fail, exit, or join at any given time. The article in [82] and [81] do not analyze mobility and performance. QL-TDMA considers such characteristics as a topology change. The QL-TDMA simulates many network topologies. For example, QL-TDMA would view a linear topology with an average degree $\bar{d}_e(a)$ during episode e with a new node added as mesh or random one during $e + 1$ of a maximum possible average degree of $\bar{d}_e(a) + 1$. The QL-TDMA maintains the same frame length for a massive linear topology when a node is down, as each resulted in mini-topology would also be linear. The QL-TDMA similarly views a ring architecture with a failed node as a linear one.

For a communication link from node a over τ_i , QL-TDMA sees only two scenarios for successful transmission over such a link. (1) Each neighbor for a is assumed to receive a correct message from a during τ_i without sending back an ACK. However, such a scenario is invalid when a neighbor fails to receive such a message. This unreachable node is now a non-neighbor during τ_i because of a node or link failure, and a performs no re-transmission before the next frame. Further, when node a hears a collision, whether a neighbor's message is not received or involved in such a collision is indistinguishable. Hence, node a updates the Q-table until there is no collision and maintains or drops such a neighbor when a learning window w ends. This confusion is manageable for nodes on symmetric links with a via NACKs in Section 4.6.3 and random offset as discussed in Section 4.6.5. For asymmetric links, $d(a)$ and vacant slots, as explained in Section 4.6.5, guide the convergence. For a minimum frame length, having all collisions managed, any vacant slot confirms an asymmetric link. However, when node a hears a non-persisted one-hop collision, a must observe a vacant τ_i or more, which is assigned for a down or collided neighbor since QL-TDMA converge for a minimum frame length. Each persistent one-hop collision observed by a indicates asymmetric communication with a node. Then, a no vacant slot and collision means a has a converged neighborhood.

Later when such a down node rejoins a 's neighborhood during learning, converged neighbors may encounter a collision with such returned node and enter a re-learning phase. This neighbor is passive when receiving a correct signal on a chosen slot and sends a NACK for hidden collision detected. The QL-TDMA does not allow a node a to send a NACK for a vacant slot heard. This NACK is unnecessary overhead since such slots might be viewed as valid by a neighbor of a , and such neighbor would discard a 's NACK. This protocol, in general, is based on the following. (1) Node a always assumes all symmetric and active neighbors receive a 's message correctly. (2) QL-TDMA depends on neighbors' NACKs for hidden collision and passive listening for one-hop

collisions. The discussion in Sections 4.6.3 and 4.6.5 explains both collision classes. The QL-TDMA mimics the partial behavior of a known transport protocol in packet-switching networks called User Datagram Protocol (UDP), over which a correct, corrupted, or dropped packet is unacknowledged. Figure 4.2 shows link asymmetry and mobility scenarios. Figure 4.2a is a link symmetry deadlock scenario. Node b can avoid collision by hearing a and c slots, and both nodes can neither listen to b 's slot nor NACK. Node b considers a and c as neighbors, while both nodes do not consider b as such since no collision or NACK is heard and see b 's slot as vacant. Figure 4.2b is a link asymmetry, not a deadlock, since b hears c and informs a when colliding with c . Node b is not required to resend to c because a link failure exists. In contrast, when c receives incorrect a 's signal, a NACK will notify a . Node c views b as failed or moved node. The QL-TDMA considers node mobility as in Figure 4.2c in which node a can not hear b and vice versa. The proposed QL-TDMA allows no NACK for empty slots as in Figure 4.2d. Here, node d sends no NACK for heard vacant slots 2 and 4 since node b directly hears correct signals on both slots and would neglect any NACK for such slots.

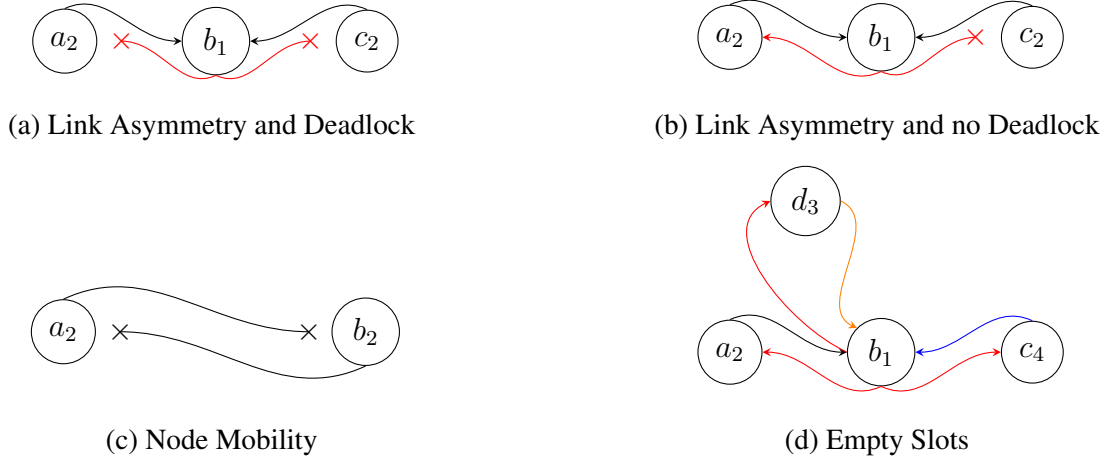


Figure 4.2: Link Asymmetry and Node Mobility without NACK Scenarios

The collision concern in wireless sensor networks is complex and expensive. This complexity is because of range irregularity in [82] and detection. The available work handles collision detection via various schemes, but random schemes dominate. The complexity in detection is because collision occurs even during ACK and NACK handshaking. This work, and as with UDP, avoids ACKs and continuous handshaking signals such as **request** and **grant** in [82]. The proposed QL-TDMA is scalable and suitable for dynamic networks. However, a neighborhood discovery is essential for QL-TDMA for nodes to determine degrees while **exploring** state space. The QL-TDMA uses node degree to detect one-hop collisions, as discussed in Section 4.6.5. For a two-hop collision, involved nodes need no adjacency information but NACKs to be informed. This scheme

is adopted as we can argue that any full-duplex handshaking, such as the Transmission Control Protocol (TCP), is less scalable when compared with UDP and suffers from long delays. Further, as the average network degree rises in dense networks, protecting signals such as **request** and **grant** from collisions is challenging. The QL-TDMA has no such scenarios, and a node can either locally detect a collision or be informed about a two-hop one.

The QL-TDMA is also abstracted to be more deterministic. The only case in which QL-TDMA and TCP are alike is when hidden collisions exist since nodes must be informed to avoid deadlocks. Thus, QL-TDMA is a hybrid collision detection scheme with no handshakes when observing no collision and NACKs when collision exists. Figure 4.3 mimics when QL-TDMA is in UDP or TCP mode. Node d sends no NACK for heard vacant slots 1, 2, and 4. Here, node b listens to a collision in blue during slot τ_1 sending period. Each arc color indicates a particular sending time. Node b sends a notify NACK in red to all neighbors a , c , d , and e . This NACK is similar to the TCP error correction mechanism. Node b also receives during slot τ_3 a correct signal from node d , for which no ACK is sent back like UDP. Further, node b does not notify node e for a vacant slot τ_4 because of a dropped signal. Hence, during such a frame, b views confusingly e as either a non-neighbor or collided over slot τ_1 . However, when b hears no collision, e would be viewed as a non-neighbor with a down-link. Therefore, node b sends no ACK to node d for a correct message on slot τ_3 and mimics UDP in such a case. This view is because QL-TDMA does not maintain neighborhood IDs to specify a particular down node while scheduling. The active degree $\delta_e(b)$ is adequate to handle such confusion. This link drop is also informable via a separate NACK from b during the slot τ_4 NACK period, but for simplicity, our work avoids such an extra overhead of NACKs.

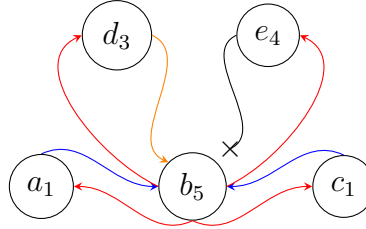


Figure 4.3: QL-TDMA Mimics TCP-UDP

4.6.1 Frame Design

Each frame has a set of slots τ as indicated in Section 4.5. The minimum frame length required for a particular TDMA allocation is $F^* = \tau^*$ which represents a length of a set of minimum length among all available τ sets. The study in [16] indicated that a spatially correlated contention persists for short periods, and obtaining an optimal or suboptimal schedule is not always required. This assumption can be valid for less loaded networks. However, as nodes continuously send, contentions would exist for a long time. The proposed QL-TDMA is for such overloaded networks.

To increase the allocation speed, we defined a redundancy factor ω for an increased frame length $F^+ = \lceil \omega F^* \rceil$. Having $\omega = 1.3$ for a linear network of $F^* = 3$ modifies the frame length to $F^+ = 4$ slots. The proposed QL-TDMA uses a slot of length 2.5 milliseconds. To be practically applied, many possible architectures for a frame of F^+ slots could be used. The reason behind such a modification is to operate closely from a minimum required frame length and, consequently, a smaller delay. Then such a minimum frame reduces the handshake signals or acknowledgments and so provides a minimum flow burden using such a frame. The literature is rich in conventional TDMA protocols that operate above minimum frame lengths and like ACK/NACK-based scheduling schemes. QL-TDMA allows wireless ad hoc network nodes to learn a successful policy that leads to a collision frame allocation using minimum NACK overhead. Each node collects collision statistics from previous frames. This information guides a node toward such a collision-free frame. Each newly chosen slot for the next learning frame in our work context can be examined only over the next frame and not a currently examined one. This enforced order may raise our QL convergence time. For a certain node with an assigned current slot τ_i whose synchronized time has elapsed of a frame of a current episode e , such a node is allowed to choose a slot τ_j such as $j > i$ of the same frame of e . This could occur as many times as a τ_j exists over the rest of this frame. Nonetheless, we postponed such a relaxed and expedited convergence for future analysis.

4.6.2 Hidden or Two-Hop Collision

Hidden or two-hop neighbors are a primary cause of collision [81]. The discussion in Section 4.1 highlights collision management via contention-based protocols. Hence, our concern is from having continuous RTS/CTS overhead even when sensor packets are small in size [81]. This overhead can occupy forty to seventy-five percent of channel capacity in sensor networks [81, 73, 107]. The study in [46] confirms a protocol called Five-Phase Reservation Protocol (FPRP) as presented in [116] suffers from an increase in allocation requests and control overhead among one-hop and two-hop nodes when the network size increases. The work in [81, 82] proposed TDMA collision-free schedules, after which no collision avoidance mechanism is needed. However, [81, 82] while scheduling continues using similar RTS/CTS signals, which reduce sensor energy. These studies provide no energy consumption compression between ZMAC, DRAND, and contention-based overheads.

4.6.3 Hidden Collision Handling

The QL-TDMA is a minimum delay and scheduling overhead collision handling via NACKs and random offsets over learning episodes. The nodes switch between only two modes of sending or hearing over each episode. For a send, a node sends a packet within a chosen slot τ_i and back to an idle hearing immediately to detect any one or two-hop collision. This hearing mode can be further controlled based on a node's previous collision detection behavior (no hearing when there is no collision in the last frames). Here, a node can confirm whether a chosen slot suffers a collision

or not by:

- I- For a node a with an episode average degree $\bar{\delta}_e(a)$, such node must receive correct $\bar{\delta}_e(a)$ ACKs. Thus, a missing ACK is likely a one-hop collision since a node is deaf when sending. There is no NACK for a no one or two-hop collision slot. This paradigm is common in wireless sensor networks and usually incorporates NACKs for dropped packets. These ACKs cause an additional overhead [87].
- II- This paradigm summarizes our handshaking system as follows. Node a receives no ACK for successful send as in [46]. Further, when a one-hop neighbor b hears a collision on a 's slot, b will notify each involved one-hop neighbor, including a , about such a hidden collision via NACK. This approach is a No success or collision ACK (NACK) across our work.

Each node during a window of size w frames maintains a collision data record as illustrated for node b of Figure 4.4 in Table 4.1 in which for simplicity $w = 2$. This record stores b 's slot when a one or two-hops neighbor collides with b . The record also has any collision and collision-free slots heard. This record is applied by our immediate reward $R_{\tilde{a}}(\tau_i)$ to move each node towards a collision-free slot via a maximum Q-value move in black of Figure 4.4 and as discussed in Section 4.5.

	One-Hop	Two-Hops	Heard	Free
e_1	2	2	1	5
e_w				1, 3, 4, 5

Table 4.1: Node b Collision Hearing Report

The ACK-based approach is expensive since each ACK must be received correctly for a node to confirm a successful packet delivery. The scenario in which a node a performed a successful transmission and ACKs of two neighbors or more collide when received by a causes confusion. The node a is now confused between having a successful send or suffering from a one-hop or two-hop collision. Each node can handle such confusion via ACKs so that when receiving a successful packet during τ_i , an ACK is sent back after a random offset. This paradigm asks for $2.5 \times (F^+ + d \times F^+)$ milliseconds while $\bar{\delta}_e$ as mentioned is a network average node degree. Hence, over a frame of length $F^+ = 6$ slots and $d = 2$, a node waits forty-five milliseconds before deciding on a new slot. However, via our proposed NACK-based collision paradigm and over $F^+ = 6$ and $d = 2$, we only need $2 \times 2.5 \times F^+ = 30$ milliseconds. During a NACK period τ_i^n as in Figure 4.10a, any heard correct or collided NACK means a neighbor attempt to report a two-hops collision. Nodes send no ACKs and instead hear neighbors until no collision exists. This paradigm benefits from collided signals. Each frame of length $F^+ = \omega \times F^*$ where ω is redundancy factor as mentioned

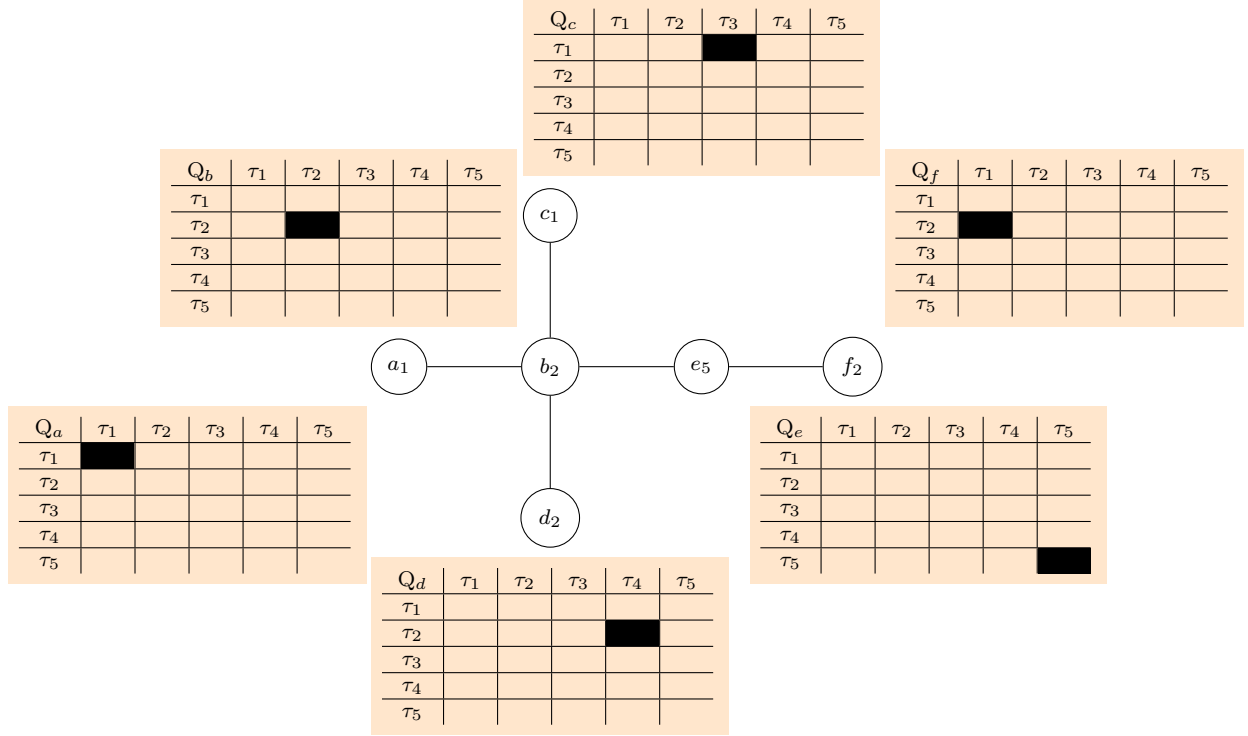


Figure 4.4: Maximum Q-Value Transitions

in Section 4.6.1 and F^* is a minimum frame length for a certain TDMA allocation. For instance, certain circular networks need $F^* = 4$, and for $\omega = 1.3$, our extended frame has $F^+ = 6$. The length of each slot is $\tau_i = 2.5$ milliseconds. Figure 4.5 abstracts two reasonable modifications as discussed above for F^+ both approaches, and summarized by an example. In Figure 4.5a, a node can not confirm a successful chosen slot or not until $\bar{\delta}_e$ ACKs come back. Paradigm-I is implementable by setting a random mini-slot to ACK a successfully received packet by each node. Each slot of Figure 4.5b is detailed in Section 4.6.5.

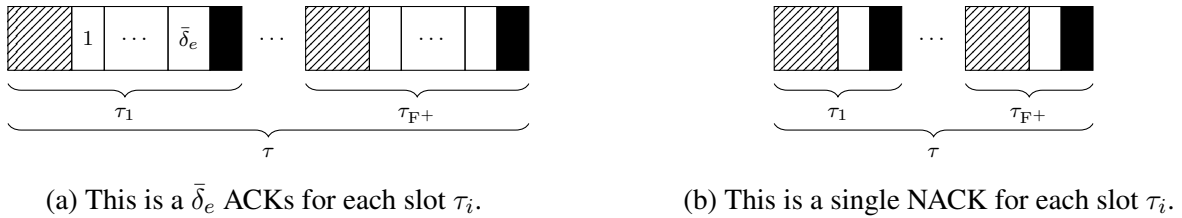


Figure 4.5: Frame Comparison

4.6.3.1 Piggyback Notification

This section compares the conventional piggyback collision system and our proposed NACK-based protocol. Figure 4.6 illustrates a two-hops collision in a linear network of four nodes. Each

node has a unique identification label and a chosen slot ID indicated as a subscript. Each node sends over a specific slot τ_i of a frame τ drawn as two adjacent big squares, each with a left mini-slot carrying a node label. The right mini-slot of each slot has any previous collision slot IDs heard as a piggybacked packet. Figure 4.6 shows four nodes competing over a frame of only two explored slots. The minimum for such a network is three slots. Nodes a and c suffer from a hidden collision issue heard by node b on slot τ_1 . These nodes a and c can not catch such a collision individually. Further, nodes b and d experience a similar collision heard by node c during slot τ_2 . Node b piggybacks slot τ_1 as a collision one and b 's ID before sending during a chosen slot τ_2 . This piggybacked information works as a delayed negative acknowledgment or NACK after performing a checksum error detection for the received message during slot τ_1 . This mechanism notifies both collided nodes a and c . However, only node a hears node b message correctly and infers such a collision on slot τ_1 . Node c cannot extract such essential knowledge since nodes b and d now collide over slot τ_2 as shown in Figure 4.6a. Each outer rectangle includes both mini-slot is a frame slot. This example illustrates a non-converged learning process in which QL-TDMA has scanned so far only two slots. The inner rectangle is either a sent or received node packet. Each Packet color corresponds to a node signal during a dedicated slot, a node subscript. Each packet is composed of two portions. The left one of each packet is sent over a mini-slot and carries a sender ID, and the right half contains all previous collision slots heard (such data is small in size when operating over a minimum frame length). For such a frame, blue signals piggyback heard collisions between red signals. Packets in a sending mode are correct and have no markers (a node is deaf when sending). The \times and \checkmark mark a packet portions status when received by a node either as a correct or collided one. Here, b during a chosen slot τ_2 attempts to inform nodes a and c about their two-hops collision. Node a receives a correct packet with b 's ID, and all two-hops away from a heard collision slots. In contrast, node c while hearing on slot τ_2 receives a collided piggybacked report between b and d . Each sender assumes any empty piggyback portion as a noisy signal that causes a collision. The blue piggybacked information will help select red signal slots for the next frame as with node a by avoiding any piggybacked slot such as τ_1 and any informative slot such as b 's slot. The collided packets from nodes b and d heard by node c during slot τ_2 will be piggybacked by c over a red signal of an immediate next frame to inform such collision sources.

Thus, node c cannot perform a reasonable slot switching from slot τ_1 over the next frame. However, node c would notify collided neighbors, nodes b , and d over the next frame, as mentioned above. This work proposes a new frame structure for early collision detection by sending both portions of each piggybacked packet over distinct data and NACK mini-slots. The data one carries as before a sender ID and a NACK with possibly a vacant data field over a smaller mini-slot called NACK period. Figure 4.6b is frame architecture that allows node b , c , and d to immediately notify collisions on current slots of a frame. Here, we show signal flows over only slots 1 and 2 of a

frame of a minimum of three slots required while learning (not converged yet) is in progress. This design informs each node by the end of its current sending slot when having a collision. The design differs from current ACK-based systems by eliminating all ACK handshaking while scheduling. The piggybacked packet from b to inform c about a heard collision on slot τ_1 also collides with d 's packet as shown in Figure 4.6a. Thus, c remains unaware of its slot collision. The packet from b carries only its ID in one portion, while a receiving node as a knows from listening b 's slot ID. This packet includes any previously heard collision slot ID, like slot τ_1 as a piggybacked portion. The \times and \checkmark mark whether or not a received signal (not a NACK) is correct or collided. The NACK mini-slot is not intended for any data but for an alarm signal, during which a node can infer any two-hop collision over its current slot. This NACK is always meaningful whether correct or collided signal. However, a NACK from b may include its ID to inform a receiving node like a about the place of a collided two-hop neighbor. This early collision awareness allows for quick scheduling schemes in which nodes can change a collision slot immediately within current frame and examine new ones soon after if they exist. For instance, if node b in Figure 4.6a sends on a slot τ_j that is away in time from a 's current slot τ_i , node a would wait for $(j - i) \times \tau_i$ where τ_i is a slot period.

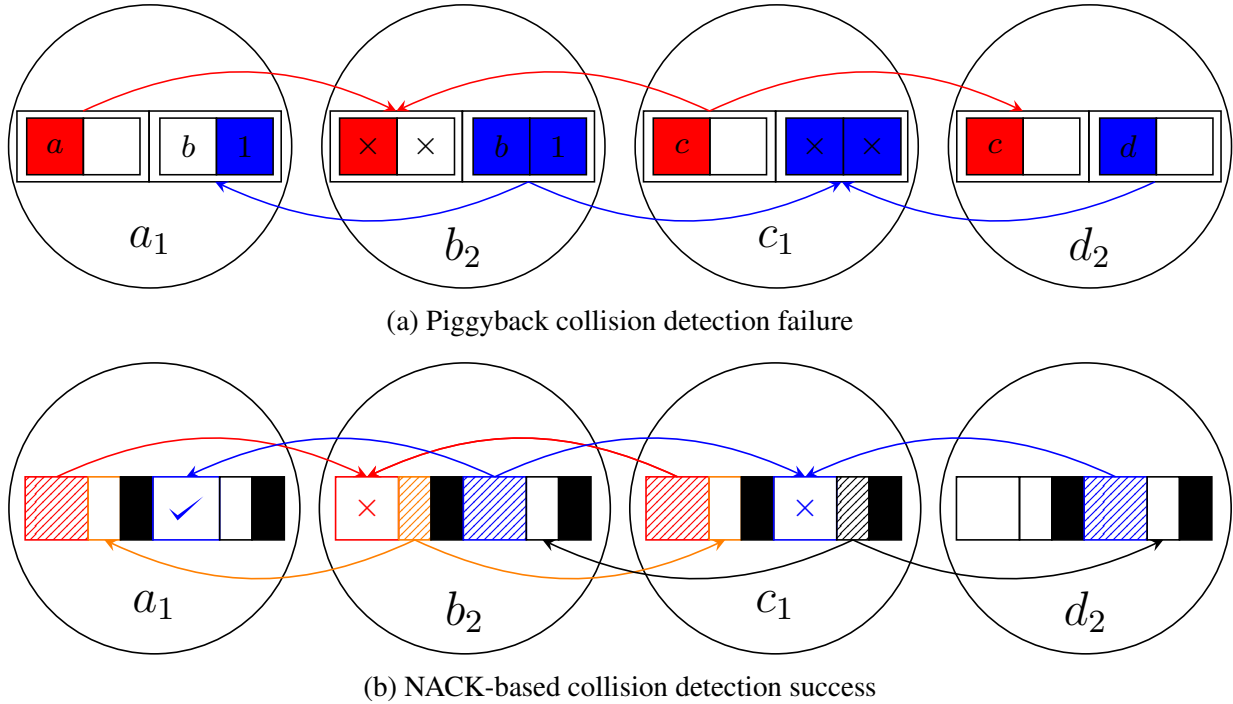


Figure 4.6: Frame τ Architecture with Piggyback and NACK Collision Detection Comparison

4.6.3.2 Random Transmission Offset

The QL-TDMA requires all nodes to be perfectly synchronized. This synchronization is essential once a network is deployed and adjusted during each learning process. The synchronized nodes in advance know the frame length or the number of slots and slot duration as convergence parameters. The QL-TDMA assumes that when two-hop neighbors choose the same synchronized slot, a one-hop neighbor of both can listen to their collision. Random transmission is a resilient option to reduce the likelihood of such a hidden collision. Each node now, instead of transmitting precisely by a start of a slot, such a sending is delayed by a random offset. This mechanism, however, is only assured of avoiding some collisions as a transmission overlap may occur even under such a scheme, especially in dense networks. These dense networks will constrain any random offset, so a remainder of a slot time after an offset is sufficient for data transmission. Nonetheless, random transmissions eventually perfectly overlap over slots as a network size rises. These offset overlaps and vacant slots control detecting one-hop collisions, as discussed in Section 4.6.5.

4.6.3.3 Minimum NACK Time

The only assured approach to detect hidden collisions comes at the cost of increased frame length as [46] applies a slot of fifty milliseconds. This period is divided as (1) For data forty-three milliseconds. (2) For ACKs, four milliseconds. (3) There is a time guard of three milliseconds. Here, we allow a minimum listening period of one millisecond for NACKs and another as a guard following each data mini-slot of τ_i . Figure 4.10a illustrates a breakdown of a QL-TDMA slot, and a random offset is used for detecting one-hop collisions. Each node, after transmission over a specific slot, switches automatically to a receive mode to listen for a NACK, confirming any hidden collision on such chosen slot or a distinct one. The existence of a common one-hop node between two collided neighbors is used to notify for a collision between both neighbors. Each sender on τ_i from any heard correct or collided NACK can confirm a collision on τ_i . Figure 4.7a is an illustration b 's view of a minimum frame of length τ composed of slots of length τ_i each. Each mini-slot marker color refers to the same signal color that is either sent or received during such a mini-slot. The hatched mini-slot is a deaf zone, during which b sends and can not receive, as indicated by a small bar on each unheard signal. The vacant mini-slot means no signal is heard. The \times shows a heard collided signal of arcs of the cross and the same color in Figure 4.7b. Each node sends precisely by the beginning (with no offset) of a dedicated slot typed as a node subscript in Figure 4.7b. In Figure 4.7b, Nodes a , b and c send their signals by the start of slot τ_1 (node subscript) or data mini-slot colored blue in Figure 4.7a with a zero offset. The standard arrow or bar with a particular color indicates a complete or incomplete portion of its signal with the same color is correct data. Thus, such nodes can not hear while sending, and their direct collision is not learned. Node b informs nodes d and e about their hidden collision during NACK duration of slot τ_2 hatched in orange via an orange signal or collision NACK. Nodes a , b , and c over the subsequent

frames may choose different random offsets so their collisions on slot τ_1 can be detected.

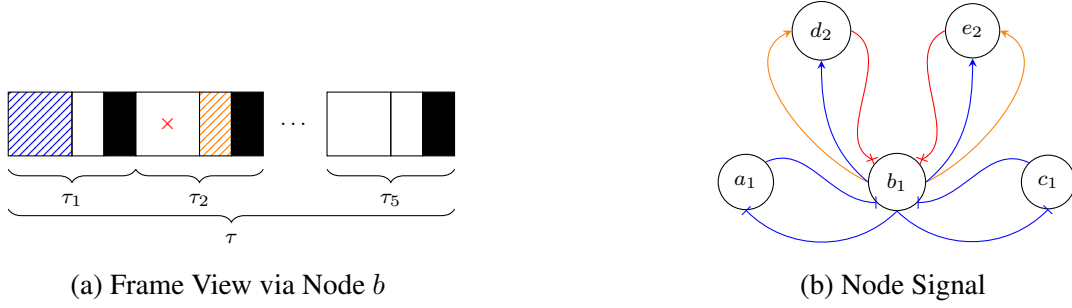


Figure 4.7: Frame with Collision NACK

Figure 4.8 is a NACK-free scenario. Figure 4.8a again illustrates b 's view when no collision is heard over a minimum frame of length τ of slots of length τ_i each. Node b hears correct signals from nodes d and e . Node b also can infer a direct collision over its slot since two neighbors, a and b , do not appear on the frame. These neighbors have a zero chance that they collided as no collision is heard over the frame while having slots 2 and 3 as vacant ones. Nodes a , b and c of Figure 4.8b send signals by the start of slot τ_1 (node subscript) or data mini-slot colored blue in Figure 4.8a with a zero offset, and so their one-hop collision remained undetected over this frame. Each marker is as explained above for Figure 4.6b. Node b sends no NACK for nodes 4 and 5 since no collision is heard.

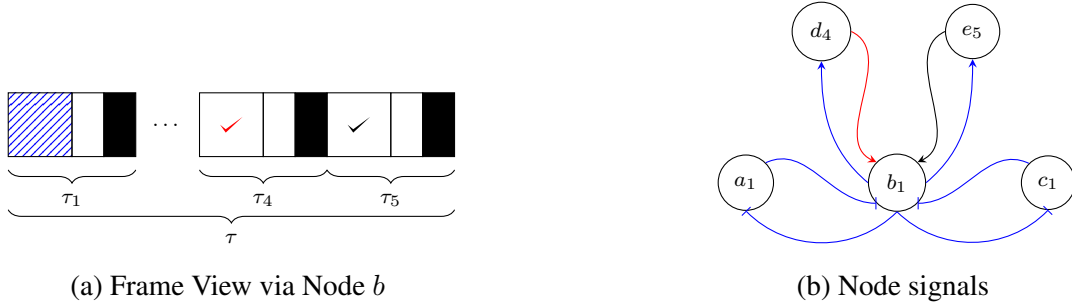


Figure 4.8: Frame without Collision NACK

Figure 4.9 shows a node benefits from a collided NACK. This is an illustration of b 's view over τ of slots of length τ_i in the case when a two-hops collision exists and another collision is heard. Node b hears a collided signal during slot τ_1 and sends a NACK for senders, nodes a , and c during the hatched blue mini-slot. Node b sends during slot τ_2 in red and is deaf in such mini-slot. Node b hears a collided NACKs during the NACK mini-slot of slot τ_2 and infers a two-hops collision as a result. Each remaining slot is vacant. Nodes a and c send signals when slot τ_1 (node subscript)

begins. This is meant as a data mini-slot in the blue of Figure 4.9a again with a zero offset. Node b sends a NACK for a and c to inform their collision. Each marker in Figure 4.9b is as explained above for Figure 4.6b. Node a informs nodes b and d for a two-hops collision, and similarly, node c informs nodes b and e for a similar collision. The zero offset NACKs collide when received by b , and such a collided NACK is enough for b to conclude having a two-hops collision.

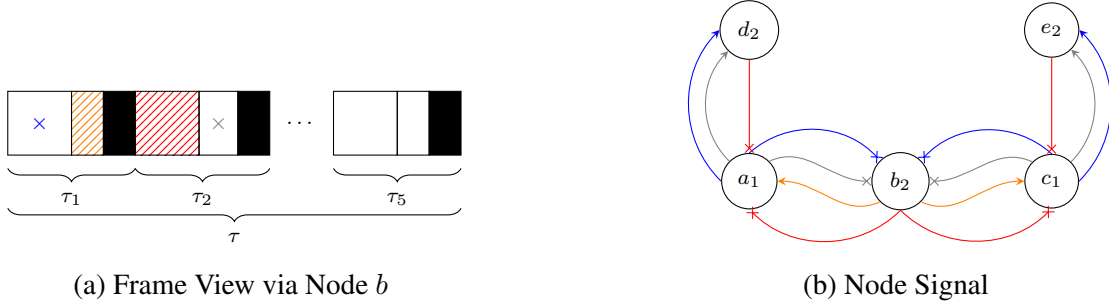


Figure 4.9: Frame with a Benefit of Collided NACKs

4.6.4 Direct Collision

The previous sections only discuss detecting a one-hop or hidden collision among two-hops neighbors via a common one-hop neighbor. The random offset one-hop collision detection, however, was not detailed as all previous examples use a zero offset. This section illustrates by example such a second class of conflict in which one-hop sending neighbors on the same τ_i are not aware of competing on a τ_i simultaneously. Nonetheless, such a scenario is not an actual collision and instead is a hardware limitation that causes deaf nodes and lost packets. Each collision case is composed of two transmitters or more that send to a particular receiver over the same slot. This is because our work assumes a node is deaf and cannot listen while sending on a slot. The proposed NACK signal is used when a node hears a collision to inform any one-hop neighbor of a collision. However, for one-hop collided neighbors, all packets over a certain τ_i are dropped while each node assumes a correct delivery. Therefore, Figure 4.10a represents a random transmission offset within each data mini-slot but not within NACK. Each deaf node cannot catch a neighbor sending over the same random offset of the same slot, and consequently, no NACK is issued for such a neighbor. The constraint of our frame design is functioning via a minimum signal overhead for a minimum network delay. Thus, any approach involving an additional mini-slot is not accepted. The transmission mode for a node starts after a random offset of a dedicated slot while a receive mode is elsewhere in the frame, as shown in Figure 4.10a. Each marker in Figure 4.10b is as explained above for Figure 4.6b.

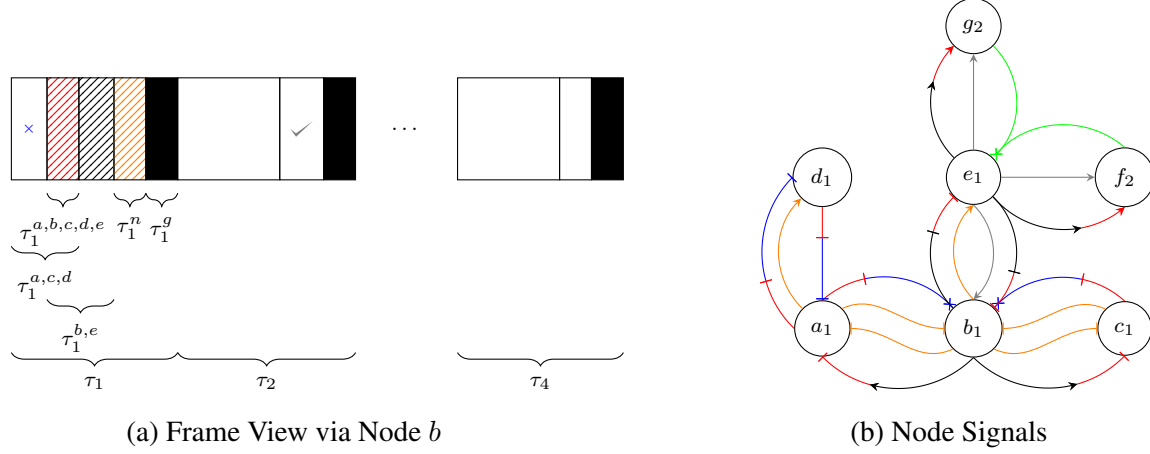


Figure 4.10: Frame with Random Offset and NACK

4.6.5 Direct Collision Handling

Hidden collisions can be detected for two nodes that send over a slot τ_i by a one-hop neighbor sending over a distinct slot to notify such nodes. This section discusses what is called deaf (no listening when sending) node collision detection. This problem occurs because of hardware limitations in conventional sensor networks. There are three scenarios for our one-hop collision detection paradigm. (1) The dense networks raise a node chance over τ_i with a high complete offset overlap with a one-hop neighbor (cannot catch collision) to have at least one neighbor with a partial offset overlap over such τ_i to catch collision. (2) However, when such a scenario does not hold for such a node such as d in Figure 4.10b, another neighbor like a with a **partial** offset overlap with node b would inform d via a NACK on a two-hops collision instead. Nodes a and d also recognize their one-hop collision since no node would hear the other on another slot. Node e also would be informed via b via NACK. However, node e cannot recognize b 's one-hop collision since b may be in a collision on slot τ_2 with nodes f and g when not heard on a certain slot. The learning process makes nodes aware of recent neighbors. Nodes f and g will work to solve such two-hops collision, and over a new frame node, e will work its collision with a if an exact offset again between b and e occurs. (3) Each node works to manage any detected one or two-hop collision within the neighborhood first. The degree-based listening can inform a node on any one-hop collision when a complete offset overlap occurs and no collision is heard, as introduced above and discussed below in Section 4.6.5. Figure 4.10a is an illustration of a slot of length τ_1 viewed by b . Nodes a and c send when τ_1 begins, and stop when $\tau_1^{a,c,d}$ ends. The \times mark indicates a heard collided signal. Node b sends by the start of $\tau_1^{b,e}$ and ends when $\tau_1^{b,e}$ ends. The hatched zone indicates a deaf zone for b . The τ_1^n is a NACK period, for which nodes are in listening mode for nodes **usually** on distinct slots. In Figure 4.10b without nodes d and e , no NACK from nodes on the

same slot, such as a and b , is necessary. However, allowing such NACKs from a collision-aware node on slot τ_1 such as a and b informs nodes in deadlock scenarios such as d and e . This collision awareness is because of random offset. The $\tau_1^{a,b,c,d,e}$ is a deaf zone for all nodes acting in send mode. The τ_1^g is a guard or no-send zone, during which a transmit is prohibited. Node, nodes a and c hear a correct partial packet from b . The missed portion is sent during a and c 's sending hatched in red in Figure 4.10a. Thus, nodes are asked to NACK any partial correct or collided packet heard. Node b in Figure 4.10b during $\tau_1^{a,c,d} - \tau_1^{a,b,c,d,e}$ of τ_1 hears nodes a and c partial and collided packets as both send when τ_1 begins. Hence, b sends NACK to inform neighbors during τ_1^n . However, neither of a , b nor c hears a NACK during τ_1^n as a deaf zone as orange arcs show in Figure 4.10b. Further, although a and c are aware of such a one-hop collision with node b because of a shortened packet over $\tau_1^{b,e} - \tau_1^{a,b,c,d,e}$, neither a nor c is aware of having a two-hops collision without a heard NACK from b . This rule is highlighted as such. The NACK sent by a node with a detected one-hop collision via random offset such as a , b or c would not be receivable by any collided one-hop neighbor with a non-overlapped offset since such a neighbor also would send (deaf) a NACK during τ_1^n .

These NACKs from nodes a and c inform d and e for having two-hops collisions, as mentioned earlier, with a and c , respectively. The frame view via d confirms a one-hop collision with a by comparing $\delta_e(d)$ with the number of correct slots heard. Further, having d and e on slot τ_2 instead, a NACK from a and b will also help d , and e avoid slot τ_1 over a new episode. Node b hears no two-hop collision on a different slot and has a degree $\delta_e(d)$ equal three, b concludes that a number of neighbors of $\delta_e(d)$ minus the number of vacant slots is in a direct collision over its current slot. This is helpful when both a and c partial and collided packet has not been heard by b . The degree $\delta_e(a)$ of a while QL-TDMA is exploring over a minimum frame length can be estimated. This a needs to confirm whether or not its slot is a one-hop collision. Having no hidden collision heard by a and a vacant slot of $\delta_e(a)$ or more confirms a one-hop collision on τ_i . This is why a two-hops collision confuses a node one-hop collision. Thus, QL-TDMA prioritizes informed two-hops collisions to help discover offset-escaped collisions. In summary, each heard two-hops collision causes two vacant slots or more of $\delta_e(a)$. Hence, a with a sending τ_i while hearing a single two-hops collision and only two vacant slots of $\delta_e(a)$ concludes by having no one-hop collision over τ_i . However, a is confused when hearing three vacant slots of $\delta_e(a)$ along with one two-hops collision since one neighbor of such collided two-hops ones could be involved in a τ_i one-hop collision.

4.6.6 Exposed Collision

The exposed one-hop problem occurs in contention-based scheduling schemes when a packet is not sent from a node to a designated receiver because of a co-channel collision with a neighboring sender. This issue shows up as active senders back-off each other after medium sensing of an under-loaded network. The sender b in Figure 4.11 intends to send a message to a , and a sender

c is sending to node d . Node b when hearing node c on slot τ_1 , b signal in blue is backed off, although no collision would occur for a as a and c are not neighbors. IEEE 802.11 RTS/CTS mechanism helps to solve such a problem when nodes are synchronized and packet sizes and data rates are identical for senders. For a node when hearing an RTS from a neighboring node without a corresponding CTS can confirm an exposed collision. Hence, such a node is now permitted to transmit to other neighbors [14]. This scenario is manageable when a network is under-loaded, so control messages are received correctly via IEEE 802.11 RTS/CTS. For current systems, even when node b receives a collided CTS because of a neighbor out of c 's range but uses c 's slot, such a collision exists. This issue could occur in TDMA scheduling when random transmission offsets and backing-off are applied. However, in zero offset systems of unified rate and packet size, an exposed collision is a challenge. This concern while sending is not an issue for QL-TDMA since nodes are deaf and view any exposed back-off as a one-hop collision in which nodes are required to modify slots until solved.

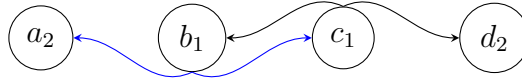


Figure 4.11: Exposed Node Collision

4.7 QL-TDMA Evaluation

This section discusses our QL-TDMA scheduling performance. This evaluation is performed by examining QL-TDMA over a set of distinct topologies as summarized in a minimum size in Figure 4.12. Here, numbers in nodes indicate particular slot IDs allocated for no collision. For a linear network, for example, only a frame of a minimum of three slots is required for no collision. The cycle one, however, such a length varies according to the number of nodes. The star network is expensive over TDMA because of requiring a frame of length equal to the number of nodes. The frame length of a random topology is a function of network degree. In Section 4.7.1, convergence in frame collision is discussed. The convergence in learning rate is presented in Section 4.7.2. The implemented reward mechanism is described in Section 4.7.4.

4.7.1 Frame Collision Convergence

The proposed QL system aims for zero collision TDMA schedule for all nodes. The learning process guides each node to avoid collision over episodes. For each learning episode or frame, Figure 4.13 shows a total number of nodes whose control message is not delivered correctly to at least one one-hop neighbor. This result indicates that nodes learn collision-free TDMA frames by reducing collision over each episode. The distributed learning update is according to each chosen action. Figure 4.13 shows a convergence in frame collision count as we vary w . Figure 4.13 also shows an early convergence in collisions as w raises since nodes would have rewards based on

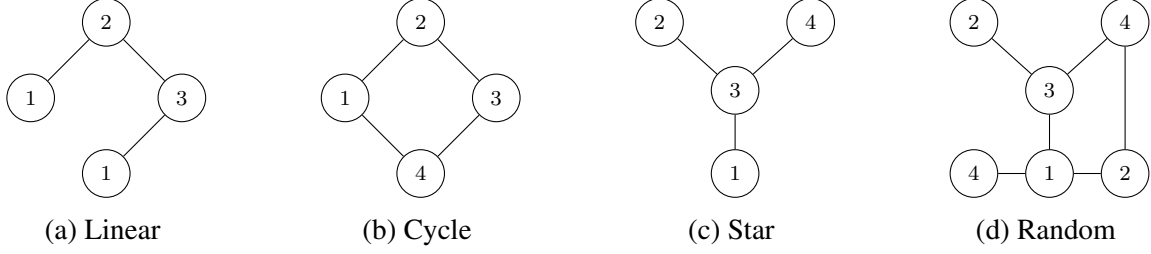


Figure 4.12: Mini Examples of Evaluated Topologies

current and more previous actions with large w values. Figures 4.14, 4.15, 4.16 and 4.17, in contrast, show a relationship between collision count and network size when having a constant w and ω . These plots indicate a longer convergence time when the frame length increases. Figures 4.14 and 4.15 are for linear and cycle topologies of distinct sizes over a frame of three slots as we notice shorter convergence when compared with the random or star topology over more extended minimum frames in figures 4.16 and 4.17 respectively.

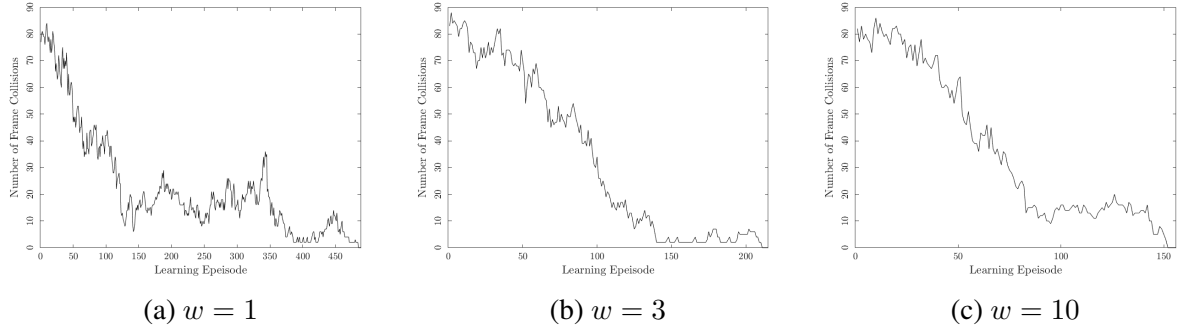


Figure 4.13: Linear of Hundred Nodes and with $\omega = 0$

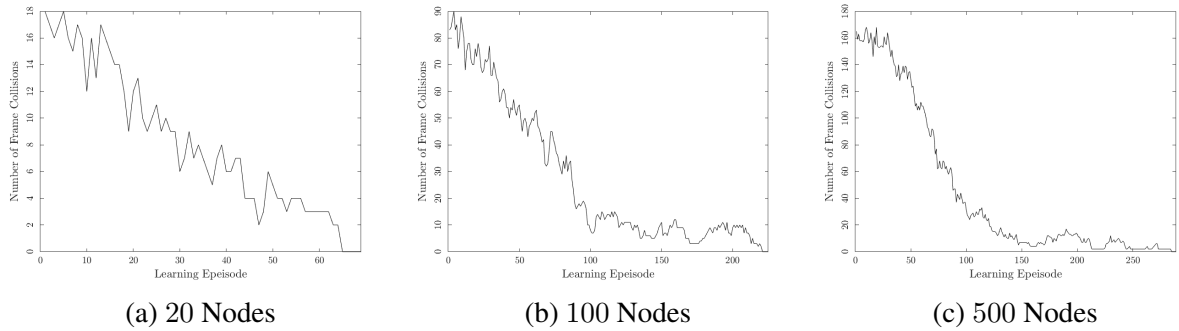
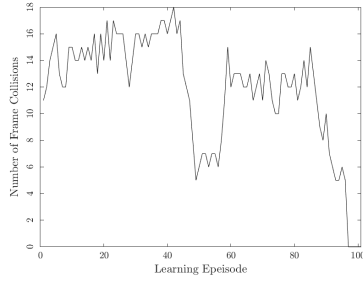
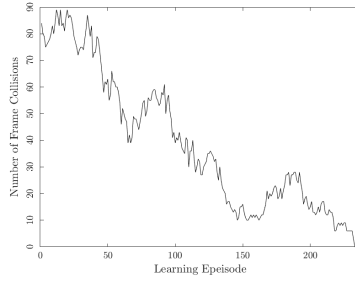


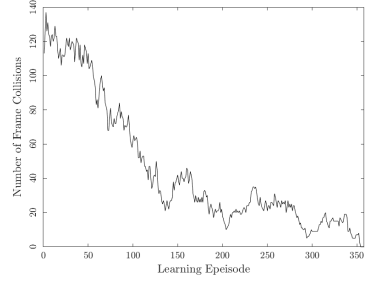
Figure 4.14: Linear over Three Slots with $w = 3$ and $\omega = 0$



(a) 18 Nodes

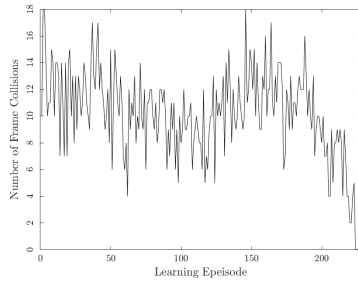


(b) 108 Nodes

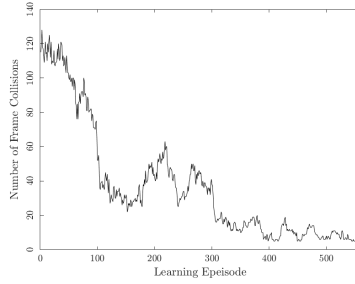


(c) 504 Nodes

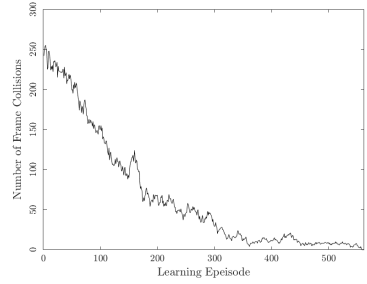
Figure 4.15: Cycle over Three Slots with $w = 3$ and $\omega = 0$



(a) 20 Nodes over Ten Slots

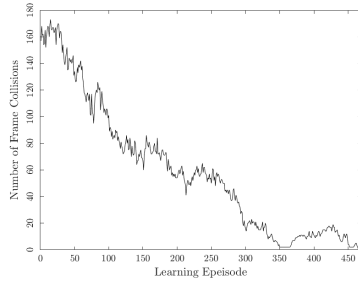


(b) 100 Nodes over Twelve Slots

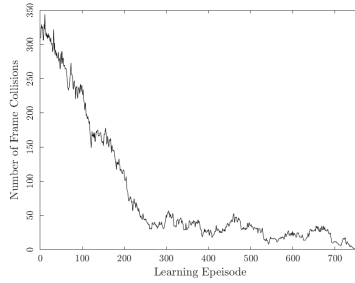


(c) 500 Nodes over Thirteen Slots

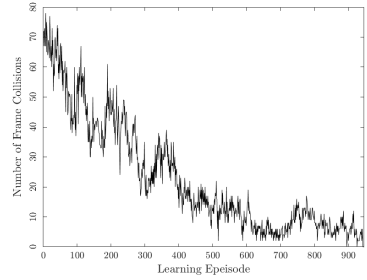
Figure 4.16: Random with $w = 3$ and $\omega = 0$



(a) 20 Nodes over Twenty Slots



(b) 100 Nodes over Hundred Slots



(c) 500 Nodes over Five-Hundred Slots

Figure 4.17: Star with $w = 3$ and $\omega = 0$

4.7.2 Learning Rate Convergence

The update of each node Q-value is governed by a learning rate α , by which a node weights the previous value Q^- and the corresponding next value Q^+ and reward update. Nodes start with proportionally high learning rates so that more weight of α is given for Q^+ , and converge when

α declines since a weight of $1 - \alpha$ is given for Q^- . This α is updated each action a according to $\alpha_e = \sqrt{\frac{\alpha_0}{\alpha_0 + c_{\tilde{a}_w^+}}}$ where $c_{\tilde{a}_w^+}$ is the number of times action \tilde{a} has been performed with no collision over w as mentioned in Section 4.5. Figure 4.18 shows α_e convergence over different networks of hundred nodes. The α convergence over e is long as expected for longer minimum frames as nodes compete over a large state space.

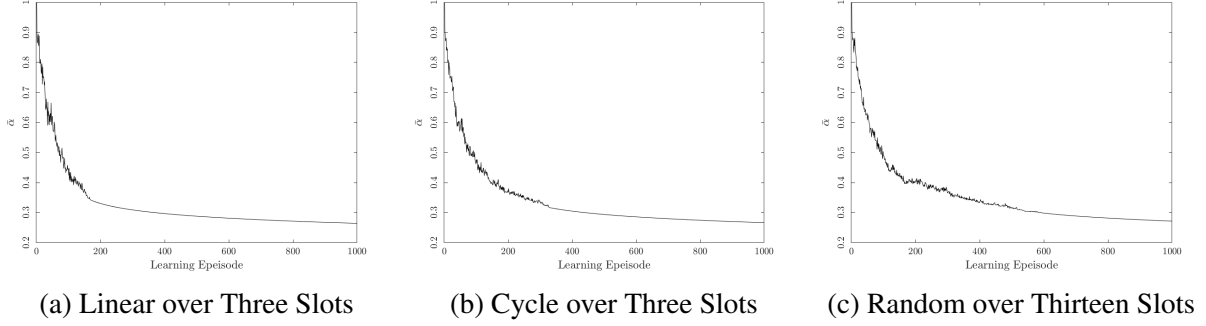


Figure 4.18: Learning Rate of 100 Nodes with $w = 3$ and $\omega = 0$

4.7.3 Q-Function Convergence

The Q-function derives learning for a collision-free TDMA frame. Figure 4.19 shows an average Q-value for each slot τ_i for an individual node. The curves exhibit a raise for a convergence slot Q-value and diminish for any other slot. This raise is caused by the immediate reward function expressed in Section 4.7.4 overall learning episodes before converging because of a decline in learning rate as learning progresses.

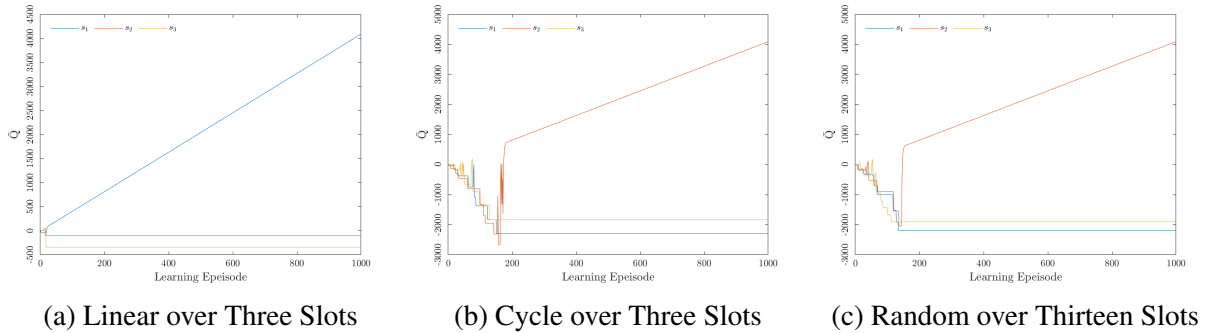


Figure 4.19: Q-Value of a Node of 100 Nodes with $w = 3$ and $\omega = 0$

4.7.4 Reward Function Convergence

This work proposes a unique piece-wise reward function for discrete parameters such as collision count over a learning frame and collision-free count for the number of listened-to slots. The

collision count over a window w of learning frames and for a particular slot is the number of frames a node chooses such a state and sees no collision. This count for a node a is a punishment or negative reward parameter for a function $\Gamma_a^- = -e d(a) \sqrt{\exp[\tau_i^{w-}]}$ where τ_i^{w-} is a collision count for τ_i over w given that τ_i is currently a collision one. The positive or success reward is a similar function $\Gamma_a^+ = e d(a) \sqrt{\exp[\tau_i^{w+}]}$. This τ_i^{w+} is a no collision count for τ_i over w . The last function is a positive reward to credit node a for any successful heard slot of a neighbor. This credit is for not choosing such a slot and causing a collision. This function is expressed as $\Gamma_a^\circ = e \sqrt{\exp[-\tau_j^{w+}]}$. The τ_j^{w+} is a count for a heard collision-free τ_j while sending over τ_i . Figure 4.20 illustrates a single node convergence in reward for a collision-free slot. The immediate reward value for a node when accessing a collision slot τ_i is expressed as follows. $R_{\tilde{a}}(\tau_i) = \Gamma_a^-$. For a no collision slot, τ_i such an immediate reward is a combination of $R_{\tilde{a}}(\tau_i) = \Gamma_a^+ + \Gamma_a^\circ$.

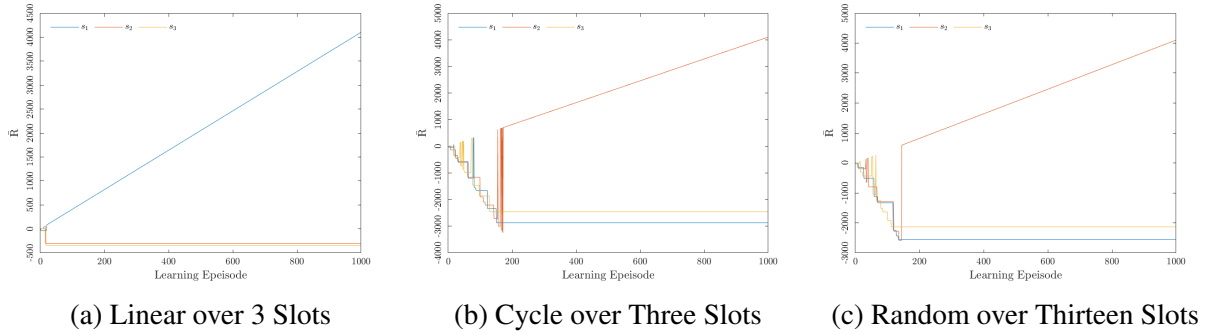


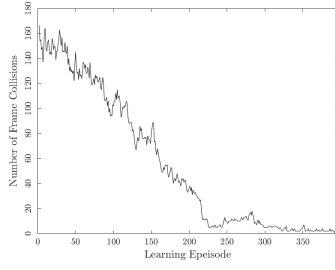
Figure 4.20: R-Value of a Node of 100 Nodes, $w = 3$ and $\omega = 0$

4.7.5 Frame Redundancy and Convergence

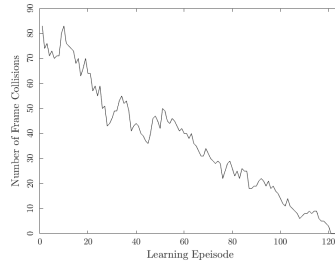
The proposed Q-learning system in this work is examined over a minimum frame length for all tested typologies indicated in Section 4.7. However, learning for such a strict demand is a challenge. Here we introduce an extra number of slots for each topology to speed up the convergence process. This number is called ω . Figure 4.21 summarizes such a performance for a linear network as we notice a decrease in convergence time as ω is increased.

4.8 Conclusion

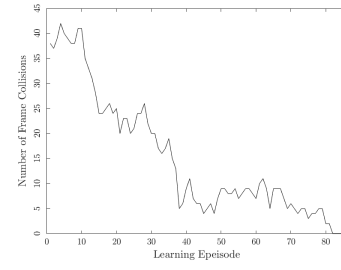
This work has shown that a distributed Q-learning-based approach is suitable for distributed TDMA frame scheduling. Results indicate such an approach with a careful parameter setting can lead to a minimum frame length. This QL-based scheduling is sensitive to two parameters. These are the collision window and frame slot redundancy. The proposed Q-learning system permit nodes to exploit and switch slots by each frame end. This, however, can increase convergence time. The next version of our model allows nodes to change slots within a frame immediately after being informed about a collision according to what slot information is available. This will adopt a more



(a) $\omega = 0$



(b) $\omega = 1$



(c) $\omega = 2$

Figure 4.21: Linear of 500 Nodes and $w = 3$

robust penalty function for any node with a high slot competition within a frame.

5 Interference Handling for Collaborative Multihop Cellular Networks

Dynamic spectrum access is proposed to utilize the scarce spectrum resources better. Recently, the FCC opened up the Citizen Broadband Radio Service (CBRS) for wireless service providers to enable dynamic spectrum access for 5G networks. However, the lack of collaboration among cellular providers has been hampering reliable access due to the increasing interference from multiple carriers. This work considers a new network architecture to foster collaboration among providers through Collaborative Multihop and Multi-Channel Cellular Networks (CMCNs). For CMCNs, we propose protocols to manage interference and congestion by utilizing minimum back-off schemes. Here, we present six policy-based scenarios that summarize the scheduling paradigm. Moreover, for the NP-hard Minimum Broadcast Delay Forwarding (MBDF), we develop a near-optimal mapping paradigm for a TDMA scheduling policy, which approximates polynomial scheduling for MBDF. The simulation results show that our approach outperforms existing models in terms of scheduling delay and message redundancy. The confidence in our proposed scheduling has been examined over distinct sets of CBRS channels by comparing each MBDF with the corresponding initial and minimum schedule demand.

5.1 Background

The cellular licensed spectrum has become very crowded due to the tremendous growth of mobile communications [100]. The current WiFi and Citizens Broadband Radio Service (CBRS) are examples of publicly available free bands, which have been the breeding ground for technology innovations. Each spectrum band is divided into a set of channels. The CBRS is administrated via three tiers of access schemes to accept the commercial share. These tiers are as follows: Incumbent Access (IA), Priority Access (PA), and General Authorized Access (GAA). Tier-1 for IA includes authorized federal users, satellite services or space-earth stations, and grandfathered wireless broadband licensees. The IA subscribers receive protection against interference from Priority Access Licensees (PALs) and GAA users. Tier-2 consists of PALs classified based on routinely competitive bidding on a county-by-county basis. Tier-3 permits open and flexible access to CBRS via a diverse group of users. The GAA users have no interference protection from all three tiers of users.

The automated Spectrum Access System (SAS) acts as a coordinator to manage access. This uses sensory data from an Environmental Sensing Capability (ESC) for detecting any military radar access. The US Federal Communications Commission (FCC) established rules for commercial CBRS usage in 2012 while preserving subband to prevent interference on military communications [28, 99, 5]. Though FCC has waived such access restrictions in 2020 [29, 43, 33]. Mobile carriers can now access CBRS as GAAs without license [10]. However, as a shared spectrum, CBRS requires unlicensed LTE systems to manage their interference to coexist with other networks.

Multihop Cellular Networks (MCNs) have been proposed as a cellular extension to incorporate the flexibility of ad hoc networks by realizing end-to-end communication among individual mobile stations [44]. MCNs enhance service quality by reducing call-blocking and bit-error-rate [44, 12]. The relay architecture in MCNs, however, is in an early scale of development and deployment. To increase capacity and better leverage the spatial reuse, more Base Stations (BSs) or eNBs as in LTE-A are desirable [44]. However, it is expensive to deploy such wired backhaul transceivers. The challenge, however, is that the CBRS databases do not manage conflict among unlicensed LTEs, and current LTE standards are equipped with centralized and private interference coordination [10] over licensed spectrum. The throughput in LTEs can degrade by a magnitude of ten without such coordination [10].

Instead of a centralized and wired backhaul, a more economical approach is to deploy a Relay Node (RN) for multihop communication between a Mobile Node (MN) and BS. The Current LTE-A relay standard is restricted to one or two hops [12]. The central reason for using one-hop standards is to reduce routing and interference handling complexity. Nonetheless, multihop relays could potentially increase capacity and coverage in future LTEs, particularly for urban and sparse deployments [12]. Furthermore, more spectrum reuse from CBRS would increase capacity due to RN's small power consumption. Existing BS \rightsquigarrow RN links in MCNs access a unique band [114, 105]. The CBRS can provide a separate relay band. Here, our vision is that mobile carriers could use CBRS as a free band to build new and less expensive Collaborative Multihop Cellular Networks (CMCN) for best-effort cellular services. However, such a shared band is a cause of access complexity. Thus, a well-designed spectrum allocation is necessary to manage the intensified relay interference over dynamic CBRS access when demanding higher data rates over single radio LTEs.

Private LTE-based such as MulteFire [77] uses unlicensed and shared UNII-3 WiFi spectrum of 5 GHz. MulteFire uses Listen-Before-Talk (LBT) interference management in coexistence with WiFi. MulteFire leverages micro cells as WiFi access points. Permissibly, such cells access UNII-3 using over-the-air contention management without regulatory units such as CBRS databases. The LBT mechanism allows coexistent networks over a shared spectrum. However, for a CMCN, the LBT mechanism cannot guarantee an overall minimum delay scheduling due to the limited communication range of RNs. In contrast, our work obligates a centralized interference view like SAS in CBRS. Each collaborative RN listens to CBRS channels, and reports available ones to databases to construct a continuously updated CMCN conflict view.

This work proposes a generic policy-based interference management protocol to handle GAAs interference within CBRS-based CMCNs. Major mobile carriers such as AT&T and Verizon can be classified as PA or CMCN class-I users. T-Mobile and Sprint, for instance, might access as GAA or class-II users. This work allows class-I users to collaborate with class-II providers for mutual

benefit. For such a collaborative paradigm, our work is a baseline for interference handling for such a benefit. The overall CBRS access of both classes remains coordinated by SAS as described above. In a nutshell, our research aims to determine an optimal CBRS channel allocation for CMCN RNs to minimize broadcast delay while achieving maximum coverage. The search for a Minimum Broadcast Delay Forwarding (MBDF*) in multi-channel and multihop systems is an NP-hard problem [50, 70, 76, 93]. For such complexity, we aim to narrow a path for a near-minimum delay scheduling for CMCNs over CBRS.

This work makes the following contributions: (1) To facilitate collaboration across multiple mobile carriers, we propose a new CMCN approach over CBRS. (2) We present a policy-based mechanism to manage conflicts between class-I and CMCN class-II users and achieve a near-minimum delay for multihop cellular networks. (3) We conduct a simulation to demonstrate the performance of the proposed approach in dense and large-scale collaborative networks.

The remainder of the work is organized as follows: In Section 5.2, we summarize related studies. The discussion in Section 5.3 describes our problem and motivation. The CMCN architecture over CBRS is discussed in Section 5.3.3. The used methodology is detailed in Section 5.4. The intra and inter-level conflict management is introduced in Section 5.6. The details of the inter-level policy **P6** such as RN dependency are explained in Section 5.7.3. The evaluation of all proposed schemes is shown in Section 5.8. Finally, our work is concluded in Section 5.9.

5.2 Literature Review

Hsu and Lin in [44] presented an overview of an MCN resource allocation framework for out-of-band relay and throughput enhancement. The study in [76] considers a Minimum Broadcast Delay (MBD) scheduling in multi-channel and multi-radio mesh networks. For MBD, [76] proposes four heuristic and centralized algorithms to search for a low-latency broadcast schedule. The simulation shows that a channel assignment mechanism designed for unicast might not work efficiently in a broadcasting scenario. The work in [113] proposes a distributed broadcast tree-based scheduling algorithm, which utilizes local information only. [10] demonstrates that fast channel allocation might not lead to an optimum schedule. The authors in [10] also described an interference management protocol for single-hop communication of Tier-3 users over CBRS. Their work considers interference as a less changeable graph since recalculating chordal graphs for each change is expensive. This is why [10] updates interference only when a new AP is introduced, which is not practical for dense CMCNs. In contrast, our work assigns each new RN to a conflict component of RNs of a previously optimized schedule. The newly introduced conflict is managed only between such component and each dependent one over a forwarding broadcast path as opposed to recomputing a whole new chordal graph. The work in [7] is an interference management protocol for mobile users who are assigned to stationary base stations that operate on a private spectrum. However, our work manages conflict among RNs of collaborative mobile carriers over the shared

CBRS spectrum.

Interference is an asymmetric phenomenon and depends on the signal strength and many environmental factors. In contrast with [10] and [7] that assume either a fixed or cumulative interference intensity at small scales, our work varies interference intensity as a driving factor of the performance of each proposed policy. Triangulation is used in [7] to chordalize a conflict graph by adding virtual interference edges to produce a chordal graph with a linear complexity when searching for a number of cliques. However, this produces a sub-optimal schedule regardless of any restoration action performed afterward.

5.3 Problem Statement and Motivation

5.3.1 Network Model

This work considers a CMCN composed of a set \mathbb{R} of relay nodes and \mathbb{B} of base stations deployed over a two-dimensional square area \mathcal{A} that is divided into squared grids of size $g \times g$. This work considers three mobile carriers who evenly contribute to both \mathbb{R} and \mathbb{B} . The proposed shared infrastructure of distinct mobile carriers as demonstrated in Figure 5.1 could benefit from existing spatial diversity without demanding a new hardware installation. Here, we assume all BSs are wired and have no bandwidth constraints. The proposed CMCN is interconnected via radio links and composed of heterogeneous RNs of distinct cellular providers. These RNs are distributed in cells according to a general 2D Poisson Point Process (PPP). The local intensity over a unit area ∂g of a grid is denoted by $\phi(\lambda)$ where $|\mathbb{R}| = \int_{\mathcal{A}} \phi(\lambda) \partial g$. Each RN in \mathbb{R} accesses a set of size \bar{c} of CBRS channels. However, a valid PPP distribution must satisfy: (1) For any area of size $\mathcal{A}' \subset \mathcal{A}$, the number of RNs is a Poisson random variable with density $\lambda' = \mathcal{A}'\lambda$ where λ is RN arrival rate over \mathcal{A} . (2) The result of any collection of RNs arrivals of a set \mathbb{G} of disjointed grids each of an area $g_k \times g_k$ where $1 \leq k \leq |\mathbb{G}|$ is mutually independent Poisson random variables each with density $\lambda \times g_k \times g_k$ of relay stations.

Hence, with $p(|\mathbb{R}| = r) = \frac{\lambda^r e^{-\lambda}}{r!}$, a CMCN would have r arrivals or active relay stations over \mathcal{A} as $|\mathbb{R}|$ is considered as a PPP, and all RNs are uniformly distributed within cells. RNs are adjacent if they are located within a communication range α of each other. RNs interfere with each other if they are placed within an interference range $\omega \times \alpha$ where ω is interference intensity. Each incumbent or licensed access activity database is modeled as a uniform process over \mathcal{A} . Each incumbent or licensed user could cause a grid GAAs or RNs to be partially or fully inactive over CBRS. Each RN a in our modeled CMCN is considered a weighted node. The weight $w(a)$ of a relay station or capacity is a summation of all connected end-customers weights. Thus, a candidate MBDF* is a frame of a minimum number of slots of milliseconds, in which RNs can be assigned to propagate a broadcast message from a source RN. The initial MBDF⁻ before placing conflict between RNs is a minimum spanning tree rooted by a source RN. The overall weight of such a schedule includes all active and covered RNs from such a source. RNs, collaboratively, determine

their conflict sets (predominately of mobile clients) over their CBRS channels. This data afterward is sent to the nearest control unit (a dedicated BS controller for instance) along with grid-specific capacity and RN parameters. Table 5.1 summarizes our protocol notations used while managing conflict among RNs.

Notation	Definition
MBDF^-	Initial minimum broadcast delay schedule with conflict
MBDF^+	Relinked minimum broadcast delay schedule with conflict
MBDF^*	Minimum broadcast delay schedule with no conflict
LVD^-	Unsolved conflict-free RNs miss dependency digraph
LVD^*	Solved conflict-free RNs miss dependency digraph
CFCR	Conflict Free Component RNs
$\bar{y}(a)$	Level or TDMA slot of RN a in MBDF^-
\bar{c}	Number of channels in a set c

Table 5.1: Notations

5.3.2 Interference Protocol Model

Interference in wireless networks over the shared spectrum is hard to eliminate and analyze. This issue of conflict occurs when an intended signal overlaps with undesired signals sent at the same time, frequency, or spatial channel. Many common assumptions are used when modeling interference in wireless networks such as spherical interference and symmetric links [58] of identical ranges for all radios [57]. There are many interference models for capturing the interference effects. The most prominent are: Protocol Interference Model or PrIM [88], Interference Ball Model (IBM), and the physical model [88]. The PrIM has been formalized and validated in many studies [88]. Following PrIM, interference occurs if an interference source resides within a certain distance range from a receiver. This range depends on a received power from the intended transmitter and a minimum Signal Interference Noise Ratio (SINR) that allows successful signal decoding. The simplicity of PrIM fails to capture additive interference aggregation or the sum of interference power from multiple interfered nodes [57]. Each RN with PrIM is not permitted to access a channel c if an IA uses c within the range $\omega \times \alpha$. Table 5.2 summarizes all PrIM interference cases handled in our simulations.

Conflict Pair	Hardness	Handling
\mathcal{I}_1 : RN v.s IA/PAL	Low	Precluding any cause RN via SAS
\mathcal{I}_2 : RN v.s RN	High	P1-P6

Table 5.2: PrIM Simulated Conflict Cases

5.3.3 CMCN Overview

This work considers a carrier-I in black and carrier-II colored brown of class-I and a carrier-III in gray of class-II as cellular providers who share private RNs of \mathbb{R} for accessing a set \mathcal{C} of non-overlapping CBRS channels of 10 MHz. The current LTE-A systems over CBRS allow such carriers to work as a CMCN. This collaboration is for: (1) Minimizing carrier-II deployment cost in a region with an installed carrier-I infrastructure. (2) More resilient and rich cellular pool of carriers among which end-users could switch services as desired. This means an end-user belonging to a carrier-II could be served via carrier-I or carrier-III LTE-A hardware. (3) Providing such an active LTE-A access system for the increased demand and high-speed technologies like the 5G evolution. (4) Mutual benefit among mobile carriers carrier-II and carrier-I for instance when carrier-II is allowed to serve new customers in regions on which no carrier-II infrastructure is installed while carrier-I in return is receiving direct credit for carrier-II's access in addition to its original revenue of such low demanded region of subscribers. (5) Perform an end-user load reduction, and overcome LTE-A failures from carrier-II via such CMCN when carrier-II services handover is limited. (6) Long haul and more continuous service for mobile users. (6) Pairing licensed and unlicensed users over shared bands like CBRS allows more collaborative and private LTEs.

Private LTEs can serve in Tier-2 at a lower subscription cost with SAS or as tier-3 cost-free systems. This cost can be determined via auctions and based on the geographical size and length of a license. This cost is more negligible, in contrast, to what mobile carriers pay for licensed spectrum [105] and [64]. This CMCN is suitable for such private LTE(s) that interconnect with existing mobile carriers when connecting long-haul branches. Therefore, mobile carriers will prosper in a more collaborative industry where individual business schemes may find it difficult to succeed. However, interference and congestion as principal challenges behind such isolated LTE-A carriers will be a more persistent dilemma. This work offers an optimistic analysis for relay conflict management in CMCNs. The aim is to achieve a minimum delay and redundancy in addition to a maximum capacity broadcast schedule for CMCNs.

End users during pandemics or system failures for instance need services adequate for their requirements. These services can be fulfilled in collaborative LTE systems of a low cost using models like CMCN over shared bands such as CBRS. Figure 5.1 is an abstract of a shared LTE-A network over a shared CBRS spectrum. This RN sharing forms a CMCN of three mobile carriers as mentioned above. The mobile RN-4 is down. Each BS \rightsquigarrow BS link exchanges request RN in red and grant RN in blue messages. This message is composed of a requester BS ID, a granter BS ID, and a requested RN ID from left to right. The databases maintain a history of such requests, and an approved request is disseminated by the first database that received the first grant message for such a request to all databases to be aware of the CMCN sharing status. These CMCN databases acquire CBRS access and RN status. Periodically, carriers can master such conversations with databases.

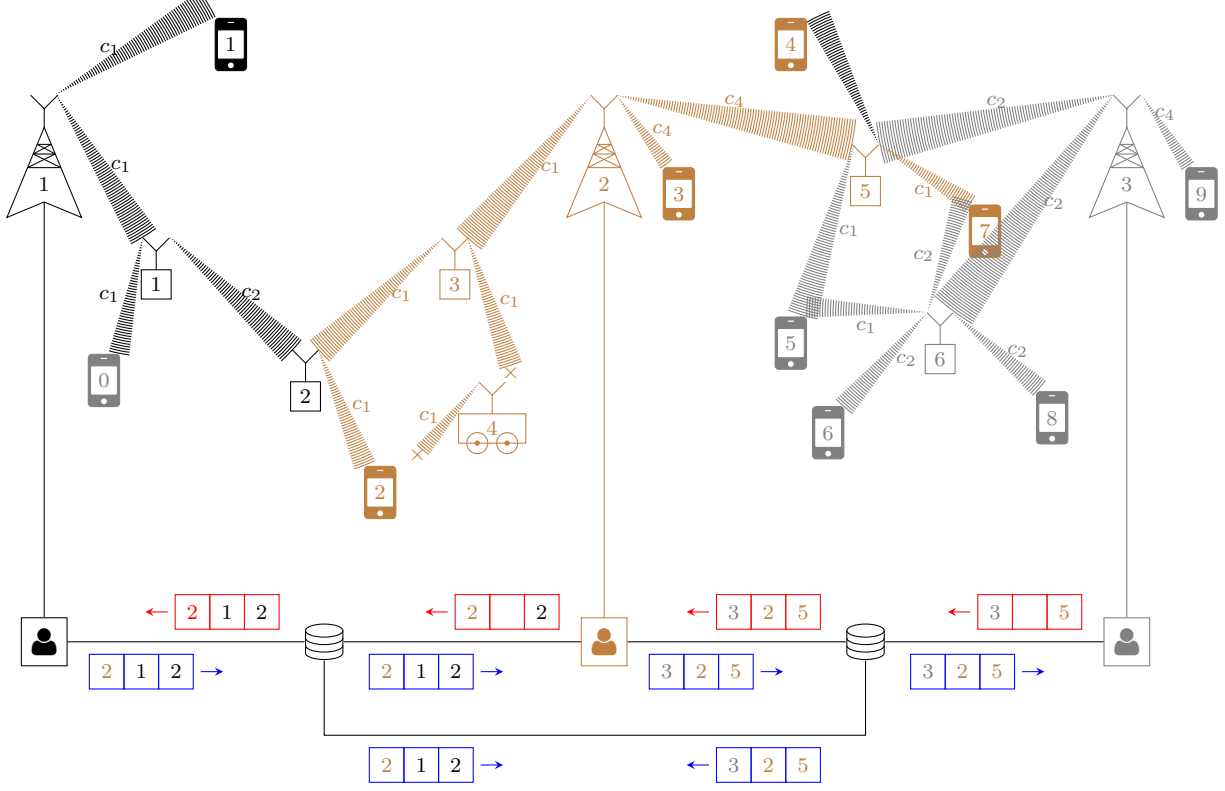


Figure 5.1: Three Mobile Carriers in a CMCN

RN-2 is granted via BS-1 to BS-2 to overcome RN-4 failure for covering MN-4. From Figure 5.1, BS-2 is owned by a mobile carrier-II, and attempts to establish a connection between two subscribers, MN-2 and MN-3. The mobile coverage station RN-4 for MN-2 is out-of-service because of a hardware failure. BS-2, consequently, sends a share request via cable to the base station that owns the nearest RN for MN-2 namely RN-2 as indicated by the request-grant messages between the two base stations. Following granting RN-2, M-2 is now on a multihop connection with MN-3 via an owned RN-3 and a temporary granted RN-2. For reducing RN-6 overcapacity over c_2 , BS-3 asks for accessing RN-5 to handle a call between MN-7 and RN-9 via RN-5 on c_1 instead of RN-6. Further, RN-6 experiences a high call-block probability due to co-channel interference on c_1 , and cannot encompass a call request from MN-4 to MN-5. Therefore, a similar request between BS-3 and BS-2 if granted, MN-4 could reach MN-5 via RN-5 on c_1 successfully. Despite the low demand for carrier-III services in a far region dominated by carrier-I, a subscriber MN-0 is still interested in carrier carrier-III. For such a case, carrier-III sends out many requests for both carrier-I and carrier-II to fulfill such a demand. These carriers in return receive credit for such a request directly from carrier carrier-III. This carrier-III would benefit from MN-0 who seeks carrier-III for immediate communication with MN-8 for example via $MN-0 \rightsquigarrow RN-6 \rightsquigarrow BS-3 \rightsquigarrow$

RN-5 \rightsquigarrow BS-2 \rightsquigarrow RN-3 \rightsquigarrow RN-2 \rightsquigarrow RN-1 \rightsquigarrow MN-8.

The non-share of RNs in our CMCN simulation is a function modeled as $\gamma(t) = \lim_{\Delta(t)} \frac{1}{\Delta(t)} P(t \leq \tau \leq \tau + \Delta(t) \mid \tau \geq t)$ when $\Delta(t) \rightarrow 0$. This function is an instantaneous no-share rate of an RN conditioned on the fact that such an RN can be in a no-share until $t < \tau$ where τ is a frame duration. $\Gamma(t) = \int_0^t \gamma(u) du$ is a cumulative non shares until t time units has elapsed. For CMCNs, conflict management as in [10] is essential for a million mobile hosts per square-kilometer capacity and over a hundred Mbps average rate. This work proposes a benchmark for minimum broadcast delay and call-blocking conflict management.

The idle or no-share status of a BS is also needed if not able to synchronize an RN view with all SAS databases within sixty seconds. This BS can rejoin a corresponding CMCN over the next frames. For simplicity, our simulated SAS databases use PRIM instead of SINR as described in Section 5.3.2. Eventually, all databases converge to a similar conflict view. Each channel is accessed by RNs according to their weights or the number of active users over sixty seconds frame. The minimum delay schedule includes assigned channels and transmission powers and is sent back by a CMCN controller to all RNs via standard CBRS messaging protocol. This process requires a synchronization via Network Time Protocol (NTP) [10].

5.4 Methodology

This work is a conflict management simulation, which provides a minimum broadcast delay MBDF* schedule for CMCNs over CBRS. The simulator examines CMCNs over a set of six distinct broadcast policies and conflict parameters. The communication range, interference intensity, and other parameters such as the number of CBRS channels and shared RNs of each mobile carrier (nodes in MBDF⁻), capacity or weight of each RN, deployment area, IA and PAL distributions. **P1-P5** are centralized heuristics and minimum broadcast policies to minimize broadcast delay, RN usage times, and packet redundancy. Their performance is examined in Section 5.8 as a conflict avoidance with no cellular negotiation like [23] and [8]. The work in [75] and [91] uses Voronoi Tessellation for analyzing interference and channel reuse for multihop wireless networks.

Related work performs minimum independent sets of nodes and cell tessellation and coloring while assuming no simultaneous RN transmissions to simplify \mathcal{I}_2 conflict cases of Table 5.2. Here such an assumption is not used. Tessellation solutions suffer from idle slots [47], and such an issue needs synchronized stations. This explains having a high broadcast delay in [47]. For 400 stations in [47], an average broadcast delay of five seconds was obtained. Here, each of **P1-P6** obtains a schedule of an average of twenty milliseconds. **P1-P6** use SAS databases to back-off any conflict RN of \mathcal{I}_1 on licensed users, and avoid any \mathcal{I}_2 conflict case in MBDF⁻. **P6** in contrast manages conflict links with no tessellation or coloring and obtains a conflict-free schedule of a minimum schedule length or \bar{y}_* . **P6** reconnects conflicted and missed RNs over each optimization loop to achieve a minimum delay schedule MBDF*. **P6** manages conflict in MBDF⁻ over two directions

to minimize such a delay and maximize coverage.

5.5 CMCN System Architecture

This work is based on a similar data architecture as in [10] by asking each RN to send: (1) location and weight or number of active MNs every sixty seconds via a message of two bytes. (2) IDs of adjacent RNs using a message of four bytes for each neighbor to build conflict digraphs. There is an overhead of hundred bytes per RN over sixty seconds interval [10]. Each BS of a CMCN collects such information from corresponding RNs and each participant mobile carrier is responsible for uploading such data to private or SAS databases. The sharing of such information is with databases' providers, and such databases are provided by a trusted entity to local operators of CMCN. These databases assure the confidentiality of such information and are only shared with members of CMCN. The main CMCN operators like carrier-II in Figure 5.1 receive a complete CBRS view from SAS databases and compute a minimum broadcast schedule on a frame of a maximum duration of one minute. This is because (1) For a meaningful and consistent conflict view, CBRS enforces database synchronization within sixty seconds [106]. (2) The LTE-A connection dynamics occur within a similar time frame since an LTE-A radio remains connected for around twenty seconds after sending the last packet for the data plane setup overhead [45]. (3) Minimize channel switching overhead below a possible sixty seconds throughput [10].

For a BS of a CMCN, remaining in idle mode over CBRS is required if not able to synchronize an RN view with all SAS databases within sixty seconds. This BS could rejoin the same CMCN over the next frames instead [10]. Further, simulated SAS databases use the Interference Protocol Model (PrIM) described in Section 5.3.2 instead of Signal to Interference and Noise Ratio (SINR) in [10] for simplicity in detecting conflicts. Eventually, all databases converge to a similar conflict view and are guaranteed to lead for a particular schedule. This is ensured by sharing in advance a pseudo-random number generator. Each allocated channel can be accessed proportionally by a weighted RN serving a number of active users within a certain frame of sixty seconds. The computed minimum broadcast delay schedule, which includes assigned channels and transmission power according to distances among RNs afterward is propagated to all RNs using standard CBRS messaging protocol by the CMCN controller. This process requires synchronization within milliseconds, and so Network Time Protocol (NTP) is suitable [10].

5.6 Interference and Congestion Handling

LTE-A networks leverage channel reuse to mitigate conflict. To handle interference and congestion of neighboring cells, BSs, and RNs negotiate a channel for access until permitted by an occupant station. This scheme is called back-off conflict handling when a conflict source access request is denied. Existing systems like [10] adopt such a back-off. However, such a back-off mechanism introduces a delay burden via vacant or underutilized time slots. Further, if the network load is uniformly distributed, back-off networks cause a lack of fairness by prioritizing a node

over another. The literature to date lacks demonstration and analysis of the performance of conflict management protocols for multihop LTE-A networks over shared spectrum such as CBRS. The CBRS currently has no interference control among GAA users. The conflict studied in our work is between relay stations over a broadcast schedule, in which contention between RNs for a certain RN receiver is not permitted. This context could easily be mapped onto a special case where no contention from an RN is permitted to affect an MH assigned to another RN. However, we have analyzed $RN \rightsquigarrow RN$ contention in general. Here, we propose two versions of CMCN scheduling for a MBDF*. The version-1 is an intra-level policy, in which RN coverage is maximized over each level (scheduled RNs into a slot) individually without any coordination among levels. This version is solved by **P1-P5**. The version-2 scheduling is inter-level scheduling via **P6**, in which the number of RN-misses of MBDF⁻ is minimum in each conflict optimization step, and coordination among levels exists.

5.6.1 Intra Level Policy Driven Conflict Management

This section summarizes five policies for packet broadcast via CMCNs. Each Policy is examined over four metrics: coverage (hit), delay (schedule interval in slots of milliseconds), RN usage-times (maximum number of times an RN is used in a schedule), and packet redundancy (number of redundant broadcast packets in a schedule). Each policy obtains a minimum delay schedule MBDF* and has unique performance and back-off schemes.

P1 is a coverage-based and partial back-off policy. **P1** backs off an RN from accessing a channel during a level \bar{y} when causing a Co-Channel Interference (CoCI) or Congestion (CoCC) for coverage of a previously scheduled RNs during \bar{y} . For a MBDF* of length \bar{y}_* , an RN with \bar{c} accessible channels could send over a \bar{c} slots or levels of MBDF* with a likelihood $\binom{\bar{y}_*}{\bar{c}} p^{\bar{c}} (1-p)^{\bar{y}_*-\bar{c}}$ where p is the probability of sending in a level. **P1** cannot ensure a maximum level coverage like **P4-P6**. This policy schedules an RN when a non-backed-off coverage channel with no conflict to the same level scheduled RNs exists. The slot duration of each level is τ , in which an RN a is scheduled when having a non-empty coverage or $H_{\bar{y}(a)}^+(\mathbf{c}) \neq \emptyset$ over a set \mathbf{c} of CBRS channels. Hence, **P1** is a First-in-First-Transmit (FIFT) scheduling policy for depicting the worst delay scenario. This mimics a scenario, in which an RN with a smaller new coverage or $H_{\bar{y}(a)}^+(\mathbf{c})$ might relay a broadcast message before another one at the same level with larger new coverage.

P2 is a dual parameter-based policy. In addition to maximizing RN coverage over levels, **P2** considers for a specific RN an interference power I_p on a cardinality of prospective hit set of RNs, which is interfered by such an RN when accessing a certain CBRS channel during \bar{y} of scheduling. **P2** partially backs off any conflict channel of an RN candidate before being considered as a member of \bar{y} sending pool similarly as in **P1**. **P3** is a partial back-off of an RN from accessing a channel c_k when causing a conflict for a prospective coverage on c_k of a scheduled pool of RNs or duplicate message during \bar{y} . **P3** mimics an RN equipped with a Single Radio Access (SRA)

by preventing any RN from using c_k if any prospective hit of RNs of a scheduled RN over a channel set \mathcal{c} would experience a conflict. There is no slot ordering for channels as illustrated via dotted curves. Each channel in Figure 5.3a can be accessed arbitrarily during τ . The intersection between channels indicates our allowed simultaneous channel overlap during scheduling. The $H_{\bar{y}}^-(\mathcal{c}) = \sqcup H_{\bar{y}'}^-(\mathcal{c}) \ \forall \bar{y}' \leq \bar{y} - 1 \wedge \mathcal{c} \in \mathcal{c}$ is a union, which includes also each previous level's exclusive RN-hit or $H_{\bar{y}'}^+(\mathcal{c})$.

P3 backs off an RN b on c_k when causing CoCI, CoCC, or an Adjacent Channel Interference of Congestion (ACI or ACC) because of a power of c_k on an adjacent channel of \mathcal{c} of a scheduled RN a when: `any(ismember(H_a(c), F_b(c_k)))` is true for any $c_k \in \mathcal{c}$. Here, $H_{\bar{y}(a)}^+(\mathcal{c})$ as mentioned above represents a 's new cumulative coverage of RNs during \bar{y} on each channel in \mathcal{c} , and $F_{\bar{y}(b)}(c_k) = C_{\bar{y}(b)}(c_k) \sqcup I_{\bar{y}(b)}(c_k)$ is b 's overall RN conflict (congestion and interference) with $H_{\bar{y}(a)}^+(\mathcal{c})$ when accessing c_k . The $C_{\bar{y}(b)}(c_k)$ and $I_{\bar{y}(b)}(c_k)$ are congestion and interference, and each one can be a co-channel or adjacent channel effect caused when b accesses c_k during \bar{y} of duration τ . Figure 5.2 depicts how **P3** backs off an ACC and CoCC channel during \bar{y} . Each set is IDs of RNs. Here, c_1 and c_2 are adjacent channels. The single line arcs indicate CoCI or CoCC. The dashed arcs are exclusive for ACI or ACC. Each single-line arc indicates a back-off of its tail-RN's channel because of either a CoCC like $C_{\bar{y}(b)}(c_1)$ or an ACI such as $I_{\bar{y}(b)}(c_1)$. Each double-line arc is a non-back-off one. The left and right dots indicate sending over c_1 and c_2 can arbitrarily occur over τ . The ACC arcs exist when both channels overlap in time. Each policy mentioned above **sequentially** assigns RNs to raise a level's coverage which depends on previously scheduled RNs. This might not maximize \bar{y} 's coverage against a Maximal Coverage Clique (MCC) maximization, and consequently, a non-minimized delay is possible. **P4** and **P5** are level-based MCC without considering RN dependency among levels. Performing a Maximal Weight Coverage Clique (MWCC) for a level instead is useless without such a dependency.

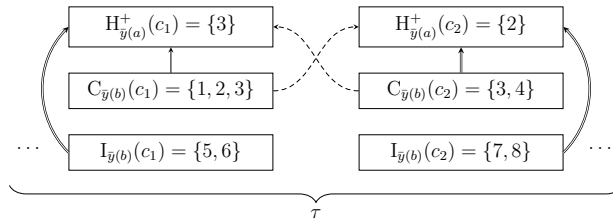


Figure 5.2: **P3** Backs off ACC and CoCC Channels during τ

The advanced spectrum access and efficient energy applies Ultra-Dense Heterogeneous Networks (UDHNs) as a principle block for 5G systems [6]. Future UDHN-based CMCN can allow multihop cellular services over CBRS as a Diverse Radio Access Technology (RAT). **P4** and **P5** are distinct OFDM-A versions of radio capabilities to mimic a CMCN efficient spectrum access. The

user in our OFDM-A context is a set of conflict-free RNs. **P4** schedules a pool of RNs while maximizing level coverage. Each level's slot τ is divided into \bar{c} mini-slots of length τ' so $\tau = \tau' \times \bar{c}$. For a τ' , RNs send only over a certain subset of channels $c_{\tau'} \sqsubset c$ when compared with **P5**. This is a valid slot architecture as each mini-slot τ' in **P4** and **P5** is considered as a newly introduced sub-level. **P4** can show idle slots for a constrained channel set $c_{\tau'}$ as described for Figure 5.3b.

Here, we considered each sub-level τ' to equal τ of any previous policy for a valid comparison among policies. Figure 5.3b is a level slot breakdown for **P4-P5**. **P4** as an OFDM optimizes $H_{\bar{y}}^-(c_k) = H_{\bar{y}}^-(c) \sqcup H_{\bar{y}}^+(c_{k'}) \forall k' \leq k$. For **P5** as an OFDMA, in contrast, $H_{\bar{y}}^-(c_k) = H_{\bar{y}}^-(c) \sqcup H_{\bar{y}}^+(c_{k'}) \forall k' \leq \bar{c}$. During τ' for channel c_k , each RN of $H_{\bar{y}}^-(c_k)$ is permitted to send once over any $c_{k'} \in c$. However, an RN can resend over $c_{k'}$ during the next τ' of \bar{y} to recover a prospective and dropped hit of RNs because of a conflict during a current mini-slot. The τ' in Figure 5.3b equals τ in Figure 5.3a as mentioned above.

The sending pool of each mini-slot c_k is initialized with previous levels unscheduled coverage $H_{\bar{y}}^-(c)$ over a set of channels c as each upper arrow shows. The c_k new coverage $H_{\bar{y}}^+(c_k)$ is copied for each $c_{k'}$ where $k' \geq k + 1$ mini-slot as indicated by lower arrows. **P5** is a more exhaustive spectrum approach than **P4** for mimicking UDHN in 5G systems. **P5** accesses CBRS with a high channel switching rate when RNs send over any $c_{k'}$ during a τ' for c_k . Each subset of RNs of $H_{\bar{y}}^-(c_k)$ sends via **P5** on any channel $c_{k'}$ simultaneously instead of a subset c' of channels with **P4**. Hence, **P5** can assure a maximum level coverage but not a minimum schedule delay or $\mathcal{L}(\text{MBDF}^*)$ due to a lack of among levels coordination. This local view can cause a dense and backed-off RN to experience a high delay and consequently, a non-minimized overall schedule length. **P5** by doing so allows \bar{y} covered RNs to send within \bar{y} 's next mini-slots instead of next levels. This causes **P5**'s optimization burden to decay over time as RNs are given chance at early mini-slots of τ , and before $H_{\bar{y}}^-(c_k)$ grows quickly, in contrast, to **P4**. The performance of **P4-P5** is based on a lexicographic channel order over τ' , in contrast, to any previous policy. However, other reordering schemes of c' may raise **P4** complexity. In general, **P4** and **P5** allow an RN to re-access a channel over many levels. This occurs when an RN of a scheduled clique of \bar{y} misses a covered RN because of conflict, and a resend for such an RN after \bar{y} is required. For **P1-P3** over a certain channel, each RN appears once in a MBDF^* . **P4** and **P5** aim for a minimum broadcast delay, but again RN dependency among levels is not considered as in **P6**. Further, except **P6**, a delay may not monotonically increase as interference intensity ω raises as indicated above. **Therefore, increasing ω inversely could result in a minimum delay schedule with P1-P5 or any existing level-based allocation scheme.**

5.6.2 Inter Level Policy Driven Conflict Management

Each aforementioned policy is a level-based coverage maximization without ensuring a monotonic increase of the minimum broadcast delay forwarding or $\mathcal{L}(\text{MBDF}^*)$ as a function of ω as discussed in Section 5.6.1. **P6**, in contrast, is an RN-miss minimization policy over two dimen-

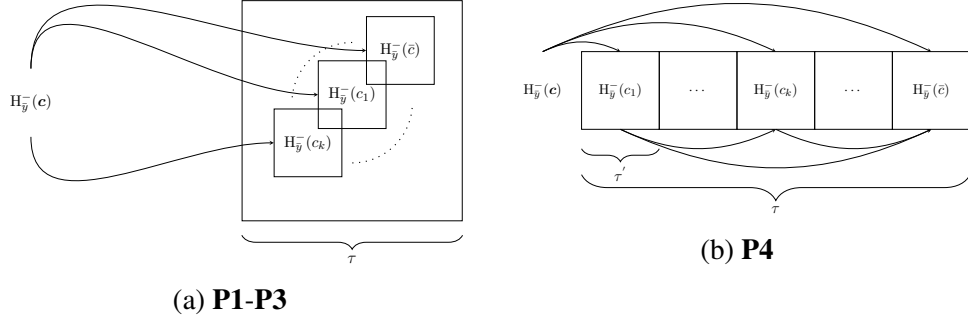


Figure 5.3: **P1-P5** Level \bar{y} Slot Architecture

sions, which is composed of intra-level coverage maximization (MBDF* width) and inter-level delay minimization (MBDF* height or \bar{y}_*). **P6** is proposed for producing a minimum broadcast delay and RN usage, and packet redundancy. **P6** provides more reliable and resilient scheduling for a CMCN over CBRs by maintaining a possible maximum number of conflict-free channels for each communication link. **P6** begins with an initial minimum delay broadcast schedule MBDF⁻. This MBDF⁻ is processed via a set of algorithms each with a certain aim. Each Conflict Component (CC) of MBDF⁻ as defined in Section 5.7.1 is solved also solved over two dimensions as described in Section 5.7.2. This ends our intra-level optimization over dimension-1. The enumerated solutions or vertices of all CC's afterward are mapped into a vertex dependency digraph or LVD⁻ as explained in Section 5.7.4. The LVD⁻ is handled for a minimum RN-miss via an inter-level or dimension-2 process before relinking RNs as indicated in Figure 5.4, which is a flowchart for obtaining MBDF* via **P6**. The discussion in Section 5.7 details **P6**.

5.7 P6 Conflict Management

P6 is an enhanced minimum broadcast schedule for CMCNs. Sections 5.7.1 and 5.7.2 provide a horizontal MBDF⁻ optimization. In Sections 5.7.3 to 5.7.5, a vertical delay minimization process is proposed. Table 5.3 summarizes our policies in terms of a sender RN choosing parameter and what channel is backed off.

5.7.1 Conflict Component

The conflict in the context of our work on a certain RN occurs when: (1) This RN experiences congestion or interference on a channel used by another RN; (2) There is a hosted MN by such an RN, and experiences similar conflict from another RN. Either case will cause packet collision for RNs. The interference and congestion component in a CMCN is a bipartite digraph composed of two subsets of RNs of \mathbb{R} residing in two adjacent levels of a corresponding MBDF⁻. For simplicity, such a component is called a conflict component. The level of a CC's conflict RNs indexes such a

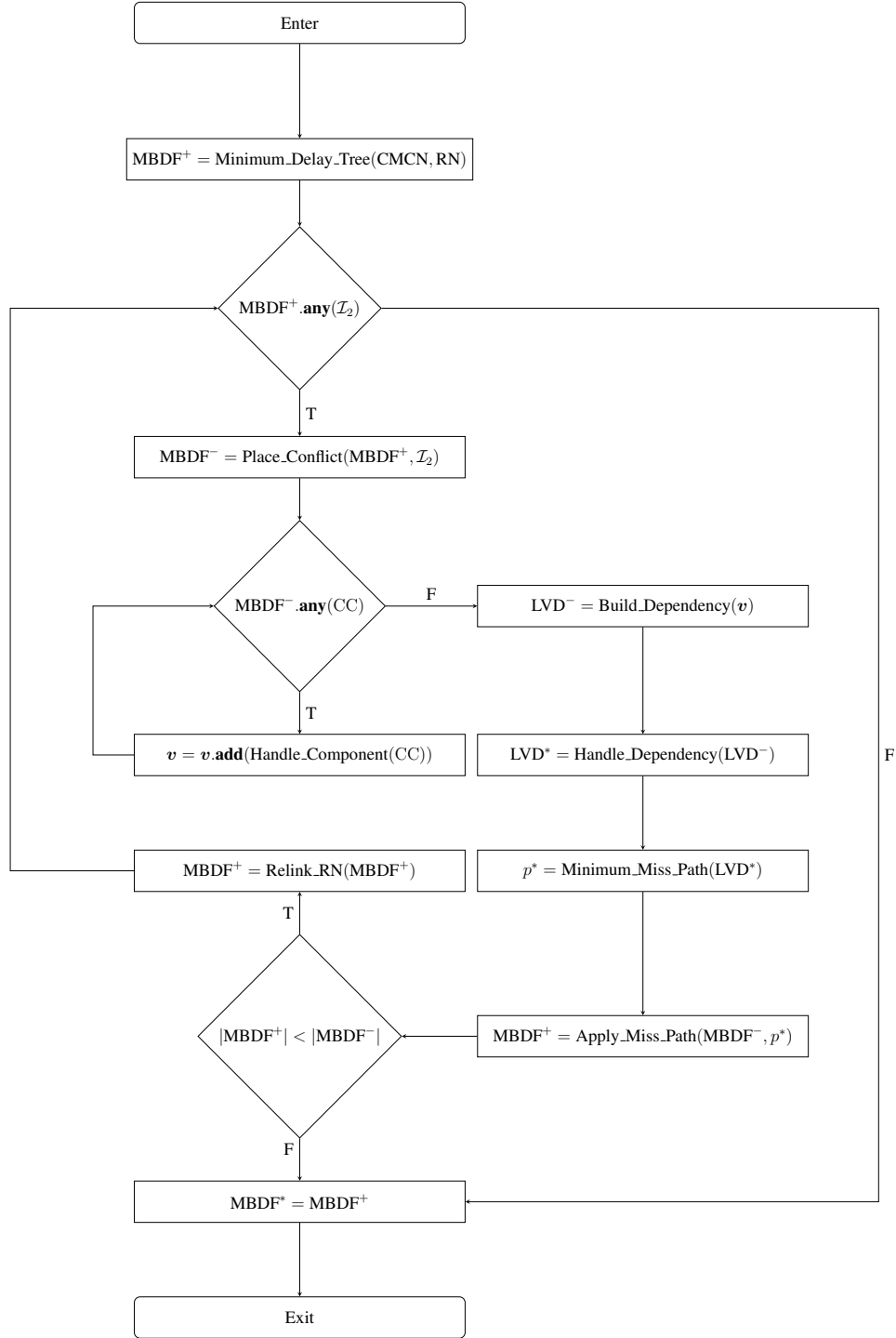


Figure 5.4: The Flowchart of **P6**

CC. Many conflict components may reside in a level. This causes a **non-polynomial** complexity in search for a minimum miss set of solutions in $MBDF^-$ as discussed in Section 5.7.6. Links of a CC are classified into three types of communications, congestion, and interference.

Policy	RN Scheduling Parameter	Back-off Channel
P1	Number of not backed-off channels $\neq 0$	Conflict
P2	Minimum number of interfered RNs	Conflict
P3	Maximum new sender coverage	(A/Co)-conflict
P4	Maximum new coverage via OFDM	Conflict
P5	Maximum new coverage via OFDM-A	Conflict
P6	Maximum overall coverage	Conflict

Table 5.3: Policy Scheduling Parameters

Figure 5.5 is conflict management in a collaborative CC of two mobile carriers carrier-I with black RNs and carrier-II with gray RNs. The communication arcs are in black, red links represent interference, and blue links show congestion. Edge labels represent channels. Figure 5.5a and Figure 5.5d have all valid solutions when **P6** lists each maximal clique. Figure 5.5e misses RN-5 when considering congestion as interference. Figure 5.5c is a valid maximal clique solution with no RN-miss when allowing RNs as CMCN and handing over RN-4 to RN-2. The two conflicts have been solved via tail-RN channel disabling, in which conflict channels are turned off from conflict sources. The solution in Figure 5.5b offers a more reliable communication link for RN-3 of two channels instead. Figure 5.5a is an example of a CC with all three types of links. The black arc is an actual communication link. Each red link is an interference case from a head-RN on a tail-RN. The congestion arc in blue is a second possible communication (handover) link for a head RN. The head RN receives a correct packet if and only if no red or blue arc is directed to such an RN.

Handing over calls in conventional LTEs occurs when a BS or an RN switches black with blue one-hop arcs to mobile users. Here, our work as an MCN considers handing over RNs with their mobile users to other RNs. The solving of a CC of three types of links is a one-dimensional and NP-hard optimization problem. There is no optimized model for wireless conflict miss with no simplification on such three types of links. Related work either manages congestion as interference or backs off nodes to reduce conflict. This model would result in a partially optimized delay for CMCNs because of: (1) Each backing off conflict management is a level-based (local) optimized coverage, in which a missed RN is relinked in the next level of MBDF⁻. (2) Handling congestion as interference is to simplify miss dependency among CCs. Therefore, precise channel scheduling for MCNs should distinguish solving congestion and interference to avoid any unnecessary RN-miss.

Handling conflict for the minimum delay is a complexity v.s. optimality trade-off. **P6** is designed as a baseline RN-miss minimization policy, in which each RN receives a broadcast packet within a minimum individual RN delay of a minimum delay schedule MBDF*. Each aforemen-

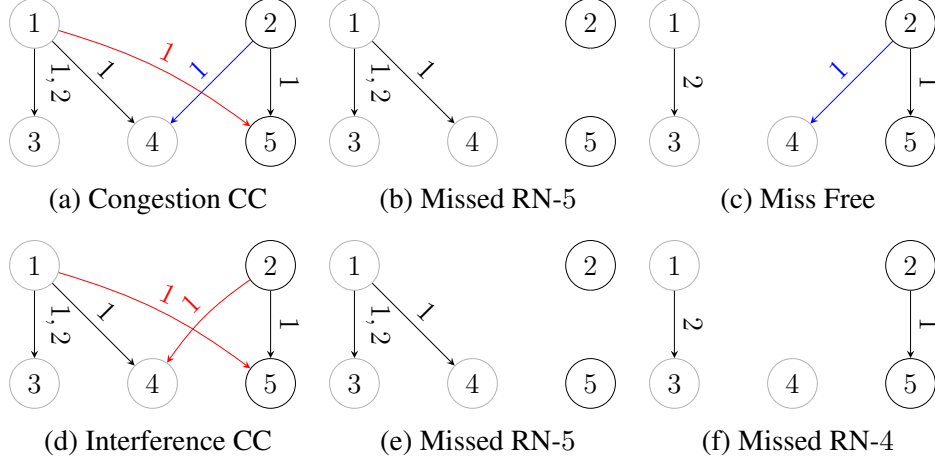


Figure 5.5: Conflict Management in a Collaborative CC of Two Mobile Carriers

tioned policy backs off RNs on a level basis for a local RN-miss minimization. In contrast, **P6** backs off RNs with coordination among levels (global) to minimize the overall RN miss. **P6** is consistent with CBRS regulations and databases as **P1-P5** for Tier-3 users. Therefore, MBDF* is achievable given the following conditions: (1) RNs support LTE-A's eNB functionality by terminating radio protocols of the E-UTRA radio interface on the access link as well as the S1 and X2 interfaces on the backhaul link [68]. (2) RNs can handle layers-1, 2, and 3 relay routing, Radio Resource Control (RRC), and Non-Access Stratum (NAS) protocols to connect as donor DeNB and donee EeNB. This study presents a conflict management framework with a minimum back-off over shared bands like CBRS that require rapid access while preventing any interruption for ITA or PAL users. For each CC in MBDF⁻, our clique solver returns a list \mathbf{v} of solutions where each $v \in \mathbf{v}$ is a set of conflict-free RNs to be scheduled within a CC's particular level or τ of a MBDF*. Each v or a CC solution is characterized by the weight and a set of conflict-free links, for which an extra inter-level miss minimization process is performed.

Each parent-RNs in a CC is not affected while solving conflict links and is excluded from component solutions. Meanwhile, each conflict link is solvable via tail or head-RN channels. Handling solving of Figure 5.5a conflict links gives a set of solutions or maximal cliques. The solution in Figure 5.5c is done by solving RN-1 \rightsquigarrow RN-5 via RN-1 (tail) disabling channel-1 of each link tailed by RN-1. In contrast, a head-RN solving causes one communication child as RN-4 in Figure 5.5b to turn off channel-1. Incorrect **congestion** management for Figure 5.5d causes miss solutions like 5.5e and 5.5f. These solutions can lead to suboptimal MBDF* as in Fig. 5.7. This is a dependency mapping, in which link (10, 15) is solved as interference in Figure 5.7a, and congestion in Figure 5.7b. LVD⁻ is a mapping from RN to vertex dependency with minimum miss paths highlighted in red. Thus, managing congestion or handover links as communication ones

can avoid RN-miss as in Figure 5.5c. These saved RNs may belong to carriers with delay-sensitive services, for which such management adds benefit.

However, such management could introduce more complexity by increasing virtual copies of conflicted RNs to be handled as discussed in Section 5.7.2. Each RN a of MBDF^- is assigned an overall weight of $w(a) = w^*(a) + w^+(a)$, and consequently, each MBDF^- solution is weighted. Here $w^*(a)$ is a weight of RNs, and contains all next levels RNs connected with a in MBDF^- . The $w^+(a)$ corresponds to the number of MNs served via such an RN. Therefore, a weight of a solution v is as follows: $w(v) = \sum_{a \in v} w(a)$. Each interference arc with a subset of channels c' of c and with size \bar{c}' has $2^{\bar{c}'}$ possible ways of being solved. Therefore, a component of $\mathcal{I}' \subseteq \mathcal{I}_2$ interference arcs has $|\mathcal{I}'| \times 2^{\bar{c}'}$ possible solutions. Thus, enumerating all CC solutions is an NP-hard problem as explained in Section 5.7.2. This exponential number needs conflict relaxation methods such as contention-based band sensing combined with conflict avoidance like back-off schemes. This relaxation reduces interference intensity and approximates a minimum length schedule.

Despite such complexity, **P6** when minimizing broadcast delay for CMCNs can be considered by considering congestion and interference conflict links solve for two copies of MBDF^* . For MBDF^* , an individual RN a might experience a higher delay $\bar{y}(a)$ in either copy. Misrepresenting conflict occurs when a communication link is falsely endorsed as a congestion link. Further, a solution might be canceled by an upper-level conflict component of a MBDF^- . Hence, any CC with a congestion link should be promoted by a set of solutions in which original communication and congestion links exist. Following our discussion over each candidate MBDF^- , a minimum delay schedule of a set MBDF^- by distinguishing congestion and interference is equivalent to achieving: $\text{MBDF}^* = \arg \min \mathcal{L}(\text{MBDF}^-) \forall \text{MBDF}^- \in \text{MBDF}^-$ while assuring a minimum $\sum \bar{y}(a) \forall a \in \text{MBDF}^-$. Here, $\mathcal{L}(\text{MBDF}^-)$ is a schedule length. In contrast, solving congestion as interference is constrained by $\bar{y}(a) \leq \mathcal{L}(\text{MBDF}^-)$ instead. Figure 5.7 of Section 5.7.3 compares both scenarios over a certain CMCN size.

5.7.2 Conflict Free RN

To calculate a solution v of a conflict-free set of RNs using a tail-head RN solving is not a practical procedure. Hence, a CC is mapped into a new channel-based conflict digraph called Conflict Free Component RNs (CFCR) for each component conflicted RNs by a set of conflict mapping rules. Figure 5.6 is two valid CFCR representations for Figure 5.5a. Figure 5.6a merges RNs to reduce the number of edges for clique listing. Figure 5.6b merges no RNs. Each node ID is an RN in Figure 5.5a. The weight of conflict node 4, 5 is as follows: $w(4, 5) = w(4) + w(5)$. Edge labels represent conflict-free channels, on which tail-RN can send without causing a conflict for a head-RN. These rules are: (1) For each RN a in a CC on a certain channel c , we define a set of conflict sources $\phi(a)$ from upper-level RNs. This set contains (I) a 's communication parent when acting as **conflict** source on other RNs in CC or having a or a sibling of a under congestion over c

and (II) conflict parents of a . Each RN whose communication parent is not a conflict source does not experience conflict, but has a sibling who is under congestion over c as mentioned in (1) is considered as a conflict RN. Thus, such an RN is not a conflict node if no sibling under congestion exists. (2) From an overall list ϕ of each corresponding $\phi(a)$, for each c , RNs with an identical **communication** parent and a union of (I) and (II) are merged into one conflict node in CFCR. Because of our distinct congestion management a size $|\text{CFCR}|$ can exceed the actual number of conflicting RNs in a CC as each congestion link causes a copy of its head RN. In contrast, if all conflict links are interference ones, $|\text{CFCR}|$ is **always** upper bounded by such an actual number. This work considers neither expressing an upper bound for such a size nor merging RNs of distinct communication RN-parents. Figure 5.6a is a CFCR instance with merged RNs of Figure 5.5a. Figure 5.6b is a valid CFCR without merging with more arcs than in 5.6a. The node labeled as 4, 5 is a merging scenario of a congestion copy of RN-4 and RN-5 over channel-1. The two adjacent conflict nodes represent RN-3 and RN-4. The arc's tail-RN can send over a certain conflict-free channel (arc label) while the head-RN is using all labels of egress arcs. The weight of a conflict node equals a summation of the weights of all contained RNs. This merging aims to minimize $|\text{CFCR}|$, and consequently, reduce computation complexity when listing all solutions of a conflict component. Therefore, discarding channel reliability (maximum number of available channels for an RN) in advance via a **risk-based conflict channel relaxation until an RN-miss is inevitable** would reduce the size of CFCR or the number of conflict nodes.

P6 is designed as a baseline scheduling policy while a risk-based relaxation version of **P6** is beyond the scope of our work. Furthermore, our RN relinking algorithm could experience an increased linking burden, and consequently, $\mathcal{L}(\text{MBDF}^-)$ would deviate from the minimum. In general, for a CC located in a level \bar{y} of a MBDF^- with a mean degree d , a maximum number of conflict nodes or $|\text{CFCR}| \leq \bar{c} \times d^{\bar{y}-1}$ where \bar{c} is number of available CBRS channels. Each CC solution is a maximal clique in CFCR, and solving a CC in our work means enumerating all maximal cliques in CFCR. For any conflicted pair of RNs, arcs can be asymmetric in the label (channel) which maps any CFCR into a conflict graph. Non-adjacent conflict nodes do not form a maximal clique. This conflict management is novel in our work, and in particular, requires any congestion link to have a copy of all congested head-RNs like 4 in Figure 5.5a. This is a worst-case scenario, and by avoiding such a copying, a CFCR can be an invalid mapping of a CC. Further, for a MBDF^- , solving conflict links carefully results not only in a minimum $\mathcal{L}(\text{MBDF}^*)$ but also a more channeled (reliable) links for a CMCN schedule. This is because congestion links can allow RNs in a minimum MBDF^* to receive from many parent-RNs as opposed to single-parent tree-based scheduling algorithms.

The Listing Maximal Clique Problem (MCLP) is an NP-hard problem [62], [53] and [86]. Many non-polynomial algorithms exist for handling MCLP such as Bron–Kerbosch

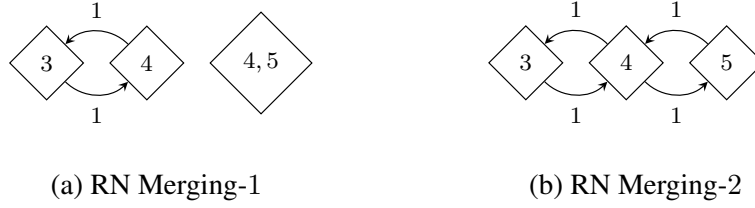


Figure 5.6: Two Valid CFCR Representations for Figure 5.5a

in [18], [102], [31], [48] and [97]. The Bron–Kerbosch algorithm is recursive and does not require any graph partitioning like [31] and [15]. The GPU-based Bron-Kerbosch algorithm in [104] solves MCLP in parallel by optimizing a decomposition process and computing resource usage while speeding up via coalesced memory accesses. This GPU-based algorithm warps reductions to increase bandwidth and reduce memory access. This work proposes a new iterative minimum path-based algorithm for MCLP in MBDF^- . For a pivot RN v^* with a maximum density and a set of neighbors Γ_{v^*} , our clique listing considers a clique path $p := v^*, \dots, \Gamma_{v^*}(i)$ as a current maximal clique candidate and a subset $\mathbf{a} \subseteq \Gamma_{v^*}$ of exploration nodes for extending p . Each $a_k \in \mathbf{a}$ is already adjacent to all nodes in p . The maximal clique path occurs when adding $\Gamma_{v^*}(i)$ causes $\mathbf{a} = \emptyset$. For **P4** and **P5**, a clique node is an RN in MBDF^- and vertex of RNs for **P6**. The clique solving consists of two subroutines, a pivot selector, and a maximal clique path builder. The Pivot selector chooses a path source node with a possible maximum and yet not discovered maximal clique paths for each call of the clique path subroutine. The clique listing afterward updates a maximal clique list \mathbf{q} . Further, each candidate for a clique path is propagated in advance via \mathbf{q} for each next node in \mathbf{a} to eliminate duplicated cliques when the next node is used for extending a clique. The worst-case scenario requires storing any clique path candidate for each corresponding index $a_k \in \mathbf{a}$ of \mathbf{q} . However, minimizing such a memory overuse is possible via a smaller subset $\mathbf{a}' = \{a' \mid \forall A(a', \mathbf{a} \setminus a') = 1 \wedge a' > \Gamma_{v^*}(i)\}$, from which a smaller set acts for indexing \mathbf{q} 's memory. Here, A is CFCR's adjacency matrix. This clique solver is not a recursive like [18] but a cycle-based maximal clique path search for enumerating all maximal cliques that include a certain pivot node v^* . The pivot scheme defined is distinct from all previous metrics used in related studies like $v^* = \arg \max |\mathbf{P} \cap \Gamma_v|$ in [97] where \mathbf{P} is a clique extension candidate set of nodes. The second layer of node degeneracy order is not required in our clique listing. Instead, we defined a new metric called node density, by which we determine for each node a $\rho(v) = \sum \mathbf{a}(v \sqcup \Gamma_v)$, which is equal to the number of edges in an induced subgraph via a set $v \sqcup \Gamma(v)$ of nodes. Further discussion on our clique listing is beyond our presented work.

Each conflict component in MFBD^- is mapped onto a conflict-free digraph as described for enumerating maximal cliques (solutions). Finding solutions instead of only a minimum RN-miss

one for each CC is essential for our **vertical** miss minimization among all levels of MBDF^- . This process can cause an RN-miss of an upper-level CC to be a predecessor of a lower-level RN-hit of another CC's miss clique. Hence, an overall minimization should not rely on a single minimum miss solution but needs a complete list of all solutions for each CC. The number of solutions for a conflict component depends on the conflict range and intensity (α, ω) and the schedule number of RNs $|\text{MBDF}^-|$, which all control a component conflict graph. This number of solutions has a relaxed upper bound as discussed in Section 5.7.1, and for a CMCN with a coverage aim instead of channel reliability, $|\text{CFCR}|$ sizes of $|\text{MBDF}^-|$ can be significantly reduced via conflict relaxation as indicated above. Performing such a reduction comes by applying a risk-based conflict channel cancellation. However, when a CC has a zero miss solution, such a CC is excluded from consideration in our secondary RN-miss minimization after applying one zero miss clique in MBDF^- . For a component with multiple zero miss solutions, **P6** chooses a solution with maximum channel reliability to be applied in MBDF^- .

5.7.3 Types of Vertex Dependency

Table 5.4 summarizes dependency links between a pair (v, v') of vertices of MBDF^- . For instance, P/F(HH)-NMM indicates either partial or full of v 's remaining coverage or hit predecessors or $\neg H(v)$ are in $H(v')$ of a LVD^- linking. The NMM means such a (v, v') link has no RN-miss dependency. The $\Delta(e) = \bar{y}(v) - \bar{y}(v')$ refers to the distance in levels between a head-tail and tail-vertex of a dependency arc. LVD^- is constructed in a reverse order with respect to MBDF^- by having \bar{y}_* and \bar{y}_0 solutions as root and leaf vertices in LVD^- respectively. Each dependency link carries a miss of RNs dependency between v and v' . Traversing LVD^- via a miss dependency path $v \rightarrow v'$ with a miss dependency arc (v, v') and only over $\Delta(e) = 1$ links requires certain vertices to be copied for solving (v, v') dependency on such a path between v and v' . Figure 5.8 shows when Dijkstra's algorithm fails when an RN-miss dependency exists. This is a subgraph of a whole LVD^- . Having RN-miss sets for each vertex like $M(v_1) = \{11, 10\}$, $M(v_3) = \{8, 9, 14\}$, $M(v_4) = \{12, 6\}$, $M(v_5) = \{7\}$, $M(v_7) = \{8, 9, 13, 14\}$ would cause Dijkstra's algorithm to output incorrect minimum miss path highlighted in gray instead of a correct path highlighted in green. The numbers in each set are RN IDs. The path from $v_1 \rightarrow v_5$ in Figure 5.8b has a either a miss cost of: $|M(v_1)| + |M(v_4) \setminus M(v_1)| + |M(v_5) \setminus M(v_4)| = 2 + 2 + 1 = 5$ via v_4 or $|M(v_1)| + |M(v_3) \setminus M(v_1)| + |M(v_5) \setminus M(v_3)| = 2 + 3 + 1 = 6$ missed RNs via v_3 . Thus, Dijkstra's algorithm would set v_5 miss cost as five missed RNs. To extend such a path to v_7 and without backtracking, Dijkstra's algorithm would calculate a false miss cost of $5 + |M(v_7) \setminus \{M(v_3) \sqcup M(v_5)\}| = 6$ over $p := v_1, v_4, v_5, v_7$. In contrast, since v_3 is not traversed when backtracking p from v_5 , v_3 must not be included in the arc $a(v_5, v_7)$ cost and as a result, an actual miss cost of $5 + |M(v_7) \setminus M(v_5)| = 9$ is now calculated. However, v_3 is not avoided when using the second path to v_5 via v_3 , and a minimum RN-miss of $6 + |M(v_7) \setminus \{M(v_3) \sqcup M(v_5)\}| = 7$

is not achievable by minimum cost path algorithms without vertex copying as in Figure 5.8c.

Following such copying, an indicator for $M(v)$ is placed on each copied ingress arc to v' . In Figure 5.8c, v_3 label is placed on all end arcs of paths from $v_3 \rightarrow v_7$. Each dependency link with $\Delta(e) > 1$ needs to be solved similarly before searching for a minimum miss path using conventional minimum cost path algorithms over links of $\Delta(e) = 1$. The link with ID-5 is a dummy dependency link introduced for connecting a fully isolated CC(s) with all adjacent levels within $\Delta(e) = 1$. For connecting CC(s) located on one level, a similar dumb arc is used. In either case, no actual miss dependency exists, but for ensuring any minimum RN-miss path contains exactly one vertex of each component. From Table 5.4, ID-6 dependency link is not necessarily solved and has been excluded from LVD^- because each CC is represented by a minimum of two solutions. Hence, for a CC, a vertex v whose $\neg H(v)$ is a proper subset of $M(v')$ while having at least one predecessor of a node in $\neg M(v)$ that is not in $M(v')$, there must exist at least one v peer solution v^* , in which partial or full $\{\neg M(v) \setminus M(v')\}$ is in $\neg H(v^*)$. The ID-2 and ID-4 arcs remain in LVD^- , and are solved by vertex copying since indirect miss dependency exists in P/F(MM) and FMM of each one respectively. Having arcs with $\Delta(e) = 1$ like ID-1, ID-3, and ID-5 carries no indirect miss dependency but is added for considering all possible vertex permutations while exploring a minimum RN-miss path.

Key	Dependency link e	$\Delta(e)$	ID	Representation
N: No	P/F(HH)-NMM	$= 1$	1	$\{\neg H(v) \sqcap H(v')\} \neq \emptyset \wedge \{\neg M(v) \sqcap M(v')\} = \emptyset$
P: Partial	N/P/F(HH)-P/F(MM)	≥ 1	2	$\{\neg M(v) \sqcap M(v')\} \neq \emptyset$
F: Full	NHH-NMM-N/P(HM)	$= 1$	3	$\{\neg H(v) \sqcap H(v')\} = \emptyset \wedge \{\neg H(v) \setminus M(v')\} \neq \emptyset \wedge \{\neg M(v) \sqcap M(v')\} = \emptyset$
H: Hit	FHM-FMM	≥ 1	4	$\{\neg H(v) \sqcup \neg M(v)\} \sqsubset M(v')$
M: Miss	NH(H/M)-NM(H/M)	$= 1$	5	$\{\neg H(v) \sqcup \neg M(v)\} \sqcup \{\neg H(v') \sqcup \neg M(v')\} = \emptyset$
\neg : Remaining	FHM-(N/P)MM	≥ 1	6	$\{\neg H(v) \sqsubset M(v')\} \wedge \{\neg M(v) \setminus M(v')\} \neq \emptyset$

Table 5.4: Types of Vertex Dependency

For a pair (v, v') , placing ID-4 arc between either adjacent or non-adjacent levels occurs when a destructive relationship exists. This means $M(v')$ is a predecessor(s) of $\neg H(v) \sqcup \neg M(v)$. Hence, applying a head-vertex v' of a destructive arc for solving a certain CC entails solving by deletion of any successor component with a destructive dependency with v' such as v . This means a head solution v' is enough for solving two end components for such an arc. Finally, ID-5 arc connects an independent or isolated component in \bar{y} with another one in $\bar{y} - 1$ (a **next** adjacent level) such that $\Delta(e) \geq 1$ in $MBDF^-$. Each independent component has no H/M relationship with any other component at all levels. In contrast, for an isolated component with adjacent levels, placing such $\Delta(e) = 1$ arc does not imply the absence of indirect P/F(MM) with upper levels and does not change LVD dependency. Further, having independent conflict components results

in a disconnected LVD^- . ID-5 is for a virtual linking all LVD^- 's components to avoid any exponential complexity for minimum miss path searching when dealing with a disconnected LVD^- as described in Section 5.7.6. The dependency ID-5 could also show up for components in the same level of $MBDF^-$, among which no H/M dependency exists. The independent component with a set of v solutions is computationally useful in $MBDF^-$ since only a minimum RN-miss vertex (v_*) needs to be included in LVD^- .

Figures 5.7a and 5.7b represent two distinct sketches of a $MBDF^-$ for an exact broadcast demand. The two plots are identical except on link RN-10 \rightsquigarrow RN-15, because of which we show two distinct $MBDF^*$ schedules after managing conflict links. The two plots include three types of links connecting twenty-three RNs. Each communication link in black, interference arc in red, and congestion arc in blue uses a CBRC channel during a particular level or slot. For simplicity, all links in $MBDF^-$ operate on one channel. Tracing backwards each $MBDF^-$ creates a LVD^- via certain H/M dependency links of Table 5.4. Figures 5.7c and 5.7d illustrate each corresponding vertex dependency LVD^- of Figures 5.7a and 5.7b respectively. Handling of dependency links of Figures 5.7c and 5.7d is in both LVD^* plots. Each $MBDF^-$ contains four components each with a subset of solutions (colored in $MBDF^-$) or vertices in LVD^- . For instance v_1 and v_2 of Figure 5.7c represent solutions of a lower-left component of Figure 5.7a. In a partitioned and dense MCN, such a component could span over many cells, in which RNs of each level of a $MBDF^*$ can only access a single and synchronized one-millisecond slot. Each vertex v solves a corresponding conflict component and is characterized by six distinct sets of RNs: Remained conflict and conflict-merge, conflict-safe, conflict-free, miss, and miss-merge RNs.

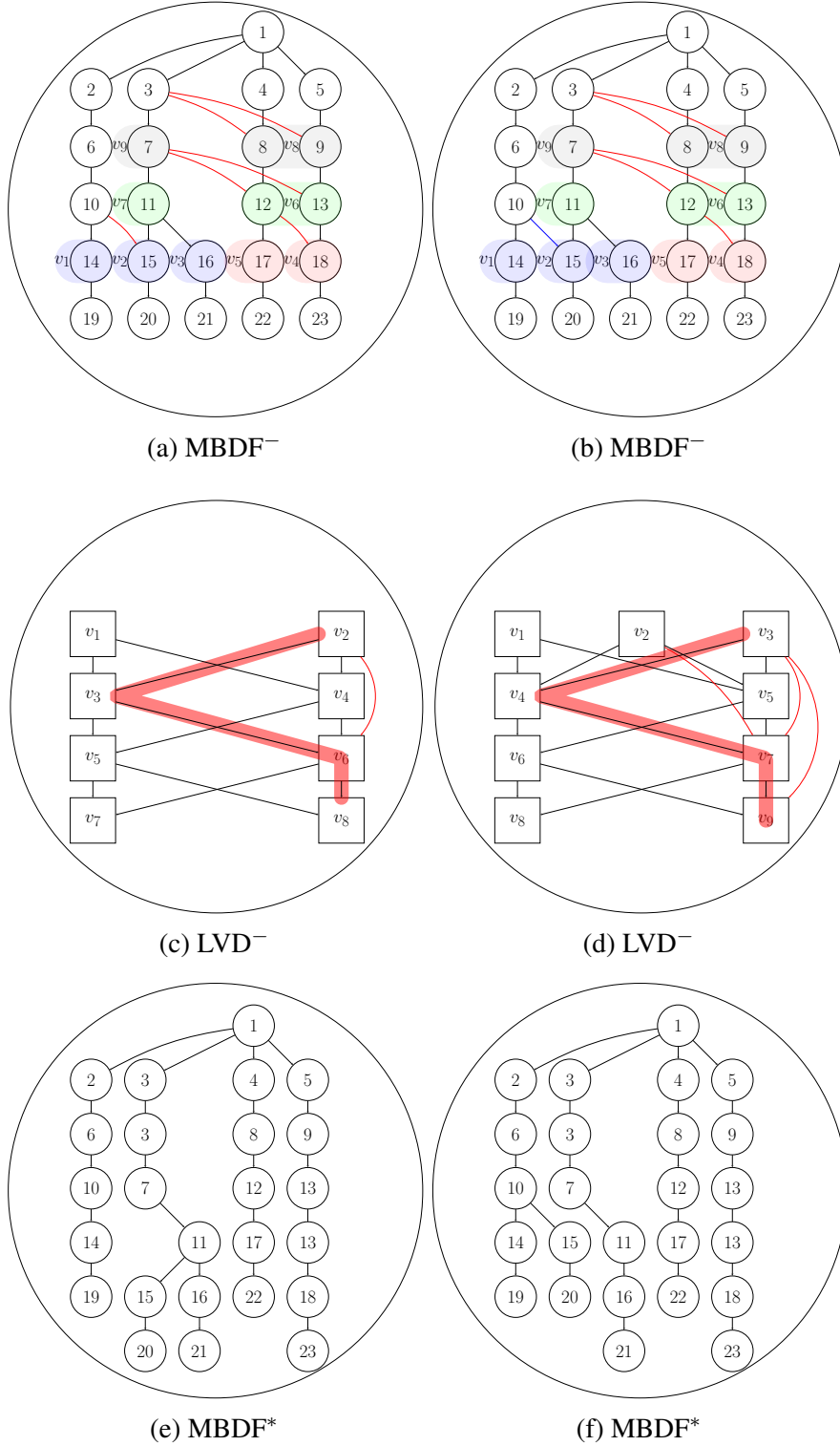


Figure 5.7: Dependency Mapping and Handling

The naïve solution for MBDF⁻ is obtained by forming a Level RN Dependency graph LRD⁻

at RN but not vertex-granularity, and searching a maximum mass solution, which is equivalent for a minimum RN-miss clique. However, with state-of-the-art approximation algorithms, for a dense CMCN, obtaining such a solution is computationally expensive since each Maximum Clique Problem (MCP), Maximum Mass Clique Problem (MMCP), and Maximal Clique List (MCL) is confirmed of being NP-hard [102]. Recursive backtracking schemes like [48] and [83] have eliminated redundant recursive calls of $O(2^{|V|/3})$. Therefore, searching over a vertex-granularity in LVD^* as opposed to RN-granularity in $MBDF^-$ has less burden when projecting onto a Minimum Miss Path (MMP). Note, over a vertex granularity either MCP or MMCP is inapplicable because of vertex dependency. Listing maximal cliques via current algorithms such as [18] is however applicable within a non-polynomial complexity. From [62], every $|V|$ -vertex graph has at most $3^{|V|}$ maximal cliques [62] and [97]. Tomita algorithm is designed based on [18] and reported effective in practice over approaches using [31]. However, [18] and similar schemes pose real challenges when solving a non-sparse graph $G(V, E)$ of $|E| \lesssim \frac{1}{2}(|V|^2 - |V|)$ in our evaluations. Lower bounds for challenging MCP algorithms of forty nodes have been established in [19]. This work, in contrast, proposes a novel transformation paradigm for any **broadcast** schedule candidate with a conflict and analysis at a vertex-granularity as opposed to RN-granularity in order to avoid computationally expensive approaches. More expensive approaches use MCL algorithms for vertex-dependent systems as detailed in Section 5.7.6. Two distinct transformations \mathcal{L}_1 and \mathcal{L}_2 are performed on $MBDF^-$ before having an MCP or MMCP at RN-granularity transformed into an MMP instance at vertex-granularity. First, transforming an instance F of $MBDF^-$ to an instance \hat{F} of LVD^- has been performed via a transformation function $\mathcal{L}_1 : F \Rightarrow \hat{F}$ by adding dependency arcs of Table 5.4 via a dependency algorithm. This mapping function \mathcal{L}_1 has a complexity of $O(|V|^2)$. The second transformation $\mathcal{L}_2 : \hat{F} \Rightarrow \hat{\hat{F}}$ where $\hat{\hat{F}}$ is a MMP instance of $MBDF^*$ with all vertex dependency arcs solved as indicated in Section 5.7.5.

5.7.4 Vertex Dependency Digraph

Each solution as described in Section 5.7.2 pertains to a certain local conflict component. These solutions regardless of RN-miss status need to be connected via a secondary **vertical** minimization process that selects a single solution for each remaining component. Linking solutions via such a process minimizes the overall RN-miss of $MBDF^-$ of each relinking loop in Figure 5.4. This provides a minimum schedule length $\mathcal{L}(MBDF^*)$ of size $|MBDF^*|$ of RNs. Performing a brute-force search over $MBDF^-$ is a tedious enumeration of all minimum RN-miss sets of solutions as discussed in Section 5.7.6. Further, such an exhaustive paradigm is impractical for massive CMCNs when neglecting RN-miss dependency among vertices. For adjacent levels, considering fully cascaded and connected solutions of a $MBDF^-$ produces an acyclic dependency digraph called LVD^- . Henceforward, each vertex of LVD^- under discussion is a conflict component solution. This would transform a brute-force of all combinations of solutions into a minimum RN-miss path discovery

problem. Even such a scheme would fail to produce a minimum RN-miss path when dependency exists as indicated in Section 5.7.6. Therefore, we introduce a new vertex linking paradigm for transforming our problem from being exponential in enumerating and consequently incorrect minimum RN-miss path searching to an exact minimum RN-miss path minimization problem. This mapping is performed by including all essential dependency arcs among vertices in LVD^- . This paradigm allows for solving a vertex-dependent minimum RN-miss path, in which exploring a new vertex depends on previously discovered ones. In other words, a dynamic arc cost because of vertex dependency while traversing is solvable via conventional minimum cost path algorithms after handling LVD^- as detailed in Sections 5.7.4 and 5.7.5.

For a simplified complete dependency graph $\kappa(V, E)$ of a $MBDF^-$, an unsolved dependency LVD^- of size $|V|$ is a subgraph of a $\kappa(V, E)$ while considering link ID-6 of Table 5.4 as unnecessary in $MBDF^-$. For a CC, an initial vertex reduction is performed by applying any zero RN-miss vertex without a congestion link for reducing $|LVD^-|$ before solving all remaining dependency arcs for a LVD^* . Figures 5.7c and 5.7d include only essential dependency links for each $MBDF^-$ in Figures 5.7a and 5.7b. Each dependency arc in LVD^- is an RN H/M relationship between vertices as summarized in Table 5.4 of four distinct H/M RN dependencies. For a $MBDF^-$, a level may host many conflict components as mentioned in Section 5.7.3. These components have no H/M correspondence. The connection of their solutions is performed via a virtual level augmentation dependency link so a level in LVD^- **must** host solutions for one component. Note, a level in LVD^- does not imply τ in $MBDF^-$. This process is very important and performed because: (1) The components in one level are independent, and consequently, placing each one in a private dependency level would not modify $MBDF^-$ RN dependency. (2) Minimum RN-miss path contains one vertex of each component. For conflict components in one \bar{y} of $MBDF^-$, any $\Delta(e) = 1$ dependency arc remains as $\Delta(e) = 1$ arc for one end component of \bar{y} , and becomes a $\Delta(e) > 1$ arc for other components. Further, an original $\Delta(e) > 1$ dependency arc would not change. This level augmentation reduces search complexity for a minimum RN-miss path by: $O(|v_{\bar{y}_0}| |v_{\bar{y}^*}| |V|^2)$ where $|v_{\bar{y}_0}|$ and $|v_{\bar{y}^*}|$ are the number of solutions of the end components in LVD^- respectively. However, solving without such a process needs another clique-based RN-miss minimization among paths that do not consider all conflict components.

5.7.5 Dependency Link Handling

Enumerating $|V|$ -maximal cliques or solutions (vertices in LVD^-) for each conflict component ends each level or **horizontal** solving step for Figures 5.7a and 5.7b. Each component is now replaced by a set of solutions for non-conflicting RNs in CMCN. Placing RN-miss dependency links between vertices of LVD^- as discussed in Section 5.7.5 is necessary to consider such an RN-miss among levels via **vertical** solving step before searching for a minimum RN-miss path composed of exactly one solution for each component. Here, we propose a vertex copying scheme

for handling each dependency arc in LVD^- . The dependency handling algorithm solves each dependency arc once using a minimum vertex copying in order for minimum cost path algorithms to be implemented correctly. Each dependency arc such as (v_2, v_6) of Figure 5.7c needs solving via vertex copying as in Figure 5.8c. Each of v_3 and v_4 are copied for only paths from v_2 , and a corresponding dependency search matrix α is updated so that $\alpha(v_4, v_6) = \{v_2, v_4\}$. This enforces when going from v_4 to v_6 a minimum RN-miss path search algorithm to exclude any missed RN of v_6 when included in either v_2 or v_4 missed RNs. This allows Dijkstra's algorithm for instance to list all minimum RN-miss paths from such a vertex to all lower-level ones correctly. Each indirect dependency arc $e = (v, v')$ with a $\Delta(e) > 1$ causes copying a set v of vertices whose $\bar{y}(v') < \bar{y}(v) < \bar{y}(v)$ so that no other vertex at $\bar{y}(v)$ except v could reach such v while searching for a minimum miss path. Before solving each dependency arc whose $\Delta(e) > 1$, our dependency handling algorithm initializes each dependency arc $\alpha(v, v') = v \ \forall (v, v')$ in LVD^- . Because of vertex copying, such an ID adjacency matrix α can exceed LVD^- original size. Hence, a longer search time for a minimum RN-miss path is possible over LVD^* with any polynomial algorithm like Dijkstra's and Bellman–Ford. However, such a size increase concern can be addressed by two approaches: (1) For each conflict component, certain solutions or vertices are used in LVD^- . (2) For two vertices v_1 and v_2 of one level and identical adjacency such that for all (v_1, v') and (v_2, v'') dependency links with $\Delta(e) = 1$, $v' = v''$, we handle only (v_1, v') for any $\Delta(e) > 1$ pair of dependency links (v_1, v') and (v_2, v') . The minimum RN-miss path search when exploring from v_1 immediately will explore from v_2 using (v_1, v') handled sub-adjacency. The Pseudocode in Appendix ?? summarizes our RN-miss dependency handling.

5.7.6 Dependency Path Finding

RN-miss dependency among vertices causes a variable weight of any dependant vertex v as in Section 5.7.1 and based on what next set v is considered by v for an RN-miss path. Hence, MCP algorithms can not report a correct v for a minimum RN-miss of all levels in LVD^- . Clearly, a solution for an upper-level component vertex might solve a lower-level one by including all lower-level conflict RNs when an ID-4 dependency exists between such solutions. Thus, a minimum RN-miss v can be of a size less than the number of conflict components in MFD^- , and as a result, obtaining v is a maximal clique listing problem, in which an examination of each RN-miss clique weight is required. Many algorithms as mentioned in Section 5.7.2 exist for searching maximal a clique list. However, Bron–Kerbosch in [18] has an excessive running time in massive and dense graphs when recursively solving for MCL. Existing schemes like [97] and [31] rely on partitioning a graph into non-maximal independent sets to reduce the number of search branches for a clique. However, such algorithms are heuristic and struggle to handle dense and massive graphs as in [102]. The count or the number of maximal independent sets in a graph is NP-complete [48] and [53], and starting with independent branching sets is a non-deterministic complexity. In our

work, any conflict component has a much smaller size than $|\text{MFD}^-|$ so that MCL algorithms can act more efficiently on a component basis instead of an entire MFD^- .

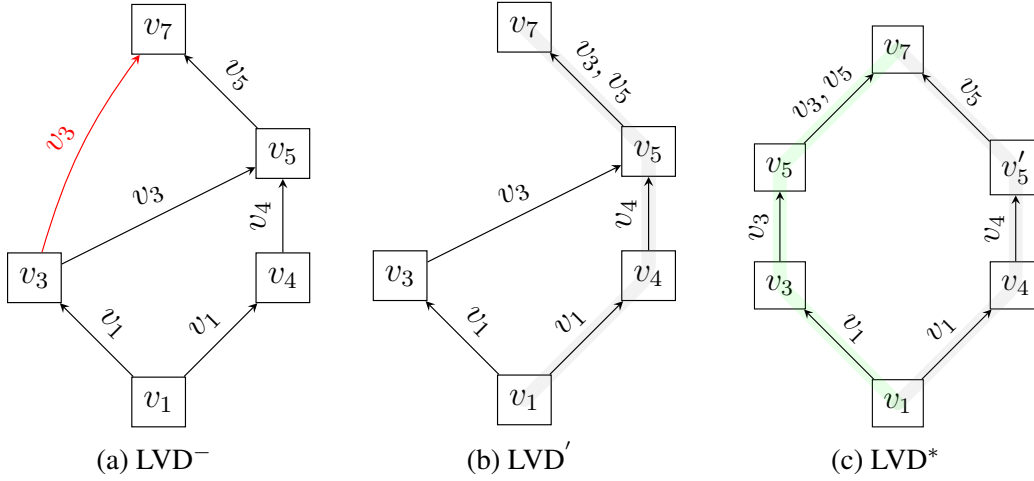


Figure 5.8: Dijkstra's Algorithm and an RN-miss Dependency Example

This work introduces a distinct cost function for conventional minimum cost algorithms running on dependent vertices. For instance, a computed cost for an arc (v, v') with a label set $\alpha(v, v') = v$ is expressed as $|\text{M}(v') \setminus \sqcup \text{M}(v)|$, which represents a size of v' exclusive missed RNs instead of counting a previous RN-miss of a dependent and explored vertex of v . There are two variations of the k -minimum cost path problem. Paths in one can visit a node multiple times, and thus, cycles exist. In another variation, paths must be acyclic. For a digraph $G(V, E)$, Eppstein's algorithm in [30] solves the cyclic version in $O(|E| + |V| \log |V| + k)$. The acyclic path is solvable by Yen's algorithm in [111] within a pseudo-polynomial complexity $O(k|V|(|E| + |V| \log |V|))$. In Section 5.7.6, our proposed LVD^* is acyclic, and thus, Yan's approach is suitable when k is not large for enumerating all possible paths between upper and lower-level vertices.

Here, we summarize three approaches for searching a minimum miss vertex set in LVD^- by: (1) Having $\text{LVD}^- \setminus \{e \mid \Delta(e) > 1\}$ by excluding all dependency arcs and setting each link cost as one allows using Yan's algorithm for listing all possible paths between LVD^- 's end components. From such a list afterward, we can obtain a path with an actual minimum RN-miss by directly evaluating each path into MFD^- for an actual minimum RN-miss. In dense graphs, such a process is computationally expensive. (2) For a $\text{LVD}^-(V, E)$, performing k -clique listing in $\psi(V, E') = \kappa_{|V|} \setminus \{e \mid \Delta(e) = 0\}$ (no links within a level) for all cliques of size $k = \mathcal{L}(\text{LVD}^-)$, which is equivalent to the number of levels or conflict components in LVD^- before picking one with a minimum RN-miss via a similar clique miss evaluation as in approach-1. The complexity of listing k -cliques is proportional to $O(k|E|a(\psi)^{k-2})$ where k is as defined above and $a(\cdot)$ is the arboricity of a graph [25]. The arboricity of a graph is the minimum number of forests, into which its links

can be partitioned. In more detail, any $|V|$ -vertex forest has at most $|E| = |V| - 1$ edges as proven. Thus, the arboricity of $\psi(V, E')$ is at least $\lceil |E'| / (|V| - 1) \rceil$. This lower bound is minimal for complete graphs, and $O(k|E'| |V|^{k-2})$ is the least worst scenario for complete graphs. In general, k can be large for dense MCNs and adds extra linear complexity for clique evaluation.

In contrast, a determined RN-miss via approach-3 of a dependency path reflects an actual miss value. This is because when applying vertices of such a path into CMCN needs no additional RN-miss confirmation as mentioned above. Further, our designed LVD^- in Figure 5.8a is solvable as in Figure 5.8c via vertex copying for an actual minimum RN-miss path searching, for which a modified version of Dijkstra's algorithm can be applied. Figure 5.8b shows how the existence of RN-miss dependency link mistakes minimum miss path searching between vertex v_1 and v_7 . For each relinking loop in Figure 5.4, our work uses a modified cost for Dijkstra's algorithm for searching a minimum RN-miss path p^* over a specific miss adjacency \mathbf{a} and congestion exclusion matrices \mathbf{e} . This cost function is: $|M(ID(v_2)) \setminus \sqcup \{M(v) \sqcup \mathbf{e}(N(v), ID(v_2)) | v \in \mathbf{a}(v_1, v_2)\}|$. Here, ID is a matrix for storing the original IDs of each copied vertex. The number of minimum miss paths or cliques required has been reduced only to the minimum RN-miss paths between the vertices of the lower and upper-level components. For approaches (1) and (2), solving congestion and interference without distinction causes both approaches to be impractical when compared to approach-3 as in Figure 5.7b. For instance, a whole RN-miss of clique or path consists of (v_3, v_4, v_7, v_9) considers RN-15 because of v_3 . However, such a consideration is problematic since RN-15 can remain because of v_3 congestion link or $RN-10 \rightsquigarrow RN-15$. The only way to overcome such an issue is by applying each vertex of a clique or path onto $MBDF^-$, and confirming its RN-miss accordingly. Neither approach is applicable for minimizing RN-miss when congestion in vertex-dependent graphs exists and is managed as interference. Thus, solving congestion as interference causes approach-1 and approach-2 for minimum RN-miss path searching in Figure 5.7a to be similar to approach-3 when no distinction between congestion and interference links is applied. This simplified approach-3 is considered as a **polynomial dependency placing and solving before an RN-miss dependency conditioned minimum RN-miss path searching**. Hence, approach-3 could be considered as a general framework for a minimum miss-cost searching in vertex-dependent environments when congestion exists. The only downside here is having a high memory demand because of vertex copying.

From a set consisting of minimum miss paths between upper and lower-level vertices of LVD^- of Figure 5.8c, a highlighted path $p^* := (v_1, v_3, v_5, v_7)$ over a corresponding dependency solved LVD^* minimizes RN-miss. **P6** would choose for a current candidate schedule $MBDF^-$ a miss path whose RN-miss or $M(p^*) = \sqcup M(v) \forall v \in p^*$ is minimum such that $\forall a \in M(p^*)$, a relinking delay constraint $\sum \bar{y}(a_{MBDF^+}) - \bar{y}(a_{MBDF^-})$ is minimum, where $\bar{y}(a_{MBDF^-})$ is a 's level in $MBDF^-$ of a relinking loop before applying p^* while $\bar{y}(a_{MBDF^+})$ is a 's level in after being relinked in $MBDF'$.

Thus, relinking all missed RNs makes MBDF^+ a next schedule candidate or MBDF^- . The latter constraint is essential since for any two paths p^* and p' with respective RN-misses $M(p^*)$ and $M(p')$ such that $|M(p')| < |M(p^*)|$, p' may cause a longer MBDF^+ when compared with p^* . This occurs when a missed RN of $M(p')$ causes a non zero deviation in schedule length or $\mathcal{L}(\text{MBDF}^+) > \mathcal{L}(\text{MBDF}^-)$ after relinking.

5.7.7 RN-Relinking

The demonstration in Figure 5.9 is a relinking example of a missed RN from a previous candidate schedule MBDF^- because of applying p^* during a current optimization loop of Figure 5.4, and to reconstruct a new candidate or MBDF^- for a next optimization loop. This relinking mechanism preserves a minimum delay constraint on p^* as mentioned in Section 5.7.6. Handling a MBDF^- as discussed in Sections 5.7.3, 5.7.4, 5.7.5 and 5.7.6 may result in a disconnected schedule or MBDF^+ when applying p^* . Note, using such a path without our discussed constraints in Section 5.7.6 could increase the MBDF^- number of levels for the next loop. For instance, an alternative RN-miss path p' whose miss set $M(p') = \sqcup M(v) \forall v \in p'$ is not minimum in size compared with $M(p^*)$ may also result in a minimum length MBDF^- because of such a high relinking chance of $M(p')$ when used p' instead of p^* . This is because such a set of missed RNs in $M(p')$ on average can also have a high relinking chance at near current levels from their previous miss levels similar to RNs of $M(p^*)$. However, **P6** applies p^* to minimize the summation of all RNs delay. Further, our relinking algorithm maintains a high RN-relinking chance for all missed RNs in $M(p^*)$ via two approaches: (1) For a missed RN from a level \bar{y} , RN-relinking preserves an exact reconnecting likelihood of such an RN at $\bar{y} + 1$ by possible reuse of each missed RN's previous MBDF^- parent in \bar{y} in addition to any existing relinking chance at $\bar{y} + 1$.

The RN-relinking algorithm receives MBDF^- as a disconnected forest of digraphs. Each resulting digraph root-RN whose previous parent-RN is still connected in MBDF^- is first relinked in MBDF^+ . (2) Each remaining RN-miss is relinked afterwards in MBDF^+ . This process minimizes both schedule length and individual RN delays namely: $\mathcal{L}(\text{MBDF}^*) = \arg \min \mathcal{L}(\text{MBDF}^-) \forall \text{MBDF}^- \in \mathbb{MBDF}^-$ and $\sum \bar{y}(a) \forall a \in \text{MBDF}^-$. The \mathbb{MBDF}^- is a set of all possible minimum delay schedules. Each root-RN of a disconnected digraph of MBDF^+ is an inevitable conflict miss. Hence, for the next relinking loop, such root-RN is relinked first after removing all communication and congestion links from all predecessors of the previous loop CMCN. Each remaining missed RN can be either an inevitable or consequential miss because of such a missed root RN. This algorithm is an iterative relinking scheme, in which a minimum length relinking path is chosen for each missed RN.

Figure 5.9a is an RN relinking example. The red links are for interference, blue arcs are congestion ones, and dashed ones are dummy RN reuse links. Links use one channel except for dashed ones with no channel. RN-3 acts as a congestion parent RN-5, and does not need

to be copied since RN-1 still resides in RN-3's level. From Figure 5.9a, RN-5 is connected via RN-2, and experiences interference from RN-1 and congestion from RN-3 as mentioned. The conflict solving of MBDF^- results in a disconnected MBDF^+ when applying a p^* as discussed in Section 5.7.6. This MBDF^+ has RN-5 as disconnected, and needs to be relinked. The RN-relinking disables all communication and congestion channels on links (2, 5) and (3, 5) since both cannot prevent such RN-5 inevitable miss. The algorithm relinks RN-5 via reusing RN-2 since RN-3 has no conflict-free channel to maintain RN-5 in place while RN-1 remains as an interference source. This relinking process may generate new conflict components that need solving as in Section 5.7.2. RN-5 is now connected with RN-2', and MBDF^- is now MBDF^* . The RN-relinking algorithm. Figure 5.11 is also influenced by such an RN reuse over many levels or slots of MBDF^- , and thus, resolves our above concern for a minimum schedule length or $\mathcal{L}(\text{MBDF}^-)$. Figure 5.11 is an example of RN-1 maximum usage and back-off over **P6**. This is a sub-schedule of a MBDF^- . Each non-dashed link has an active channel while a dashed one is a dummy link for illustrating RN reuse. RNs in red are previous conflict miss. Each missed RN is re-linked via a reused RN with a prime sign.

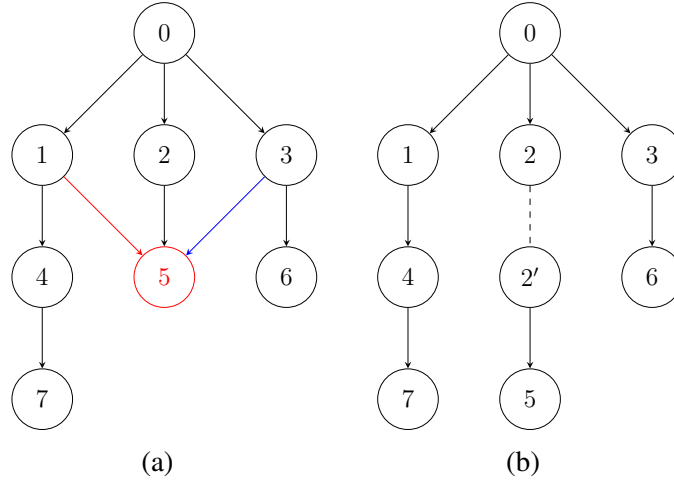


Figure 5.9: RN-Relinking

5.8 Results and Evaluation

Here we compare **P6** with **P1-P5**. The analysis shows the compatibility between **P1-P5** as local conflict optimization and back-off schemes against **P6**. For a CMCN, we examine the delay and redundancy of **P1-P6**. This analysis is based on handling \mathcal{I}_2 of Table 5.2. The study in [47] assumes only one node to send over a slot, and thus, scalability and packet size restrictions are challenges. Handling \mathcal{I}_2 is simplified in [47]. These assumptions are not placed in our work. For a network deployed over a size 15×15 , and $|\mathbb{R}| \approx 400$ nodes in in [47], we further examine performance as a function of CBRS channels. Figure 5.10 shows **P6** always outperforms **P1-P5**

with an average of 27 milliseconds in a delay over all sets of CBRS channels giving all policies achieve maximum coverage. However, in [47], the minimum delay was ≈ 50 milliseconds. Here, our CMCN simulation set of parameters is: $\mathcal{A} = 5 \times 5$, $|\mathbb{R}| = 400$, $\lambda = 1.8$, $\alpha = 1$ and $1 \leq \omega \leq 3$ for more challenging setup over [47] unless mentioned otherwise. The size of each box marker of each plot below is proportional to each policy minimum and maximum delay values.

5.8.1 Delay

Delay and each analyzed performance metric are based on maximum RN coverage constraint $\max \mathbf{H}(\text{MBDF}^*, \mathbf{c})$ over a set \mathbf{c} of channels. The simulated CBRS SAS blocks any RN of CMCN when causing any \mathcal{I}_1 instance by conflicting an IA or PAL over a CBRS channel. For a CMCN minimum delay schedule MBDF^* , an overall delay is equivalent to the diameter length of MBDF^* from a given broadcast source, and in short: $\mathbf{ID}(\text{MBDF}^*, \mathbf{c}) = \mathcal{L}(\text{MBDF}^*)$ over a set \mathbf{c} of channels. The monotonic raise in delay (box size) as interference intensity ω increases with **P1-P5** is because of the lack of a global conflict view as mentioned in Section 5.6.1. The view in Figure 5.10 shows: (1) Linear increase in delay as ω increases as boxes increase in length for each policy. (2) Linear reduction in delay as we increase \bar{c} in general for each policy **P4-P6**. This worst delay case with **P4** and extra channels is because of the ordered mini-slots τ' of one millisecond each as we explained in Section 5.6.1. Thus, an increase in \bar{c} overcomes our monotonic raise in delay as ω is magnified for each policy except in general. **P6** achieved minimum delay over all simulated channel sets.

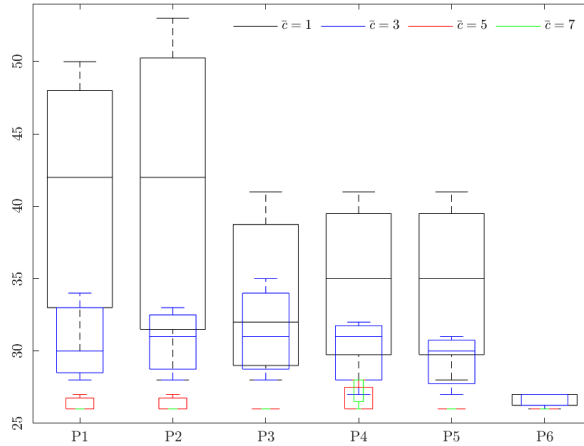


Figure 5.10: Policy or Schedule Delay [s]

5.8.2 Maximum RN Usage-Times

The RN usage-times in [47] in a schedule is defined by the maximum number of times an RN disseminates a broadcast packet and defined by: $\eta = \max \eta(a) \quad \forall a \in \mathbb{R}$ where $\eta(a)$ is the number of channels used by an RN labeled as a . Low RN usage-times indicate adequate fairness in scheduling since all nodes contribute equally to propagating a broadcast message. This number,

however, is not an indicator of the width of a broadcast schedule or the maximum number of RNs in a slot as a node a with $\eta(a) = 2$ can relay a broadcast packet to two relay stations or more. Each RN can appear at many distinct levels for covering the remaining RNs. This redundant usage can cause packet redundancy. **P1-P3** is an obvious instance of such RN usage-times, in which RNs can be reused for each backed-off channel. **P5-P6**, in contrast, can re-access a used channel to cover a previous RN-miss of a conflict clique as indicated in Section 5.6.1.

For **P1-P5** and a set \mathcal{c} of CBRS channels, an RN usage-times equals the number of \mathcal{c} accesses. The maximum RN usage among all RNs is MBDF* usage-times. **P6** as a multi-level minimum delay policy, in which RN-relinking, as described in Section 5.7.7, reuses RNs. However, **P1-P5** RN usage-times is comparable with **P6** when considering any reused RN of Figure 5.11 as redundant with a previous sending to a non-empty set of non-dummy successors on channel c_k . Hence, a usage-times of a with a copy set \mathbf{a} over a set of channels \mathcal{c} is an indicative function: $\mathbb{R}(a, \mathcal{c}) = \sum_{c \in \mathcal{c}} \sum_{a' \in \mathbf{a}} \mathbb{1} \mid \mathcal{K}_c(a') \neq \emptyset$ where $\mathcal{K}_c(a')$ is a set of non dummy successors of a' over channel c .

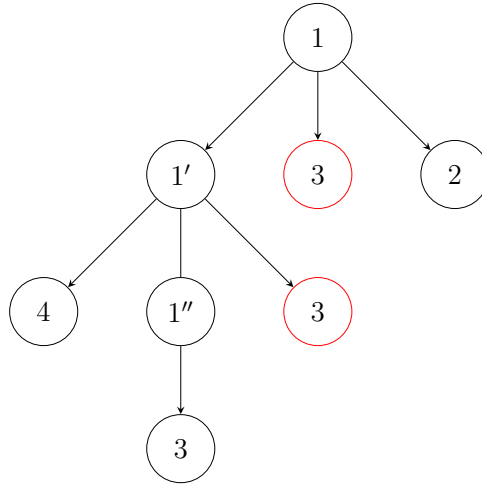


Figure 5.11: RN-1 RN Usage and Back-off over **P6**

Figure 5.11 is a usage instance of RN-1 three times. RN-3 was previously connected with RN-1 and RN-1'. Thus, for RN-1, $\mathbb{R}(1, 1) = 3$. This function is now comparable with η of any policy. For a schedule MBDF* such a maximum RN usage-times is described as: $\mathbb{R}(\text{MBDF}^*, \mathcal{c}) = \max \mathbb{R}(a, \mathcal{c}) \forall a \in \mathbb{R}$. By adding more channels into \mathcal{c} , an increase in $\mathbb{R}(\text{MBDF}^*, \mathcal{c})$ is expected since an RN can be reused over more channels. However, such usage is constrained by each RN location, for which more channels might not raise $\mathbb{R}(\text{MBDF}^*, \mathcal{c})$ as Figure 5.12 shows for **P1-P5** over a set of five channels. **P6** has a maximum RN usage as \mathcal{c} has more channels when considering RN-miss dependency between levels as discussed in Section 5.7.3. For one channel all policies have similar RN usage-times except **P4** and **P5** as both permit reusing an RN over the

same channel as explained in Section 5.6.1. For **P1-P3**, we observe no increase in $\mathbb{R}(\text{MBDF}^*, c)$ beyond three channels.

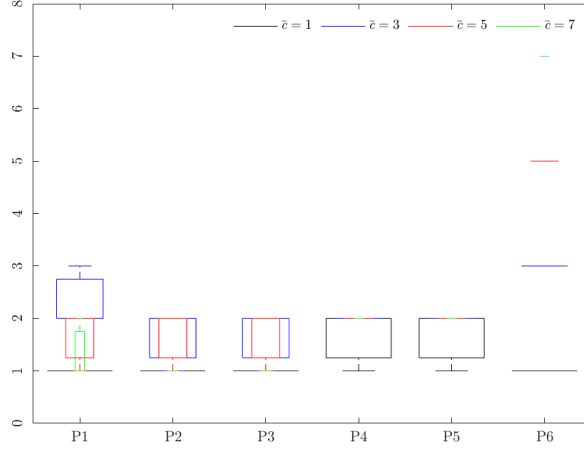


Figure 5.12: Policy RN Usage-Times

5.8.3 Packet Redundancy

Packet redundancy of channel c is a ratio measure between redundant data or badput sent via MBDF^* and all generated data or throughput $g(c)$, and determined by: $\mathbb{P}(\text{MBDF}^*, c) = \sum_{c \in c} (g(c) - g'(c)) / g(c)$. The badput of channel c is calculated as a difference $g(c)$ and successfully delivered data or goodput $g'(c)$. The previous $\mathbb{R}(\text{MBDF}^*, c)$ is not a packet redundancy indicator as $\mathbb{P}(\text{MBDF}^*, c)$. For a multi-channel cellular network, above and below a specific number of channels, packets tend to be redundant. This number depends on the network connectivity. For our simulated CMCN, three channels have shown a minimum number of redundant broadcast packets as indicated in Figure 5.13.

5.8.4 Policy Back off

Policy back-off $\mathbb{B}(a, c)$ is a measure of average RN delay deviation in MBDF^* . The $\mathbb{B}(a, c)$ for an RN a with a copy set \mathbf{a} in MBDF^* and a predecessor RN on channel c is an indicator function defined as: $\mathbb{B}(a, c) = \sum_{a' \in \mathbf{a}} \mathbb{1} \mid \mathcal{K}_c(a') = \emptyset \wedge \mathcal{K}_c(\bar{a}) \cap \mathcal{K}_c^-(a') \neq \emptyset$ where $\mathcal{K}_c(a')$ again is as defined earlier, $\mathcal{K}_c(\bar{a})$ is a non dummy successors of \mathbf{a} 's last RN and $\mathcal{K}_c^-(a')$ contains a set of predecessors of a' in previous schedule candidates. Figure 5.11 is also a back-off instance, in which RN-1 has three copies, of which RN-3 was previously connected with RN-1 and RN-1'. However, $\mathbb{B}(1, 1)$ or back-off of RN-1 on channel-1 is zero since RN-1 in each previous copy of RN-1'' namely RN-1 and RN-1' who were previous parents of RN-3 is still sending on channel-1. Thus, RN-1 has a zero back-off count in such a scenario. Figure 5.14 shows a cumulative back-off count of all RNs in \mathbb{R} , and is defined by: $\mathbb{B}(\text{MBDF}^*, c) = \sum_{c \in c} \sum_{a \in \mathbb{R}} \mathbb{B}(a, c)$. Now, we can compare each back-off scheme of **P1-P5** and **P6**. $\mathbb{B}(\text{MBDF}^*, c)$ for **P1-P5** is a summation of all

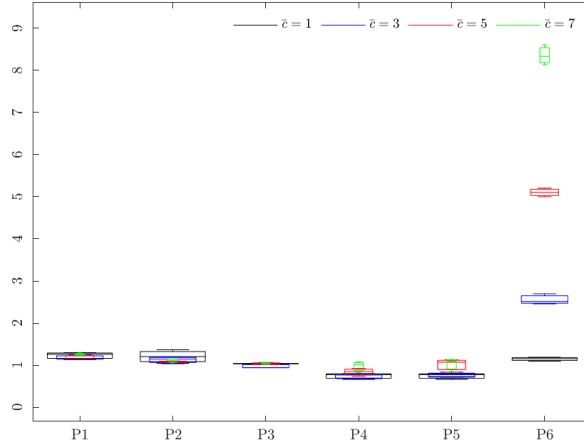


Figure 5.13: Policy Packet Redundancy [Packets]

back-off cases of \mathbb{R} over all channels. Figure 5.14 indicates as c increases in size, policies become more similar in backing off RNs. The back-off reduction increases with more CBRs channels. **P6** avoids backing off RNs even over a small set of CBRs channels.

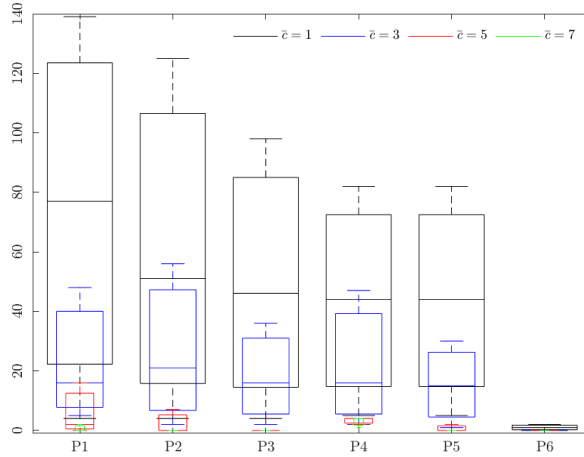


Figure 5.14: Policy RN Back-off Times

5.8.5 RN-Delay Distribution

The delay of a schedule discussed in Section 5.8.1 indicates an overall schedule length or $\mathcal{L}(\text{MBDF}^*)$. However, as we mentioned in Section 5.7.1, RNs in a MBDF^* may not be served via minimum delay paths. Theoretically, a measure of RN delay in a schedule MBDF^* with $\Pi(\text{MBDF}^*, c) = \min \sum_{a \in \mathbb{R}} \mathcal{L}(p^*(a))$ and minimum $\mathcal{L}(\text{MBDF}^*)$ over a set c of channels is sufficient to confirm minimum delay schedule. Here, $p^*(a)$ is a minimum serving path for RN a in MBDF^* . Therefore, a simple distribution comparison is conducted for proving **P6** is indeed minimum over both aspects of individual RN delay and schedule length when compared with other

policies. Here, against **P6** we choose **P1-P3** along with **P4** and **P5** for individual RN delay comparison. **P6** has an early convergence in Figure 5.15 when \bar{c} and ω are high. This increase in CBRS channels offers **P6** additional links to reduce delay.

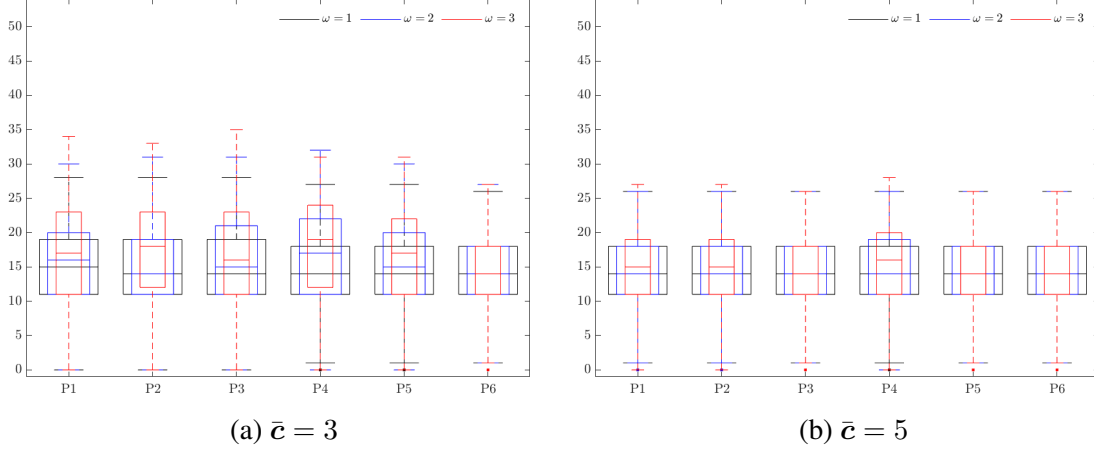


Figure 5.15: Policy RN Delay Convergence [s]

The view of RN average delay in Figure 5.16 indicates individual RN delay in contrast with the overall schedule delay discussed in Section 5.8.1. Each policy except **P6** shows a range of average RN delay as ω varies as each channel box represents. **P6** in contrast has a minimum and steady broadcast RN delay over each channel. Figure 5.10 is a direct relationship between schedule delay and the number of channels. Figure 5.16, in contrast, is an inverse relationship between individual RN delay and the number of channels. This is because individual RN delay is a secondary optimization parameter, in contrast to, a schedule delay, for which **P6** may relink conflicted RNs via longer delay paths in MBDF^+ while still minimizing $\mathcal{L}(\text{MBDF}^*)$ as mentioned in Section 5.7.7.

5.8.6 Confidence Factor

For **P6**, we propose a relative confidence factor $\mathbf{F}(\text{MBDF}^*, c)$ for a minimum delay schedule MBDF^* . This factor is concerning an initial delay schedule MBDF^+ and over a set c of CBRS channels. $\mathbf{F}(\text{MBDF}^*, c)$ does not imply a lack of delay optimality in **P6**, but determines by how much MBDF^* deviates in length from such an initial conflict schedule. To recall, the length of the longest delay path or diameter of MBDF^* from a given broadcast source is $\mathcal{L}(\text{MBDF}^*)$. Figure 5.17 shows $\mathbf{F}(\text{MBDF}^*, c)$ over distinct sets c of channels. Figure 5.17 shows a decay in $\mathbf{F}(\text{MBDF}^*, c)$ when \bar{c} is small and interference intensity or ω increases. This occurs because of new levels added for reused RNs especially when contention over c is high. Increasing a channel width improves speed and throughput for CMCN broadcasting. The work in [10] demonstrates a duplication in throughput when an LTE client switches from a 5 MHz to a 10 MHz channel. However, CoCI and

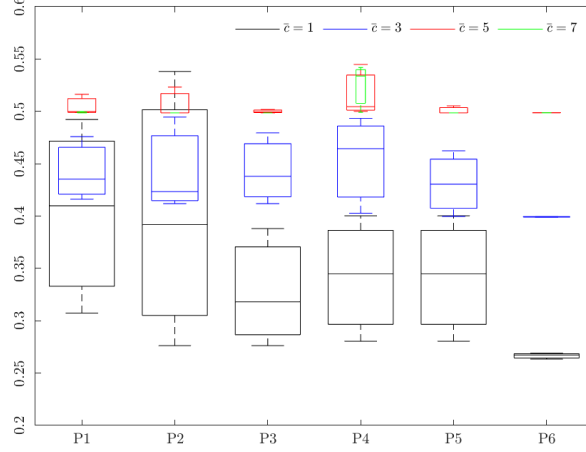


Figure 5.16: Policy RN Mean Delay [s]

ACC would be an issue with wider LTE channels. Here, since a channel width over CBRS depends on the number of used channels, **P6** shows as \bar{c} is magnified, a $\bar{c} = 5$ is sufficient for achieving a minimum delay schedule with a $\mathbb{P}(\text{MBDF}^*, c) = 100\%$.

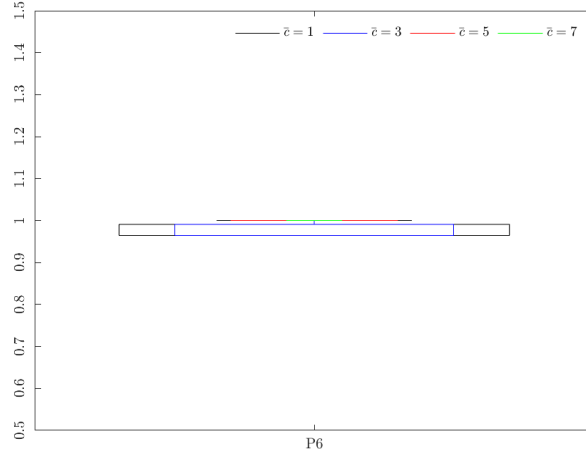


Figure 5.17: **P6** Confidence Factor

5.8.7 Delay v.s. Network Size

The previous discussion encounters how a schedule length varies as interference intensity or CBRS channel set size varies. Table 5.5 summarizes such a delay as the number of RNs grows. For each policy, we evaluated $\mathcal{L}(\text{MBDF}^*)$ over $\lambda = 1.8$, $1 \leq \omega \leq 2$ and $\bar{c} = 5$. These values are selected to simulate performance when either congestion only occurs $\omega = 1$ or both interference and congestion $\omega = 2$ exist. The two columns beside each policy are for both ω values respectively. Table 5.5 indicates as $|\mathbb{R}|$ and ω raise, **P6** provides a minimum delay schedule. The level-based maximum coverage policies such as **P1-P5** show increased delay because of not considering con-

flict dependency between levels. The OFDM-ordered access mechanism used by **P4** also causes a delay increase.

$ \mathbb{R} $	100	200	256	403	920	1620	3641
\mathcal{A}	5×5	7×7	8×8	10×10	15×15	20×20	30×30
P1	20	23	31	30	50	66	95
	20	24	31	32	52	72	100
P2	20	23	31	30	49	66	95
	21	23	32	32	51	71	97
P3	20	23	31	30	49	66	95
	20	23	31	31	52	69	99
P4	28	31	38	38	65	86	123
	28	32	39	44	73	100	137
P5	20	23	31	29	49	67	95
	20	23	32	30	49	69	96
P6	20	23	31	29	49	62	95
	20	23	31	29	49	66	95

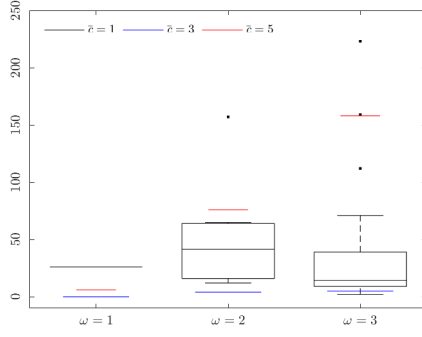
Table 5.5: Delay vs. Network Size Evaluation

5.8.8 RN-Relinking and Dependency Handling

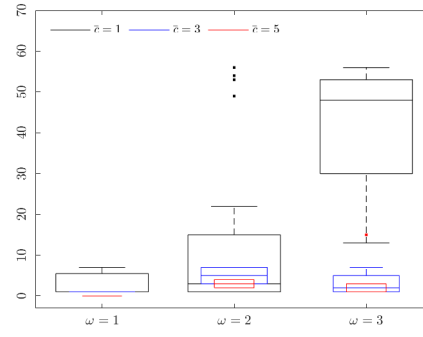
The number of RN-miss dependency arcs in LVD^- is one performance metric of **P6**. The second metric is the RN relinking mechanism as detailed in Section 5.7.7. The relinking algorithm reconnects disconnected RNs when RN-miss dependency is solved. For CMCNs of eight-hundred RNs and distinct sets of CBRs channels, Figure 5.18 summarizes both metrics. The y -axis as a count for both metrics decays over time. The minimum miss path selection criteria discussed in Section 5.7.6 causes RN relinking to exhibit a monotonic decrease proportional to interference intensity. The maximum number of dependency arcs handled via **P6** is not assumed to have a monotonic decrease as relinked RNs could add more dependency arcs among new conflict components over time. Handling miss dependency is done by copying vertices from bottom to top of LVD^- . This copies any unsolved dependency arc towards a copied vertex. Each original dependency arc is solved once, and its solution is applied for any identical arc in LVD^- in a polynomial time. Further, all dependency links from a vertex are solved in one call of our RN-miss handling algorithm.

5.9 Conclusion and Future Work

Primarily, we aimed to manage conflict for minimizing delay, redundancy, back off, and other metrics for CMCNs over shared bands. Each proposed policy handles conflict for mutual benefit and failure management between mobile carriers over free or less expensive spectrum like CBRs. For CMCNs, our work presents a polynomial baseline scheduling for MBDF. The use of conflict graphs allows our analysis to establish a new transformation for an initial instance $MBDF^+$,



(a) Dependency vs. ω



(b) Relinking vs. ω

Figure 5.18: **P6** Dependency and Relinking

in which dependency between conflict components can be easily visualized and solved. This effort can be considered a reference framework for solving many optimization problems similar to MBDF. **P6** can be simplified over many aspects. For instance, considering a subset of solutions for each conflict component and approximating congestion as interference. Each policy has been evaluated over a set of QoS metrics, distinct conflict conditions, and network scales. The CMCN is an encouraging approach over free bands to minimize hardware costs while circulating mutual benefits among mobile carriers. This benefit will be analyzed in a future study.

6 Conclusion and Future Work

This study analyzed the behavior of single session relay routes that minimize the end-to-end delay for all possible TCP pairs of the examined network. The analysis shows a common delay reduction range below hundred milliseconds. This study has confirmed the existence of a considerable number of relay routes that can minimize the physical round-trip times by the mentioned reduction range for the majority of nodes. The advantage of this relay design varies from node to node, based on their topological positions. The study summarizes the tradeoff between minimizing the end-to-end delay and the consumption of the network resources and shows the lack of delay reduction using fewer links than the underlying paradigm. In the future, we would compare the end-to-end minimum delay relay paths and another relay design that optimizes the overall network delay instead of such individual paths. This comparison aims to minimize the consumption of network resources.

This study shows relay paths are more stable in terms of HTL, and our results demonstrate that an estimation-based relay selection can help reduce the search space of better paths, and speed up locating better relay paths for streaming services. This work recommends that an estimation-based relay can enhance performance while reducing the probing overhead for minimum delay routing. The HTL as a more stable metric has been used to infer relay changes with an error of fifteen percent. For large scale-networks, we find that instead of focusing on exact bandwidth estimates, our second bandwidth scheme shows that it is highly possible to find other relay paths that can serve a demand rate quickly. Therefore, we are expanding our work to design a joint TCP-UDP scheme that could improve YouTube and VoIP services for instance.

This work has shown that our distributed QL-TDMA is a candidate for distributed TDMA frame scheduling. The results indicate such an approach with a careful parameter setting can lead to a minimum frame length. This QL-based scheduling is sensitive to two parameters. These ones are the collision window and frame slot redundancy. The proposed QL system permits nodes to exploit and switch slots by each frame's end. This switching may increase convergence time. The next version of our model allows nodes to change slots within a frame immediately after being informed of a collision according to what is available of slot information so far. This will adopt a stronger penalty function for any node with a high slot competition within a frame. The next step is to advance the collision detection capability and conditions to scale the network size and reduce the convergence period.

Primarily in our last chapter, we aimed to manage conflict for minimizing delay, redundancy, backoff, and other metrics for CMCNs over shared bands. The proposed policies manage conflict between mobile carriers over CBRS. For CMCNs, and using conflict graphs, our analysis establishes a new transformation for an initial instance MBDF⁻, in which RN dependency can be solved. This approach can solve many similar problems to MBDF. The next version of **P6** would

be simplified. The reduction of component solutions is a simplified approach. Policies have been examined for a set of QoS metrics and over many network scales. The proposed CMCN is an approach to minimize future hardware costs while having mutual benefits for mobile carriers. This benefit would be analyzed in an auction-based framework for CMCN participants.

BIBLIOGRAPHY

- [1] CAIDA.
- [2] RIPE.
- [3] Akamai.
- [4] IP Option Numbers: The Current Recommended Default TTL for Internet Protocol is 64 in RFC-791 and RFC-1122. pages 1–64, 2018.
- [5] CBRS, SAS and Spectrum Sharing: The Complete Guide. 2021.
- [6] Jianping An, Kai Yang, Jinsong Wu, Neng Ye, Song Guo, and Zhifang Liao. Achieving sustainable ultra-dense heterogeneous networks for 5g. *IEEE Communications Magazine*, 55(12):84–90, 2017.
- [7] Mustafa Y Arslan, Jongwon Yoon, Karthikeyan Sundaresan, Srikanth V Krishnamurthy, and Suman Banerjee. Fermi: a femtocell resource management system for interference mitigation in ofdma networks. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 25–36, 2011.
- [8] François Baccelli, Bartłomiej Błaszczyszyn, and Paul Muhlethaler. An aloha protocol for multihop mobile wireless networks. *IEEE transactions on information theory*, 52(2):421–436, 2006.
- [9] Michael Backes, Thorsten Holz, Christian Rossow, Teemu Ryttilähti, Milivoj Simeonovski, and Ben Stock. On the feasibility of ttl-based filtering for ddos mitigation. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 303–322. Springer, 2016.
- [10] Ghufuran Baig, Ian Kash, Bozidar Radunovic, Thomas Karagiannis, and Lili Qiu. Interference management for unlicensed users in shared cbbs spectrum. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, pages 333–345, 2018.
- [11] Andrea Baldini, Lorenzo De Carli, and Fulvio Rizzo. Increasing performances of tcp data transfers through multiple parallel connections. In *2009 IEEE Symposium on Computers and Communications*, pages 630–636. IEEE, 2009.
- [12] Abderrahmane BenMimoune, Fawaz A Khasawneh, Bo Rong, and Michel Kadoch. Dynamic joint resource allocation and relay selection for 5g multi-hop relay systems. *Telecommunication Systems*, 66(2):283–294, 2017.
- [13] Dimitri Bertsekas and Robert Gallager. *Data networks*. Athena Scientific, 2021.
- [14] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: A media access protocol for wireless lan’s. *ACM SIGCOMM Computer Communication Review*, 24(4):212–225, 1994.
- [15] Jayaram Bhasker and Tariq Samad. The clique-partitioning problem. *Computers & Mathematics with Applications*, 22(6):1–11, 1991.
- [16] Ashutosh Bhatia and Ramesh C Hansdah. A distributed tdma slot scheduling algorithm for spatially correlated contention in wsns. In *2013 27th International Conference on Advanced*

- Information Networking and Applications Workshops*, pages 377–384. IEEE, 2013.
- [17] Ashutosh Bhatia and RC Hansdah. A classification framework for tdma scheduling techniques in wsns. *arXiv preprint arXiv:2002.00458*, 2020.
 - [18] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
 - [19] Nam Nguyen Canh. On globally solving the maximum weighted clique problem. *Journal of Algorithms*, 2014.
 - [20] Goutam Chakraborty. Genetic algorithm to solve optimum tdma transmission schedule in broadcast packet radio networks. *IEEE Transactions on Communications*, 52(5):765–777, 2004.
 - [21] Lucas Chaufournier, Ahmed Ali-Eldin, Prateek Sharma, Prashant Shenoy, and Don Towsley. Performance evaluation of multi-path tcp for data center and cloud workloads. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*, pages 13–24, 2019.
 - [22] Fangfei Chen, Ramesh K Sitaraman, and Marcelo Torres. End-user mapping: Next generation request routing for content delivery. *ACM SIGCOMM Computer Communication Review*, 45(4):167–181, 2015.
 - [23] Zheng Chen and Marios Kountouris. Decentralized opportunistic access for d2d underlaid cellular networks. *IEEE Transactions on Communications*, 66(10):4842–4853, 2018.
 - [24] Hongju Cheng, Naixue Xiong, Larence T Yang, and Young-Sik Jeong. Distributed scheduling algorithms for channel access in tdma wireless mesh networks. *The Journal of Supercomputing*, 63(2):407–430, 2013.
 - [25] Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on computing*, 14(1):210–223, 1985.
 - [26] David Choffnes and Fabián E Bustamante. On the effectiveness of measurement reuse for performance-based detouring. In *IEEE INFOCOM 2009*, pages 693–701. IEEE, 2009.
 - [27] T van Dam. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the ACM Symposium on Networked Embedded Systems (SenSys, 03)*, 2003.
 - [28] Mike Dano. Verizon Aims to Test 5G, Carrier Aggregation in 3.5 GHz CBRS Band. *5G Mobile Strategies*, 2017.
 - [29] John Eggerton. FCC Says 3.5 GHz Services Are Good to Go. *Multichannel News*, 2020.
 - [30] David Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.
 - [31] David Eppstein, Maarten Löffler, and Darren Strash. Listing all maximal cliques in large sparse real-world graphs. *Journal of Experimental Algorithmics (JEA)*, 18:3–1, 2013.
 - [32] Sonia Fahmy and Minseok Kwon. Characterizing overlay multicast networks and their costs. *IEEE/ACM transactions on networking*, 15(2):373–386, 2007.
 - [33] FCC. FCC Authorizes Full Commercial Deployment in 3.5 GHz Band. 2020.

- [34] Romain Fontugne, Johan Mazel, and Kensuke Fukuda. An empirical mixture model for large-scale rtt measurements. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2470–2478. IEEE, 2015.
- [35] Alan Ford, Costin Raiciu, Mark Handley, and Olivier Bonaventure. TCP extensions for multipath operation with multiple addresses, Internet draft. pages 1–64, 2013.
- [36] Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies. *arXiv preprint arXiv:1512.02011*, 2015.
- [37] Claudio Daniel Freire, Alina Quereilhac, Thierry Turetletti, and Walid Dabbous. Automated deployment and customization of routing overlays on planetlab. In *International Conference on Testbeds and Research Infrastructures*, pages 240–255. Springer, 2012.
- [38] Forough Golkar, Thomas Dreibholz, and Amund Kvalbein. Measuring and comparing internet path stability in ipv4 and ipv6. In *2014 International Conference and Workshop on the Network of the Future (NOF)*, pages 1–5. IEEE, 2014.
- [39] Jimmi Grönkvist. Novel assignment strategies for spatial reuse tdma in wireless ad hoc networks. *Wireless Networks*, 12(2):255–265, 2006.
- [40] Shaista Habib, Fawaz S Bokhari, and Samee U Khan. Routing techniques in data center networks. In *Handbook on Data Centers*, pages 507–532. Springer, 2015.
- [41] Junghee Han, David Watson, and Farnam Jahanian. An experimental study of internet path diversity. *IEEE Transactions on Dependable and Secure Computing*, 3(4):273–288, 2006.
- [42] Robert Hebner and Valerie Zelenty. Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan networks - specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications: Higher speed physical layer (phy) extension in the 2.4 ghz band. *IEEE Std 802.11b-1999*, pages 1–96, 2000.
- [43] Jeremy Horwitz. FCC Unlocks 3.5 GHz CBRS Band, Enables OnGo in Apple and Android phones. 2020.
- [44] Yu-Ching Hsu and Ying-Dar Lin. Multihop cellular: A novel architecture for wireless data communications. *Journal of Communications and Networks*, 4(1):30–39, 2002.
- [45] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 225–238, 2012.
- [46] Irfan Jabandžić, Spilios Giannoulis, Ruben Mennes, Felipe AP De Figueiredo, Maxim Claeys, and Ingrid Moerman. A dynamic distributed multi-channel tdma slot management protocol for ad hoc networks. *IEEE Access*, 9:61864–61886, 2021.
- [47] Shouling Ji, Raheem Beyah, and Zhipeng Cai. Minimum-latency broadcast scheduling for cognitive radio networks. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 389–397. IEEE, 2013.
- [48] Tang Jian. An $O(2^{0.304n})$ algorithm for solving maximum independent set problem. *IEEE*

Transactions on Computers, 100(9):847–851, 1986.

- [49] Junchen Jiang, Rajdeep Das, Ganesh Ananthanarayanan, Philip A Chou, Venkata Padmanabhan, Vyas Sekar, Esbjorn Dominique, Marcin Goliszewski, Dalibor Kukoleca, Renat Vafin, et al. Via: Improving internet telephony call quality using predictive relay selection. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 286–299, 2016.
- [50] Xianlong Jiao, Weidong Xiao, Bin Ge, and Yuli Chen. On minimum-latency broadcast in multichannel duty-cycled wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(8):913451, 2015.
- [51] Jinoh Kim, Abhishek Chandra, and Jon Weissman. Open: Passive network performance estimation for data-intensive applications. 2008.
- [52] Andri Lareida, Daniel Meier, Thomas Bocek, and Burkhard Stiller. Towards path quality metrics for overlay networks. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pages 156–159. IEEE, 2016.
- [53] Eugene L. Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980.
- [54] Yong Liu, Yu Gu, Honggang Zhang, Weibo Gong, and Don Towsley. Application level relay for high-bandwidth data transport. *Proc. of GridNets*, 2004.
- [55] Zhenzhen Liu and Itamar Elhanany. Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks. In *2006 IEEE International Conference on Networking, Sensing and Control*, pages 768–773. IEEE, 2006.
- [56] Bruce M Maggs and Ramesh K Sitaraman. Algorithmic nuggets in content delivery. *ACM SIGCOMM Computer Communication Review*, 45(3):52–66, 2015.
- [57] Mohsen Riahi Manesh and Naima Kaabouch. Interference modeling in cognitive radio networks: a survey. *arXiv preprint arXiv:1707.09391*, 2017.
- [58] Wahida Mansouri, Khitem Ben Ali, Faouzi Zarai, and Mohammad S Obaidat. Radio resource management for heterogeneous wireless networks: Schemes and simulation analysis. In *Modeling and Simulation of Computer Networks and Systems*, pages 767–792. Elsevier, 2015.
- [59] Rob Matheson. MIT News Office. 2020.
- [60] Salim Mohamed, Saptarshi Das, Subir Biswas, and Osama Mohammed. On the significance of layer-3 traffic forwarding. In *International Conference on Wired/Wireless Internet Communication*, pages 170–181. Springer, 2019.
- [61] Salim Mohamed and Osama Mohammed. Towards stable and hybrid udp-tcp relay routing for streaming and voip services. *ICNS*, pages 55–65, 2020.
- [62] John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.
- [63] Thomas Moscibroda and Rogert Wattenhofer. Coloring unstructured radio networks. In *Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and*

- architectures*, pages 39–48, 2005.
- [64] Kyung Mun. Cbrs: New shared spectrum enables flexible indoor and outdoor mobile solutions and new business models. *White Paper*, Mar, 2017.
 - [65] Chiu Yeung Ngo and Victor OK Li. Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms. *IEEE Transactions on Communications*, 51(9):1439–1441, 2003.
 - [66] Thanh-Tung Nguyen, Taejoon Kim, and Taehong Kim. A distributed tdma scheduling algorithm using topological ordering for wireless sensor networks. *IEEE Access*, 8:145316–145331, 2020.
 - [67] Thinh Nguyen and Sen-Ching S Cheung. Multimedia streaming using multiple tcp connections. In *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pages 215–223. IEEE, 2005.
 - [68] Nutaq. .
 - [69] Huiung Park, Haeyong Kim, Kyeong Tae Kim, Seon-Tae Kim, and Pyeongsoo Mah. Frame-type-aware static time slotted channel hopping scheduling scheme for large-scale smart metering networks. *IEEE Access*, 7:2200–2209, 2018.
 - [70] Parth H Pathak and Rudra Dutta. *Designing for network and service continuity in wireless mesh networks*. Springer Science & Business Media, 2012.
 - [71] Vern Edward Paxson. End-to-end internet packet dynamics. In *Proceedings of the ACM SIGCOMM’97 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 139–152, 1997.
 - [72] Vern Edward Paxson. *Measurements and analysis of end-to-end Internet dynamics*. University of California, Berkeley, 1997.
 - [73] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, 2004.
 - [74] U Pongsakorn, Kohei Ichikawa, Hajimu Iida, Nawawit Kessaraphong, Putchong Uthayopas, Susumu Date, Hirotake Abe, Hiroaki Yamanaka, Eiji Kawai, et al. Application-oriented bandwidth and latency aware routing with open flow network. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 775–780. IEEE, 2014.
 - [75] JN Portela and Marcelo S Alencar. Cellular network as a multiplicatively weighted voronoi diagram. In *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, volume 2, pages 913–917. IEEE, 2006.
 - [76] Junaid Qadir, Chun Tung Chou, Archan Misra, and Joo Ghee Lim. Minimum latency broadcasting in multiradio, multichannel, multirate wireless meshes. *IEEE transactions on mobile computing*, 8(11):1510–1523, 2009.
 - [77] Qualcomm. MulteFire™ Technology Progress and Benefits, and How It Enables A New Breed of Neutral Hosts. 2016.

- [78] S Ramanathan. A unified framework and algorithm for (t/f/c) dma channel assignment in wireless networks. In *Proceedings of INFOCOM'97*, volume 2, pages 900–907. IEEE, 1997.
- [79] Subramanian Ramanathan and Errol L Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on networking*, 1(2):166–177, 1993.
- [80] Injong Rhee and Jangwon Lee. Distributed scalable tdma scheduling algorithm. Technical report, North Carolina State University. Dept. of Computer Science, 2004.
- [81] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-mac: a hybrid mac for wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 90–101, 2005.
- [82] Injong Rhee, Ajit Warrier, Jeongki Min, and Lisong Xu. Drand: distributed randomized tdma scheduling for wireless ad-hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 190–201, 2006.
- [83] John Michael Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.
- [84] Rodrigo Caetano Rocha and Bhalchandra D Thatte. Distributed cycle detection in large-scale sparse graphs. *Proceedings of Simpósio Brasileiro de Pesquisa Operacional (SBPO'15)*, pages 1–11, 2015.
- [85] R Rozovsky and PR Kumar. Seedex: A mac protocol for ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 67–75, 2001.
- [86] Pablo San Segundo, Jorge Artieda, and Darren Strash. Efficiently enumerating all maximal cliques with bit-parallelism. *Computers & Operations Research*, 92:37–46, 2018.
- [87] Aggeliki Sgora, Dimitrios J Vergados, and Dimitrios D Vergados. A survey of tdma scheduling schemes in wireless multihop networks. *ACM Computing Surveys (CSUR)*, 47(3):1–39, 2015.
- [88] Yi Shi, Y Thomas Hou, Jia Liu, and Sastry Kompella. How to correctly use the protocol interference model for multi-hop wireless networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 239–248, 2009.
- [89] Hanan Shteingart, Tal Neiman, and Yonatan Loewenstein. The role of first impression in operant learning. *Journal of Experimental Psychology: General*, 142(2):476, 2013.
- [90] Suresh Singh and Cauligi S Raghavendra. Pamas—power aware multi-access protocol with signalling for ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 28(3):5–26, 1998.
- [91] Abhishek Sinha, Georgios Paschos, Chih-Ping Li, and Eytan Modiano. Throughput-optimal multihop broadcast on directed acyclic wireless networks. *IEEE/ACM Transactions on networking*, 25(1):377–391, 2016.
- [92] Ramesh K Sitaraman, Mangesh Kasbekar, Woody Lichtenstein, and Manish Jain. Overlay networks: An akamai perspective. *Advanced Content Delivery, Streaming, and Cloud Services*, 51(4):305–328, 2014.

- [93] Min Song, Jun Wang, and Qun Hao. Broadcasting protocols for multi-radio multi-channel and multi-rate mesh networks. In *2007 IEEE International Conference on Communications*, pages 3604–3609. IEEE, 2007.
- [94] John A Stankovic, TE Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer C Hou. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, 91(7):1002–1022, 2003.
- [95] Ao-Jan Su, David R Choffnes, Aleksandar Kuzmanovic, and Fabián E Bustamante. Drafting behind akamai (travelocity-based detouring). *ACM SIGCOMM Computer Communication Review*, 36(4):435–446, 2006.
- [96] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132. IEEE, 1990.
- [97] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical computer science*, 363(1):28–42, 2006.
- [98] Viet-Hoang Tran, Quentin De Coninck, Benjamin Hesmans, Ramin Sadre, and Olivier Bonaventure. Observing real multipath tcp traffic. *Computer Communications*, 94:114–122, 2016.
- [99] K. Underwood. Smart Spectrum Sharing. *Signal Magazine*, 2021.
- [100] Hrishikesh Venkataraman, Dipesh Gandhi, and Vikrant Tomar. Multi-hop multi-band intelligent relay-based architecture for lte-advanced multi-hop wireless cellular networks. *Wireless personal communications*, 75(1):131–153, 2014.
- [101] Sonia Waharte and Raouf Boutaba. Performance comparison of distributed frequency assignment algorithms for wireless sensor networks. In *International Conference on Network Control and Engineering for QoS, Security and Mobility*, pages 151–162. Springer, 2004.
- [102] Yiyuan Wang, Shaowei Cai, and Minghao Yin. Two efficient local search algorithms for maximum weight clique problem. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [103] Zheng Wang and Jon Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on selected areas in communications*, 14(7):1228–1234, 1996.
- [104] Yi-Wen Wei, Wei-Mei Chen, and Hsin-Hung Tsai. Accelerating the bron-kerbosch algorithm for maximal clique enumeration using gpus. *IEEE Transactions on Parallel and Distributed Systems*, 32(9):2352–2366, 2021.
- [105] WIA. The CBRS Opportunity New Spectrum, Stakeholders, Networks and Devices. 2020.
- [106] WinnForum. Requirements for Commercial Operation in the U.S. 3550-3700 MHz Citizens Broadband Radio Service Band, Technical report. 2017.
- [107] Alec Woo and David E Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 221–235, 2001.

- [108] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 13–24, 2004.
- [109] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies*, volume 3, pages 1567–1576. IEEE, 2002.
- [110] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on networking*, 12(3):493–506, 2004.
- [111] Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.
- [112] Ossama Younis and Sonia Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, 3(4):366–379, 2004.
- [113] Dianbo Zhao and Kwan-Wu Chin. A distributed broadcast algorithm for duty-cycled networks with physical interference model. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–13, 2015.
- [114] Yue Zhao, Xuming Fang, and Zhengguang Zhao. Interference coordination in compact frequency reuse for multihop cellular networks. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 93(11):2312–2319, 2010.
- [115] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A Stankovic. Impact of radio irregularity on wireless sensor networks. In *Second International Conference on Mobile Systems, Applications, and Services (MobiSys2004)*, 2004.
- [116] Chenxi Zhu and M Scott Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. *Wireless networks*, 7(4):371–384, 2001.