1-1-2015

Predictable Real-Time Wireless Networking For Sensing And Control

Xiaohui Liu
Wayne State University,

PREDICTABLE REAL-TIME WIRELESS NETWORKING FOR SENSING AND CONTROL

by

XIAOHUI LIU

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2015

MAJOR: COMPUTER SCIENCE

Approved by:

Advisor	Date

DEDICATION

To my parents, my wife, and my sister.

ACKNOWLEDGMENTS

I owe a great deal of appreciation to my advisor Dr. Hongwei Zhang. He has an amazing breath of knowledge and depth of vision. His remarkable dedication and determination for deep creative work is admirable, especially in the face of uncertainties inherent in open-ended research. Perhaps more importantly, he constantly pushes me to go beyond my intellectual limits and accomplish tasks I deem impossible previously, a challenge I am becoming more and more confident to tackle. He also showed that even top-notch researchers have to put their boots on the ground and go through the same hard thinking process as I do. They have no magic shortcut or trick, much to my surprise. This was especially liberating when I was overwhelmed by the fact that I have to compete with established, even world-renowned, professors to publish.

My parents put me immensely in their debt for my upbringing. My mother always strives to be the best at her work among her peers, whether it is weaving shoes on a factory floor or attending crops in a field. Her work ethic of perseverance and diligence, even under extremely harsh difficulties, has been a constant source of inspiration for me. My father is always behind me whenever I make life-changing decisions, and he encourages me to pursue what my heart tells me to do irrespective of what other people say. He is the example I look up to for being a man of integrity, decency, and optimism.

I wish to express my gratitude to my girlfriend and now wife, Fei Liu, for putting up with my divided attention during the dissertation's gestation period. Contrary to the laws of wireless signal propagation, her unconditional love and support never attenuate nor fade, even over a long distance. Her company has brought me joys I cannot possibly recount here, for which I am eternally grateful.

I am also indebted to the group members over the years, including Xin Che, Qiao Xiang, Xi Ju, Yuehua Wang, Chuan Li, Yu Chen, Qing Ai, Pengfei Ren, and Ling Wang. Together with our collaborator Hai Jin, they make the lab invariably entertaining. We went through the many ups and downs of graduate research together.

I am grateful to my friends and fellow students in and outside our department. A special thanks goes to Shanshan Qiu and Yuliang Geng, who always invited me over to their homes and offered help and comfort unselfishly. It was a lot of fun talking with Jiaxi Hu, Chunhyeok Lim, Eyad Hailat, Shinan Wang, Masud Ahmed, Grace Metri, Cristina Mitrea, Farhana Dewan, Bilal Hawashin, Paul Janiczek, Vineeth Rakesh, Balaji Palaniswami, Nariman Ammar, Yuanzhe Li, Bhanukiran Vinzamuri, and Andrey Kashlev, especially over lunch. I learned something new every time I talked with Safraz Rampersaud. I learned from Sharrukh Zaman's dedication to family. It was delightful to work with Monika Witoslawski , even though I am still having trouble pronouncing her last name like everybody else is.

Besides my advisor, I was privileged to have Dr. Purvesh Khatri, Stanford Ovshinsky, and Dr. Bhaskar Krishnamachari as my tutors, offering personal advice and insights on research. My world view was forever changed by science popularizer and critical thinker Zhouzi Fang and Carl Sagan, whose enlightening works opened the door of science, skepticism, and reasoning for me.

Other thanks are due to Dr. Nathan Fisher, Dr. Loren Schwiebert, and Dr. George Yin for serving on my committee and providing invaluable feedback.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Besides deployments for open-loop sensing such as environmental monitoring, embedded wireless networks are increasingly being explored for real-time, closed-loop sensing and control. For instance, the wireless networking standard IEEE 802.15.4g has been defined for large scale process control applications such as smart grid sensing and control [5], and wireless networks are expected to serve as major communication infrastructures in neighborhood area networks and home area networks of the smart grid [48, 40, 130, 120]. In addition, wireless networking standards such as the IEEE 802.15.4e, WirelessHART, and ISA SP100.11a have been defined for industrial monitoring and control [6, 122, 112], wireless sensor networks have been deployed for industrial automation [70, 121], and the automotive industry has also been exploring the application of wireless networks to vehicle sensing and control [41, 114]. In wireless networked sensing and control, message passing (or messaging for short) across wireless networks is a basic enabler for coordination among distributed sensors, controllers, and actuators. In supporting mission-critical tasks such as smart grid control and industrial process control, wireless messaging is required to be reliable (i.e., having high delivery ratio) and in real-time [89]. This is because packet loss and large delay usually reduce the system stability (e.g., in proportional-integral control), lengthen the settling time, and increase the maximum overshoot in control [55].

Organization of this paper. Chapter 2 introduces a real-time routing protocol that predictably delivers data by their deadlines. Chapter 3 presents a scheduling protocol to ensure a certain link reliability based on the Physical-ratio-K (PRK) model, which is both realistic and amenable for distributed implementation. Chapter 4 presents an efficient greedy scheduling algorithm Best Link First Scheduling to maximize the expected number of packets delivered to the collection tree root by a deadline in lossy multi-hop wireless sensor networks. Chapter 5 concludes the dissertation.

CHAPTER 2

TAMING UNCERTAINTIES IN REAL-TIME ROUTING FOR WIRELESS NETWORKED SENSING AND CONTROL

1 Introduction

In multi-hop wireless networks, a basis for reliable, real-time messaging is real-time routing which routes data packets from their sources to destinations within specified deadlines. Nonetheless, link/path delays (i.e., the time taken to successfully deliver a packet across a link or a path) are dynamic, uncertain in wireless sensing and control (WSC) networks due to factors such as the spatiotemporal wireless link dynamics and the queueing dynamics along links/paths. The dynamics and uncertainties in link/path delays introduce fundamental *challenges* to real-time routing:

- Firstly, the dynamics and uncertainties make link/path delays probabilistic in nature. Given the delay distributions of individual links along a path, the basic problem of computing the probabilistically guaranteed path delays is NP-hard [51]; in real-time routing where nodes need to identify paths that ensure certain delay bounds, however, quantifying probabilistic path delays is a basic task, and it may well have to be executed by resource-constrained nodes in a distributed manner.
- Secondly, given that link/path delay is a highly varying metric and that it can change at a short timescale of each packet transmission, it is important to adapt to in-situ delay conditions in routing. Yet the highly-varying nature of link/path delays makes it difficult to accurately estimate path delays in a distributed and agile manner, and it has been observed that delay-

adaptive routing can lead to routing instability and low data delivery performance in general [37, 137].

Despite much work in throughput- or energy-efficiency-oriented wireless routing [16, 34, 49, 132], real-time routing is much less studied. Moreover, the existing work that do consider data delivery delay in wireless routing either only try to minimize average path delay without ensuring probabilistic delay bounds [126, 43, 54, 74, 129], or they do not address the challenges that delay uncertainties pose to the task of quantifying probabilistic path delays and the task of addressing instability of delay-adaptive routing [60, 91]. Therefore, how to enable real-time routing in the presence of dynamic, uncertain link/path delays remains an important open problem for real-time wireless networked sensing and control.

Towards enabling routing with probabilistic delay bounds in WSC networks, we propose the *Multi-Timescale Adaptation (MTA)* routing protocol that addresses the aforementioned challenges of dynamic, uncertain link/path delays in real-time routing. In MTA, nodes leverage the different timescales of dynamics to accurately estimate probabilistic path delay bounds in an agile manner and to adapt spatiotemporal data flow control at the same timescales of the dynamics themselves. More specifically, we make the following contributions:

- For accurate, agile estimation of probabilistic path delay bounds, we decompose contributors to path delay uncertainties into two factors: dynamic per-packet transmission time (which we refer to as *packet-time* hereafter) and dynamic queueing along paths. Through detailed experimental analysis, we find that, given a network condition, the distribution of packet-time is quite stable despite the quick variation of instantaneous packet-time. This enables each node to accurately estimate the mean and variance of packet-time from itself to the next-hop along a path. We also observe that the packet-time for different packet transmissions, whether from the same node or from different nodes, are uncorrelated; this enables each node to compute, for a given time instant, the variance of the path delay from itself to the destination node as the sum of the variances of the packet-times for all the packets queued along the path at that instant. Based on these observations, we develop a multi-timescale approach, denoted by *multi-timescale estimation (MTE)*, to accurately estimate the highly-varying mean and variance of path delay by accurately estimating the mean and variance of packet-time and by adapting to fast-varying queueing in an accurate, agile manner. Using the mean and variance of path delay, we evaluate different methods of upper-bounding quantiles, and we identify Chebyshev Inequality as an effective basis for computing probabilistic delay bounds in constant time.

- For enabling adaptivity while addressing instability and low-performance in real-time routing, we propose the *Multi-Timescale Adaptation (MTA)* routing protocol: to facilitate the aforementioned multi-timescale estimation (MTE) and to avoid detrimental instability while ensuring data delivery performance during adaptation, a directed-acyclic-graph (DAG) is maintained at lower frequencies based on the relatively slow-varying link property ETX (i.e., expected number of transmissions taken to successfully deliver a packet), which reflects network throughput, data delivery reliability, and the overall trend of data delivery delay [34, 38]; at higher frequencies and based on the MTE method, the data flow within the DAG is controlled on a per-packet basis to minimize ETX and to ensure packet delivery within the required probabilistic delay bound. By ensuring overall stability and performance while addressing short-term dynamics at the same time, MTA enables efficient, real-time routing in the presence of complex dynamics and uncertainties.

- We implement MTE and MTA in TinyOS, and we address the challenges of limited memory, limited CPU capability, and the lack of real-time operation support in TinyOS. Besides the running MTA protocol, these implementation strategies may well be of interest to real-time routing in general.

- We evaluate the performance of MTA and and other related work in the high-fidelity sensor network testbeds NetEye [11] and Indriya [2]. We find that MTA significantly outperforms existing protocols, e.g., improving deadline success ratio by 89% and reducing transmission cost by a factor of 9.7.

The rest of the chapter is organized as follows. We briefly introduce some preliminaries in Section 2. We present the MTE method and the MTA protocol in Sections 3 and 4 respectively, and then we present the measurement study in Section 5. Then we evaluate our design with sparse networks in Section 4. We discuss related work in Section 5, and we make concluding remarks in Section 6.

2 Preliminaries

Our study leverages two publicly available wireless sensor network testbeds NetEye [11] and Indriya [2]. In what follows, we briefly introduce the two testbeds and the traffic patterns we study.

NetEye testbed. NetEye [11] is deployed in a large lab space at Wayne State University as shown in Figure 1. We use a subset of a 15×7 grid of TelosB motes in NetEye, where every two

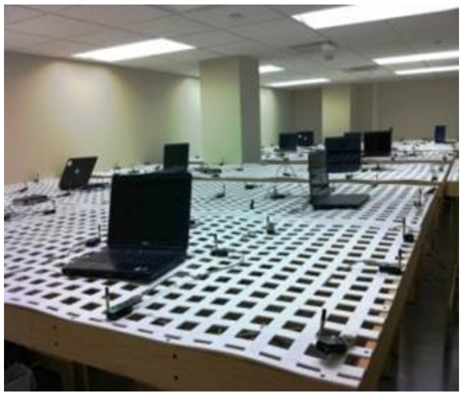


Figure 1: *NetEye* wireless sensor network testbed

closest neighboring motes are separated by 2 feet. The subset of the grid forms a random network, and it is generated by removing each mote of the 15×7 grid with probability 0.2.

Each of these TelosB motes is equipped with a 3dB signal attenuator and a 2.45GHz monopole antenna. In our measurement study, we set the radio transmission power to be -25dBm (i.e., power level 3 in TinyOS) such that multihop networks can be created and the link reliability is over 90% for links up to 6 feet long. For the transmission power of -25dBm, Figure 2 shows the boxplot of packet delivery ratio (PDR) for links of different length, and Figure 3 shows the histogram

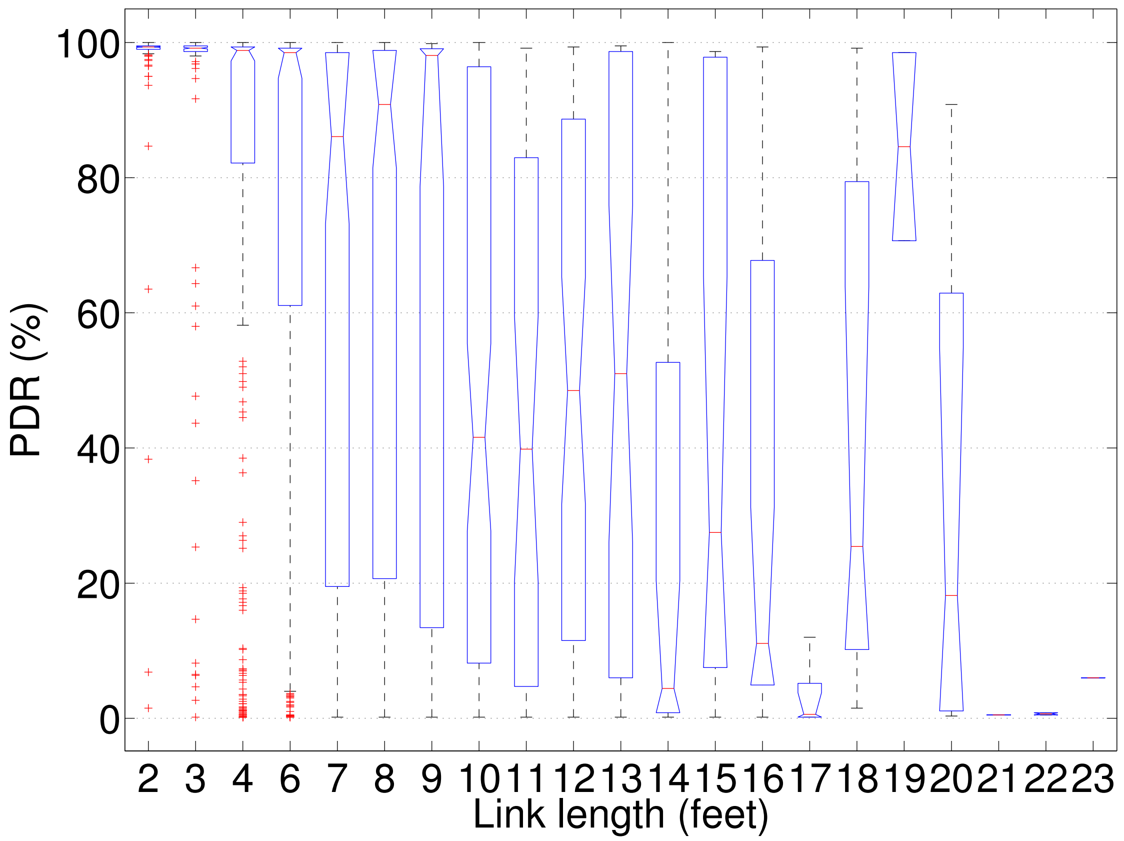


Figure 2: PDR vs. link length in NetEye when transmission power is -25dBm

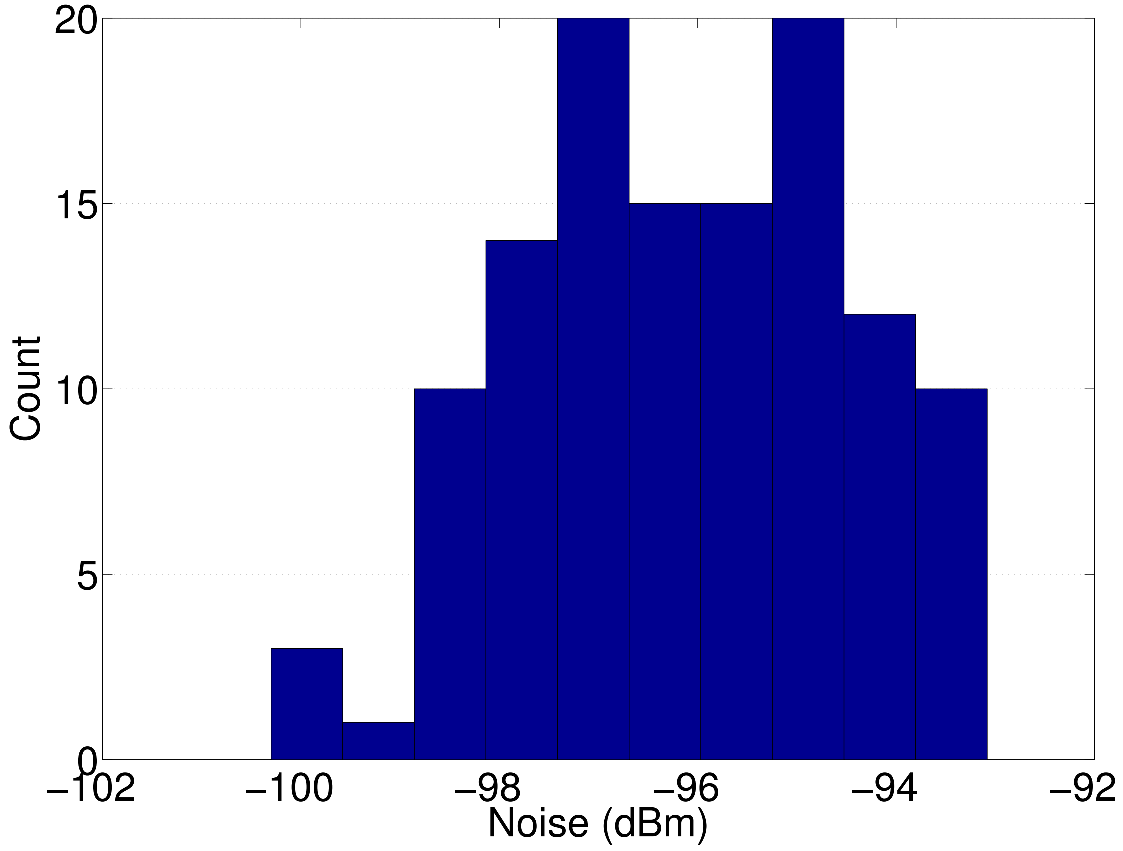


Figure 3: Histogram of background noise power in NetEye

of background noise power in NetEye. We see that there is a high degree of variability in PDR for links of equal length and in background noise power. Thus the testbed reflects non-uniform network settings as seen in practice. Given the high availability and high fidelity of NetEye, we mainly use NetEye in our measurement study, but we verify key observations using the Indriya testbed too.

Indriya testbed. Indriya [2] is deployed at three floors of the School of Computing at the National University of Singapore as shown in Figure 4. The Testbed consists of 127 TelosB sensor motes,

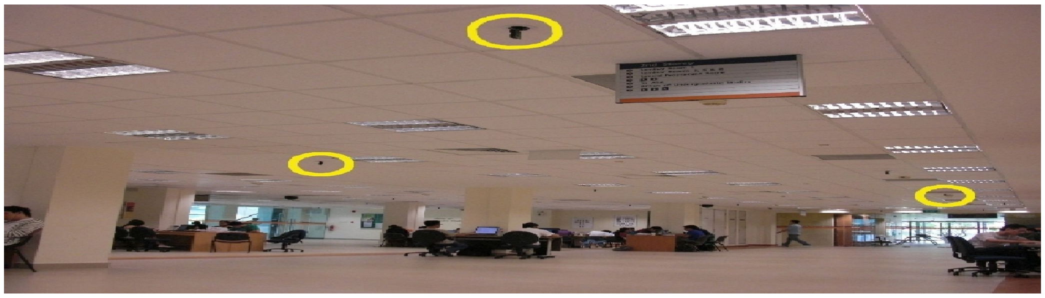


Figure 4: *Indriya* wireless sensor network testbed

and Figure 5 shows a typical three-dimensional connectivity graph between the motes.

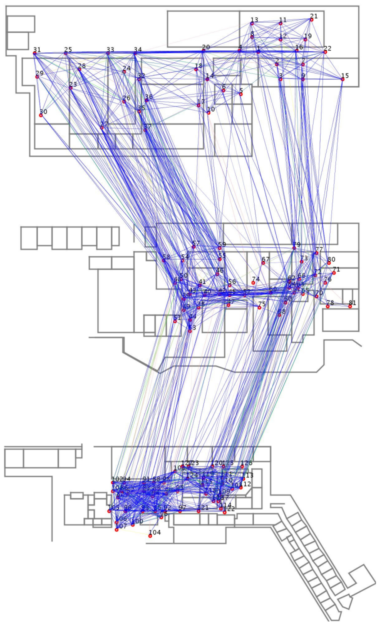


Figure 5: A typical connectivity graph for Indriya

Our measurement study uses all of its 127 TelosB motes, and we use a transmission power of

-10dBm (i.e., power level 11 in TinyOS) to generate a well-connected multi-hop network where the link reliability is over 90% for links up to 20 feet long. For the transmission power of -10dBm, Figure 6 shows the boxplot of PDR for links of different length, and Figure 7 shows the histogram of background noise power in Indriya. We see that there is a high degree of variability in link

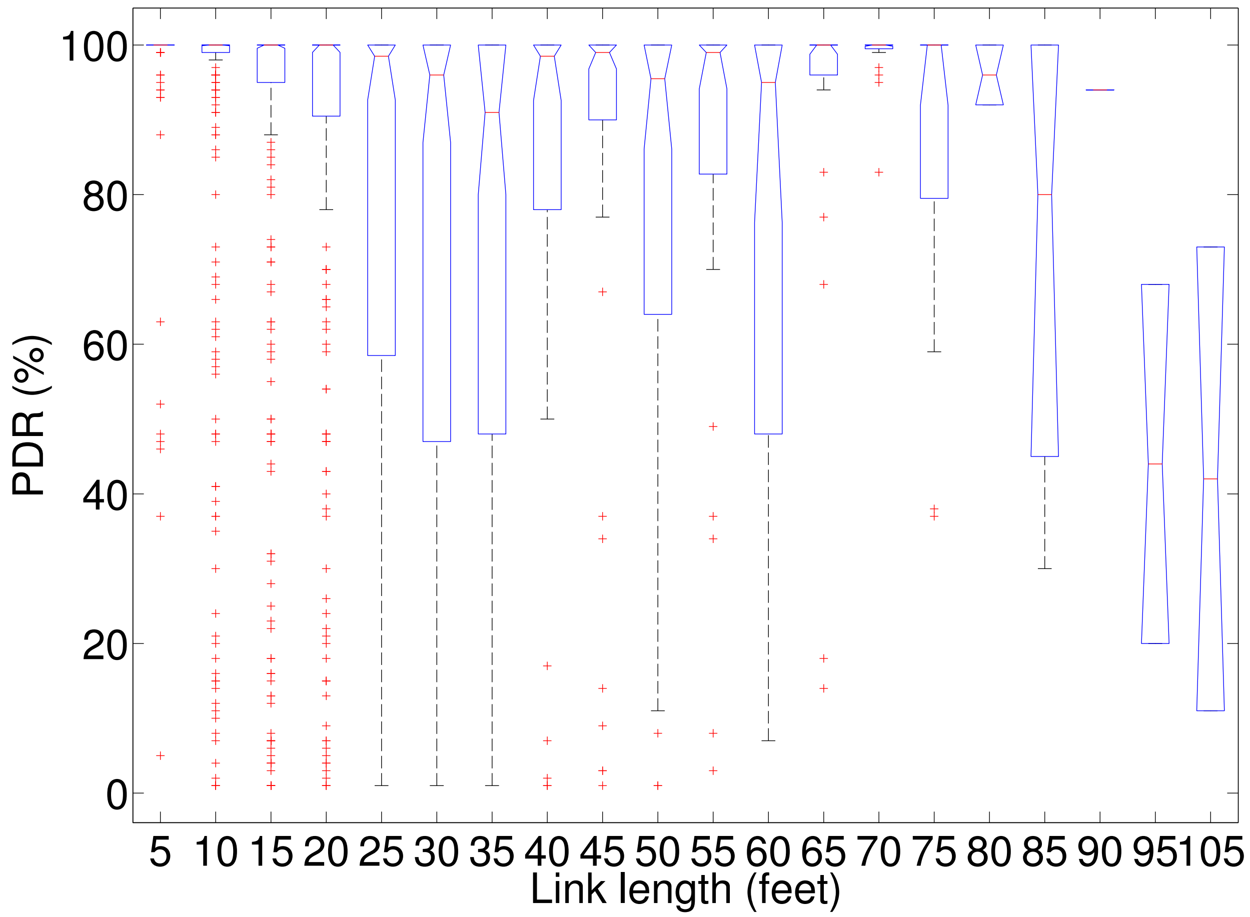


Figure 6: PDR vs. link length in Indriya when transmission power is -10 dBm

PDRs and background noise power too, which reflects real-world, non-uniform settings.

Traffic pattern. Using the two testbeds, we study both periodic and event traffic patterns.

For periodic traffic, we study three types of them based on the amount of queueing they introduce when the default TinyOS routing protocol CTP [49] is used: 1) light traffic: no queueing in network; 2) medium traffic: moderate queueing in network but with very rare queue overflow;

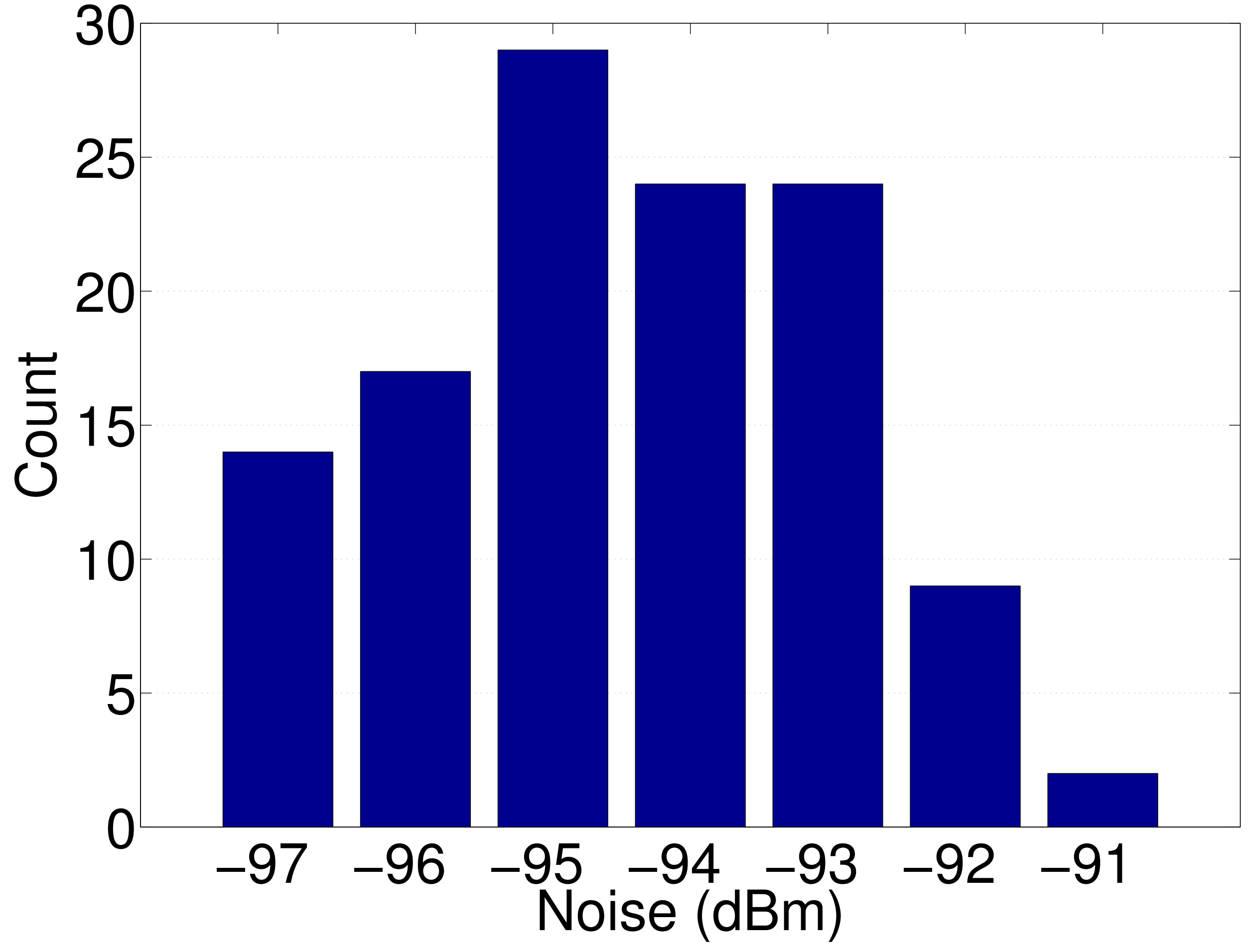


Figure 7: Histogram of background noise power in Indriya

3) heavy traffic: severe queueing in network and with frequent queue overflow. To this end, we select one node as the sink and another 10 nodes as traffic sources; the sink and the sources are nearly at opposite positions in both testbeds to create as many routing hops as possible. In NetEye, more specifically, mote 15 is the sink, and motes 61, 62, 63, 64, 76, 77, 79, 91, 92, and 93 are the sources, with each source generating a packet every 1,000ms, 400ms, and 75ms for light, medium, and heavy traffic respectively. To verify observations from NetEye and given that the medium traffic is the common case in WSC networks, we also study medium traffic scenario in Indriya as follows: mote 105 at the third floor is the sink, and motes $1 \ldots 10$ at the first floor are the sources, with each source generating a packet every 600ms.

For event traffic, we use a publicly available event traffic trace for a field sensor network deployment [131, 1] to evaluate the performance of different protocols. The traffic trace corresponds to the packets generated in the 7×7 grid of a field mote network when a vehicle passes across the middle of the network. When the vehicle passes by, each mote except for the base station detects the vehicle and generates two packets, which correspond to the start and the end of the event detection respectively and are separated by 5-6 seconds on average. Overall, 96 packets are generated each time the vehicle passes by. The cumulative distribution of the number of packets generated during the event is shown in Figure 8. (Interested readers can find the detailed description of the traffic trace in [1].) In NetEye, the 7×7 subgrid in the trace data corresponds to the 49 motes that are the farthest from the sink mote 15. In Indriya, the 7×7 subgrid is mapped to motes $1 \ldots 49$.

3 Multi-timescale estimation of path delays

In what follows, we first discuss the need for considering probabilistic path delays in real-time routing, then we present our approach to accurate, agile estimation of probabilistic path delay bounds.

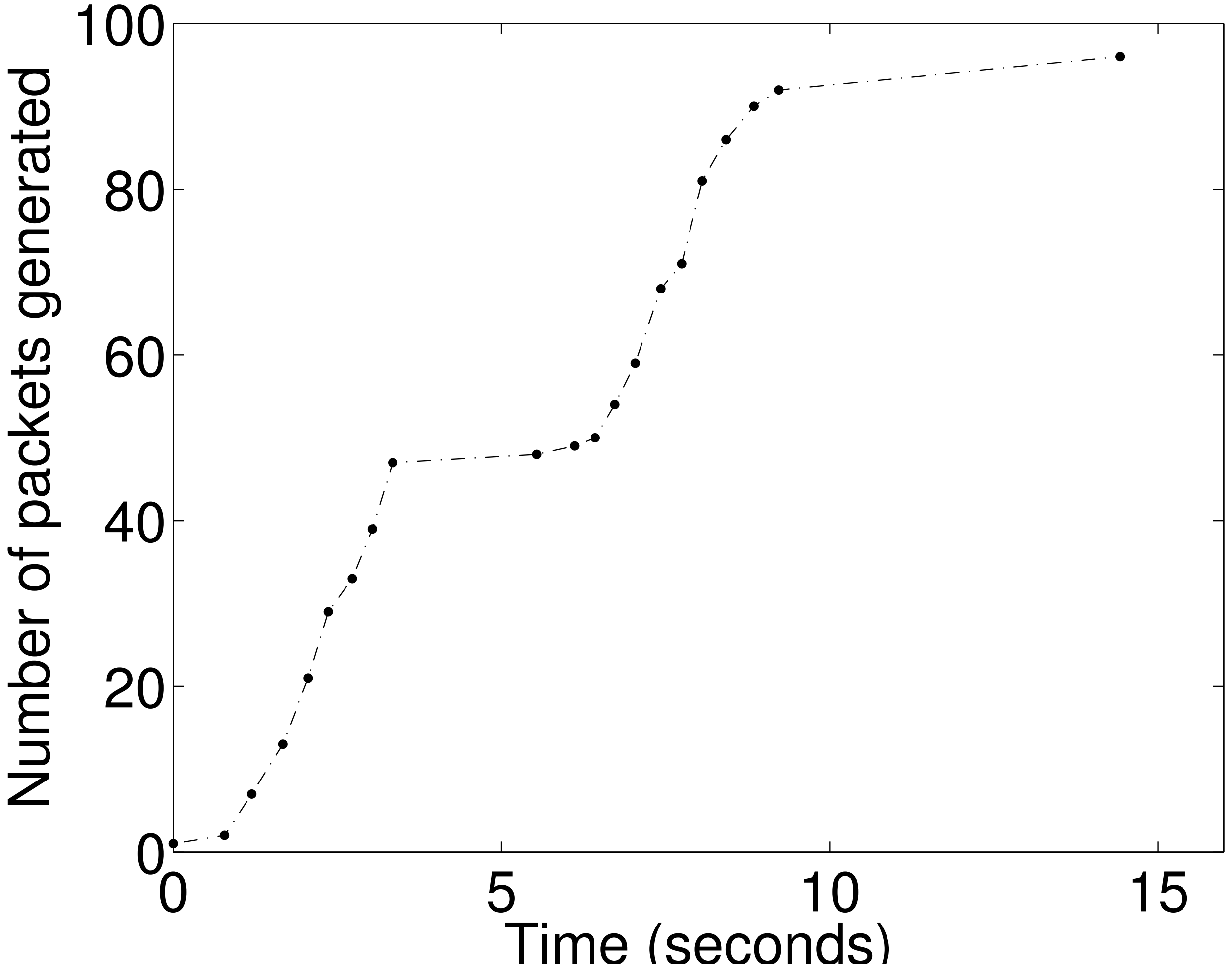


Figure 8: The distribution of packets generated in the event traffic trace

3.1 Why probabilistic path delay?

Due to inherent spatiotemporal dynamics and uncertainties in wireless communication [138, 132], the success of a packet transmission is probabilistic instead of deterministic. Thus, data delivery delays across links/paths are probabilistic in nature in wireless sensing and control (WSC) networks, and this is the case even if TDMA- instead of CSMA-based scheduling is used. Accordingly, application requirements on data delivery timeliness tend to be probabilistic in WSC networks. Based on its requirements on stability region and settling time, for instance, a control application can specify a maximum tolerable delay; by treating a deadline miss as the loss of the corresponding signal sample and by analyzing the corresponding impact on system estimation and control, the application can also specify a maximum tolerable probability of deadline miss [117, 55]. Therefore, we consider application requirements on the maximum tolerable delay and the least probability of deadline success in this study.

Most existing delay-aware wireless routing protocols [126, 43, 54, 74, 129] consider mean delay instead of delay quantiles. Yet smaller mean delay does not ensure smaller delay quantiles. For the medium traffic setting in NetEye as discussed in Section 2, for instance, Figure 9 shows the non-negligible probability that, when the mean delay across a link ℓ_0 is less than that of another link ℓ_1, the q-quantile of ℓ_0's delay is greater than that of ℓ_1. Therefore, routing based on mean delay does not ensure the use of paths with small probabilistic delays, thus leading to deadline miss and the reduction of network real-time capacity (i.e., amount of data that can be delivered within a certain deadline).

Traditional real-time system design usually considers worst cases, and, in real-time wireless networking, worst-case path delay has also been considered [91]. Yet the distribution of link/path delays tend to be heavy-tailed such that the maximum delay is significantly larger than the common-case delays. For the medium traffic setting in NetEye, for instance, Figure 10 shows the empirical cumulative distribution function (CDF) of the per-packet transmission time (which we refer to as *packet-time*) for successfully delivering a packet across a typical link in NetEye.[1]

[1]The packet-time includes time spent for channel access contention and possibly packet retransmission.

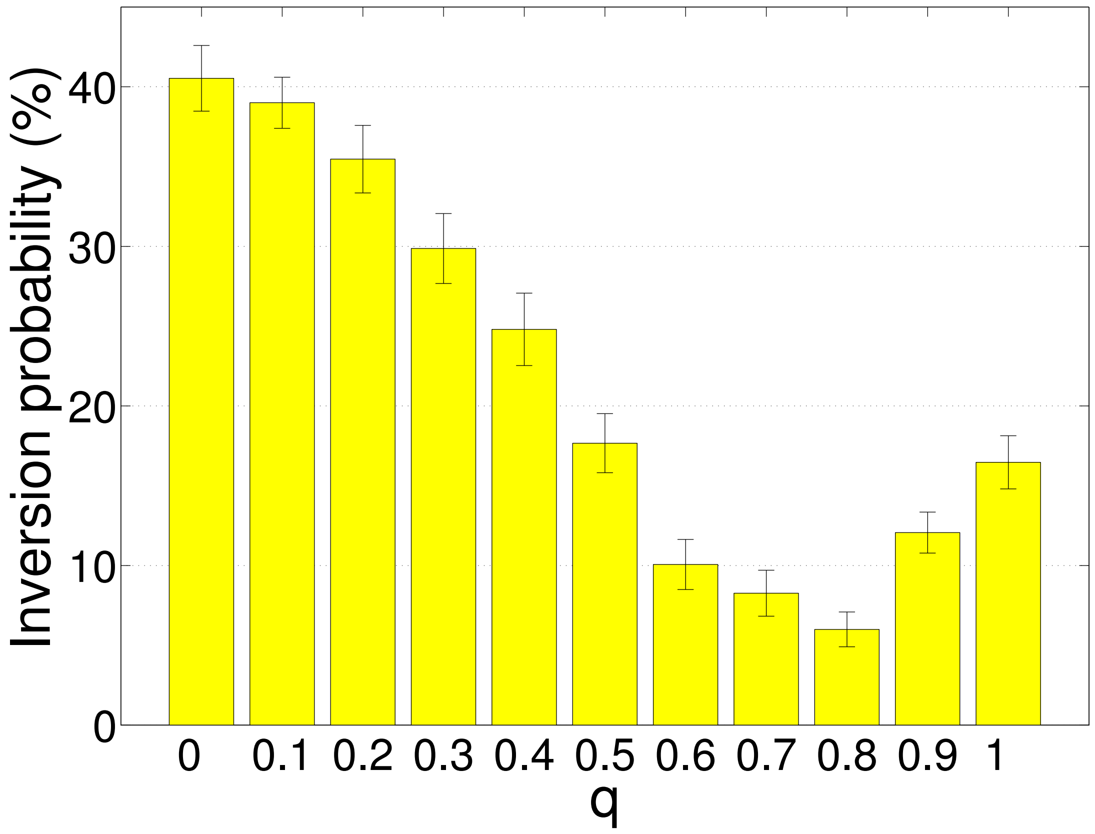


Figure 9: Goodness inversion probability and its 99% confidence interval

The median, 90-percentile, and maximum of the packet-time are 23, 86, and 728 milliseconds

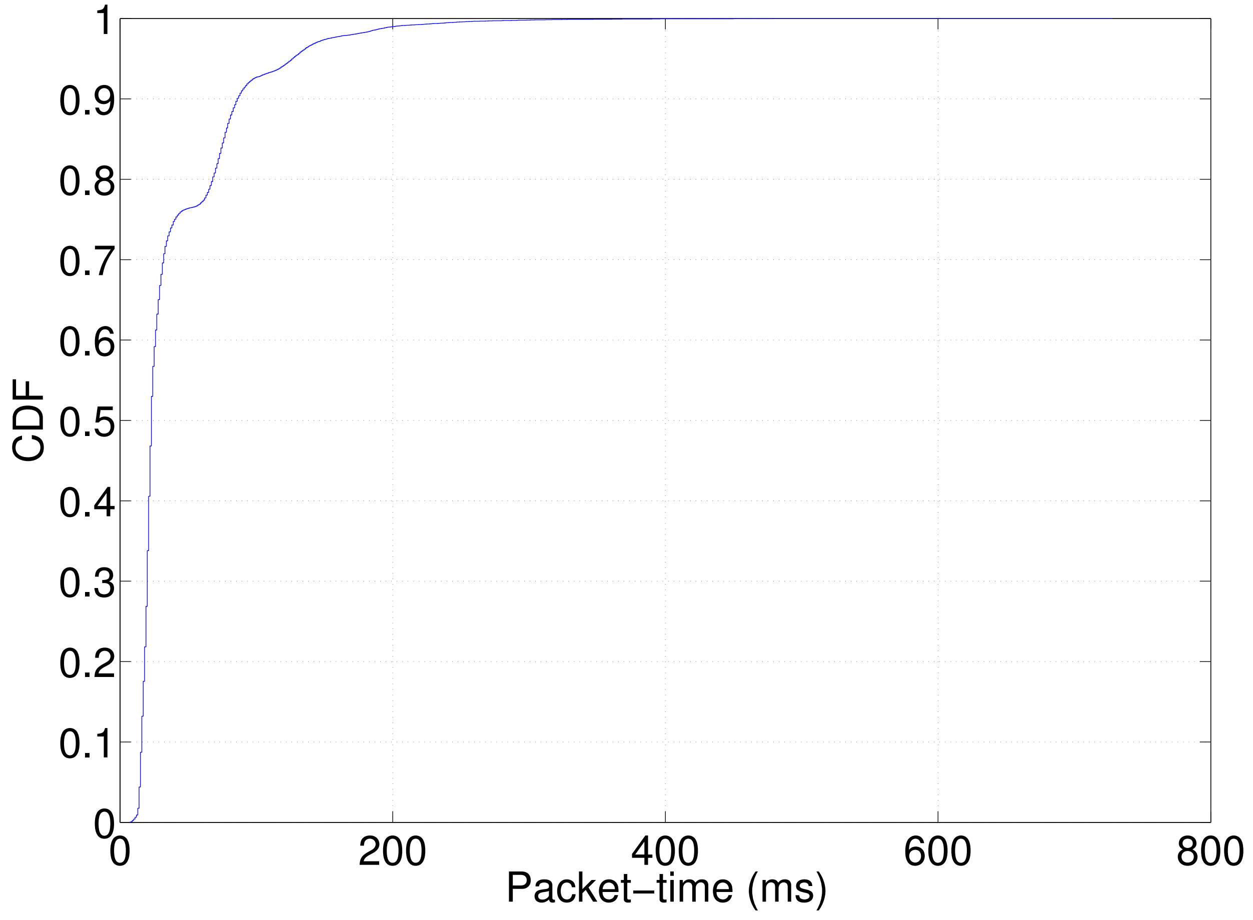


Figure 10: CDF of the packet-time across a typical link

respectively. Therefore, focusing on maximum link/path delays will not be able to utilize many paths that could have been used to support WSC applications where probabilistic delay guarantees suffice, thus reducing the application-usable real-time capacity of the network.

Therefore, considering probabilistic path delays instead of mean or maximum delays reduces deadline misses and enables effective utilization of application-usable real-time capacity of WSC networks.

3.2 Agile, accurate estimation of probabilistic path delay bound

To enable routing with probabilistic delay guarantees, a basic task is to quantify probabilistic delays along paths. Given a path $P = v_0, v_1, \ldots, v_n, v_{n+1}$ from v_0 to v_{n+1} as shown in Figure 11 and

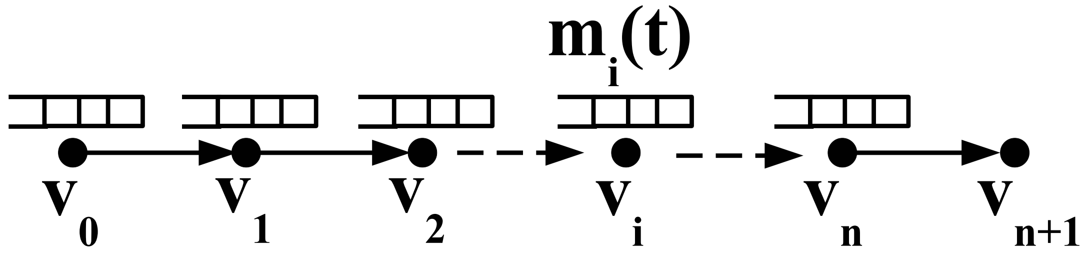


Figure 11: An example path P

assuming that the number of packets queued at node $v_i (i = 0 \ldots n)$ is $m_i(t)$ at time t and that the packet transmission scheduling algorithm at each node is first-come-first-serve (FCFS), the instantaneous delay along path P at time t, denoted by $d_P(t)$, is the delay that a packet arriving at v_0 at time t experiences in reaching the destination node v_{n+1} assuming that the number of queued packets at each node does not change while the packet is being delivered to v_{n+1}. Therefore,

$$d_P(t) = \sum_{i=0}^{n} \sum_{j=1}^{m_i(t)+1} d_i^j(t), \tag{2.1}$$

where $d_i^j(t)$ is the time taken for v_i to deliver its j-th queued packet to v_{i+1} at time t.

Challenges of highly-varying delay distribution. Given the distributions of $d_i^j(t)(i = 0 \ldots n, j = 1 \ldots m_i(t) + 1)$, it is NP-hard to compute the distribution of $d_P(t)$ [51]. Since nodes of WSC networks tend to be resource-constrained (e.g., in CPU and memory), it is infeasible to directly compute the distribution of $d_P(t)$ in general. One alternative is to first sample $d_P(t)$ and then use non-parametric approaches (e.g., the P^2 algorithm) to estimate delay quantiles based on these delay samples [62, 82]. Nonetheless, non-parametric quantile estimation usually converges slowly. For instance, Figure 12 shows, as path delay samples are collected for a typical five-hop

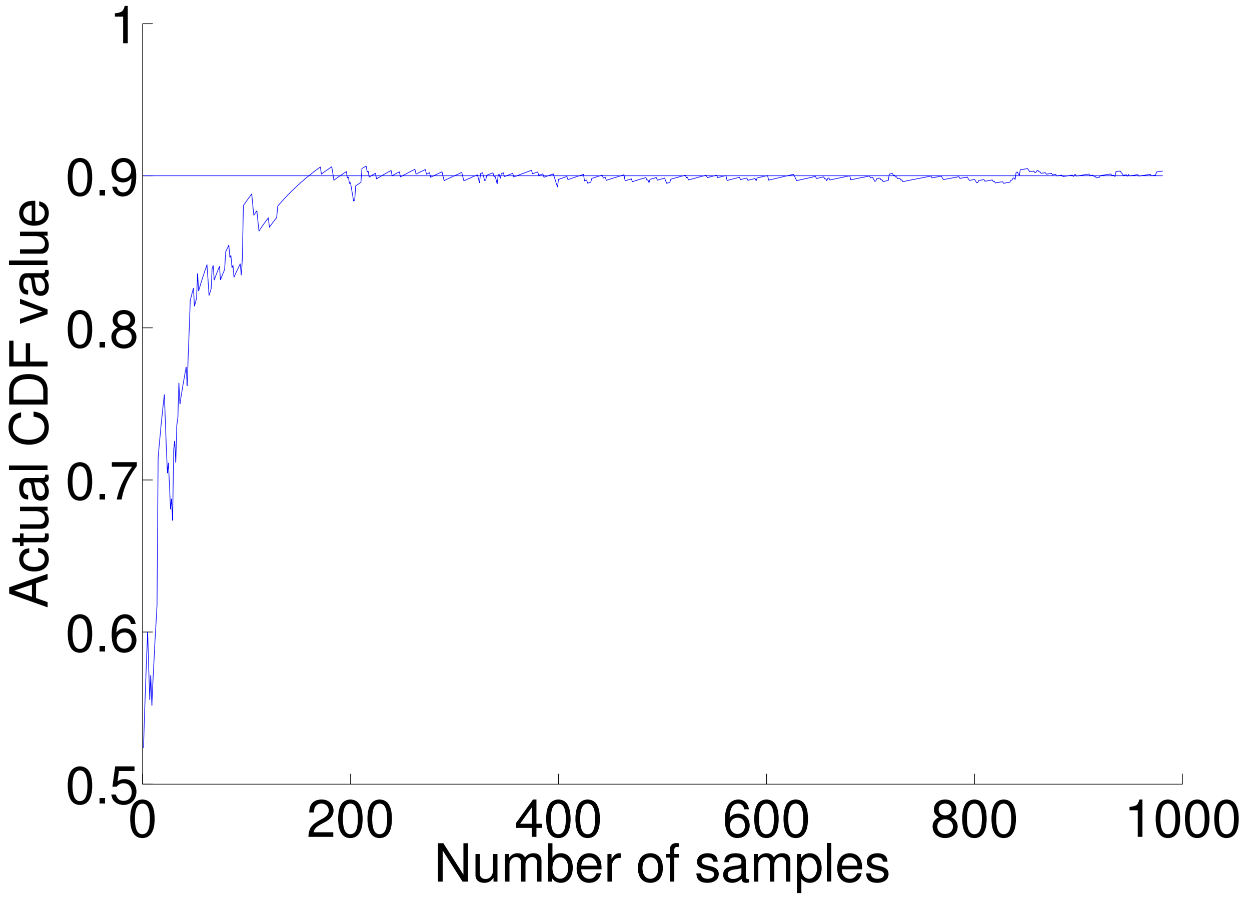


Figure 12: Actual CDF value of the estimated 90 percentile via the P^2 algorithm

path in the NetEye medium traffic scenario, the time series of the actual cumulative distribution function (CDF) values of the estimated 90 percentiles using the P^2 algorithm. We see that it takes more than 200 samples for the estimation to converge, and this observation holds in general [82]. Yet the distribution of path delays varies at a much shorter timescale than the sample size required for non-parametric quantile estimation to converge. As can be seen from (2.1), in particular, the distribution of path delays vary with the queueing levels (i.e., $m_i(t)$) along paths, and network queueing can vary quickly over time.[2] For the NetEye medium traffic scenario, for instance, Figures 13 and 14 show the histogram of the coherence window size for node and path queueing

[2]As we will discuss shortly, the distribution of packet-time (i.e., $d_i^j(t)$) is quite stable over time, and we leverage the stability of packet-time distribution in designing our multi-timescale estimation (MTE) method.

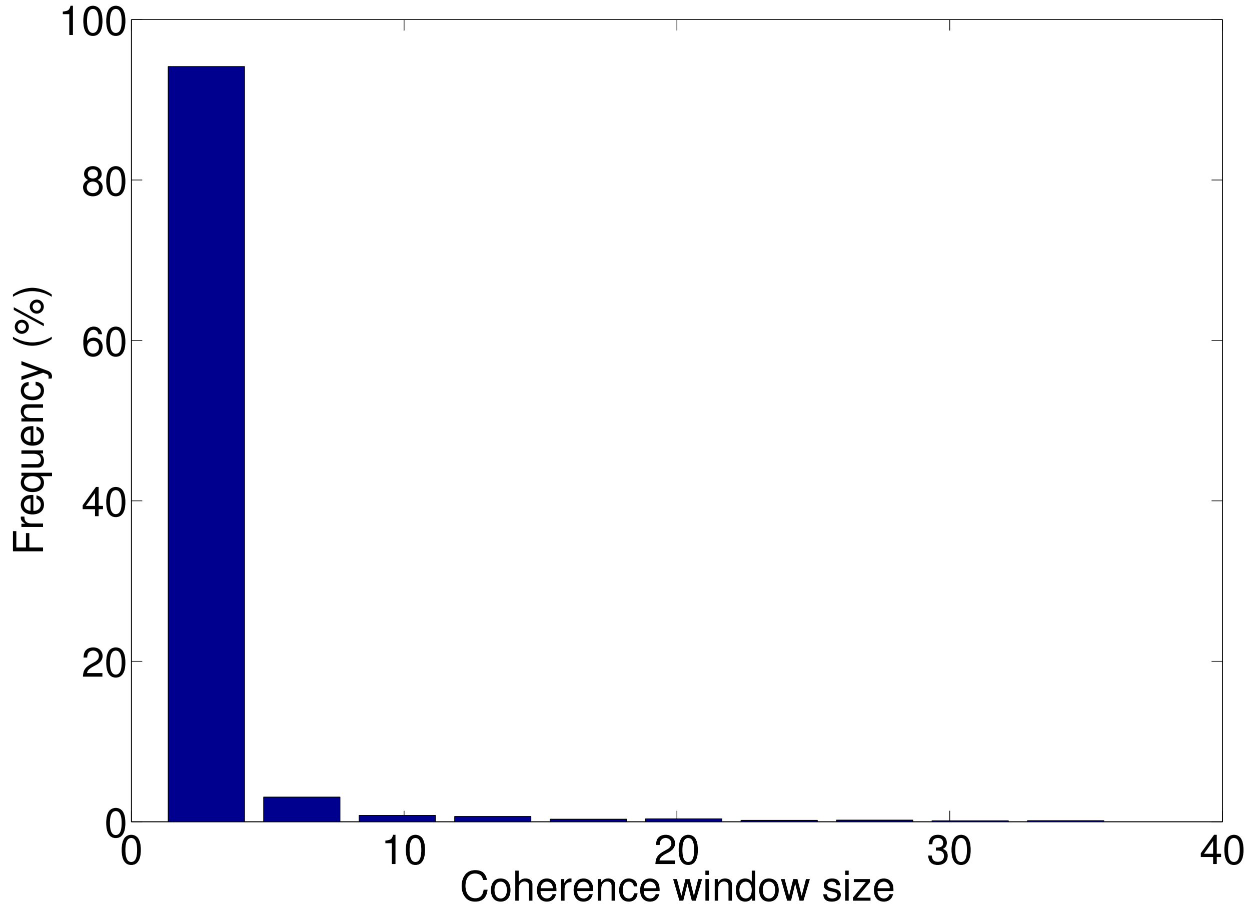


Figure 13: Histogram of the coherence window size for node queueing level

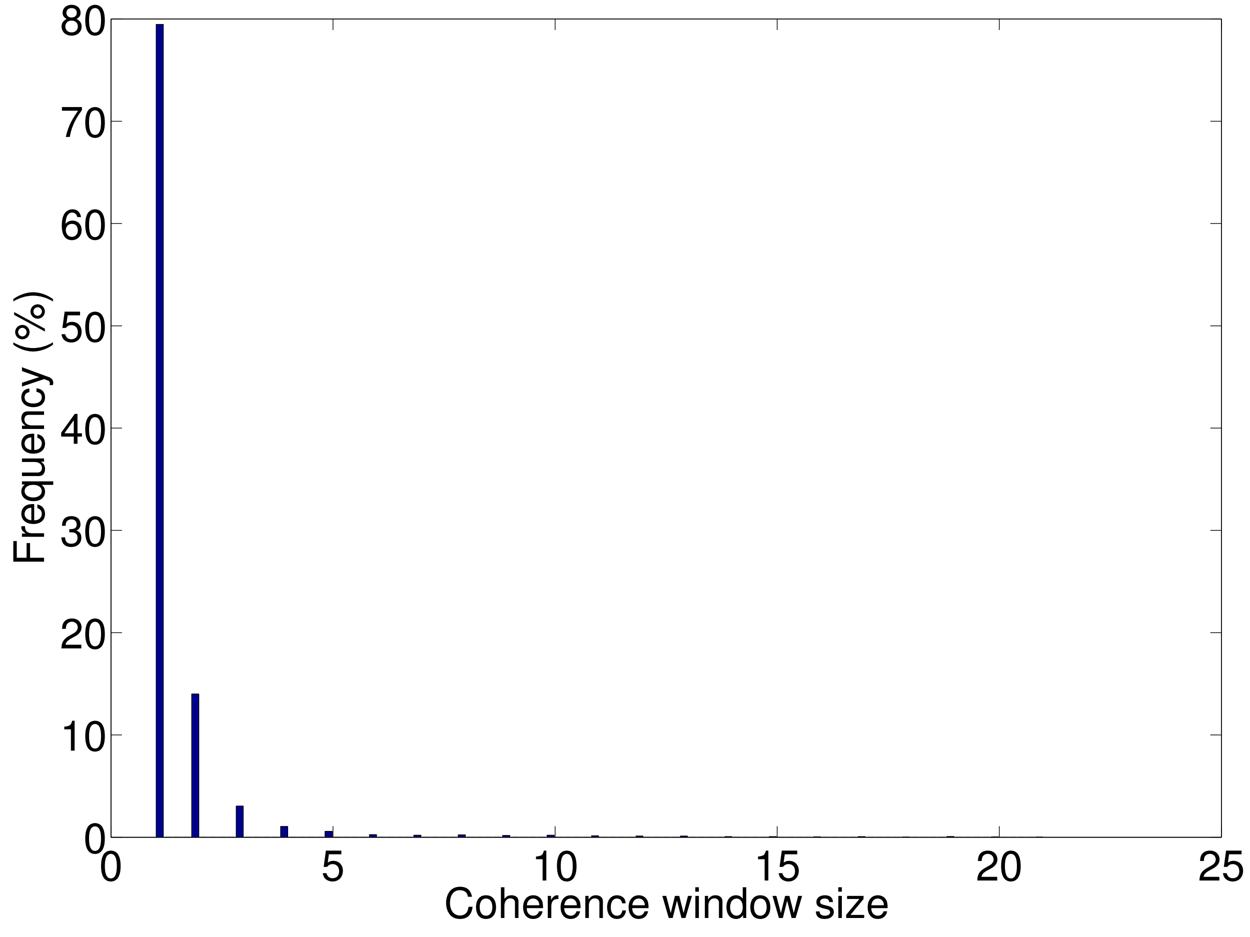


Figure 14: Histogram of coherence window size for path queueing level

respectively, where each coherence window of a time series is a maximal consecutive segment of the time series where all the sample values are the same. We see that the coherence window size is two orders of magnitude less than the sample size required for non-parametric quantile estimation to converge. Therefore, the highly-varying nature of path delay distribution makes non-parametric quantile estimation unable to accurately estimate instantaneous path delay quantiles in an agile manner; that is, the distribution of the path delay changes before the estimation converges to an accurate value.

To circumvent the computational complexity and the inability of accurate, agile estimation of exact path delay quantiles, we propose to identify upper bounds on probabilistic path delays and to use the delay bounds in identifying paths for real-time data delivery. As we will discuss later in this section , upper bounds on path delays can be derived using probability inequalities such as Chebyshev Inequality [101], and the properly identified delay bounds are still orders of magnitude less than the maximum delays, thus enabling effective utilization of network real-time capacity. Most probability inequalities use the mean and/or the standard deviation of the corresponding random variable, thus we need an accurate, agile mechanism of estimating the mean and standard deviation of path delays. Nonetheless, the sample size required for accurate estimation of path delay statistics tend to be quite large. For instance, Figure 15 shows the empirical CDF of the sample size required for estimating mean path delay at 90% accuracy and 90% confidence level. We see that the required sample size is greater than 100 in most cases and can be up to 594, which are significantly greater than the coherence window size of node/path queueing levels. In addition, node and path queueing levels tend to have low autocorrelation, not to mention staying unchanged, for time lags greater than 100 and 40 samples respectively, as shown in Figures 16 and 17. Thus, the large sample size requirement and the highly-varying nature of path delay distribution makes it impossible to accurately estimate instantaneous mean path delay in an agile manner; that is, the mean path delay changes before the estimation converges to an accurate value.

Multi-Timescale Estimation (MTE). Towards addressing the challenges of highly-varying distribution and statistics of path delay, we decompose contributors to path delay variations into two

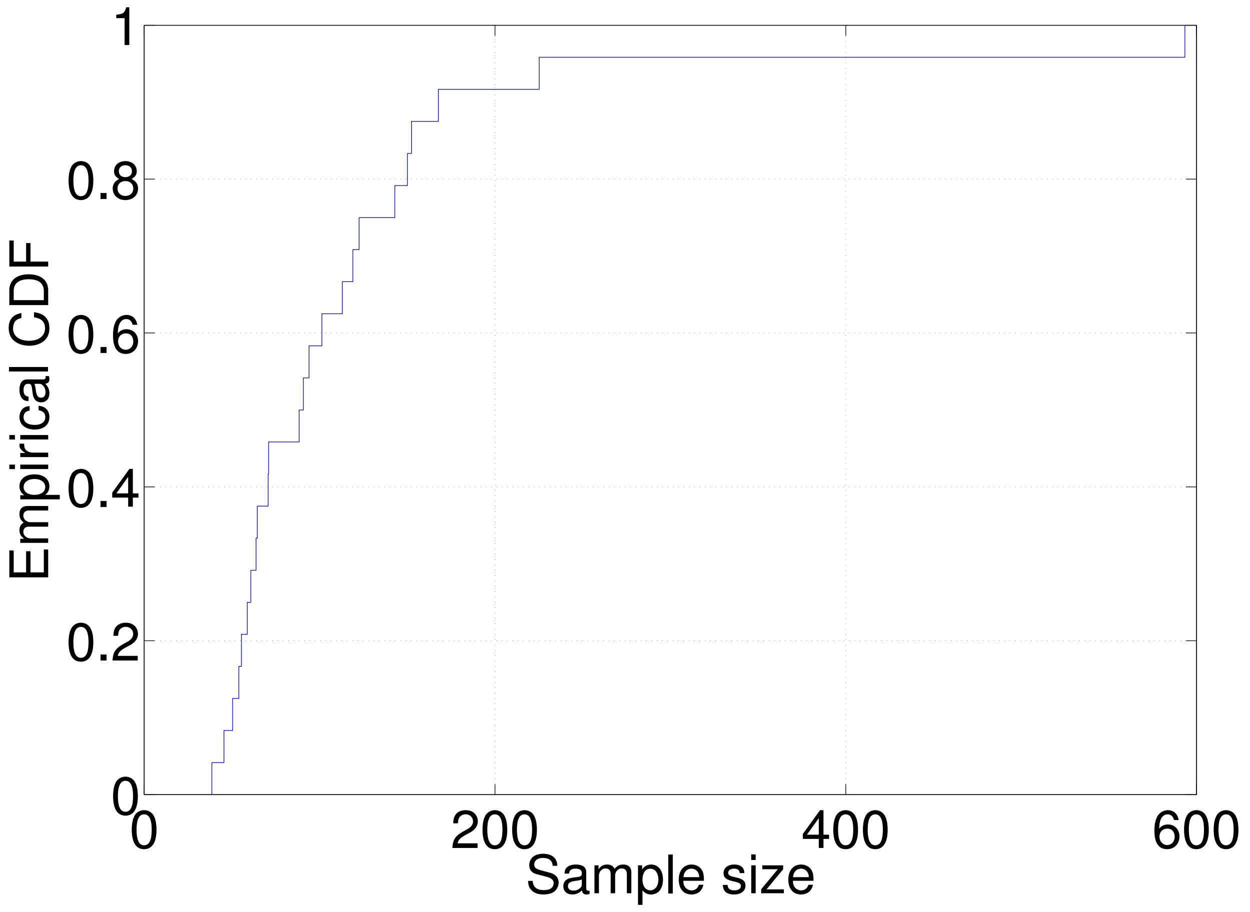


Figure 15: Empirical CDF of the sample size required to estimate mean path delay at 90% accuracy and 90% confidence level

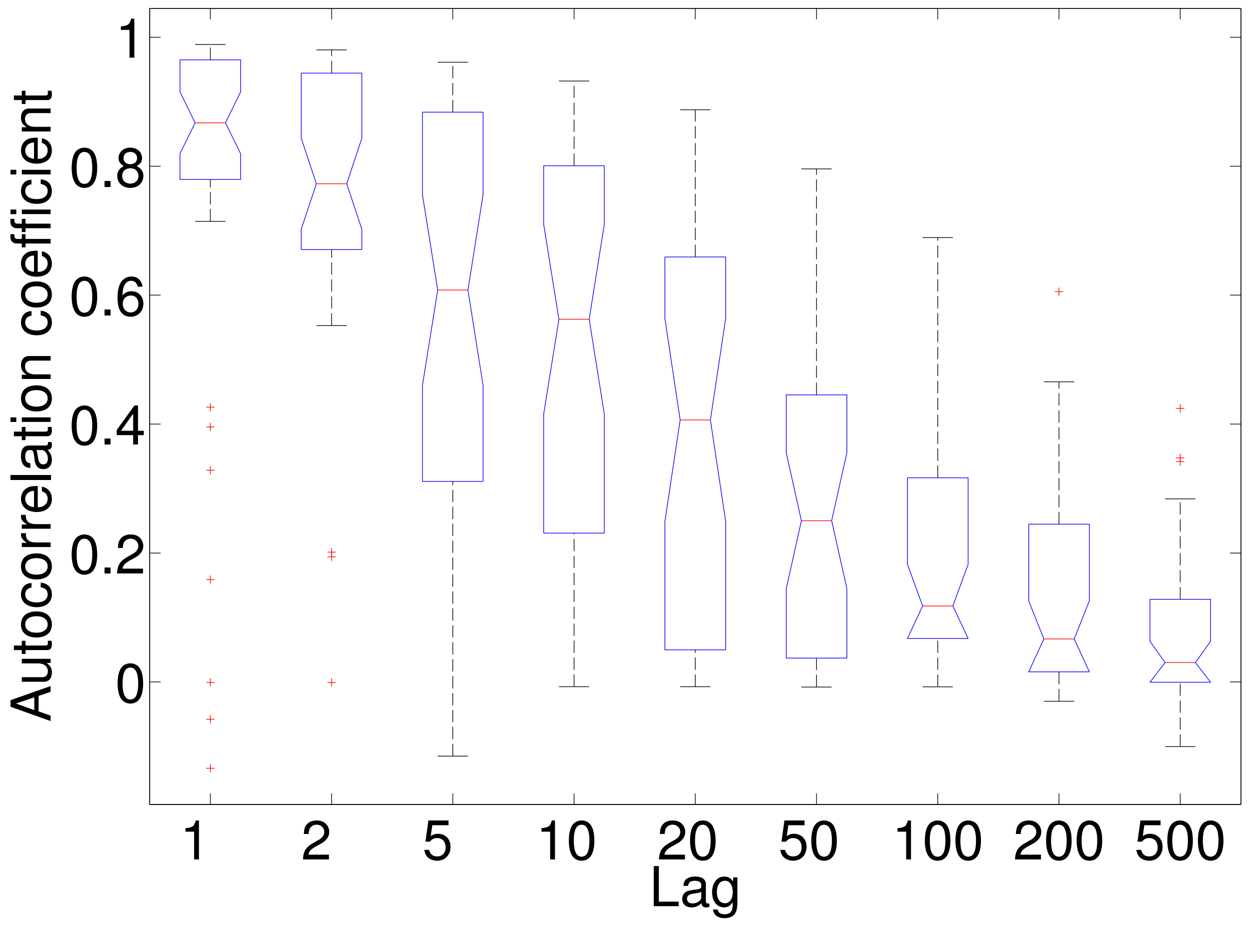


Figure 16: Autocorrelation of node queueing levels

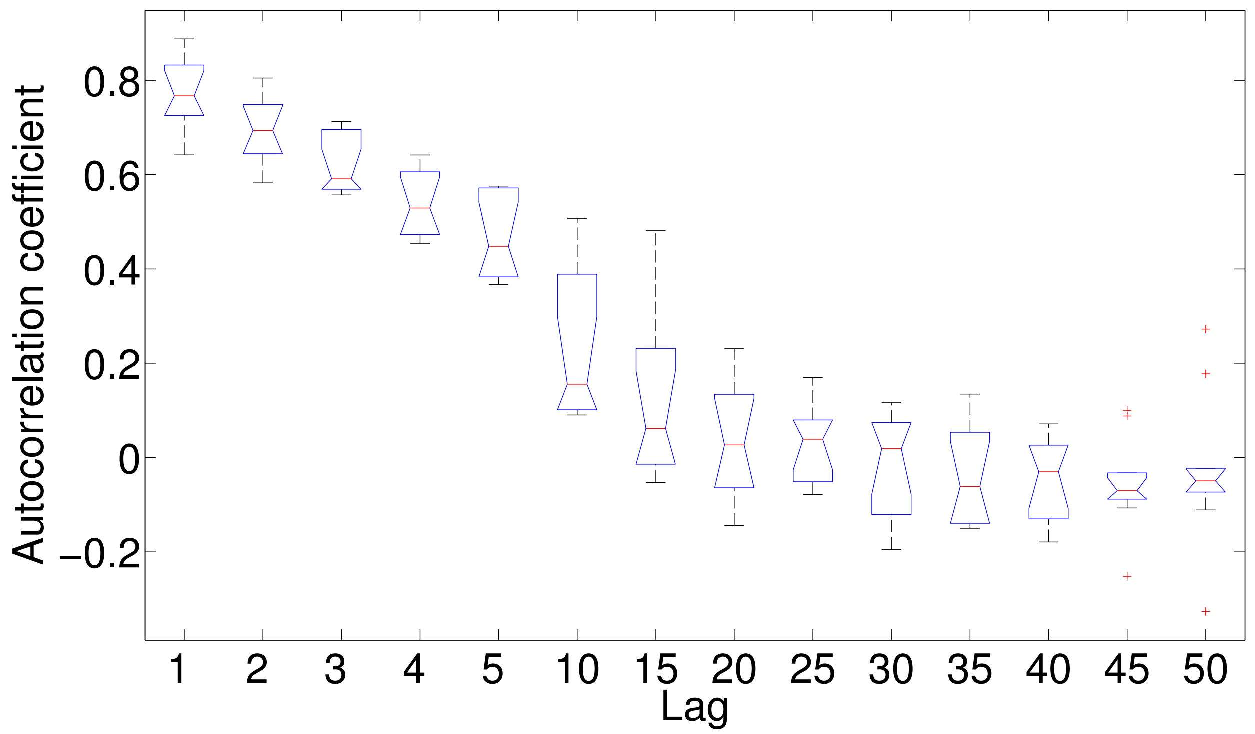


Figure 17: Autocorrelation of path queueing levels

factors: dynamic packet-time and dynamic queueing. By leveraging the different timescales at which packet-time and queueing vary, we propose the multi-timescale-estimation (MTE) method that accurately estimates the highly-varying mean and variance of path delay by 1) accurately estimating the mean and variance of packet-time at a longer timescale and 2) by adapting to fast-varying queueing at a shorter timescale. In what follows, we elaborate on the design of the MTE method.

Observation 1. *Packet-time distribution is stable.*

Through detailed experimental analysis, we find that, given a network condition, the distribution of packet-time is quite stable despite the quick variation of instantaneous packet-time. For instance, Figure 18 shows the time series of the packet-time along a typical link in the NetEye medium traffic scenario. Using the Generalized KPSS test [56], we analyze the stationarity window size of packet-time, where each stationary window of a time series is a maximal consecutive segment of the time series that is weak-sense stationary (i.e., with a constant mean and variance

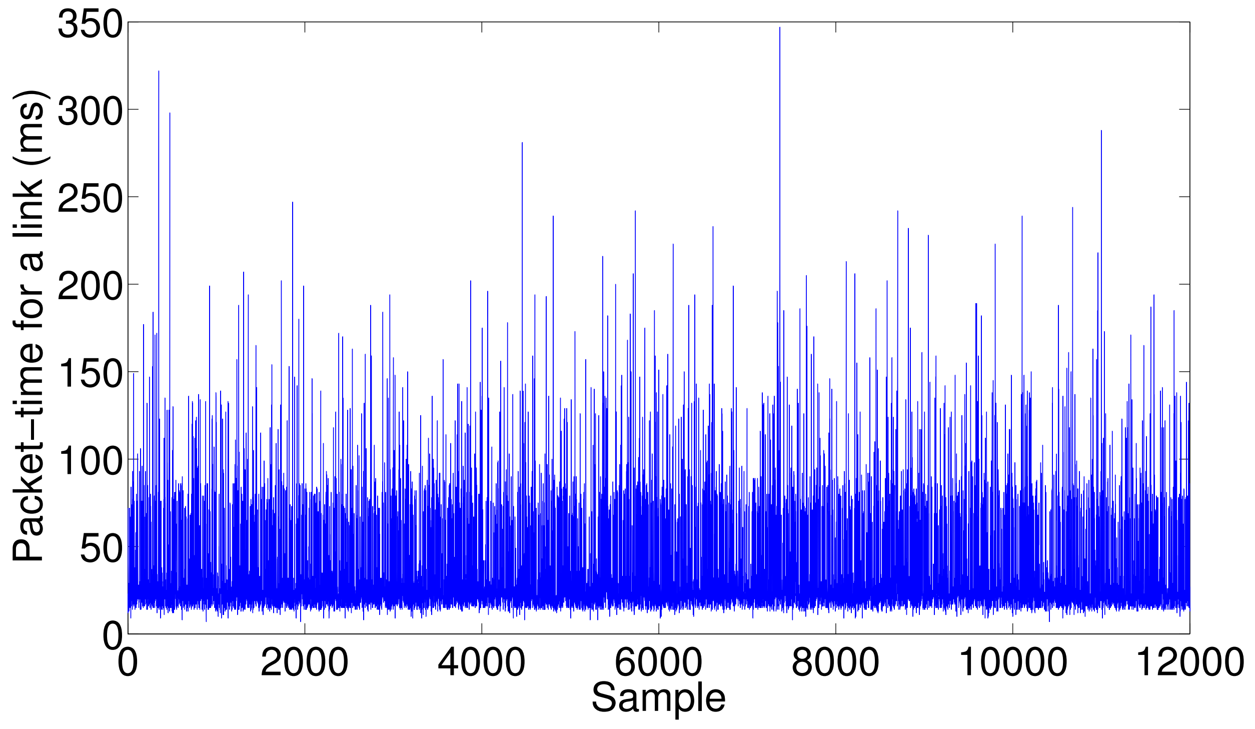


Figure 18: Time series of packet-time along a typical link

over time). Figure 19 shows the histogram of the stationarity window size for packet-time in the NetEye medium traffic scenario. The minimum and the maximum window size are 1,901 and 21,937 respectively, both of which are significantly greater than the sample size required to precisely estimate the mean and variance of packet-time. For instance, Figure 20 shows the empirical CDF of the sample size required for estimating the mean packet-time at 90% accuracy and 90% confidence level. The sample size requirement is less than 360 with over 99% probability, and the maximum sample size requirement is 600.

The stability of the packet-time distribution has two important implications:

- Given the real-time requirement on data delivery and the constrained memory size in embedded WSC networks, the stationarity window size tends to be greater than the maximum node queue size in general. This implies that, in Formula (2.1), the mean and variance for $d_i^j(t)$'s $(j = 1 \ldots m_i(t) + 1)$ are the same for a given $i (i = 0 \ldots n)$, and we denote them as

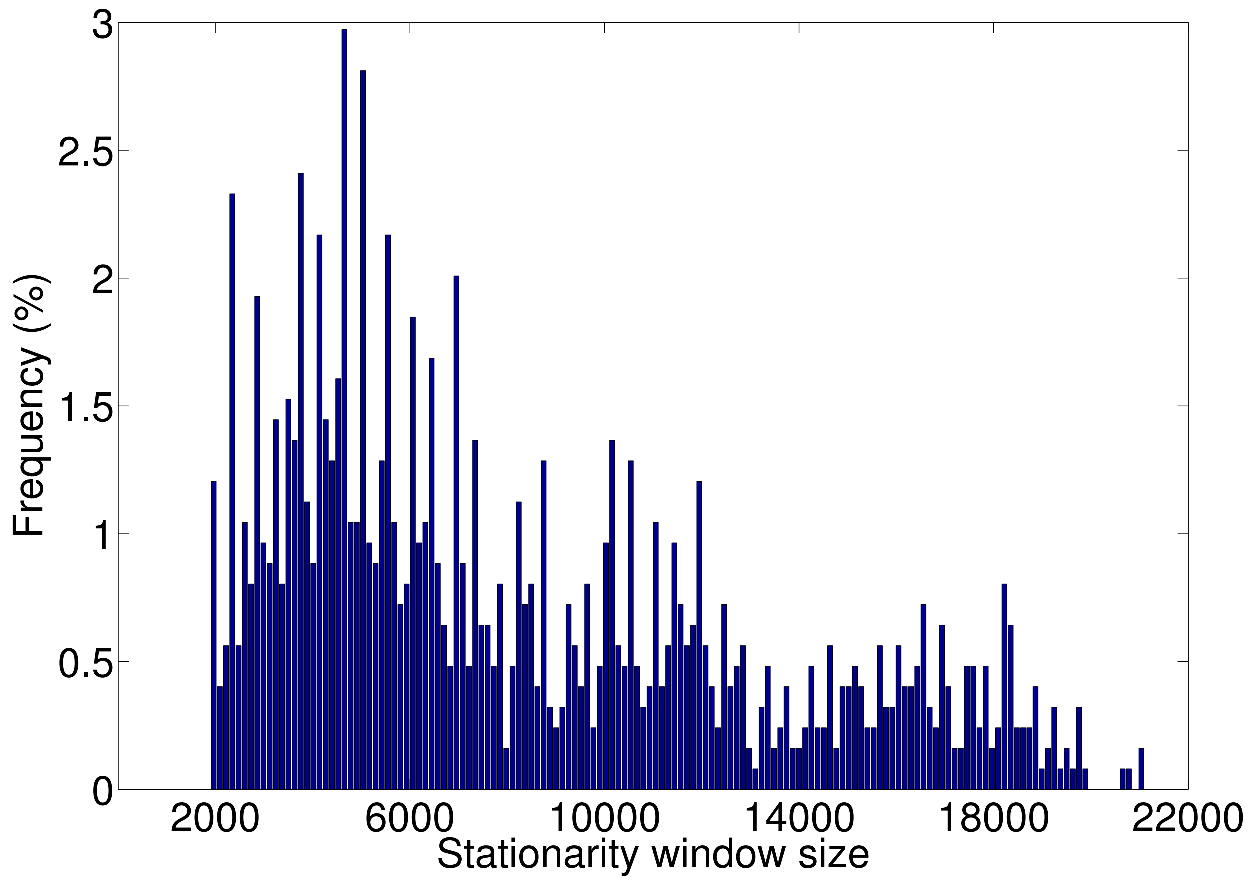


Figure 19: Histogram of stationarity window size for packet-time

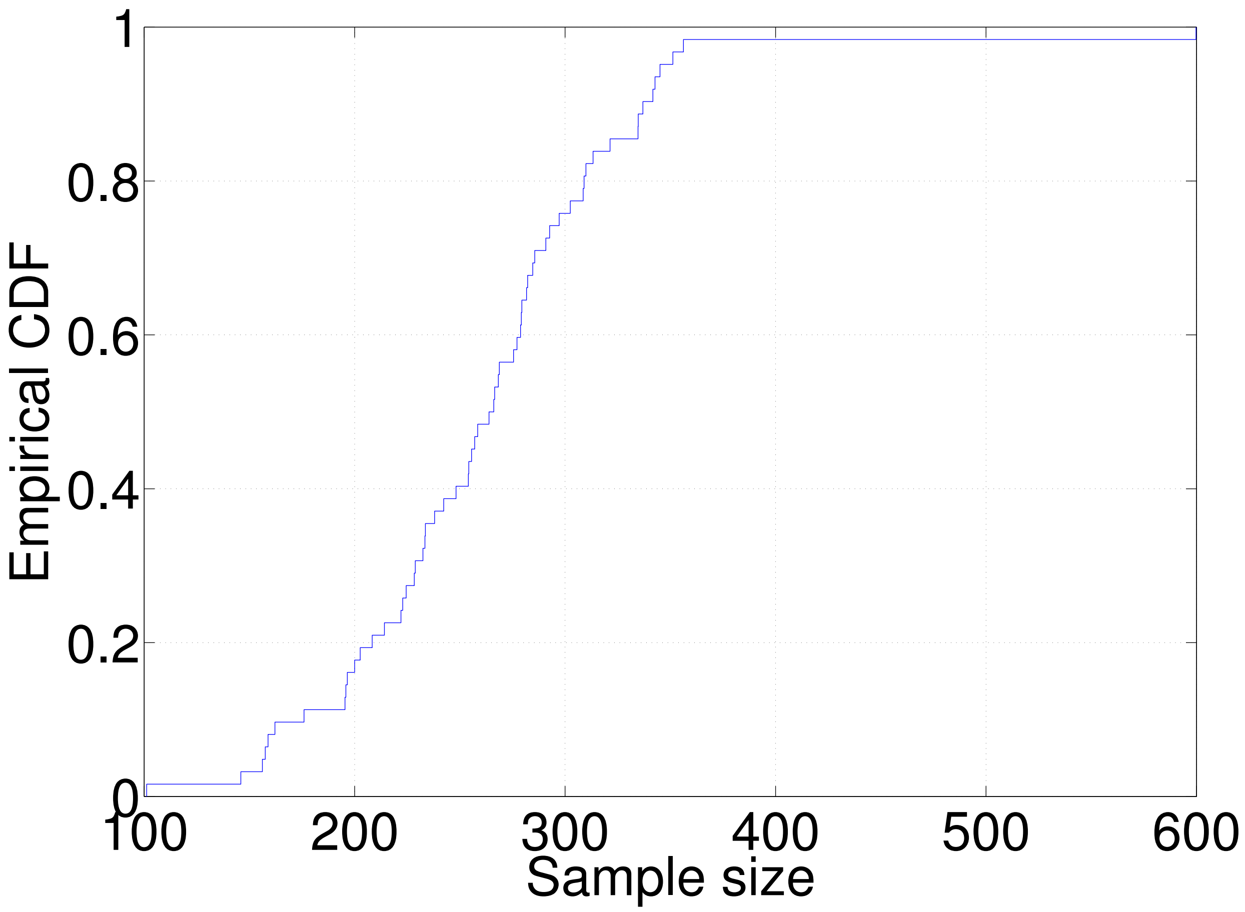


Figure 20: Empirical CDF of the sample size required for estimating the mean packet-time at 90% accuracy and 90% confidence level

$\mu(d_i(t))$ and $\sigma^2(d_i(t))$. Therefore, the mean path delay can be computed as follows:

$$\mu(d_P(t)) = \sum_{i=0}^{n}(m_i(t)+1)\mu(d_i(t)). \tag{2.2}$$

- The stability of packet-time distribution enables each node to accurately estimate the mean and variance of packet-time from itself to the next-hop along a path, and, with Formula (2.2), the mean path delay can be accurately estimated.

Observation 2. *Packet-time is uncorrelated.*

We also observe that the packet-time for different transmissions, whether from the same node or from different nodes, tend to be uncorrelated. For instance, Figure 21 shows the small auto-

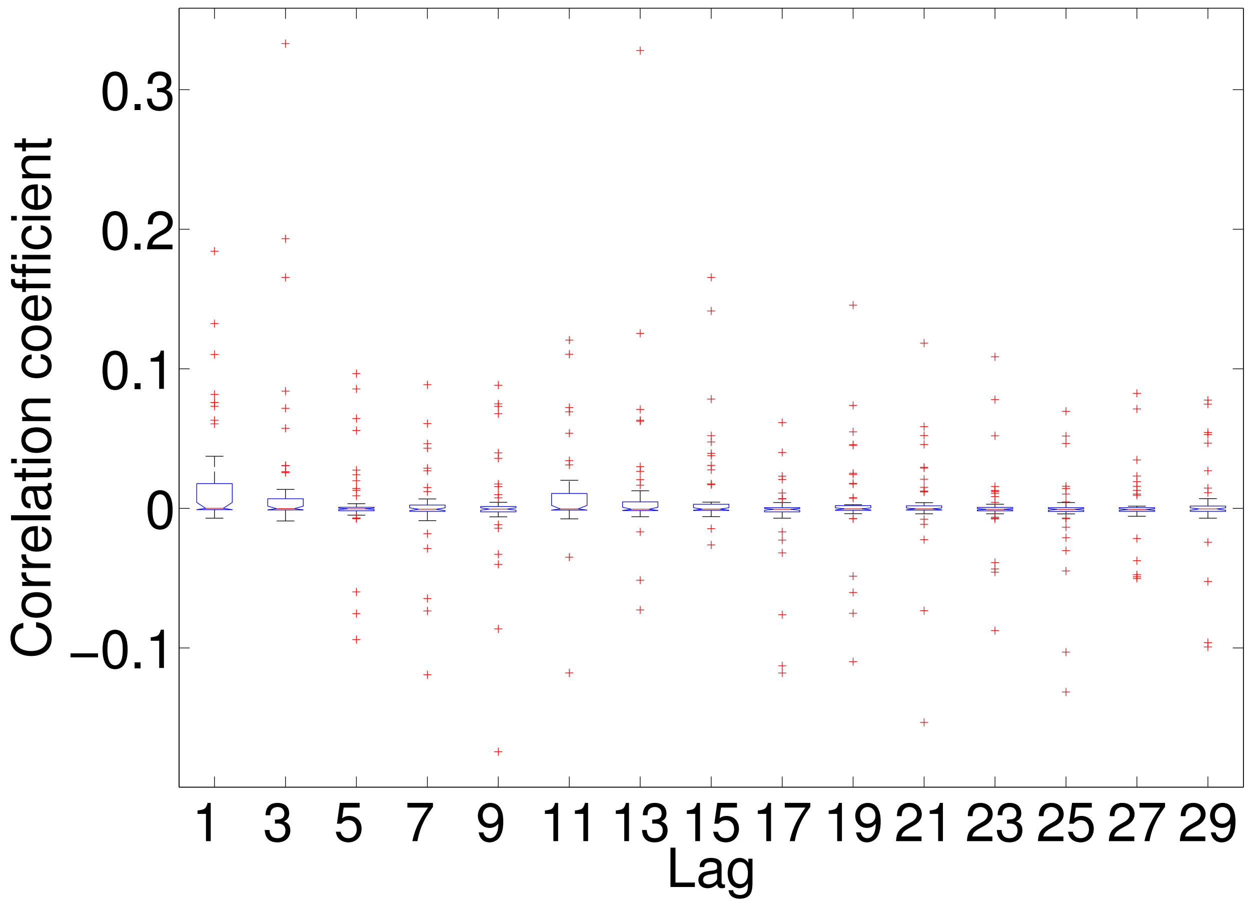


Figure 21: Autocorrelation coefficient of packet-time along the same link

correlation coefficient of packet-time along the same link, with the boxplot showing, for each lag, the distribution of autocorrelation coefficients across different links; Figure 22 shows the small

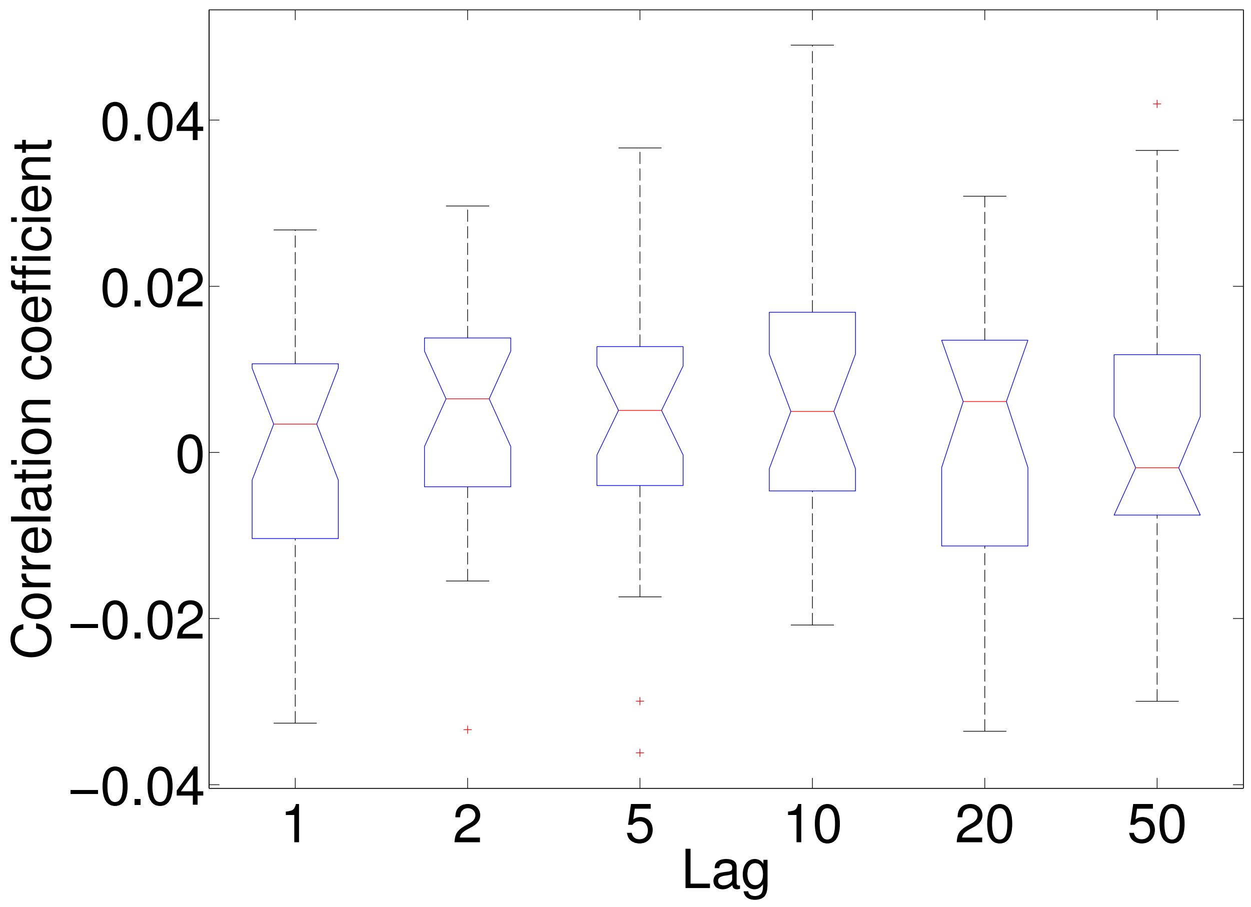


Figure 22: Correlation coefficient between the packet-time across different links along a path

correlation coefficient of packet-time along different links, where the lag is defined as the number of packets queued between two packets at different node queues as shown in Figure 11, assuming that the packets queued at time t flow through the nodes queues in a FIFO manner. We also observe that the median autocorrelation coefficient and the median cross-link correlation coefficient of packet-time is zero at 99% confidence level. An intuition for the uncorrelatedness between packet-time is that, given a network condition, the behavior of one packet transmission does not have much impact on the behavior of another packet transmission as far as the MAC protocol is concerned.

The Bienaym$\acute{e}$ Formula [84] shows that the variance of the sum of pair-wise uncorrelated random variables is the sum of the variances of the individual random variables. Thus our observations on the uncorrelatedness of packet-time along a path enables each node to compute, for a given time instant, the standard deviation of the path delay from itself to the destination node as the square root of the sum of the variances of the packet-times for all the packets queued along the path at that instant. For path P of Figure 11, for instance, the standard deviation of $d_P(t)$ can be computed as follows:

$$\sigma(d_P(t)) = \sqrt{\sum_{i=0}^{n}(m_i(t)+1)\sigma^2(d_i(t))}. \tag{2.3}$$

For a typical 5-hop path in NetEye, Figure 23 shows the histogram of the relative errors in estimat-

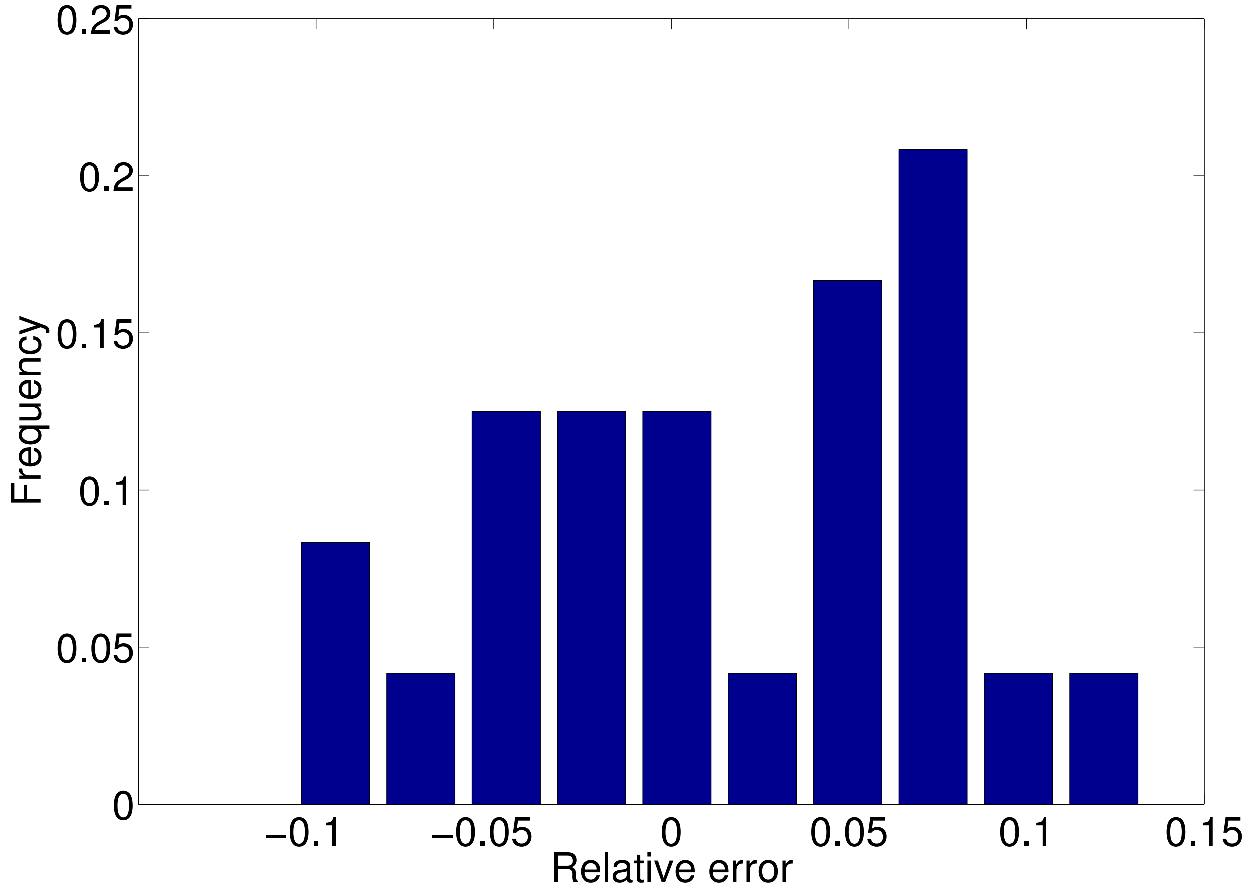


Figure 23: Histogram of relative errors in estimating the standard deviation of path delay

ing the standard deviation of path delay in the presence of different queueing levels along the path.

We see that the estimation is quite accurate, with most relative errors within the range of (-0.075, 0.075). Note that, if we directly estimate the variance of the sojourn time $d_i^{so}(t) = \sum_{j=1}^{m_i(t)+1} d_i^j(t)$ at node v_i without decomposing $d_i^{so}(t)$ into its individual components $d_i^j(t)$'s ($j = 1 \ldots m_i(t)+1$), we cannot compute $\sigma(d_P(t))$ as $\sqrt{\sum_{i=0}^{n} \sigma^2(d_i^{so}(t))}$. This is because, for $i1 \neq i2$, $d_{i1}^{so}(t)$ and $d_{i2}^{so}(t)$ are correlated due to the correlation between queueing levels at different nodes of a path.

In the simplified scenario of Figure 11, the next-hops of all the packets in a node queue are the same. In reality, a node may well use different next-hops for different packets, for instance, depending on packet deadlines as we will discuss in Section 4. Assume that, at time t, each node $v_i (i = 0 \ldots n)$ has $N_i(t)$ number of next-hops, the number of packets (including the one arriving at v_0 at time t) to be forwarded to its k-th next hop is $m_i^k(t)$, and the packet-time from v_i to its k-th next-hop is $d_{i,k}(t)$. We observe that, given v_i and t, $d_{i,k}(t)$'s tend to be uncorrelated for different ks. For instance, Figure 24 shows the histogram of the correlation coefficient between the packet-time across different outgoing links from the same node. The correlation coefficient is very small and is less than 3% most of the time. We also observe that the median correlation coefficient is zero at 99% confidence level. Then, based on Formulae (2.2) and (2.3), the mean and standard deviation of the delay along a path P can be computed as follows:

$$\begin{aligned} \mu(d_P(t)) &= \sum_{i=0}^{n} \sum_{k=1}^{N_i(t)} m_i^k(t) \mu(d_{i,k}(t)) \\ \sigma(d_P(t)) &= \sqrt{\sum_{i=0}^{n} \sum_{k=1}^{N_i(t)} m_i^k(t) \sigma^2(d_{i,k}(t))} \end{aligned} \tag{2.4}$$

Observation 3. *Network queueing is relatively stable at short timescales.*

To leverage Formula (2.4) in estimating the mean and variance of the delay along path P in a distributed manner, each node v_i can compute the mean and variance of the delay from itself to the destination node v_{n+1} based on those of its next-hop along P. Denoting the mean and standard deviation of the delay from v_i to v_{n+1} by $\mu(d_P^i(t))$ and $\sigma(d_P^i(t))$ respectively (i = 0 … n), then we have

$$\begin{aligned} &\mu(d_P^i(t)) = \mu(d_P^{i+1}(t)) + \sum_{k=1}^{N_i(t)} m_i^k(t) \mu(d_{i,k}(t)) \\ &\mu(d_P^{n+1}(t)) = 0 \end{aligned} \tag{2.5}$$

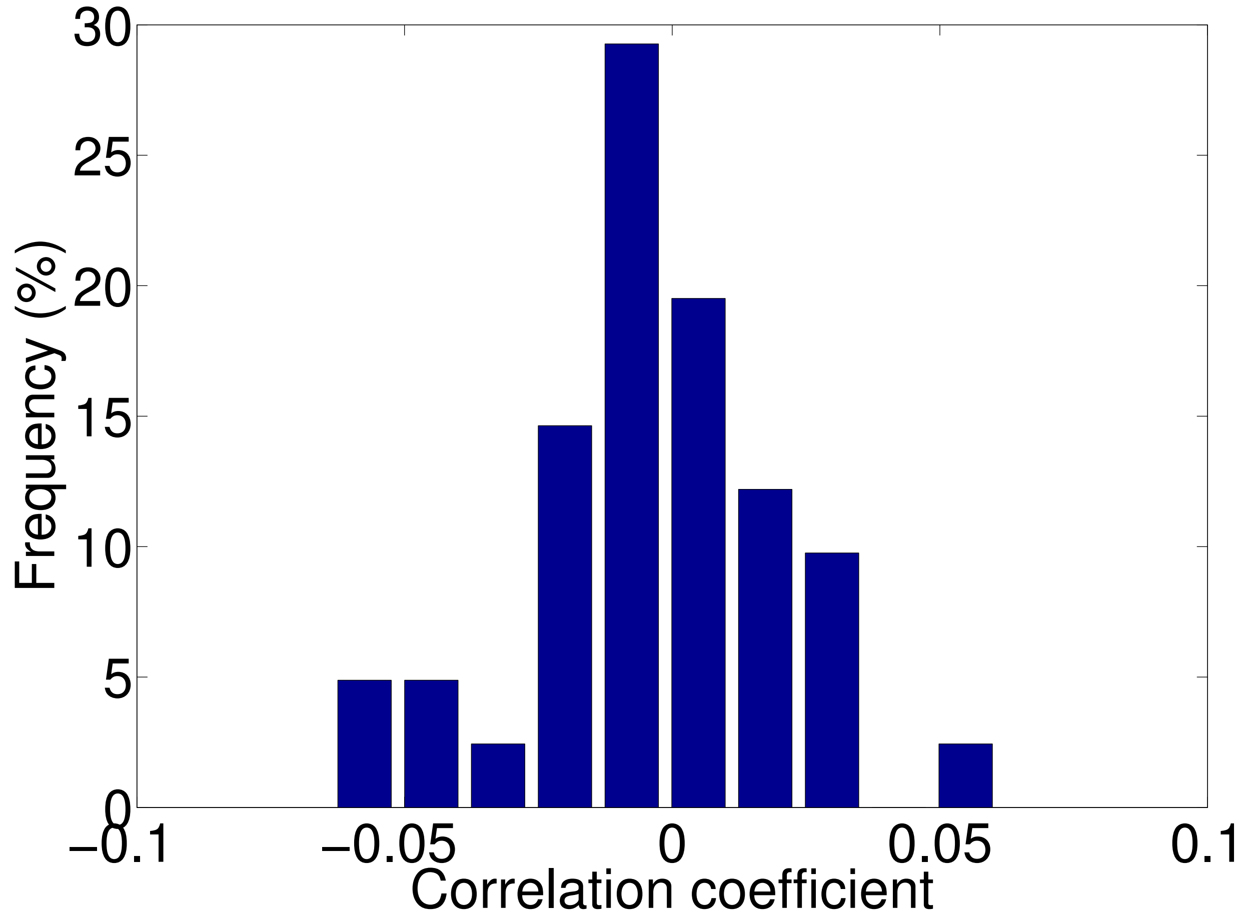


Figure 24: Histogram of the correlation coefficient between the packet-time across different outgoing links from the same node

$$\begin{aligned}&\sigma(d_P^i(t)) = \sqrt{\sigma^2(d_P^{i+1}(t)) + \textstyle\sum_{k=1}^{N_i(t)} m_i^k(t)\sigma^2(d_{i,k}(t))}\\ &\sigma(d_P^{n+1}(t)) = 0\end{aligned} \tag{2.6}$$

Formulae (2.7) and (2.8) can be implemented using a distance-vector-type routing algorithm. Due to information diffusion delay τ_i from v_{i+1} to v_i in routing, the implemented version of Formulae (2.7) and (2.8) are

$$\begin{aligned}&\mu(d_P^i(t)) = \mu(d_P^{i+1}(t-\tau_i)) + \textstyle\sum_{k=1}^{N_i(t)} m_i^k(t)\mu(d_{i,k}(t))\\ &\mu(d_P^{n+1}(t)) = 0\end{aligned} \tag{2.7}$$

$$\begin{aligned}&\sigma(d_P^i(t)) = \sqrt{\sigma^2(d_P^{i+1}(t-\tau_i)) + \textstyle\sum_{k=1}^{N_i(t)} m_i^k(t)\sigma^2(d_{i,k}(t))}\\ &\sigma(d_P^{n+1}(t)) = 0\end{aligned} \tag{2.8}$$

For accurate estimation of $\mu(d_P^i(t))$ and $\sigma^2(d_P^i(t))$, we need to make τ_i as small as possible. This can be achieved by having v_{i+1} piggyback $\mu(d_P^{i+1}(t))$ and $\sigma^2(d_P^{i+1}(t))$ onto its data transmissions as well as control signaling (e.g., broadcast of routing beacons) so that v_i can overhear the values quickly. In WSC networks, sample data about physical behavior are usually generated periodically in a continuous manner, thus τ_i is at the same timescale of inter-packet arrival interval, which enables quick diffusion of path delay statistics.

We also observe that, even though network queueing varies significantly at a long timescale of hundreds of inter-packet intervals, it is much more stable at a short timescale of a few inter-packet intervals. In the NetEye medium traffic scenario and for the time lags of 1, 5, and 10 packet transmissions, for instance, Figures 25, 26, and 27 show the empirical cumulative distribution function (CDF) of link queueing level changes (i.e., changes in $m_i^k(.)$), node queueing level changes (i.e., changes in $m_i(.)$), and path queueing level changes (i.e., changes in $\sum_i m_i(.)$) respectively. We see that, at the timescale of a few inter-packet intervals, network queueing remains relatively stable, and, with more than 90% probability, the absolute changes in link, node, and path queueing levels are no more than 1, 1, and 3 respectively.[3] To ensure enough real-time capacity for each source

[3]The routing hop count in the NetEye medium traffic scenario with MTA is around 6 hops in general.

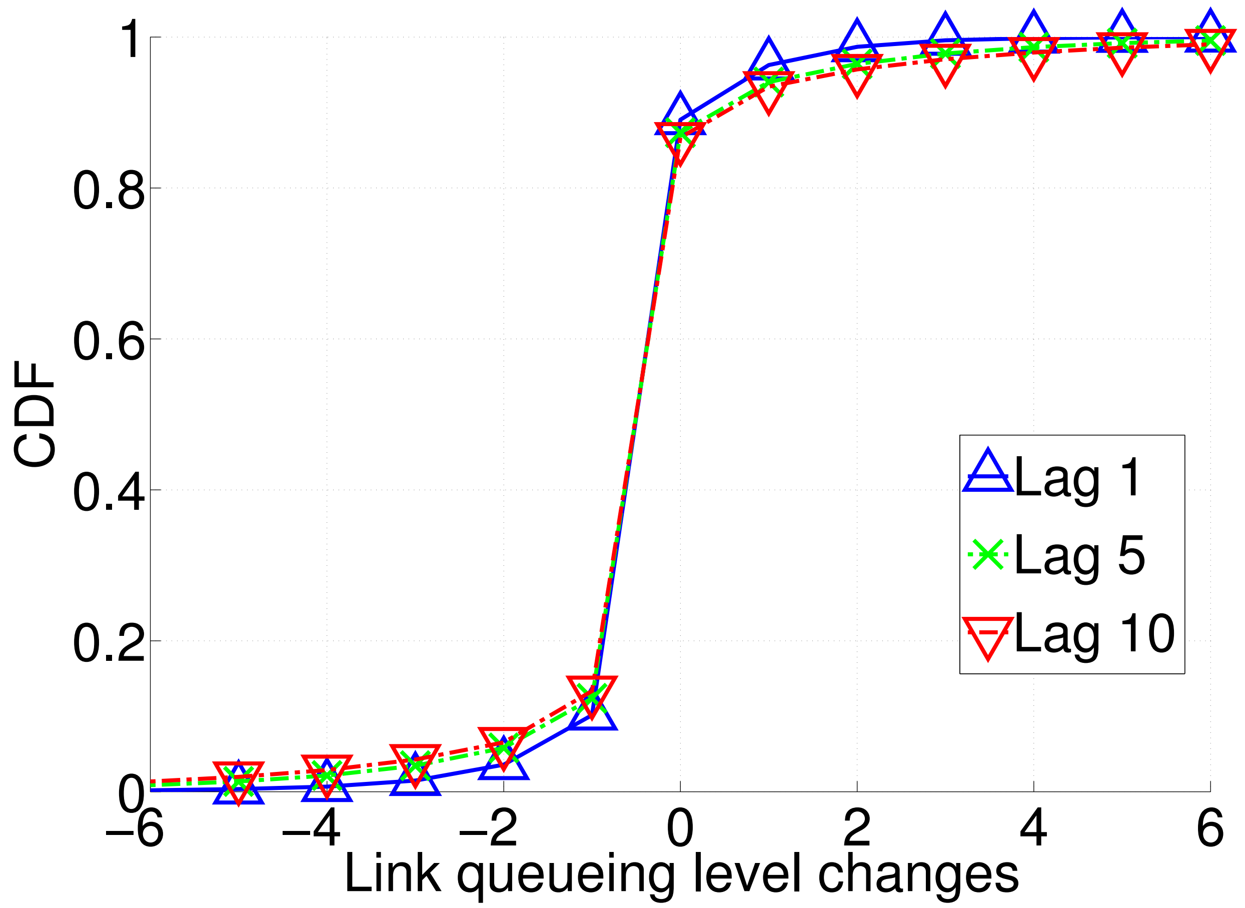


Figure 25: CDF of link queueing level changes

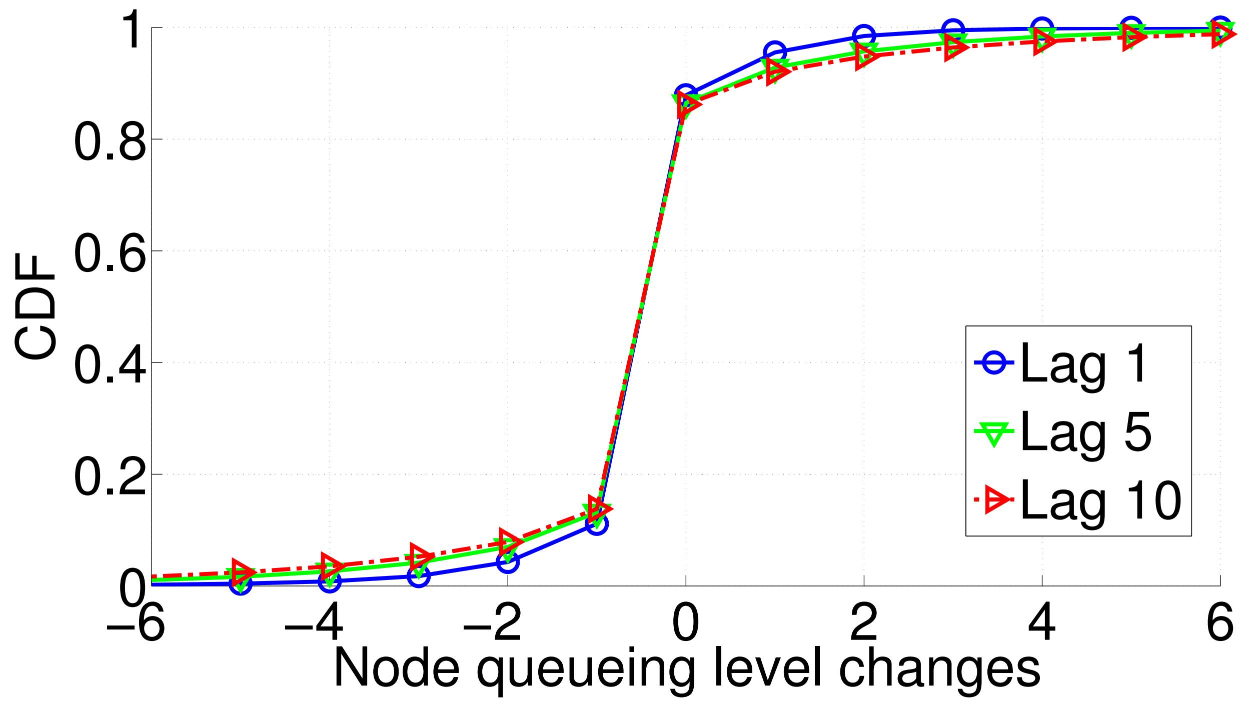


Figure 26: CDF of node queueing level changes

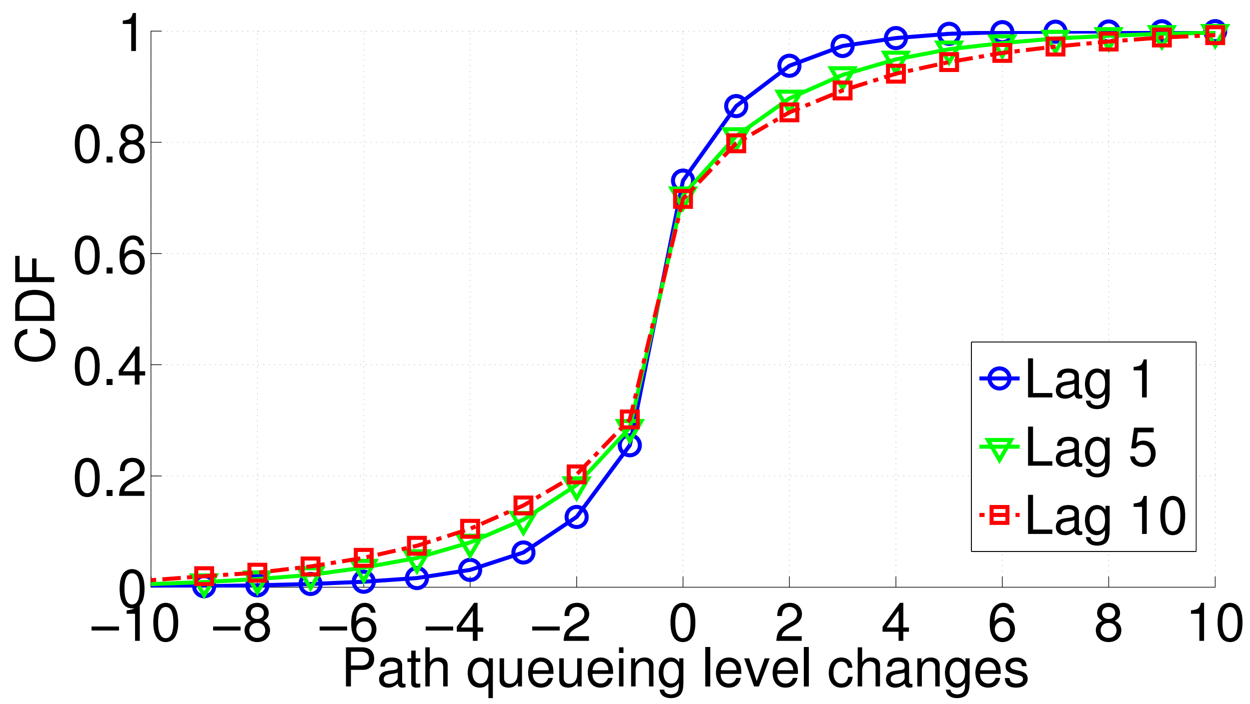


Figure 27: CDF of path queueing level changes

node, routing hops tend to be limited (e.g., less than 10) in WSC networks [15]. This fact, together with the quick information diffusion and the relative stable network queueing at short timescales, enables the MTE method to accurately estimate the mean and variance of path delays in an agile manner.

Probabilistic path delay bound. With the mean and variance of path delay estimated via the MTE method, a node can derive the probabilistic path delay bounds using probability inequalities. To this end, we have

Proposition 1. *For a random variable X, if*

$$Pr\{X \geq f(x)\} \leq g(x), \tag{2.9}$$

then $Q_X^q = f(g^{-1}(1-q))$ is an upper bound on the q-quantile of X, where g^{-1} is the inverse function of g.

Proof. Let $g(x) = 1 - q$, then

$$Pr\{X \geq f(x)\} \leq 1 - q.$$

Thus,

$$Pr\{X \leq f(x)\} \geq q$$

Since $x = g^{-1}(1-q)$, we have

$$Pr\{X \leq f(g^{-1}(1-q))\} \geq q$$

Thus, $f(g^{-1}(1-q))$ is an upper bound on the q-quantile of X. □

For $\lambda > 0$ and a non-negative random variable X with mean μ and variance σ^2, two widely-applicable probability inequalities [115] are the Markov Inequality

$$Pr\{X \geq \lambda\} \leq \frac{\mu(X)}{\lambda}, \tag{2.10}$$

and the one-tailed Chebyshev Inequality

$$Pr\{X - \mu \geq \lambda\sigma\} \leq \frac{1}{1+\lambda^2}. \tag{2.11}$$

Thus, we have

Corollary 1. *Using Markov Inequality,* $Q_X^q = \frac{\mu}{1-q}$.

Corollary 2. *Using one-tailed Chebyshev Inequality,* $Q_X^q = \mu + \sigma\sqrt{\frac{q}{1-q}}$.

For a typical five-hop path in NetEye, Figure 28 shows the ground truth and the estimated

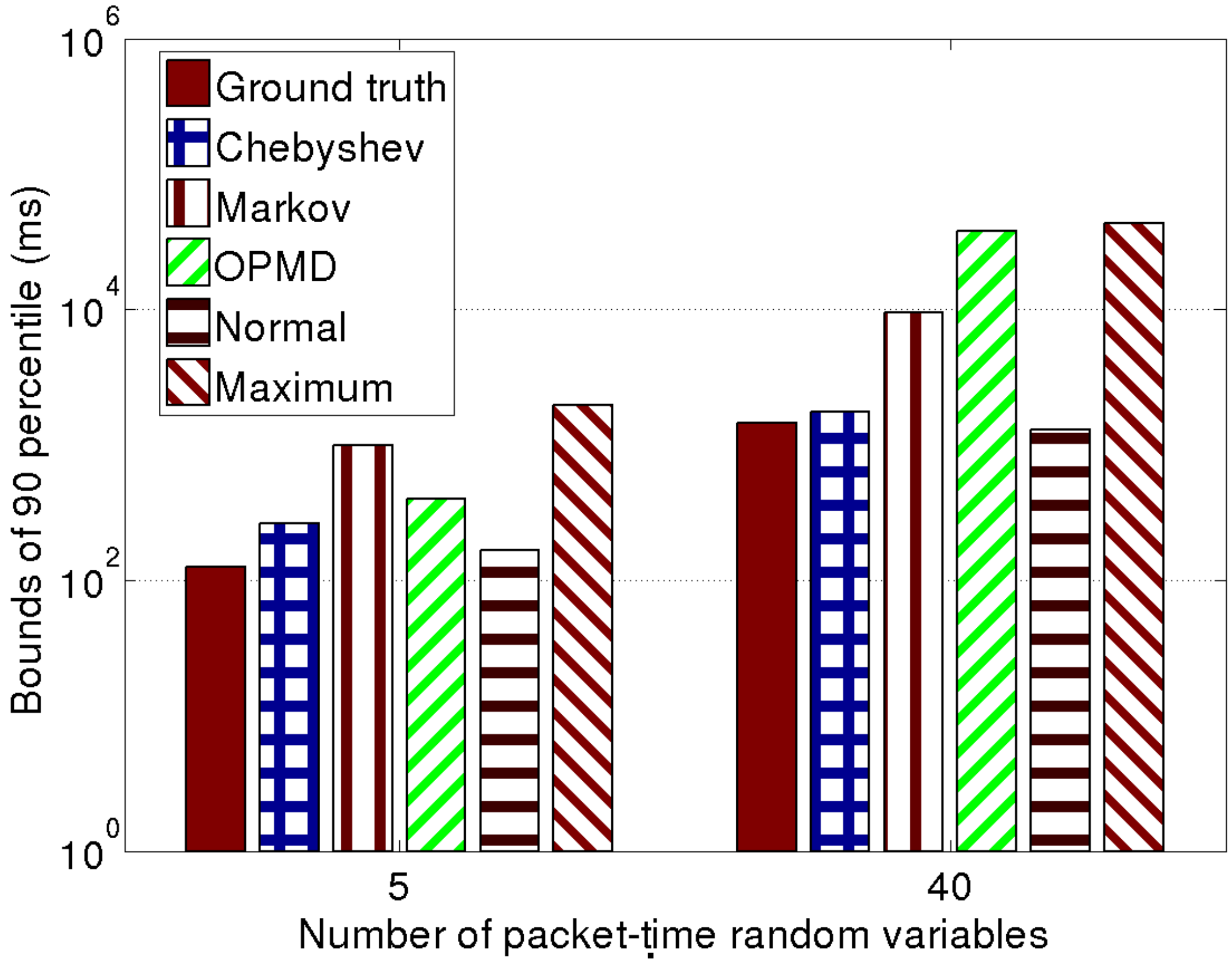


Figure 28: Bounds on 90-percentile of path delay

90-percentile of the path delay when it is the sum of 5 and 40 packet-time random variables respectively. For comparison purpose, we present the probabilistic delay bounds estimated via the optimal-partition-minimum-delay (OPMD) method [82] or by assuming path delay is normally distributed [69]; we also present the maximum path delay. Given that Markov Inequality usually gives a looser bound than one-tailed Chebyshev Inequality, the probabilistic delay bound by Markov Inequality is greater than that by Chebyshev Inequality. Compared with the method of assuming path delay is normally distributed, the bound by Chebyshev Inequality is always greater than the actual 90-percentile path delay, whereas the former method underestimates the 90-percentile for the case of 40 packet-time random variables. Given that the OPMD method is rather conservative in estimating path delay bound, the bound by Chebyshev Inequality is also much less than the bound by the OPMD method, especially when path queueing increases. We also see that the maximum path delay is orders of magnitude greater than the bound by Chebyshev Inequality, thus using probabilistic delay bound instead of maximum path delay helps improve application-usable real-time capacity. Since the bound by Chebyshev Inequality upper-bounds and is close to the probabilistic path delay, we use the probabilistic delay bound by Chebyshev Inequality in our protocol design.

From FCFS to EDF. In real-time scheduling, the earliest-deadline-first (EDF) algorithm is a commonly used algorithm, and it can ensure a smaller deadline than what is feasible with the first-come-first-serve (FCFS) algorithm. Our MTE method of computing probabilistic path delay bounds is derived by assuming that nodes use FCFS algorithm in intra-node transmission scheduling, but we observe that the delay bounds derived via our FCFS-based MTE method can also serve as the basis of selecting real-time packet forwarding paths even if the EDF algorithm is used. Formally,

Proposition 2. *If routed along a path whose probabilistic delay bound computed via the FCFS-based MTE method is less than the relative deadline of a packet, the packet will reach its destination before its deadline even if the EDF algorithm is used for intra-node transmission scheduling.*

Proof. We prove the proposition by contradiction. When a packet pkt_0 is forwarded where, each time a next-hop has to be chosen, the next-hop is chosen such that pkt_0's relative deadline is no less than the probabilistic path delay bound computed via the FCFS-based MTE method, the only way in which pkt_0 can miss its deadline is as follows: when pkt_0 is queued at some node v_i, another packet pkt_1 arrives at v_i at time t and is transmitted to a next-hop earlier than pkt_0 is transmitted. In this case, since pkt_1 arrives at v_i later than pkt_0, the relative deadline of pkt_1 at time t is no less than the probabilistic path delay bound $Q_i(t)$ from v_i to the destination assuming pkt_1 is transmitted after pkt_0. If the EDF algorithm schedules pkt_1 to be transmitted earlier than pkt_0, this implies that the relative deadline of pkt_0 at time t is greater than that of pkt_1 at t, thus being greater than the delay bound $Q_i(t)$. Note also that $Q_i(t)$ has included the time taken to transmit pkt_1 and pkt_0. Therefore, even if pkt_1 is transmitted earlier than pkt_0 (which is similar to switching their queue positions in a FIFO queue), pkt_0 can still be delivered to the destination before its deadline. Q.E.D. □

4 Multi-timescale adaptation for real-time routing

In what follows, we first present the multi-timescale adaptation (MTA) framework for real-time routing, then we address the challenges of implementing MTA in resource-constrained platforms.

4.1 MTA routing design

Overview. Real-time routing is subject to dynamics and uncertainties at multiple timescales. At a longer timescale, link properties such as ETX (i.e., expected-number-transmissions to successfully deliver a packet) vary as a result of changing environmental conditions (e.g., temperature); at a shorter timescale, data transmission delay varies on a per-packet basis, and bursty traffic may introduce sudden changes to network conditions. Robust system design usually requires adaptation to dynamics at the same or shorter timescales of the dynamics themselves. Yet we have found that, due to the highly-varying nature of path delays, routing using delay-based metrics can introduce large estimation errors and lead to routing instability as well as low performance [137]. To ensure long-term stability and optimality while addressing short-term dynamics at the same time, we propose a multi-timescale adaptation (MTA) framework for real-time routing as follows.

At lower frequencies, a directed-acyclic-graph (DAG) is maintained for data forwarding, and any path within the DAG is a candidate path for packet delivery. Given that link/path ETX reflects network throughput, data delivery reliability, and the overall trend of data delivery delay [34, 38], and that ETX-based routing structures tend to be stable even if ETX is dynamic [136], we propose to maintain the data forwarding DAG based on link and path ETX such that the DAG reflects long-term system optimality and changes relatively slowly compared with delay variation. More specifically, there is a directed edge from node v_i to v'_i in the DAG if and only if the minimum path ETX from v'_i to the destination v'_0 is less than that from v_i to v'_0. The DAG defines, for each node v_i, a set of forwarder candidates $\mathcal{R}(v_i)$ where $v'_i \in \mathcal{R}(v_i)$ if and only if link $\langle v_i, v'_i \rangle$ belongs to the DAG.

At higher frequencies, the spatiotemporal flow of packets within the data forwarding DAG is adaptively controlled to ensure reliable, real-time data delivery in the presence of short-timescale dynamics such as transient packet losses and per-packet variations of link delay. More specifically, each packet contains information about the remaining time to deadline, denoted by L, and the required real-time guarantee probability q. When the packet reaches a node v_i, v_i first finds the set of forwarder candidates within the DAG, denoted by $\mathcal{R}'(v_i, L, q)$, that can ensure the real-time requirements L and q; then v_i sets the next-hop node of the packet as the node of the smallest path ETX to v'_0 among all the nodes in $\mathcal{R}'(v_i, L, q)$, and v_i puts the packet in the transmission queue. Queued packets are then scheduled for transmission using the earliest-deadline-first (EDF) algorithm.

Using the above approach, packets are always routed along the minimum-ETX path that satisfies the real-time data delivery requirement. In the presence of heavy traffic load that induces queueing, this real-time forwarding mechanism creates the water-filling effect as follows: packets are delivered to the minimum-ETX path until the path cannot ensure the required timeliness of data delivery (e.g., due to queueing), at which point the forthcoming packets are delivered to the path with the second-minimum-ETX, and so on; once the delay along paths of less ETX decreases (e.g., due to reduced queueing), more packets will be delivered to those paths to fill them up. This process repeats at the same timescale of packet arrival process, and it enables real-time data delivery while ensuring as small ETX in data delivery as possible. Unlike traditional delay-based routing that may lead to instability and low performance, this quick adaptation of spatial packet flow is enabled by 1) the MTE method of accurate, agile estimation of probabilistic path delay bound at the same timescale of the changes in path delay distribution and by 2) the overall stability of spatial packet flow along the data forwarding DAG.

System architecture. To implement the above MTA framework, we adopt a system architecture as shown in Figure 29.

Using MAC feedback for the status (e.g., number of retries and time duration) of transmitting data packets, each node uses data-driven approaches [49, 132] to estimate the ETX as well as the mean and variance of packet-time for its local links. Based on nodes' local link ETX, the data-forwarding DAG can be established and maintained using distance-vector-routing-type diffusion computation.

Each node also maintains a neighbor table that stores information about its neighbors. In the case of limited memory, a node may only maintain information about a selected set of neighbors, e.g., those of small path ETX to the destination. For each neighbor in the neighbor table, the ETX as well as the mean and variance of the delay along different paths from the neighbor to the destination is maintained. Ideally, each node shall share information about all the paths from itself to the destination, but this is infeasible in practice due to limited memory size at nodes, overhead of exchanging path information with neighbors, and the exponentially increasing number of paths from a node to the destination as the hop distance to the destination increases. Therefore, each node v_i only shares with its neighbors a summary of its knowledge about the paths from itself to the destination, using the following path aggregation method:

1. For each path P from a neighbor to the destination v'_0, as implicitly identified by an entry in the neighbor table, v_i computes the mean and variance of the delay from itself to v'_0 through P using Formulae (2.7) and (2.8), and, based on Corollary 2 and application-required real-time guarantee probability q, compute the probabilistic delay bound for the path $\{v_i, P\}$

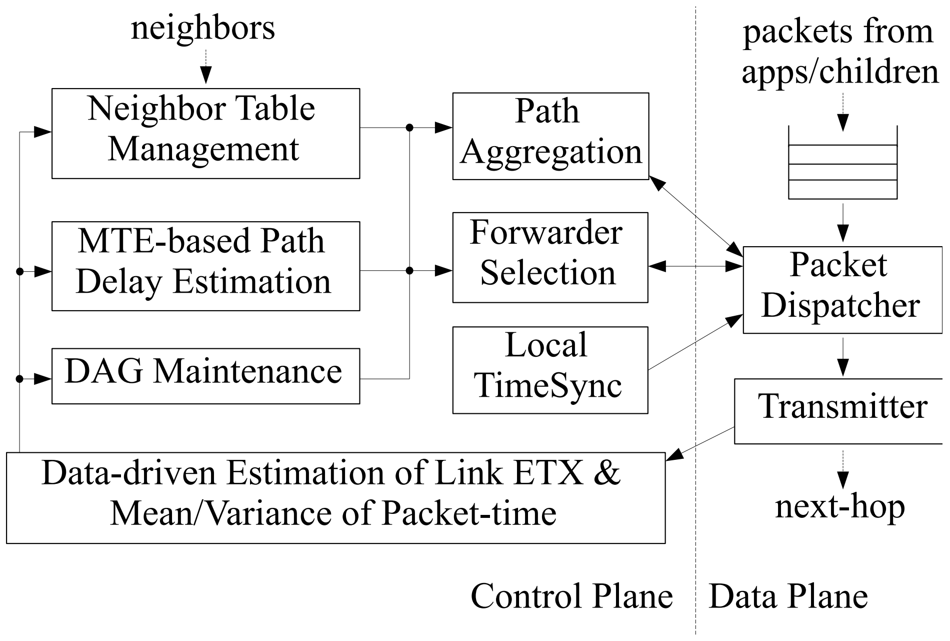

neighbors
Neighbor Table Management
MTE-based Path Delay Estimation
DAG Maintenance
Path Aggregation
Forwarder Selection
Local TimeSync
packets from apps/children
Packet Dispatcher
Transmitter
next-hop
Data-driven Estimation of Link ETX & Mean/Variance of Packet-time
Control Plane
Data Plane

Figure 29: Architecture of MTA-based real-time routing

from v_i to v'_0 through P;

2. Compute the set $\mathcal{P}$ of all valid paths from v_i to v'_0, where a path $\{v_i, P\}$ is valid if there is no other path $\{v_i, P'\}$ that has both smaller ETX and smaller path delay bound (as computed above);

3. Order the paths in $\mathcal{P}$ in a non-decreasing order of the delay bounds they ensure, and select a subset of them as the path summary to share with neighbors; in our implementation, a node selects the path summary such that the delay bounds ensured by the selected paths cover the complete range of delay bounds ensured by the original set $\mathcal{P}$.

The computed path summary is piggybacked onto data packet transmissions by v_i, where each entry of the summary records the ETX as well as the mean and variance of the delay along the corresponding path.

When a node receives a packet from a local application or a child, the packet dispatcher first uses a local time synchronization service [10] to compute the time elapsed since the packet was transmitted by its sender, which enables the dispatcher to derive the remaining time to deadline (a.k.a. relative deadline) for the packet. Based on the packet's relative deadline, the dispatcher puts the packet in a corresponding position in the local queue. Whenever the queue is not empty and the wireless channel becomes available, the dispatcher selects a packet to transmit using the EDF algorithm (a.k.a. least laxity first algorithm).

4.2 Implementation

In this subsection , we discuss challenges and our solutions in implementing MTA architecture (as shown in Figure 29) in TinyOS.

Computation overhead & task management. Unlike wired Internet routers, embedded WSC network nodes tend to be resource-constrained, yet the many control plane and data plane functions (e.g., link estimation, next-hop selection) need to be computed in real-time and on a per-packet basis to ensure accuracy and agility in estimation and routing. Therefore, the different computation tasks inside each node need to be managed well to ensure correct operation of the protocol. In fact, the computation tasks can consume so much CPU time that the entire protocol collapses in TelosB due to its failure of executing critical tasks. For instance, timers may fail to fire at specified time instants, and packet receptions may not be signaled to upper layers and thus lost. To address these issues, we split heavy-weight computations (e.g., computing the next-hop for a packet) into multiple subtasks such that each computation can be interrupted between its subtasks to ensure responsiveness of the system to events such as timer firing and packet reception. Our experience suggests that it suffices to have each subtask take no more than a few milliseconds of time to complete.

Be default, TinyOS employs a FIFO queue to store all pending tasks. It does not differentiate urgent tasks (i.e., data packet transmission) from normal tasks (i.e., beacon transmission) in terms of real-time communication. To address this issue, we have implemented priority schemes for managing the tasks of the control and data planes to ensure real-time packet forwarding and control adaptation.

Global vs. local time synchronization. Global time synchronization is sufficient to provide a time reference across the entire network. Nonetheless, it comes at substantial cost of memory, computation, and bandwidth overhead, leaving even less resource for MTA in already resource-constrained devices. Even worse, it can take a long time for a global time synchronization protocol such as FTSP [87] to converge, during which the network cannot be used to deliver any real-time traffic.

Upon closer scrutiny, we discover that only neighboring nodes, not all nodes, have to agree on a common time as far as MTA/MTE is concerned, which enables estimating packet-time and tracking absolute/relative deadline when a packet is delivered from one node to another. Thus, we employ the packet-level time synchronization mechanism in TinyOS, which automatically converts a time value from the sender's local time to the receiver's local time. Only a few bytes are added into the header a data packet, and no additional control packets have to be sent, significantly reducing the overhead compared with global time synchronization.

Packet-level time synchronization requires timestamping packet reception. But the default radio stack for TelosB stores the incoming packet and its reception timestamp in two separate FIFO queues and then match them by taking head-of-queue elements from both queues. Occasional mismatches of a packet and its timestamp are observed due to disparate operations on the two queues. To address this issue, we store the reception timestamp of a received packet directly in its metadata instead of using separate FIFO queues.

5 Measurement evaluation

We evaluate the behavior of MTA through measurement study in the NetEye and Indriya testbeds. In what follows, we first present the methodology, then we study the design decisions of MTA and compare MTA with existing routing protocols.

5.1 Methodology

Protocols. We have implemented the MTA framework (including the MTE method) in TinyOS. To understand the design decisions of MTA, we comparatively study MTA with its following variants:

- *M-DS*: same as MTA but directly estimate path delay quantiles using non-parametric method P^2 [62] and path delay samples that are collected in a distributed manner;
- *M-DB*: same as MTA but estimates the mean and variance of path delay directly through path delay samples;
- *M-ST*: same as MTA but estimates the mean and variance of path delay as the sum of the mean and variance of the sojourn time at each node of the path, without decomposing the sojourn time into the individual packet-times;
- *M-MD*: Same as MTA but maintains the data forwarding DAG based on mean link/path delay instead of link/path ETX;

- *M-mDQ*: same as MTA but forwards packets to the next-hop candidate with the minimum path delay quantile instead of the one with minimum path ETX;

- *mDQ*: same as M-mDQ but does not use the data forwarding DAG of MTA for stability control;

- *M-FCFS*: same as MTA but uses FCFS instead of EDF for intra-node transmission scheduling.

Towards understanding the benefits of MTA, we also comparatively study MTA with the following existing protocols:

- *MCMP*: a multi-path QoS routing protocol where end-to-end QoS requirements on reliability and timeliness are uniformly divided into per-hop reliability and timeliness requirements, upon which a node chooses the minimum number of next-hops to satisfy the per-hop requirements in data delivery [60];

- *MM*: the geographic routing protocol MMSPEED [43] that routes and schedules packet transmissions based on nodes' distances to destinations, packet delivery deadlines, and mean link delays; MMSPEED also tries to improve packet delivery reliability by transmitting packets along multiple paths; (Note: we denote MMSPEED as MM for the readability of figures to be presented in the next subsection .)

- *MM-CD*: same as MMSPEED but, instead of using the mean link delay, uses a conservative estimate of link delay that equals the sum of the mean delay and three times the standard deviation of the delay;

- *SDRCS*: similar to MMSPEED but, instead of using geographic distance, uses data-forwarding hop-count as the measure of distance, where the hop-count is computed based on received-signal-strength (RSS) between nodes [126]; data forwarding is through receiver contention similar to that in opportunistic routing;

- *CTP* : an ETX-based non-real-time routing protocol in TinyOS [49].

(Note: for MM and MM-CD which use node locations, we configured each node with its correct location in our TinyOS programs.)

Traffic & real-time requirements. For studying the design decisions of MTA, we use the common case of the medium traffic scenario in NetEye; for comparing MTA with existing real-time routing protocols, we use all the periodic and event traffic scenarios discussed in Section 2. In our experiments, we use 90% as the required real-time guarantee probability by default, but we have also experimented with the real-time guarantee probability of 99% and observed similar phenomena [83]. For differentiating the performance of different protocols, the deadline for each traffic scenario is chosen so that it is neither too stringent (that no protocol can support) nor too loose (that all protocols can support). In NetEye, the deadlines for light, medium, and heavy traffic are 250ms, 2 seconds, and 7.5 seconds respectively, and the deadline for event traffic is 10 seconds. In Indriya, the deadlines for periodic and event traffic are 2 seconds and 2.5 seconds respectively.

Metrics. For each combination of protocol, testbed, and traffic scenario, we run it for 10 times and evaluate protocol performance in terms of the following metrics:

- *Deadline success ratio (DSR)*: ratio of packets delivered to the sink before their deadlines;
- *Packet delivery ratio (PDR)*: ratio of packets delivered to the sink;
- *Number of transmissions per packet delivered (NTX)*: total number of transmissions, including retransmissions, divided by the number of unique packets delivered to the sink.

To understand protocol behavior in more detail, we also analyze the different causes for a packet to miss its deadline:

- *Overflow*: packet discarded due to node queue overflow;
- *Transmission failure*: a packet not delivered to the next hop even after the maximum number of retransmissions at a node;

- *Rejection*: no candidate path can ensure the required real-time delivery guarantee (i.e., deadline and probability) of a packet;
- *Expiration*: deadline expired before the packet reaches the sink, whether or not the packet is delivered to the sink.

5.2 Design decisions of MTA

For periodic medium traffic in NetEye and for protocol MTA and its variants, Figures 30, 31, 32, and 33 show the deadline success ratio, packet delivery ratio, number of transmissions per packet delivered, and packet delivery status respectively. We see that MTA consistently enables the

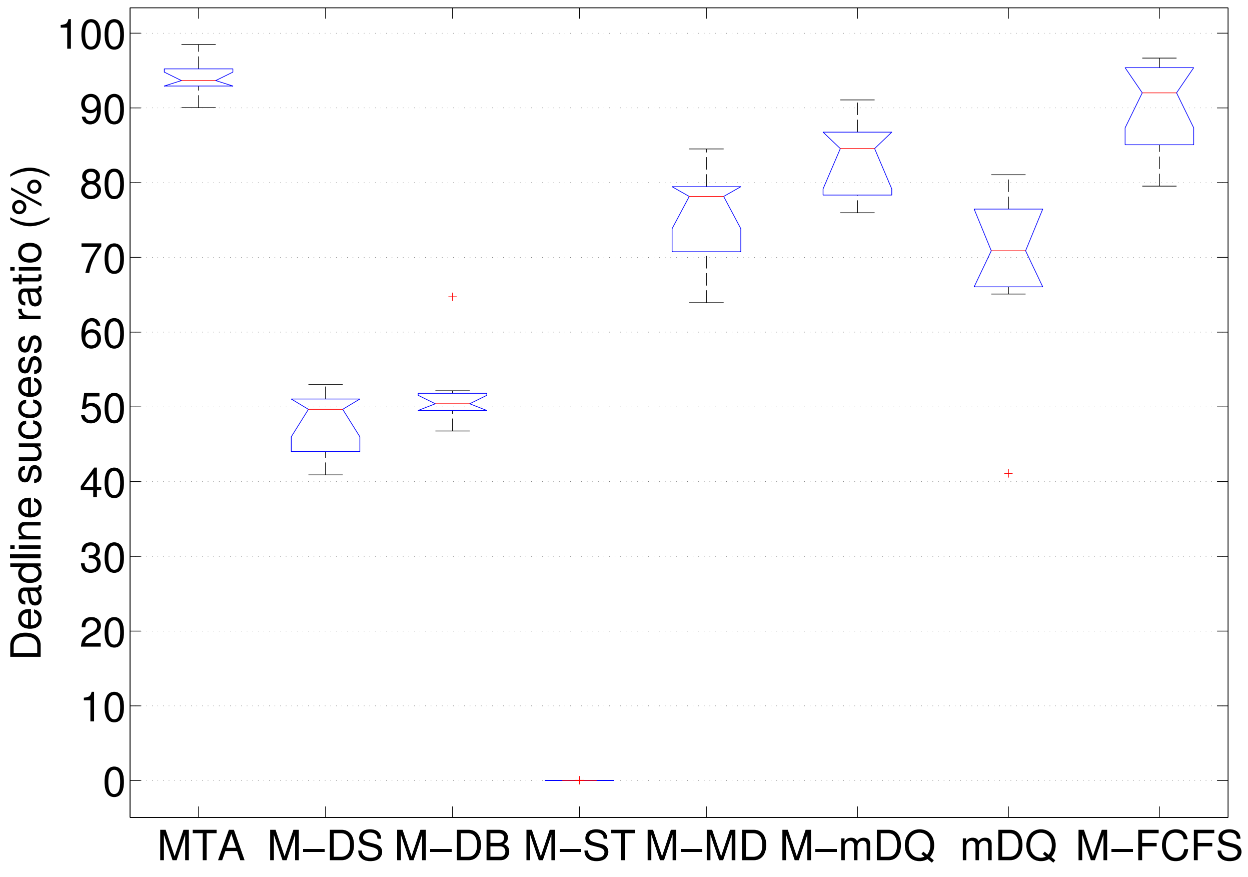


Figure 30: Deadline success ratio: MTA and variants

highest deadline success ratio and packet delivery ratio as well as the lowest transmission cost.

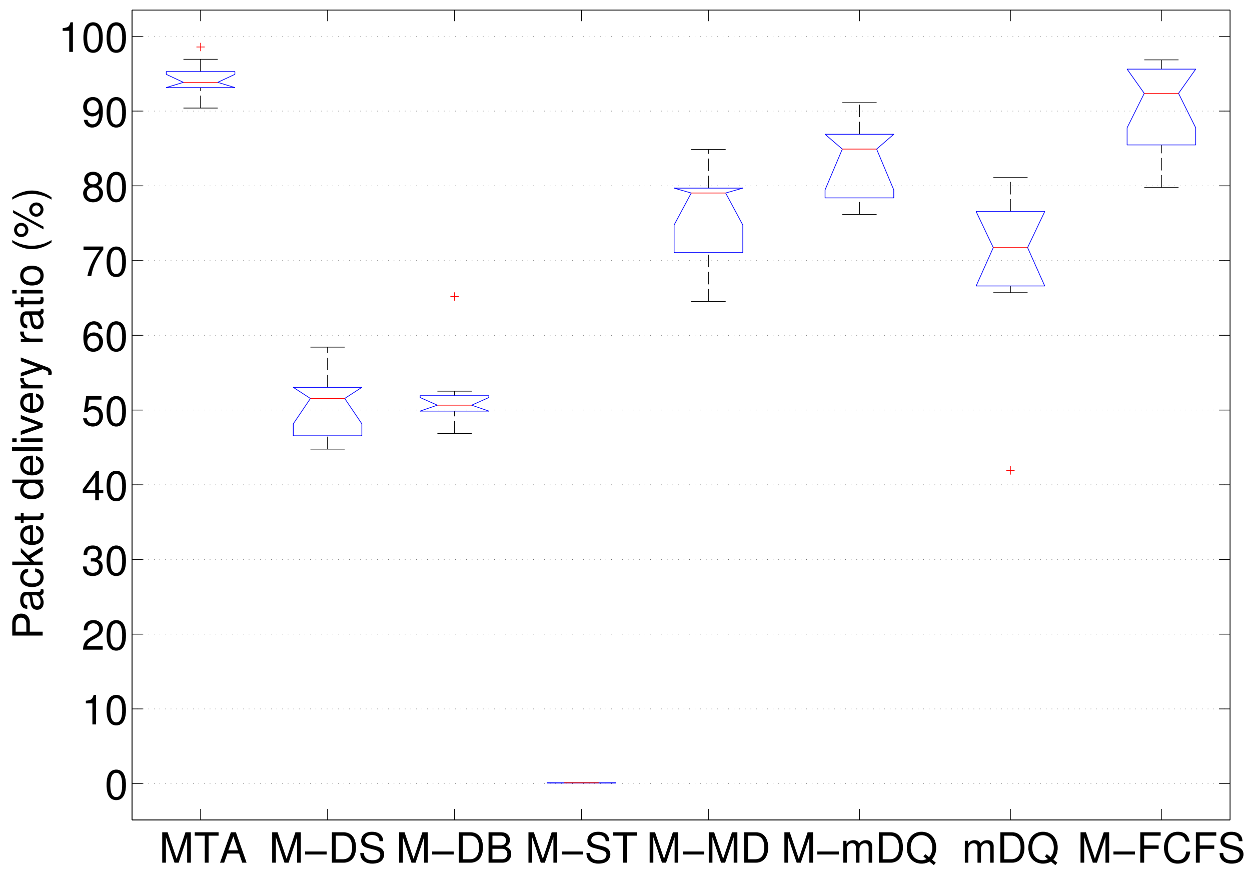


Figure 31: Packet delivery ratio: MTA and variants

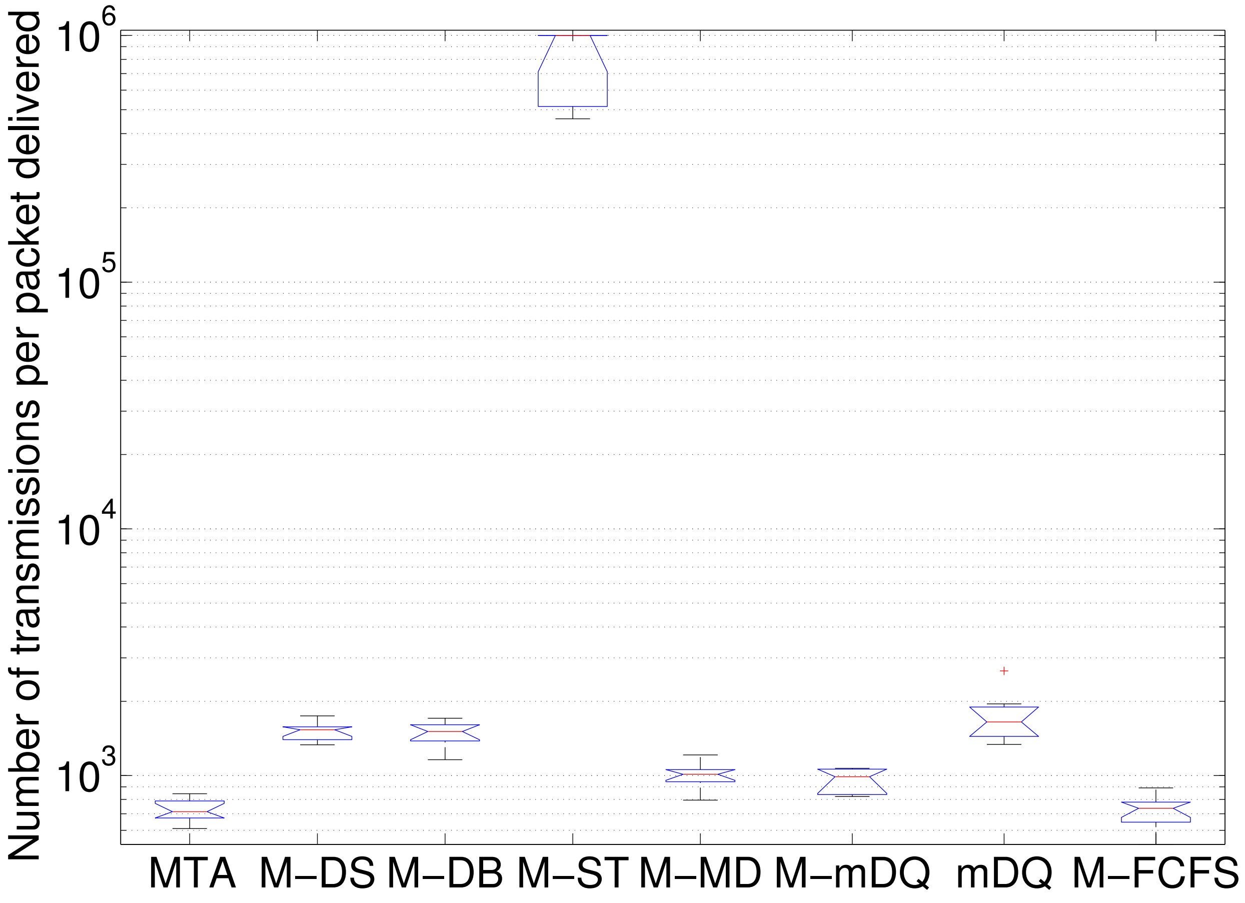


Figure 32: Number of transmissions per packet delivered: MTA and variants

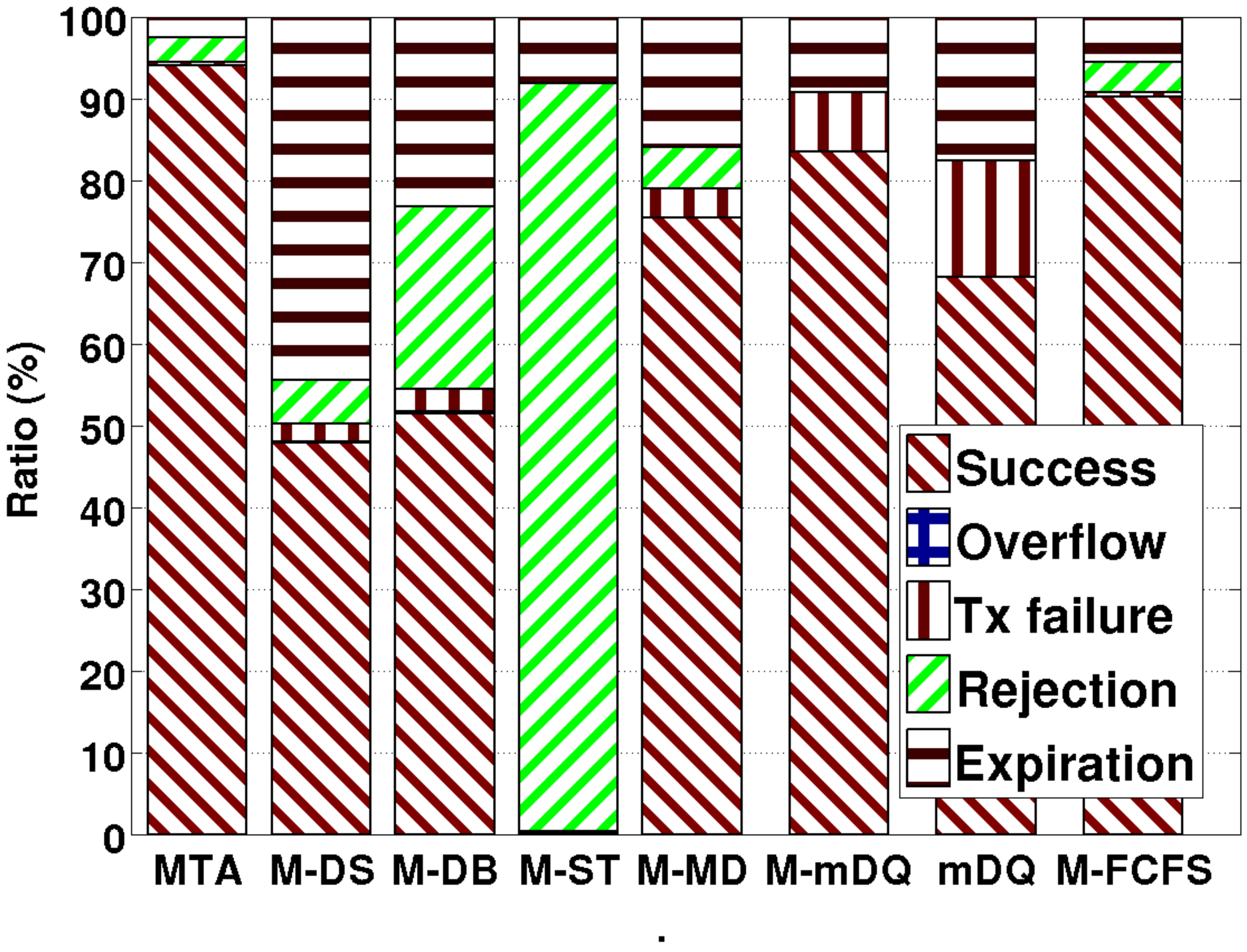


Figure 33: Packet delivery status: MTA and variants

Compared with MTE as used in MTA, M-DS, M-DB, and M-ST all underestimate path delay quantiles such that packets are routed along paths that cannot ensure the required data delivery deadline, which makes packets rejected in the network or their deadlines expire before reaching the sink node (as shown in Figure 33). For a typical four-hop path in NetEye, for instance, Figure 34 shows the empirical cumulative distribution function (CDF) of the relative error in estimating the

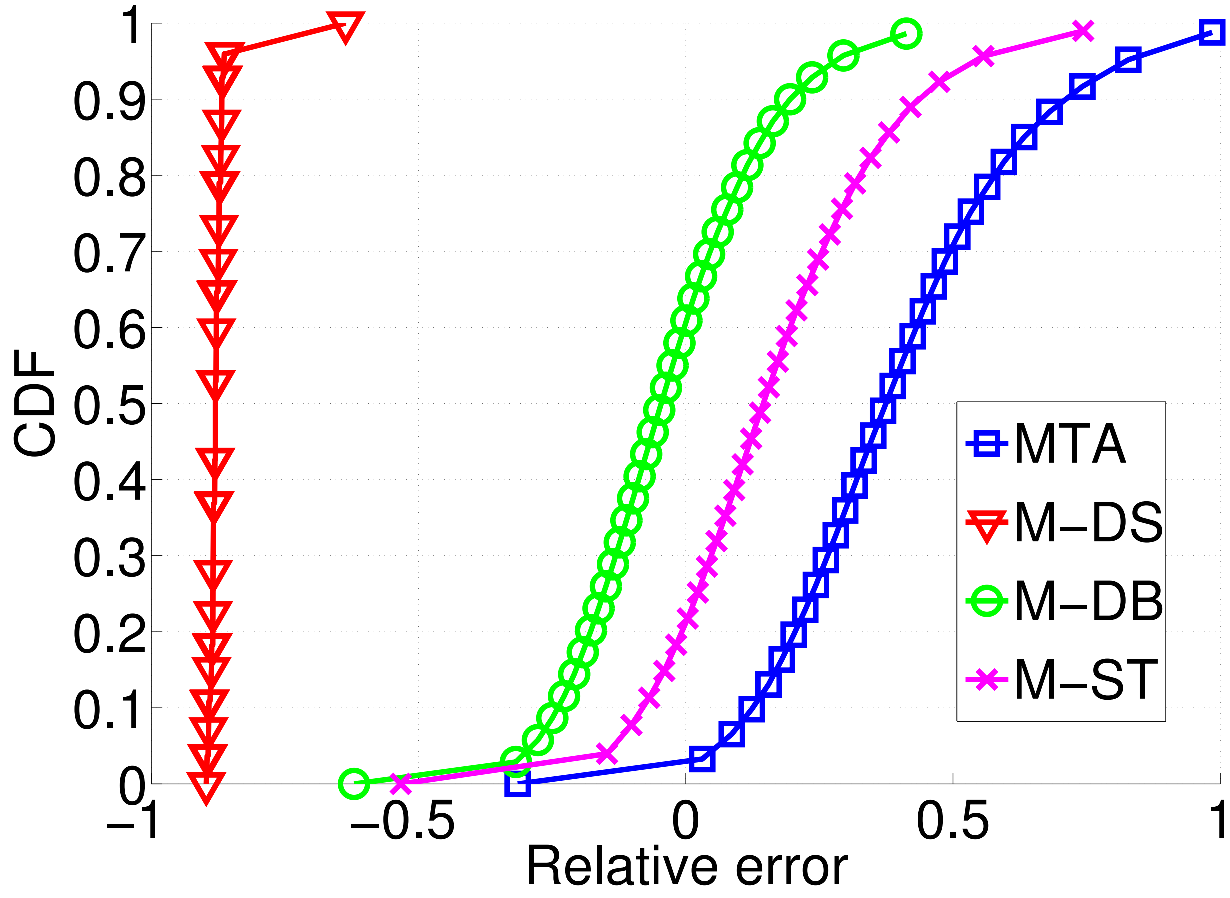


Figure 34: CDF of the relative error in estimating the 90 percentile of path delay

90 percentile of path delay using MTA, M-DS, M-DB, and M-ST respectively, where the relative error is defined as the estimated percentile minus the actual percentile and then divided by the actual percentile. We see that M-DS, M-DB, and M-ST underestimate the 90 percentile for 100%, 60%, and 20% of the time respectively. M-DS and M-DB tend to underestimate because, when sampling path delay in a distributed manner, a sample of the delay along a link can be used multiple

times due to diffusion delay; this tends to reduce the variability of the collected path delay samples, thus causing the underestimation of path delay variance which in turn leads to the underestimation of delay quantiles. M-ST tends to underestimate because the sojourn time at different nodes are positively correlated due to positive correlation in packet queueing at nodes; since the variance of the sum of the sojourn time at different nodes of a path equals to the sum of their individual variance plus their pair-wise covariance, path delay variance tends to be underestimated in M-ST, which in turn leads to the underestimation of path delay quantiles. In contrast, MTA only underestimates for 2% of time thanks to the MTE method. The reason why MTA does not completely avoid underestimation is because, just as in any possible estimation method, there are inherent errors (e.g., due to the EWMA estimator) in the MTE method that leads to small errors in estimating the mean and standard deviation of path delay which in turn can offset the conservativeness of Chebyshev inequality.

It is known that ETX-based routing enables higher delivery ratio and lower transmission cost than delay-based routing [137], thus MTA enables higher delivery ratio and lower transmission cost than M-MD, M-mDQ and mDQ. M-mDQ and mDQ use paths with minimum path delay quantiles; this introduces more route changes than in MTA, leading to larger errors in estimating path delay quantiles and thus reduces deadline success ratio. Even though M-MD, M-mDQ, and mDQ do not try to minimize path ETX as M-DS, M-DB, and M-ST do, they outperform M-DS, M-DB, and M-ST which do not use MTE as the path delay estimation method; this shows the importance of accurate, agile estimation of path delay quantiles via MTE.

The median deadline success ratio in M-FCFS is higher than the required real-time guarantee probability of 90%, but there are a few cases when the deadline success ratio is slightly below 90%. This is because, just as any estimation method, the MTE method is not perfect; it has to use estimators such as exponentially-weighted-moving-average which has inherent estimation errors, and the delay in diffusing network state (e.g., mean and variance of path delay from a node to the sink) introduces slight estimation errors in MTE. The fact that MTA always ensures a deadline success ratio of at least 90% shows that the temporal packet flow control enabled by EDF helps ameliorate the impact of the minor imperfections in MTE. Thus an interesting direction to explore is the joint optimization of spatial and temporal packet flow in real-time networking, and we will study this in more detail in our future work.

5.3 Different routing protocols

NetEye. For periodic medium traffic in NetEye, Figures 35, 36, 37, and 38 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively.

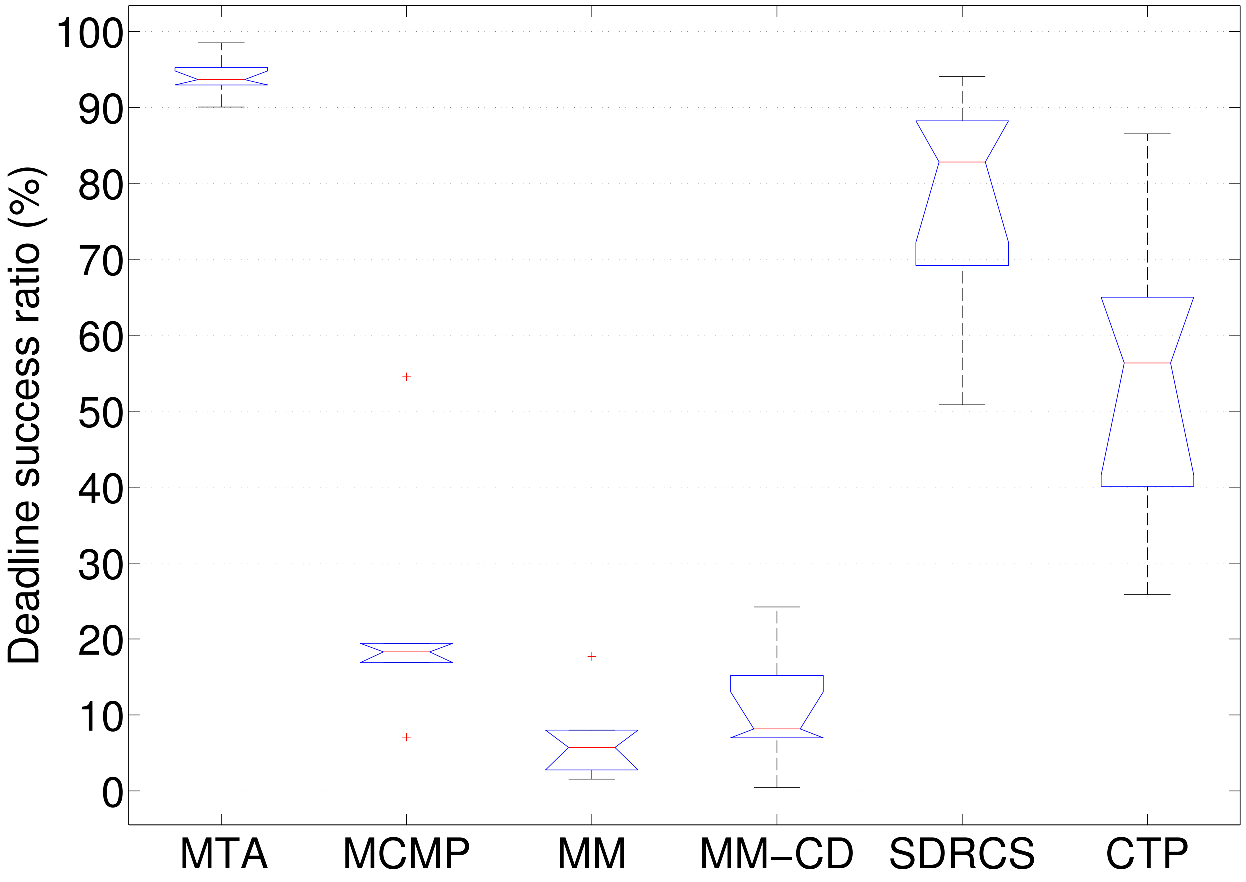


Figure 35: Deadline success ratio: MTA and existing protocols in the NetEye medium traffic scenario

We see that MTA always ensures the required real-time delivery performance, and the median deadline success ratio in MTA is 76%, 89%, 86%, 11%, and 38% higher than that in MCMP, MM, MM-CD, SDRCS, CTP respectively. The median number of transmissions per packet delivered in

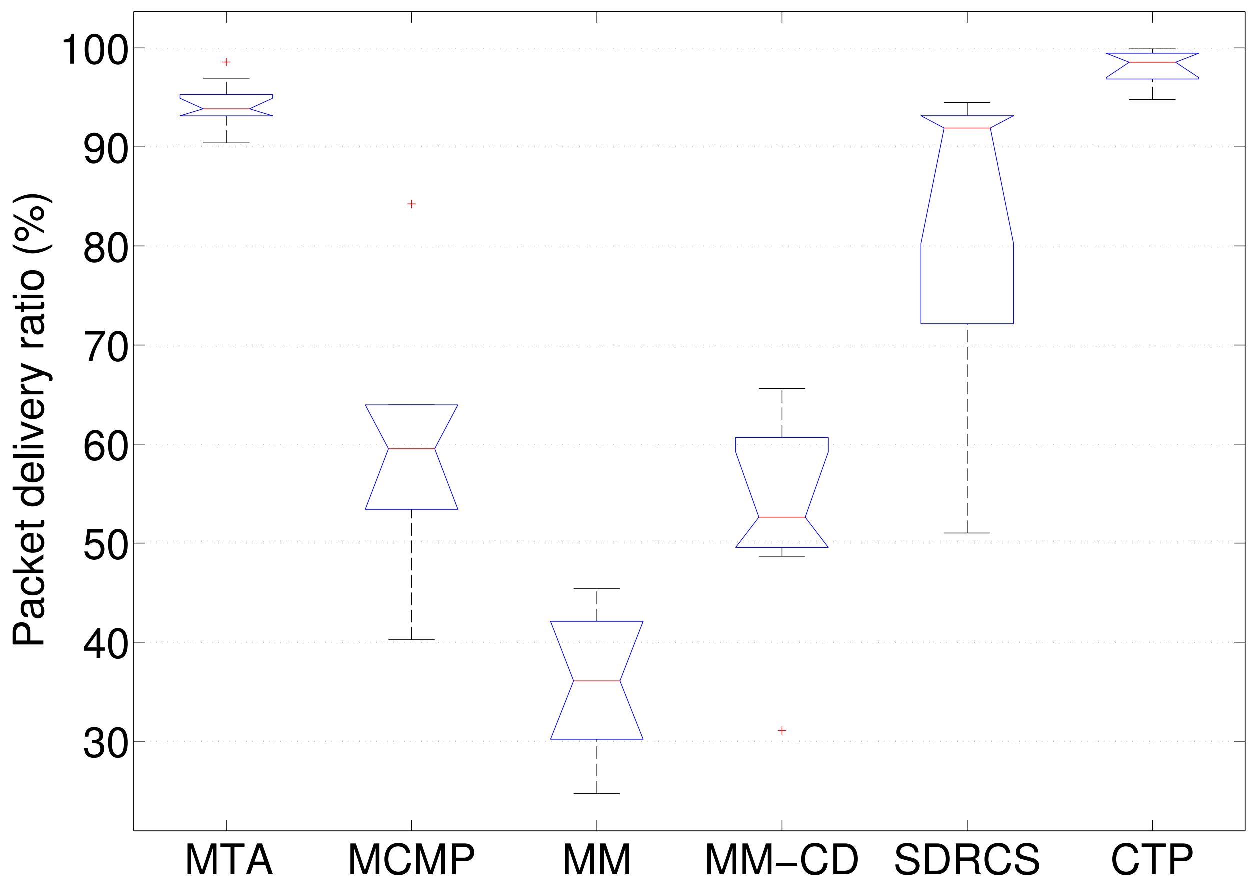


Figure 36: Packet delivery ratio: MTA and existing protocols in the NetEye medium traffic scenario

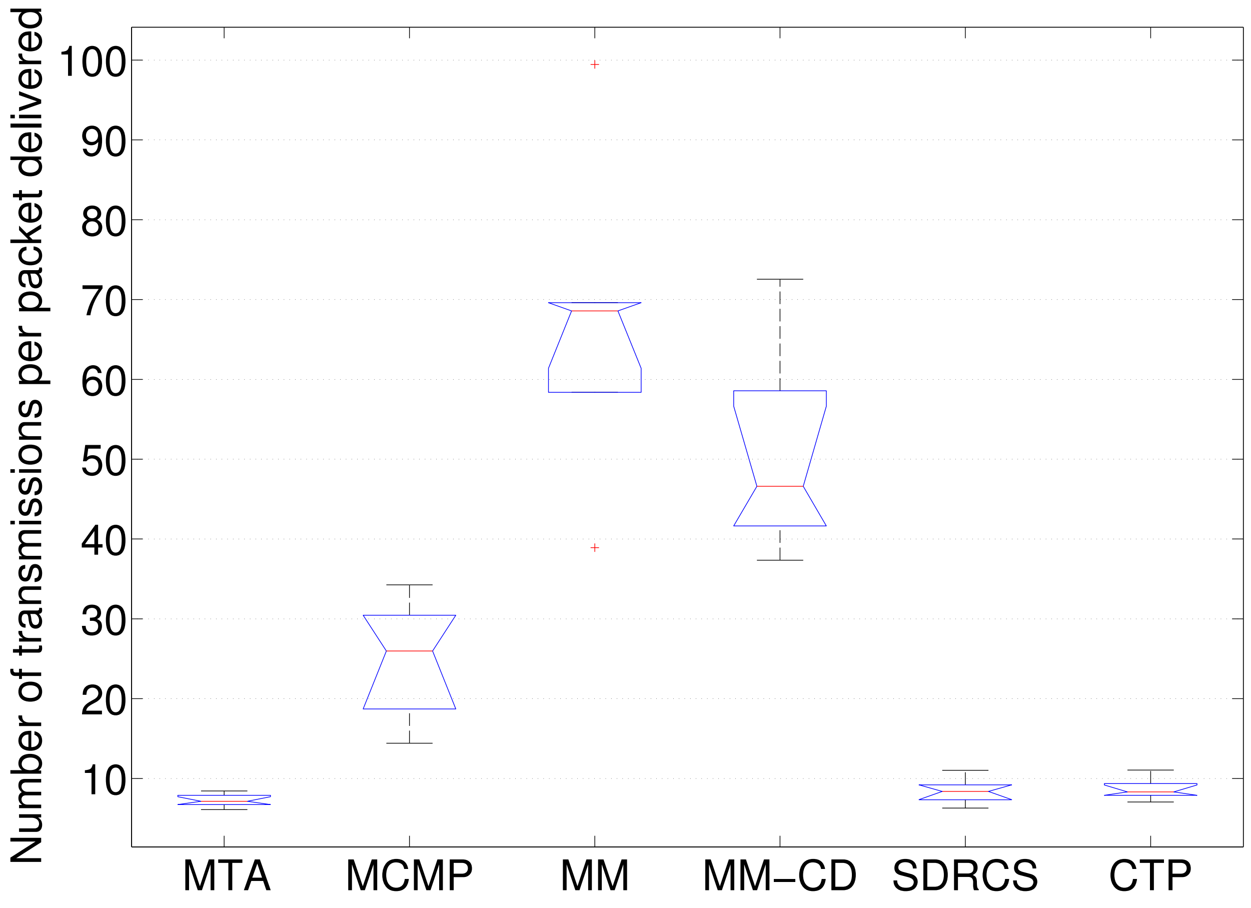


Figure 37: Number of transmissions per packet delivered: MTA and existing protocols in the NetEye medium traffic scenario

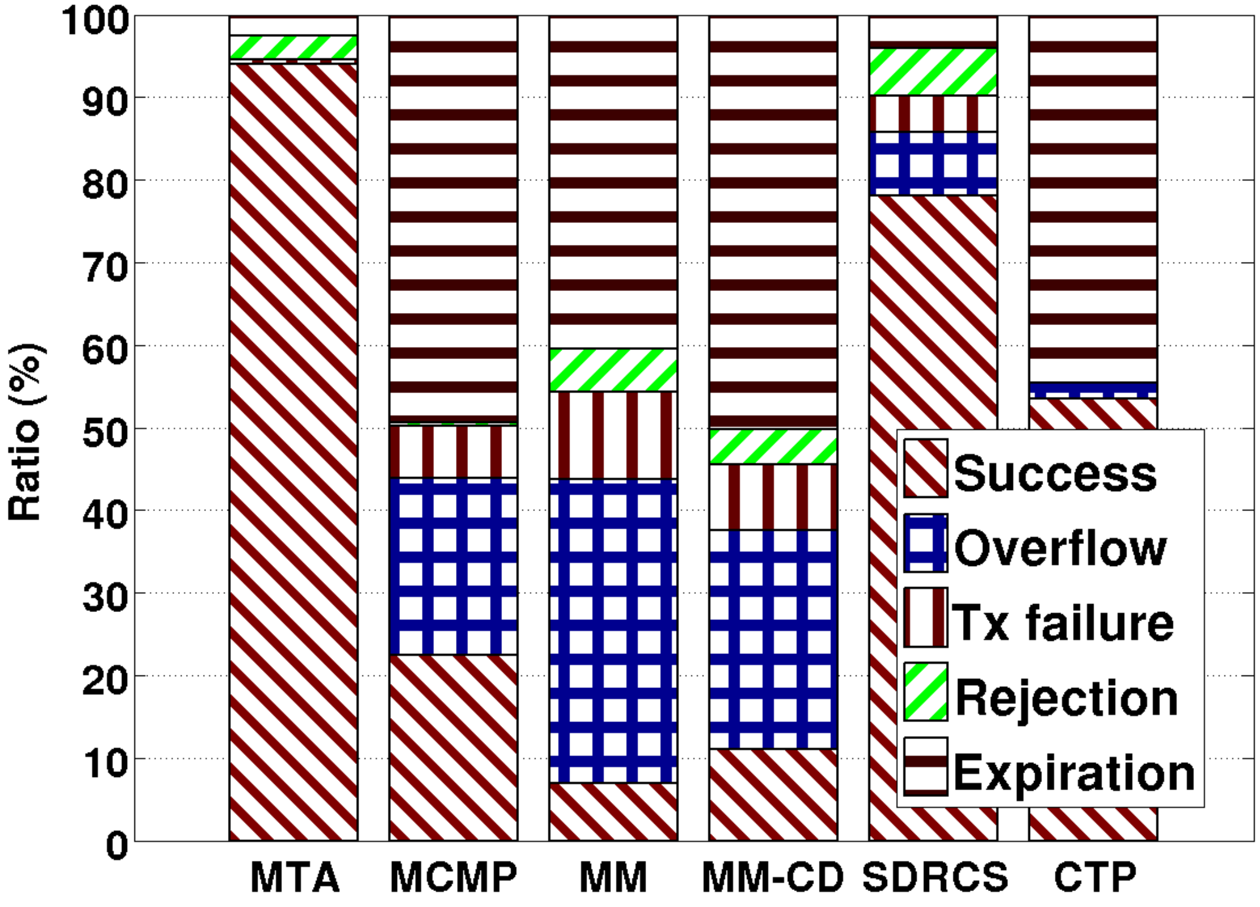


Figure 38: Packet delivery status: MTA and existing protocols in the NetEye medium traffic scenario

MTA is less than that in MCMP, MM, MM-CD, SDRCS, CTP by a factor of 3.7, 9.7, 6.5, 1.2, and 1.2 respectively.

One reason why the deadline success ratio in MCMP, MM, and MM-CD is very low is because a large fraction of packets are lost due to queue overflow as can be seen in Figure 38. They all try to use multiple paths to ensure data delivery reliability: at the sources, multiple copies of a packet can be sent; at the next hop, each of these copies can be multiplied again, and so on. This multipath routing mechanism can lead to exponentially increasing number of copies of a packet, thus causing severe queue overflow and large transmission cost as shown in Figures 38 and 37 respectively. SDRCS does not have this problem because it does not use multipath routing, instead it uses a data forwarding mechanism similar to opportunistic routing.

Another reason why MCMP, MM, MM-CD, and SDRCS do not perform well is because, by evenly dividing end-to-end QoS requirements into per-hop requirements, they implicitly assume that network conditions are uniform across the network which is usually not the case. Among these protocols, the negative impact of this assumption is relatively less severe in SDRCS because it uses signal strength as the basis of measuring forwarding distances and signal strength is a better metric for measuring wireless link quality than geographic distance. MTA does not have this problem because MTE enables accurate, agile estimation of end-to-end delay quantiles without assuming uniform network conditions. A third reason for the low performance of MM, MM-CD, and SDRCS is because they only consider mean delays instead of the probabilistic distributions of delays.

Compared with MTA, CTP has higher packet delivery ratio, but CTP only enables a median deadline success ratio of 56% which is lower than the real-time guarantee probability of 90% and much lower than the 93% probability guarantee by MTA. This is because CTP only considers path ETX in routing, and it is delay-unaware. Even if a low-ETX path is experiencing large delay due to queueing, CTP still uses the path, thus leading to large data delivery delay and deadline miss. Through accurate, agile estimation of path delay via MTE, in contrast, MTA can switch to a less congested path whenever it detects the inability of the current low-ETX path to deliver packets before their deadlines, creating the water-filling effect as we have discussed in Section 4. For instance, Figure 39 shows the histogram of the ETX of the paths taken by all the packets. We see

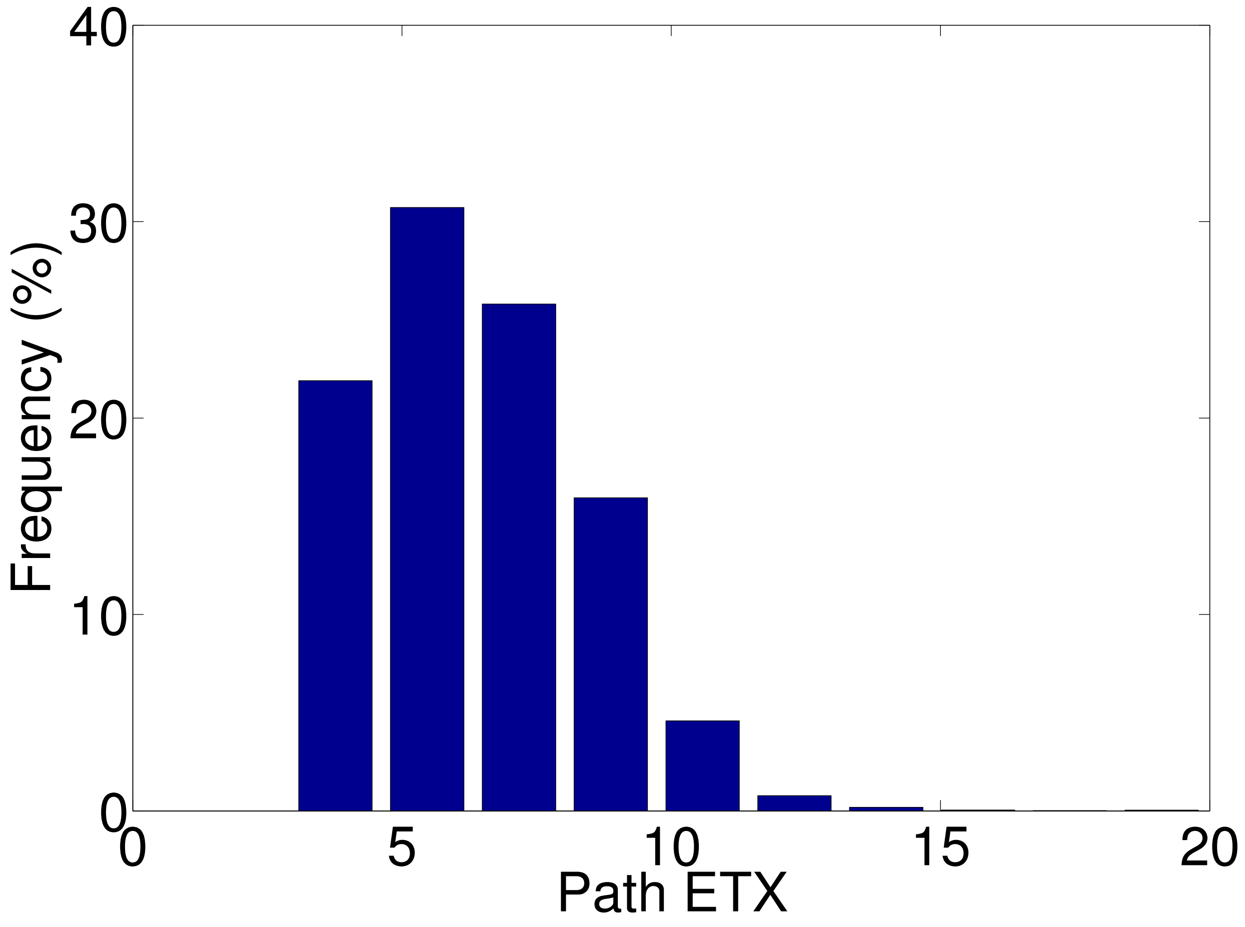


Figure 39: Histogram of the ETX of the paths taken by all the packets

that MTA tends to use paths of lower-ETX with higher probability, even though the minimum-ETX path will not always be used with the highest probability (e.g., when the capacity of the minimum-ETX path is reduced due to the shared path segments with other paths).

For periodic light traffic in NetEye, Figures 40, 41, 42, and 43 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively. The results are similar to those of periodic medium traffic except that CTP's performance is now indistinguishable from MTA's while it is worse than MTA's in medium traffic. Both protocols yield close to 100% packet delivery and deadline success and similar transmission costs. This is because, in light traffic, there is almost no queueing. Hence, a path's ETX is a good indicator of its delay, and MTA tends to choose minimal ETX path as CTP does, resulting in similar performance.

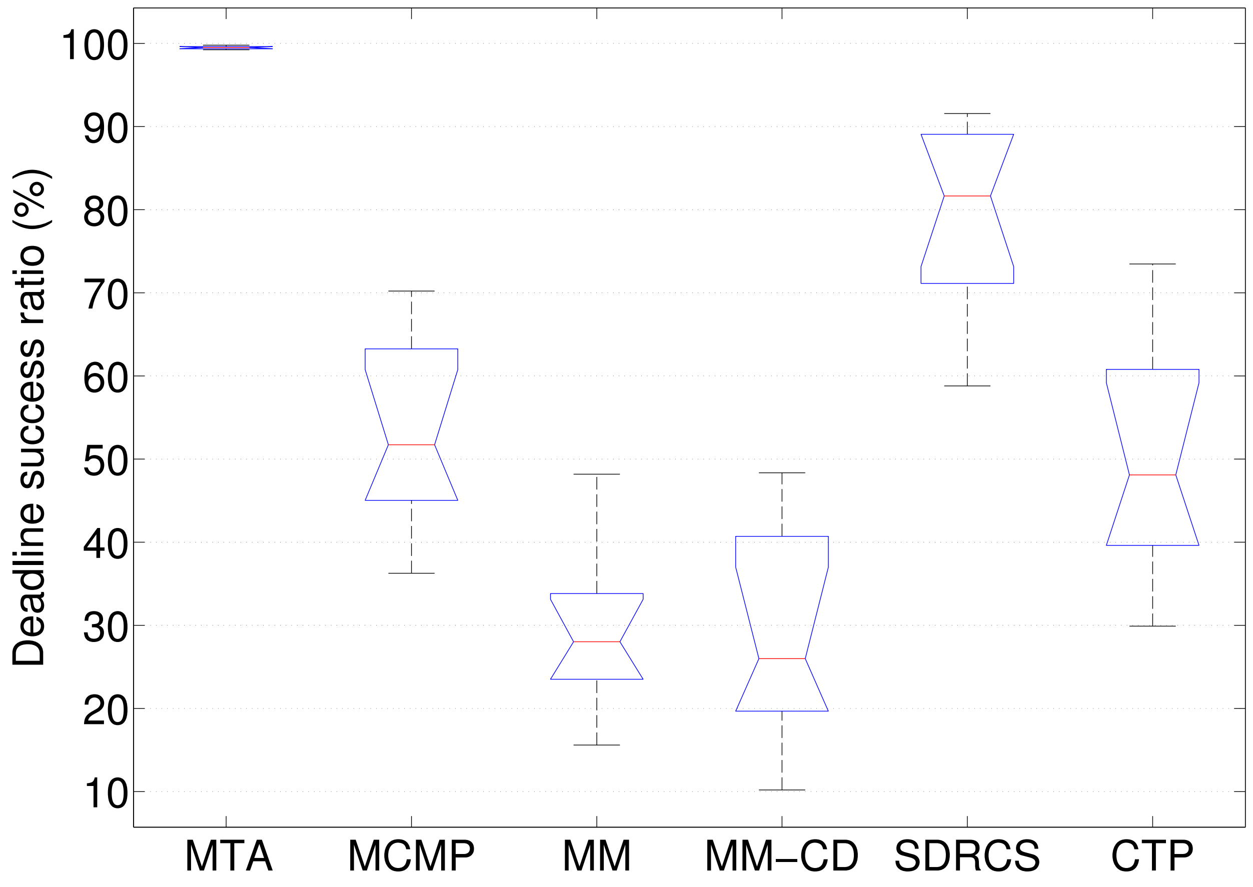


Figure 40: Deadline success ratio: MTA and existing protocols in the NetEye light traffic scenario

For periodic heavy traffic in NetEye, Figures 44, 45, 46, and 47 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively. Again, the results are similar to those in medium traffic, and MTA consistently guarantees deadline success ratio over the required 90%. In CTP, however, more packets are lost due to queue overflow in the presence of heavy traffic, thus leading to a median packet delivery ratio of around 80% (compared with its packet delivery ratio of around 100% in medium traffic).

For event traffic in NetEye, Figures 48, 49, 50, and 51 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively. The comparative results closely resembles those in medium traffic, and MTA ensures a deadline success ratio greater than the required 90%.

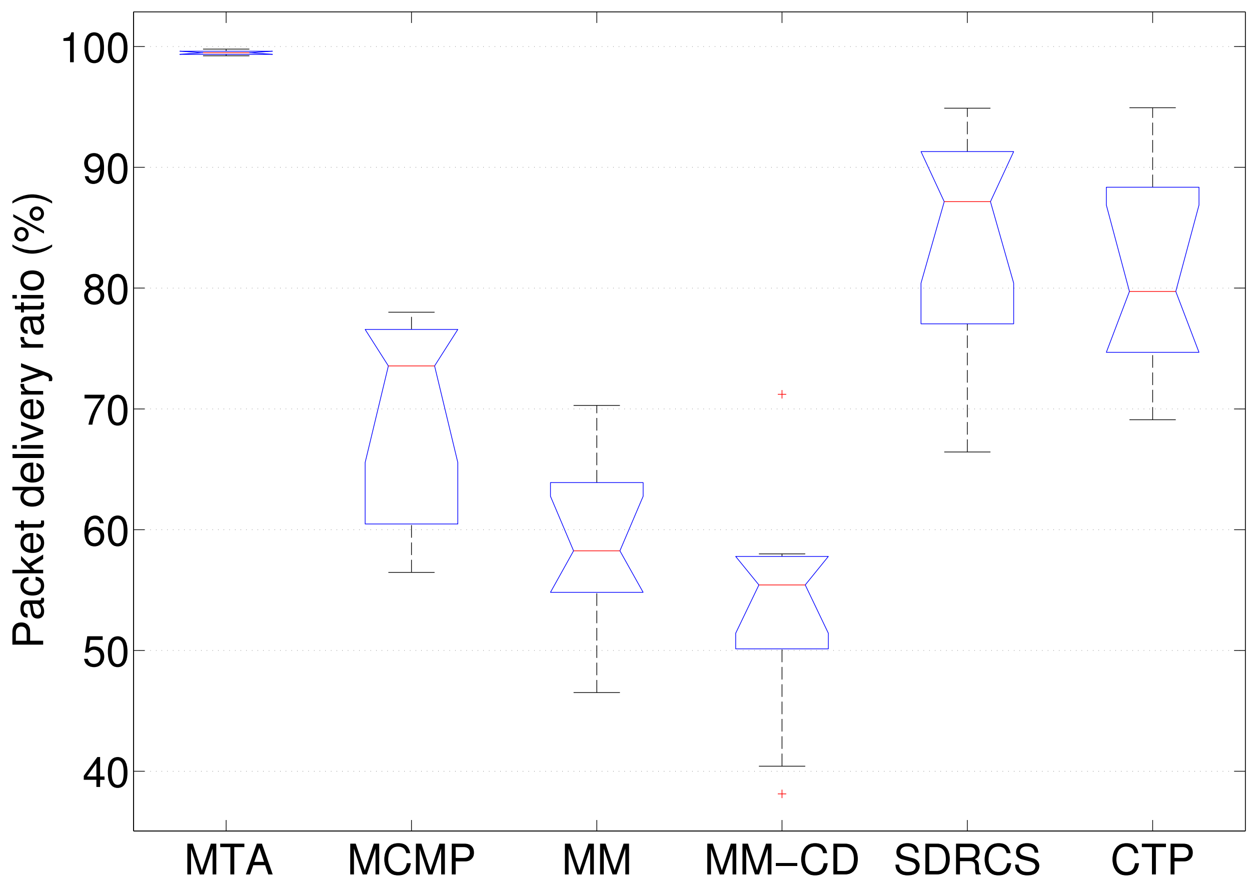


Figure 41: Packet delivery ratio: MTA and existing protocols in the NetEye light traffic scenario

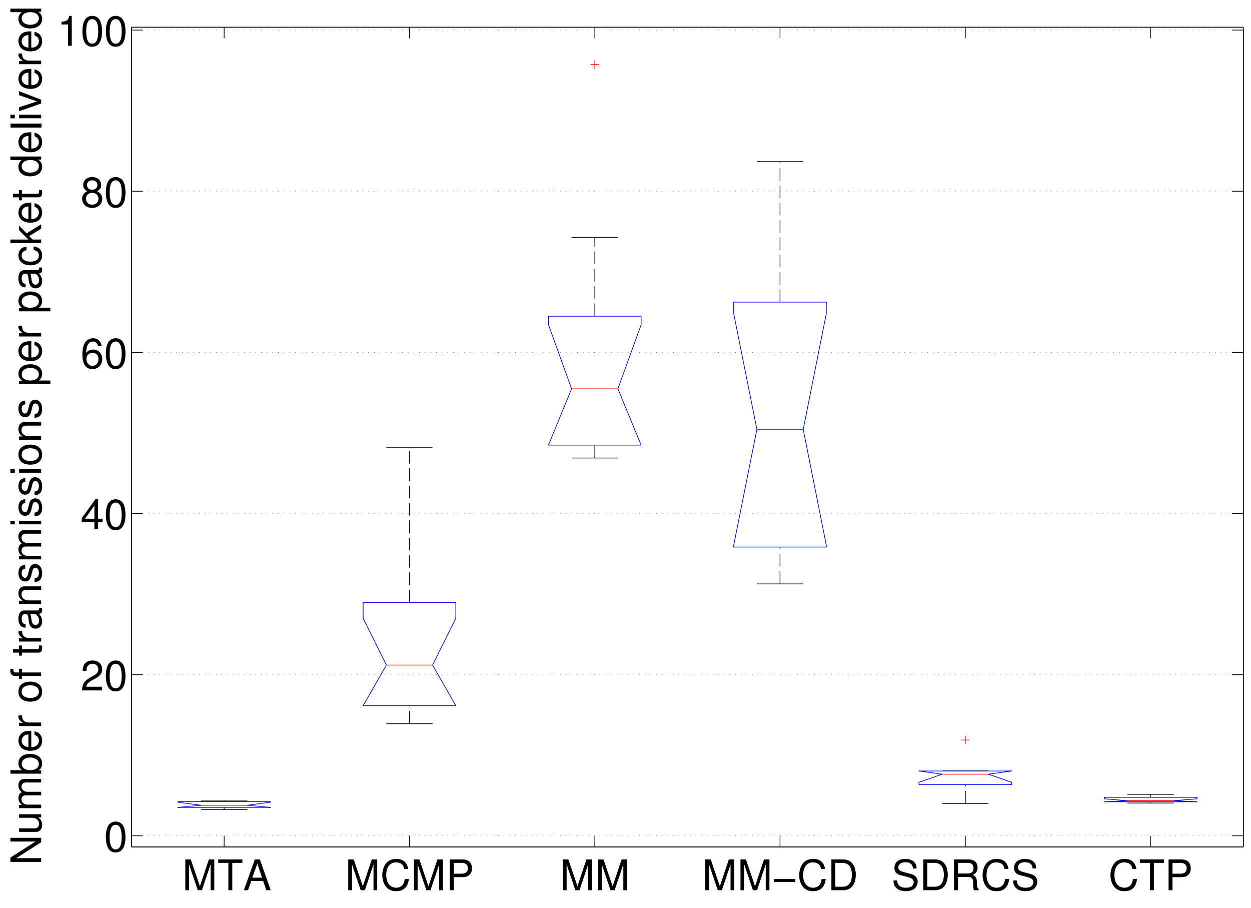


Figure 42: Number of transmissions per packet delivered: MTA and existing protocols in the NetEye light traffic scenario

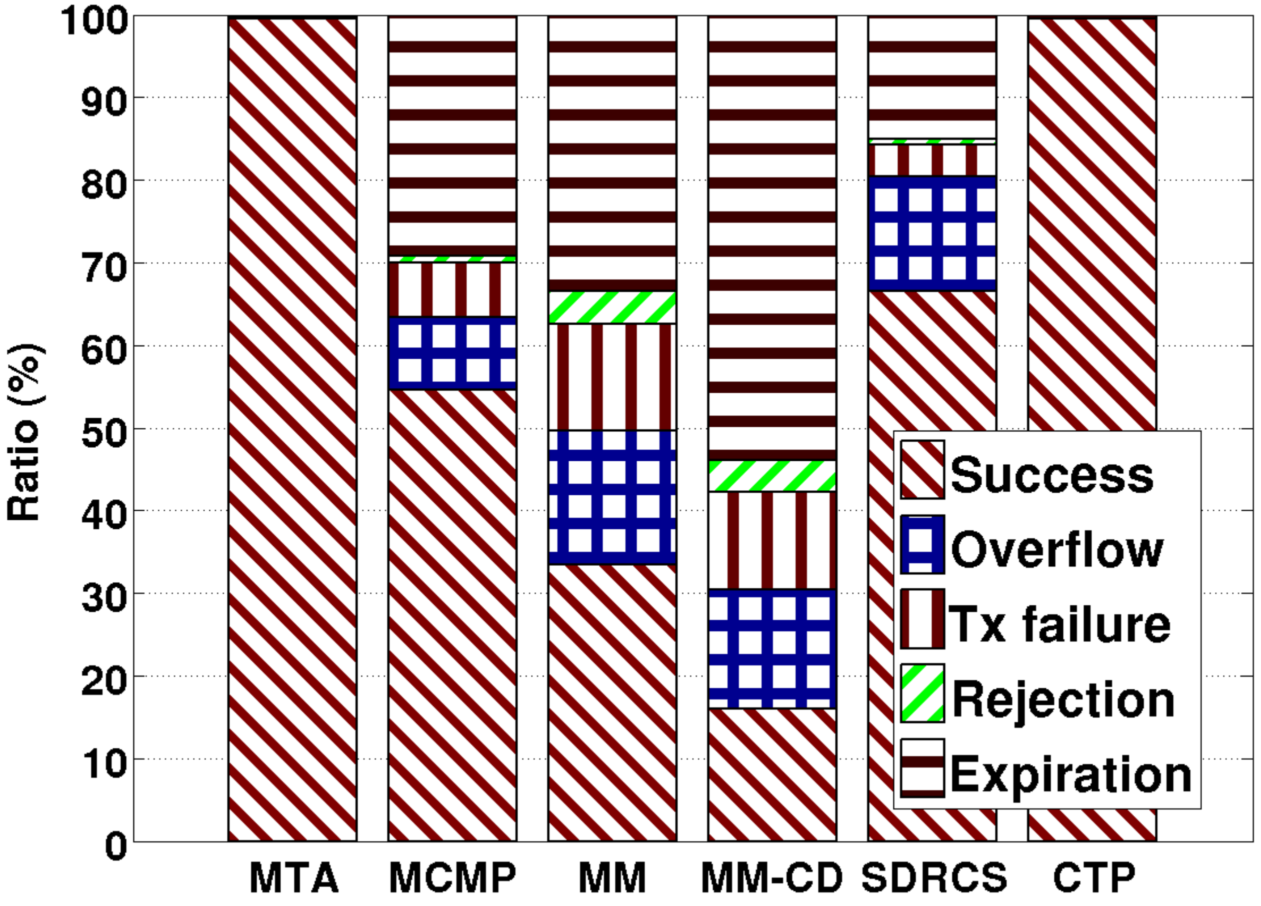


Figure 43: Packet delivery status: MTA and existing protocols in the NetEye light traffic scenario

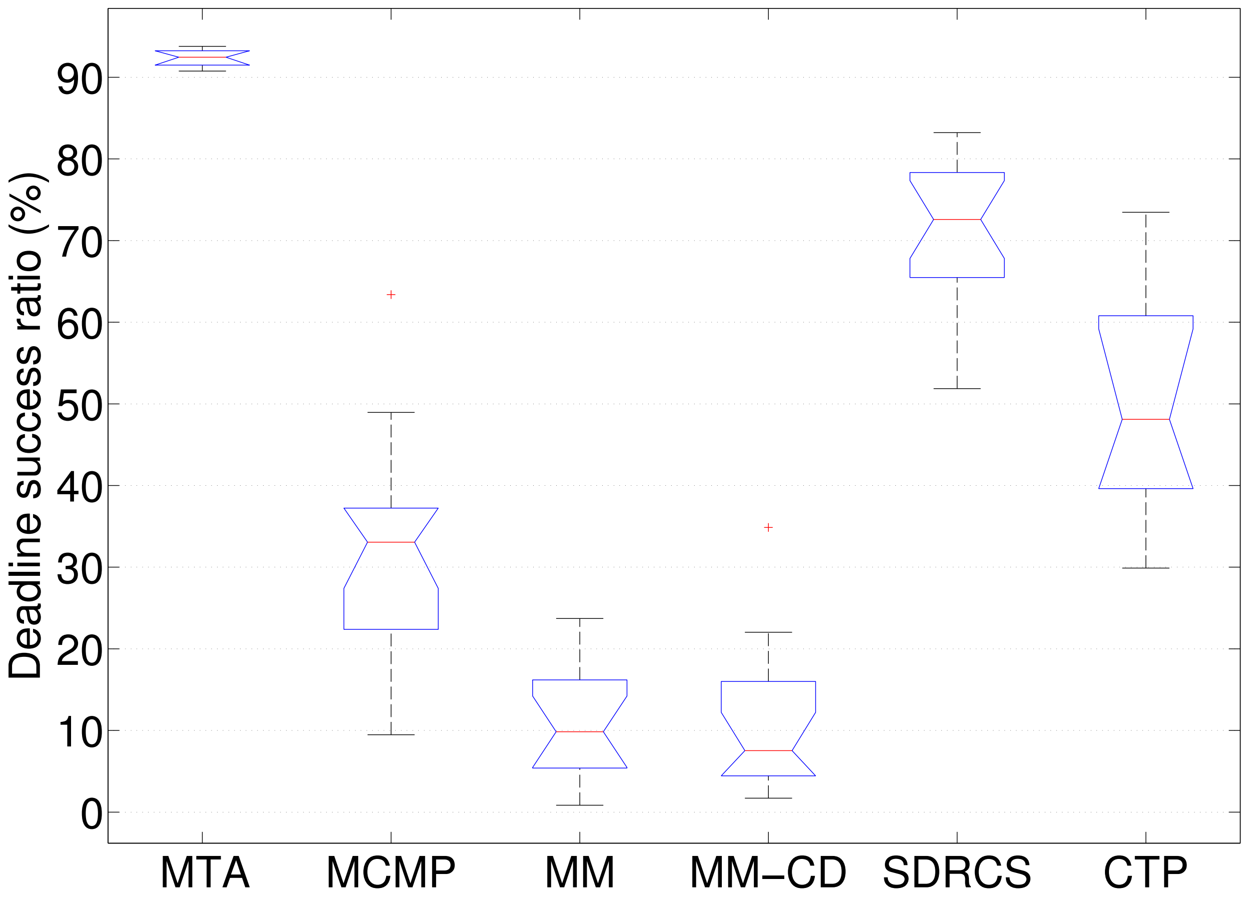


Figure 44: Deadline success ratio: MTA and existing protocols in the NetEye heavy traffic scenario

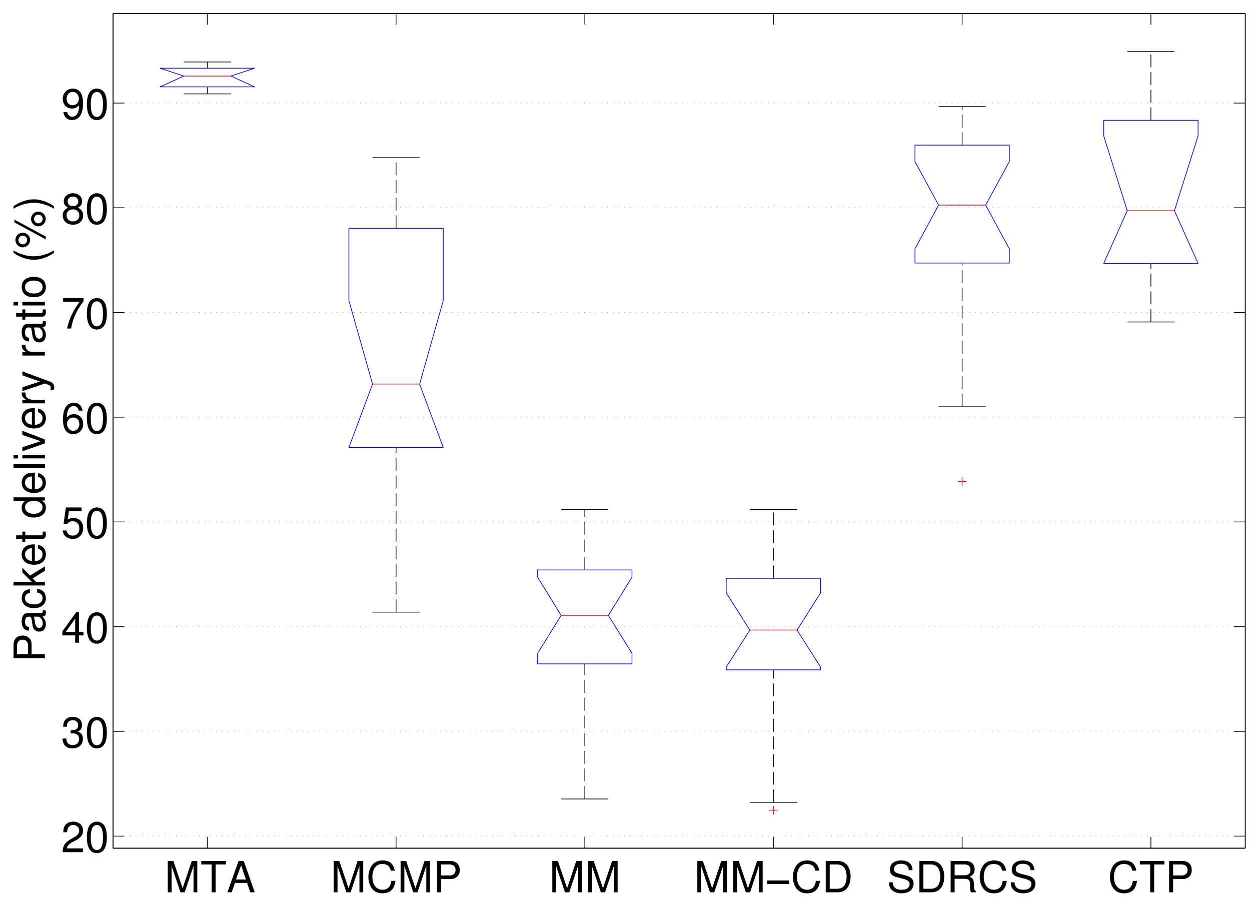


Figure 45: Packet delivery ratio: MTA and existing protocols in the NetEye heavy traffic scenario

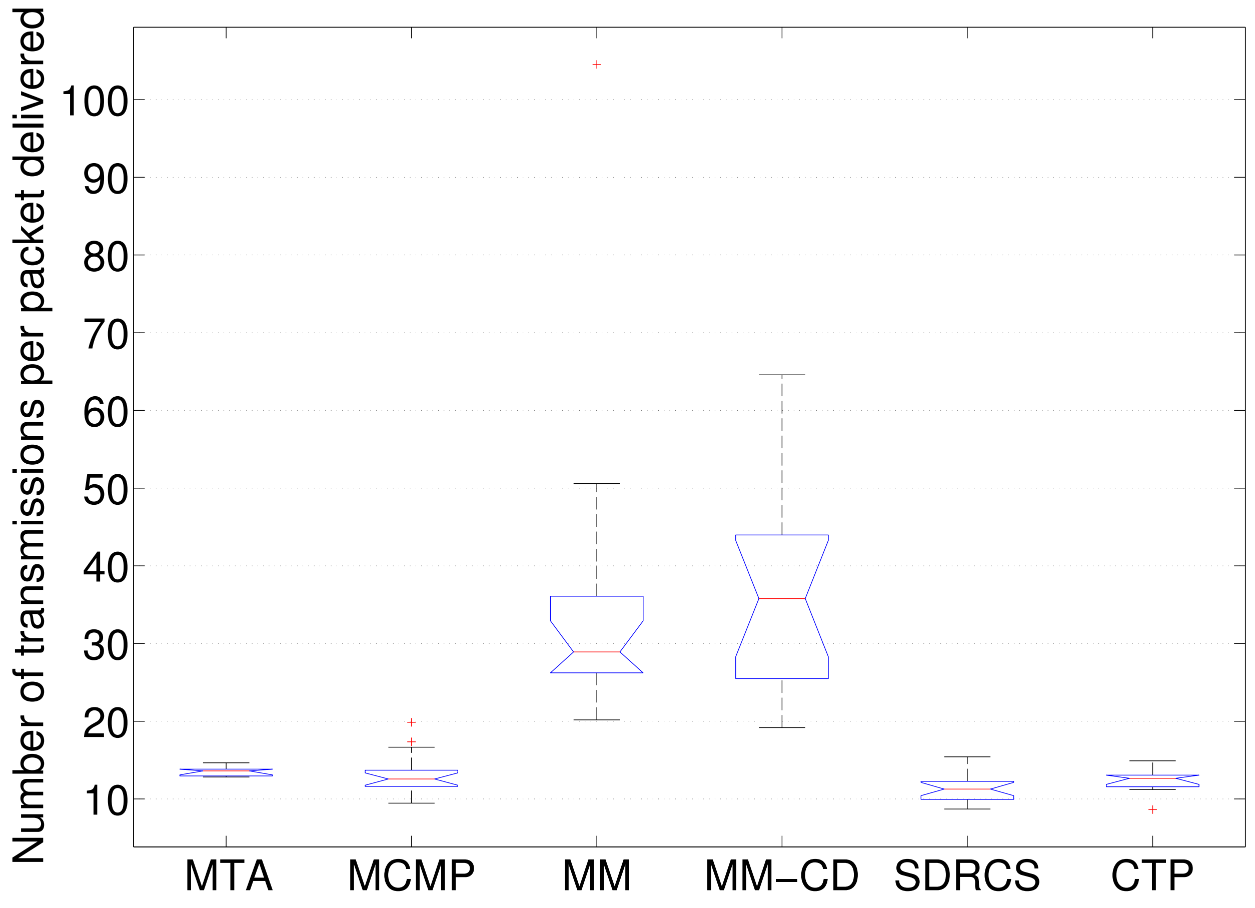


Figure 46: Number of transmissions per packet delivered: MTA and existing protocols in the NetEye heavy traffic scenario

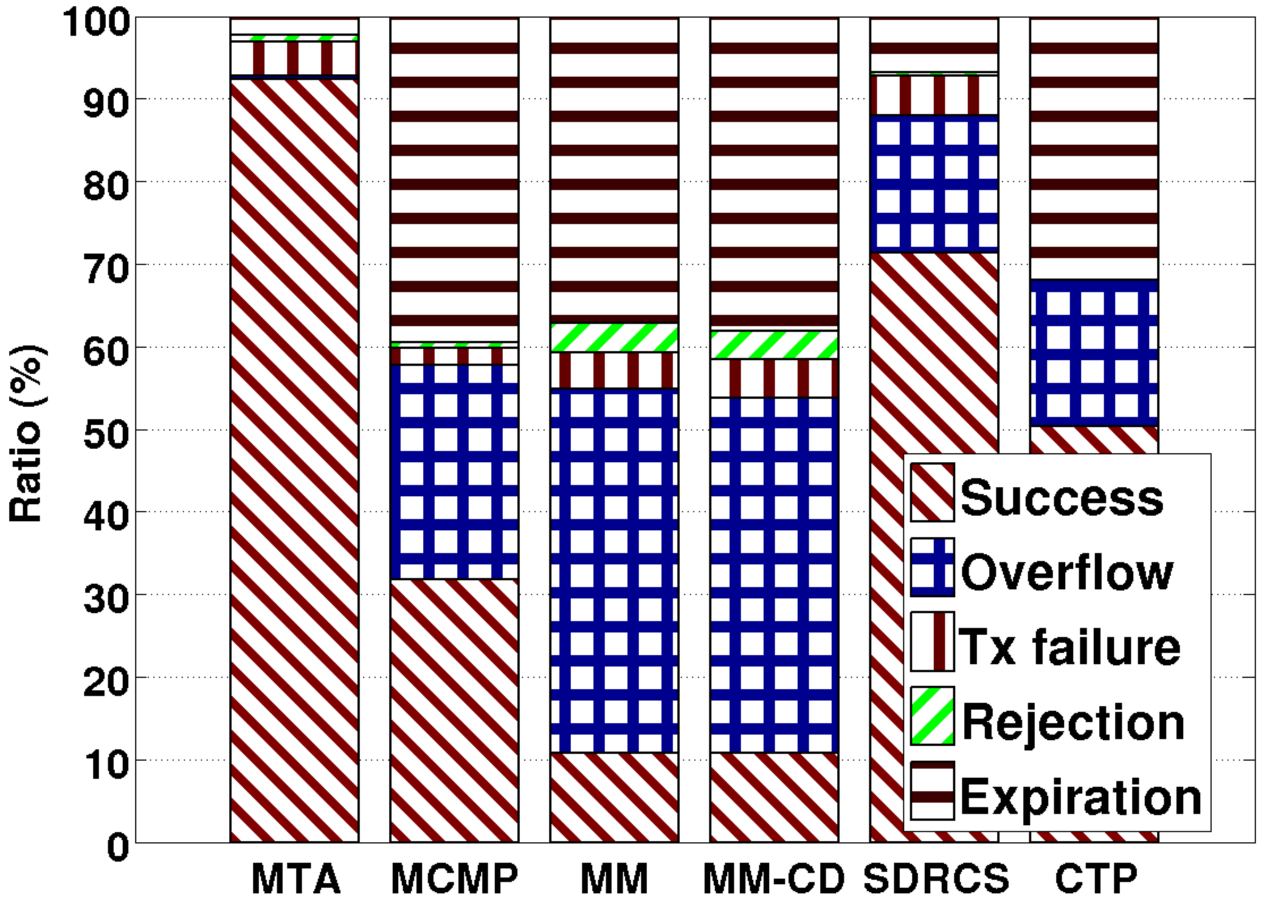


Figure 47: Packet delivery status: MTA and existing protocols in the NetEye heavy traffic scenario

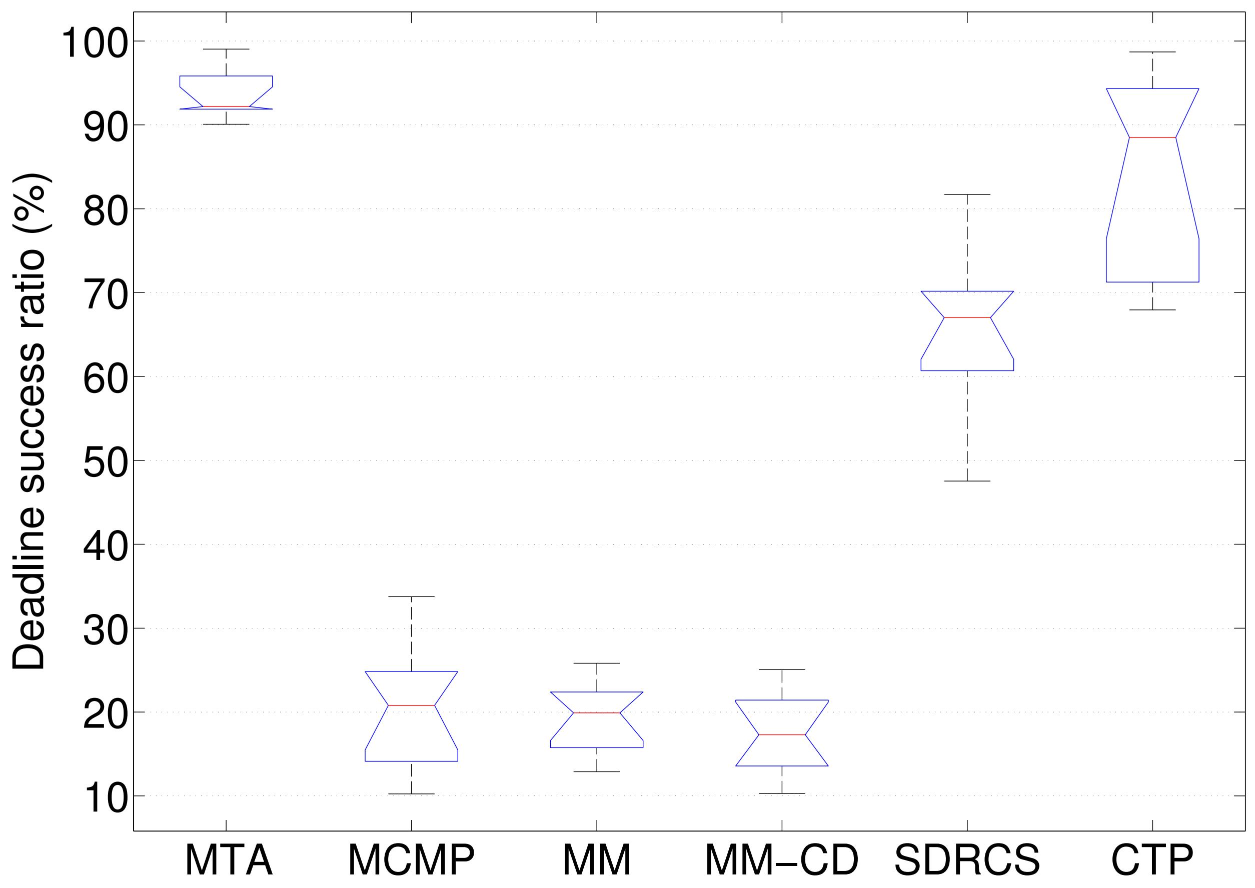


Figure 48: Deadline success ratio: MTA and existing protocols in the NetEye event traffic scenario

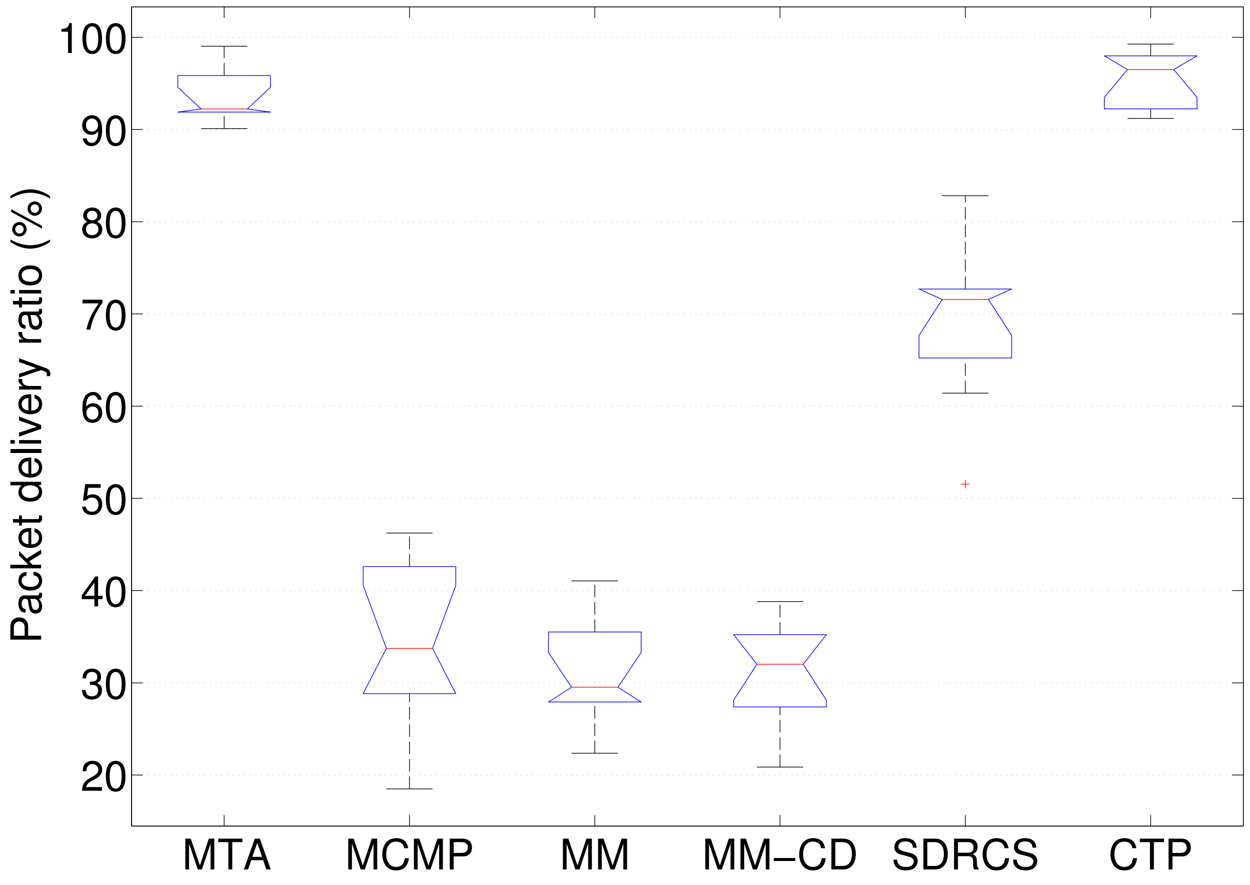


Figure 49: Packet delivery ratio: MTA and existing protocols in the NetEye event traffic scenario

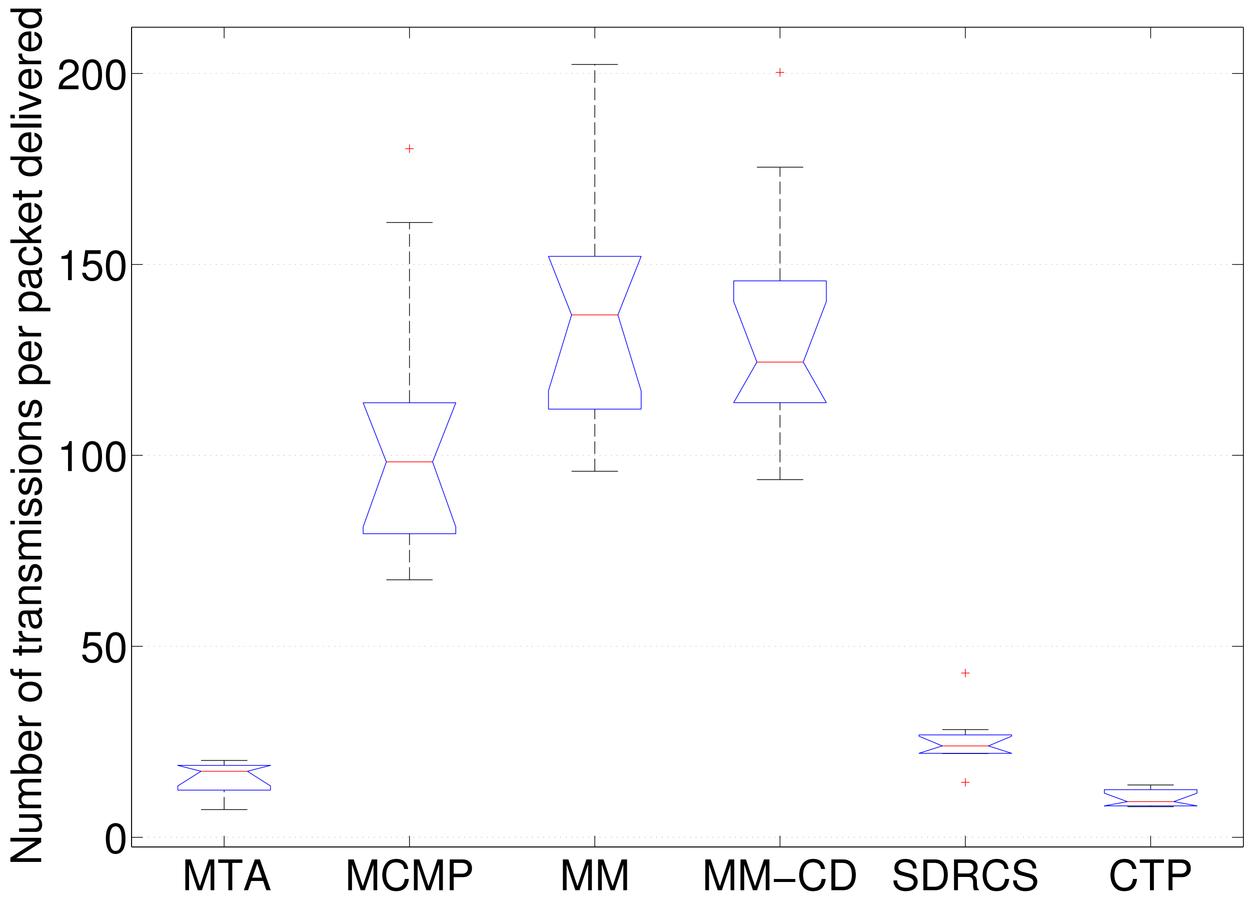


Figure 50: Number of transmissions per packet delivered: MTA and existing protocols in the NetEye event traffic scenario

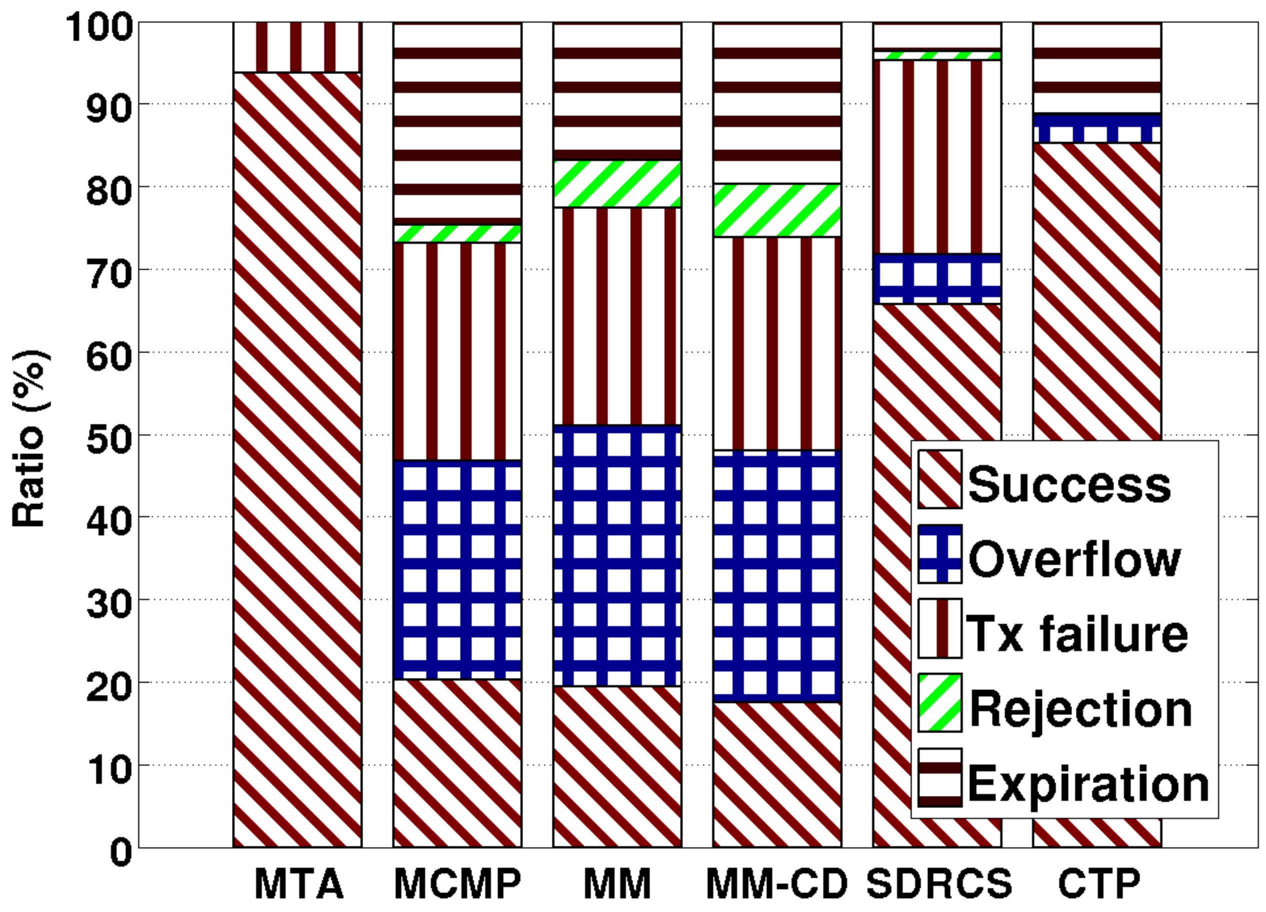


Figure 51: Packet delivery status: MTA and existing protocols in the NetEye event traffic scenario

All the previous traffic set the required real-time probability as 90%. To demonstrate that MTA works for arbitrarily required probability guarantee, we raise the probability to 99% and set the corresponding deadline as 7.5 seconds. For the same periodic medium traffic in NetEye, Figures 52, 53, 54, and 55 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively. Even the required probability guarantee of 99% is more stringent than 90%, Figure 52 shows that MTA still consistently meets the deadline. All the relative performance comparisons are similar to those of periodic medium traffic with required probability 90%.

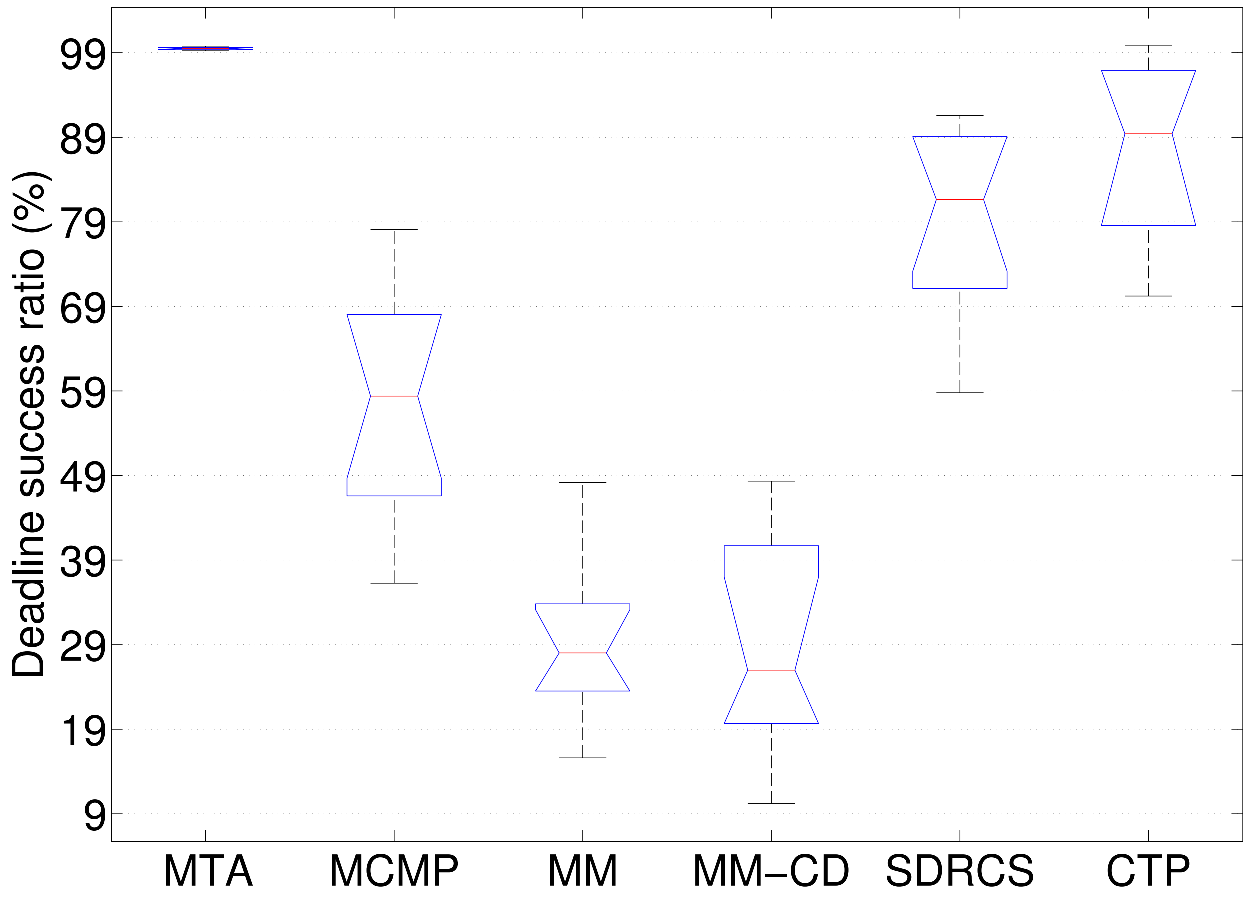


Figure 52: Deadline success ratio: MTA and existing protocols of under medium traffic (real-time probability 99%) in NetEye

Indriya. Figures 56, 57, 58, and 59 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status for periodic traffic in Indriya respectively. The

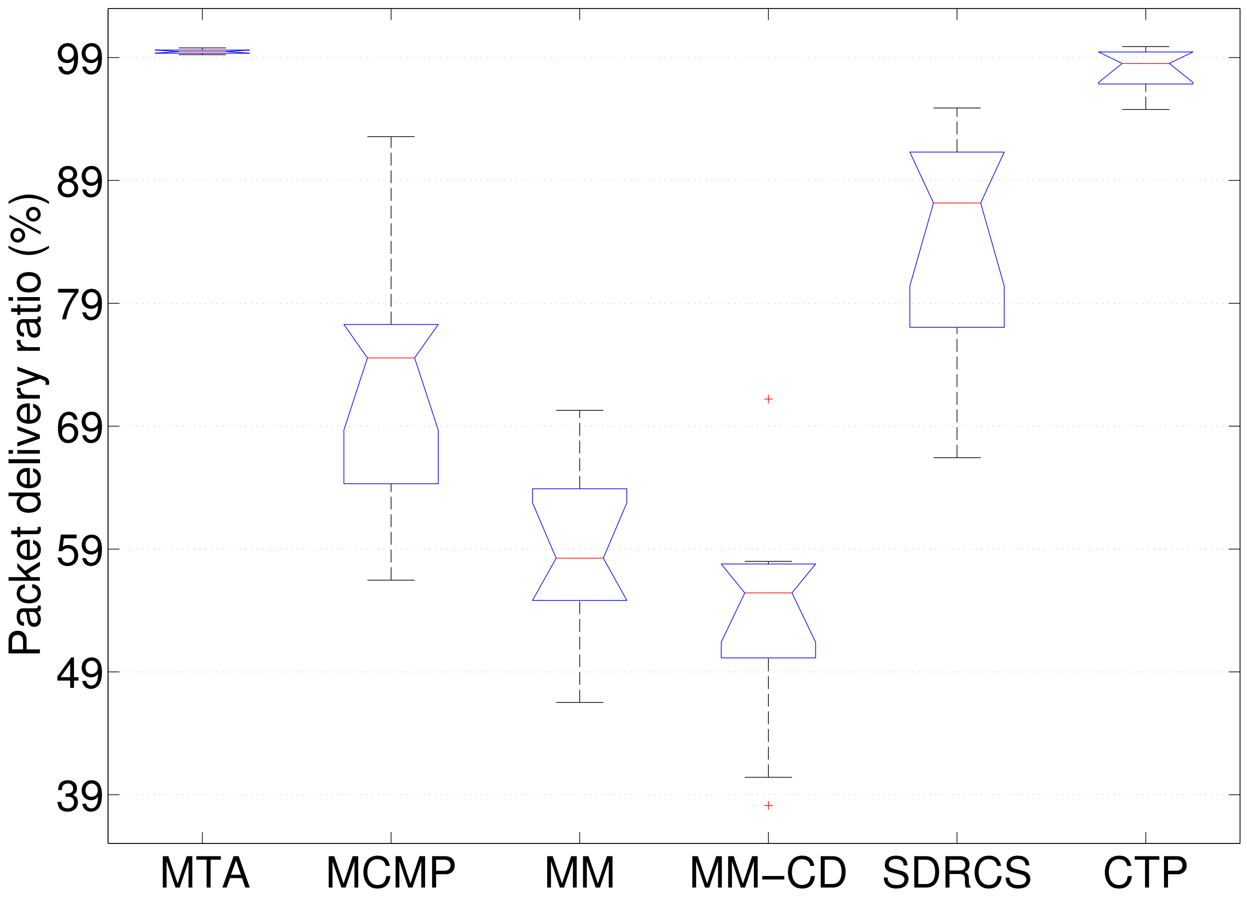


Figure 53: Packet delivery ratio: MTA and existing protocols under medium traffic (real-time probability 99%) in NetEye

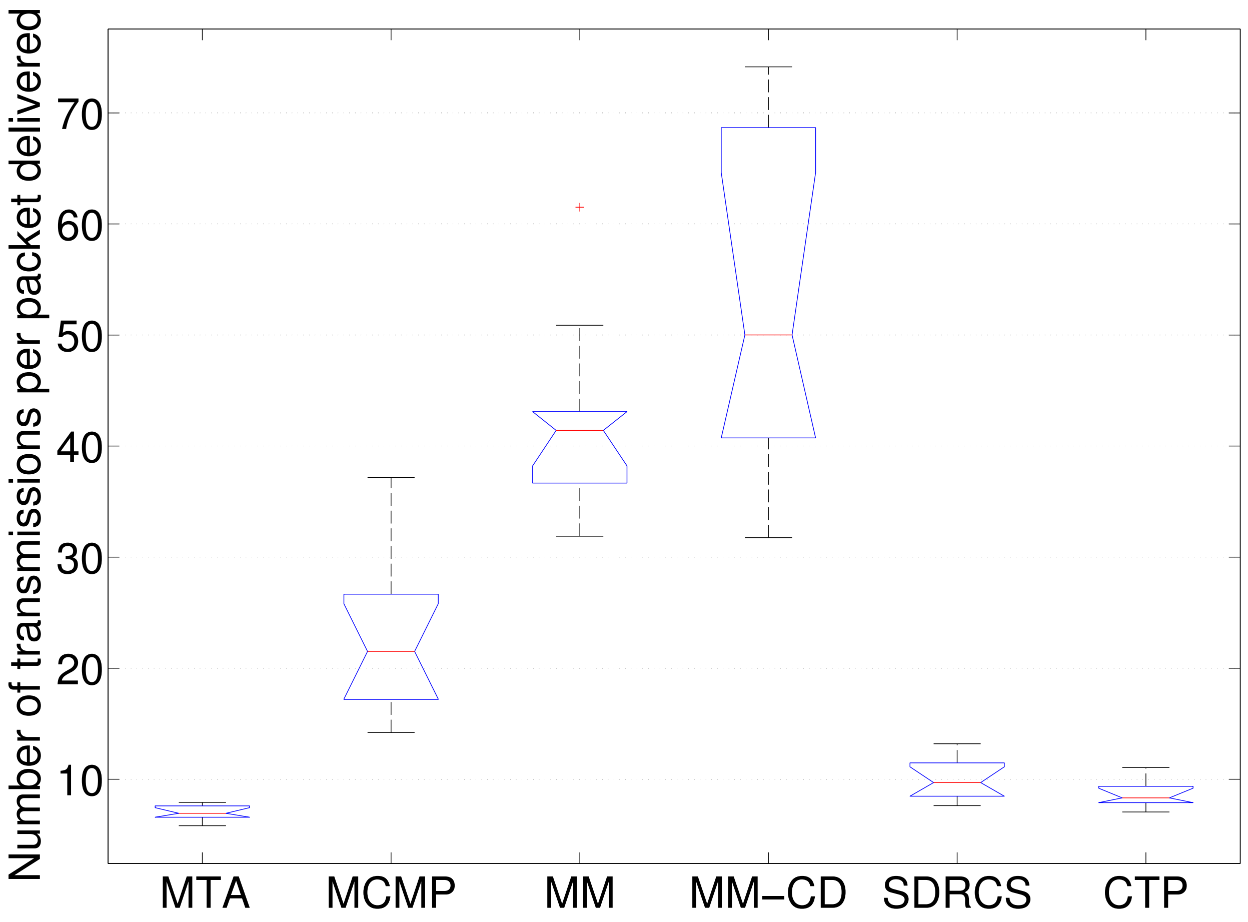


Figure 54: Number of transmissions per packet delivered: MTA and existing protocols under medium traffic (real-time probability 99%) in NetEye

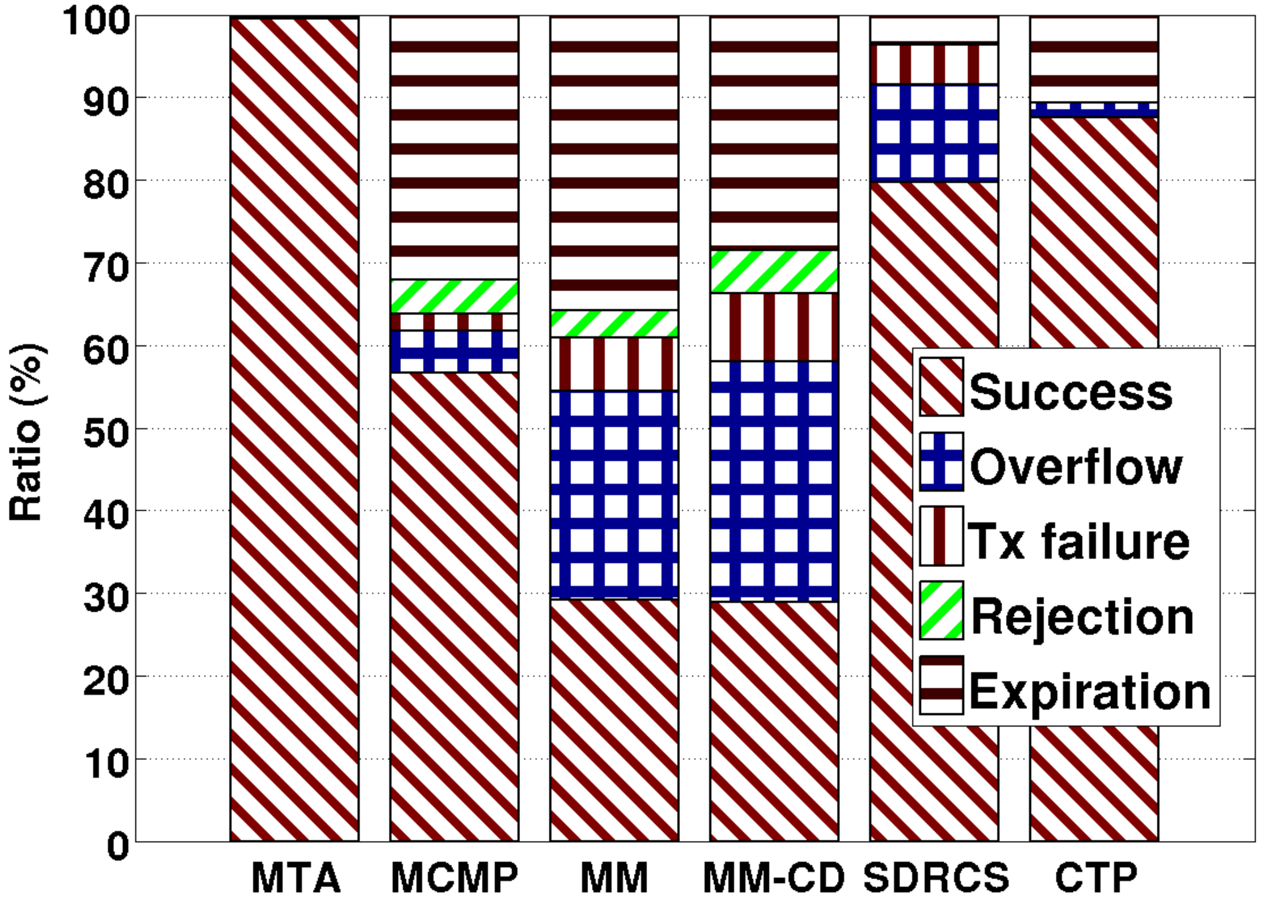


Figure 55: Packet delivery status: MTA and existing protocols under medium traffic (real-time probability 99%) in NetEye

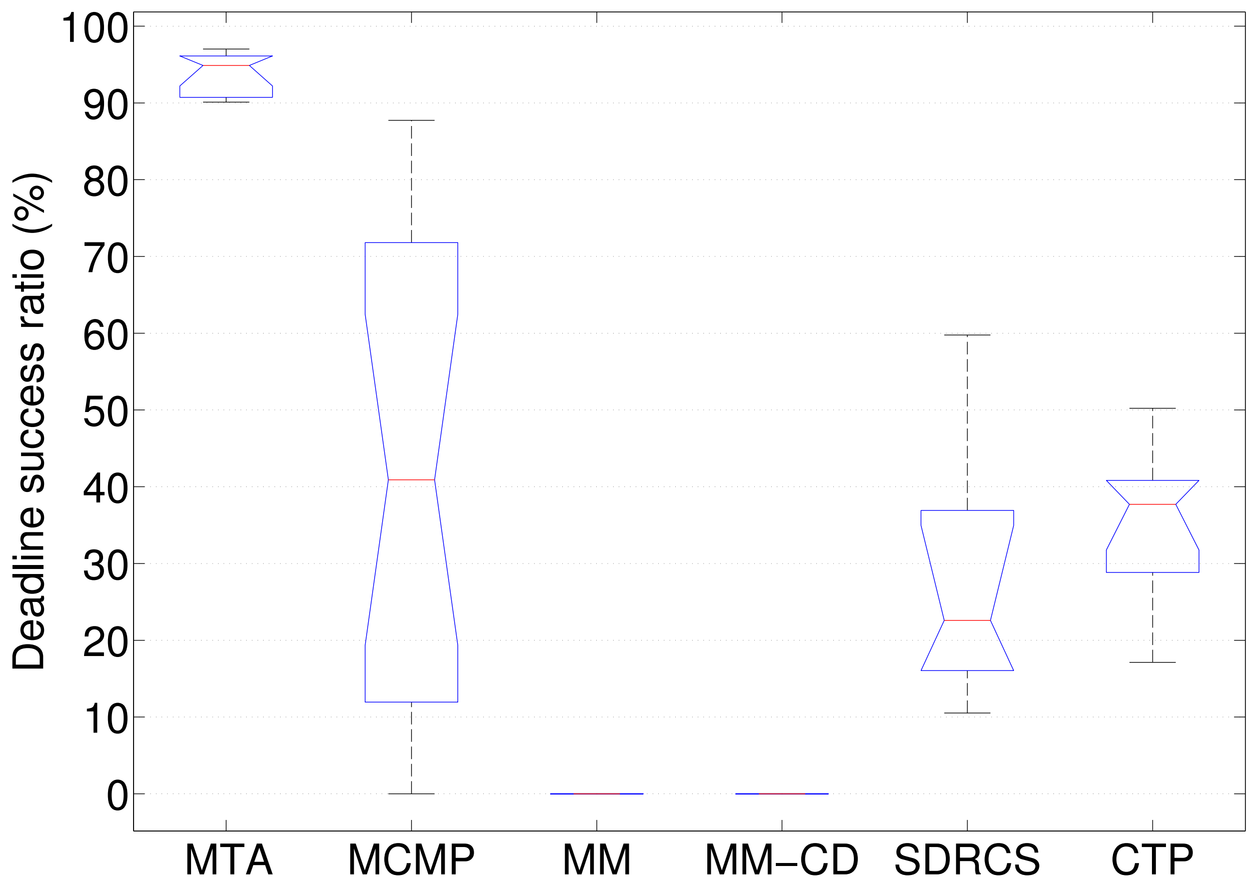


Figure 56: Deadline success ratio: MTA and existing protocols in Indriya

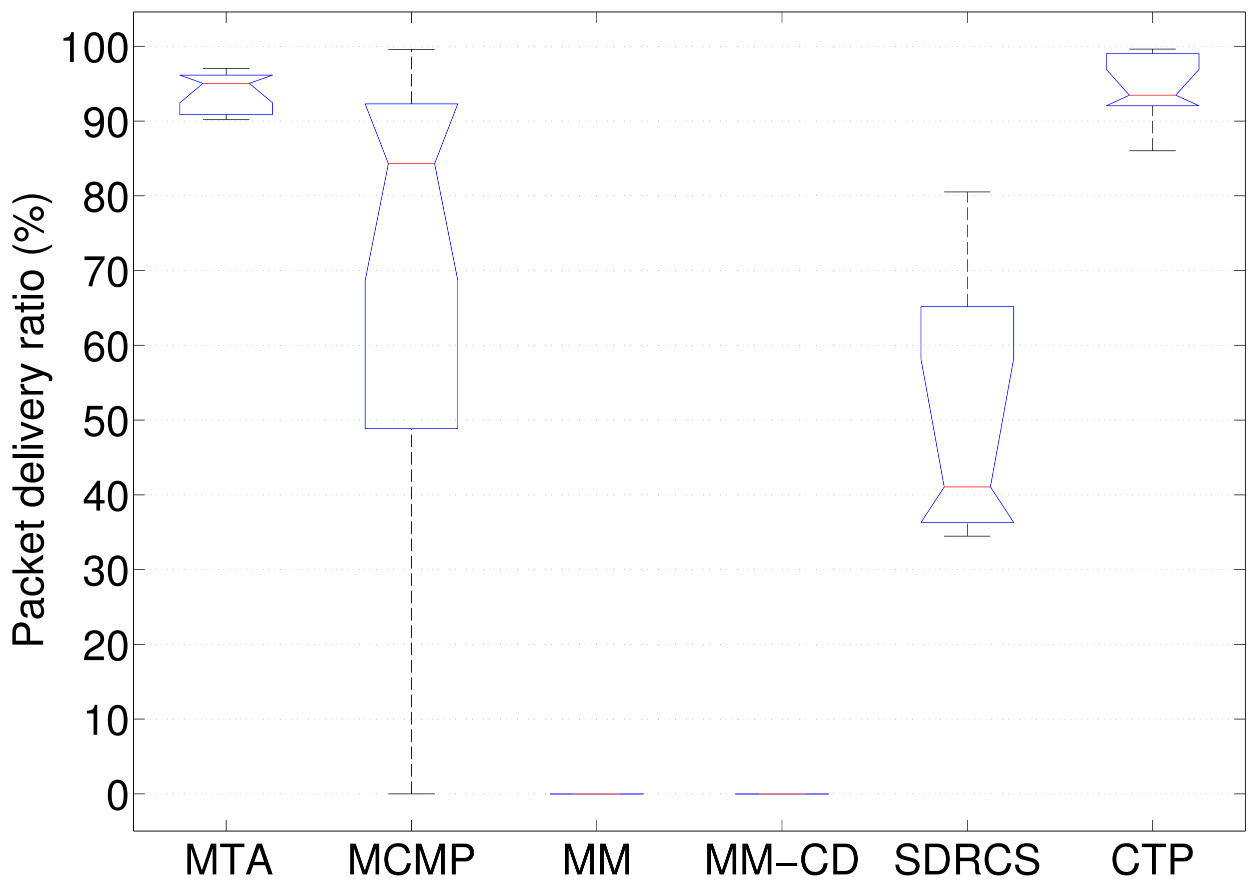


Figure 57: Packet delivery ratio: MTA and existing protocols in Indriya

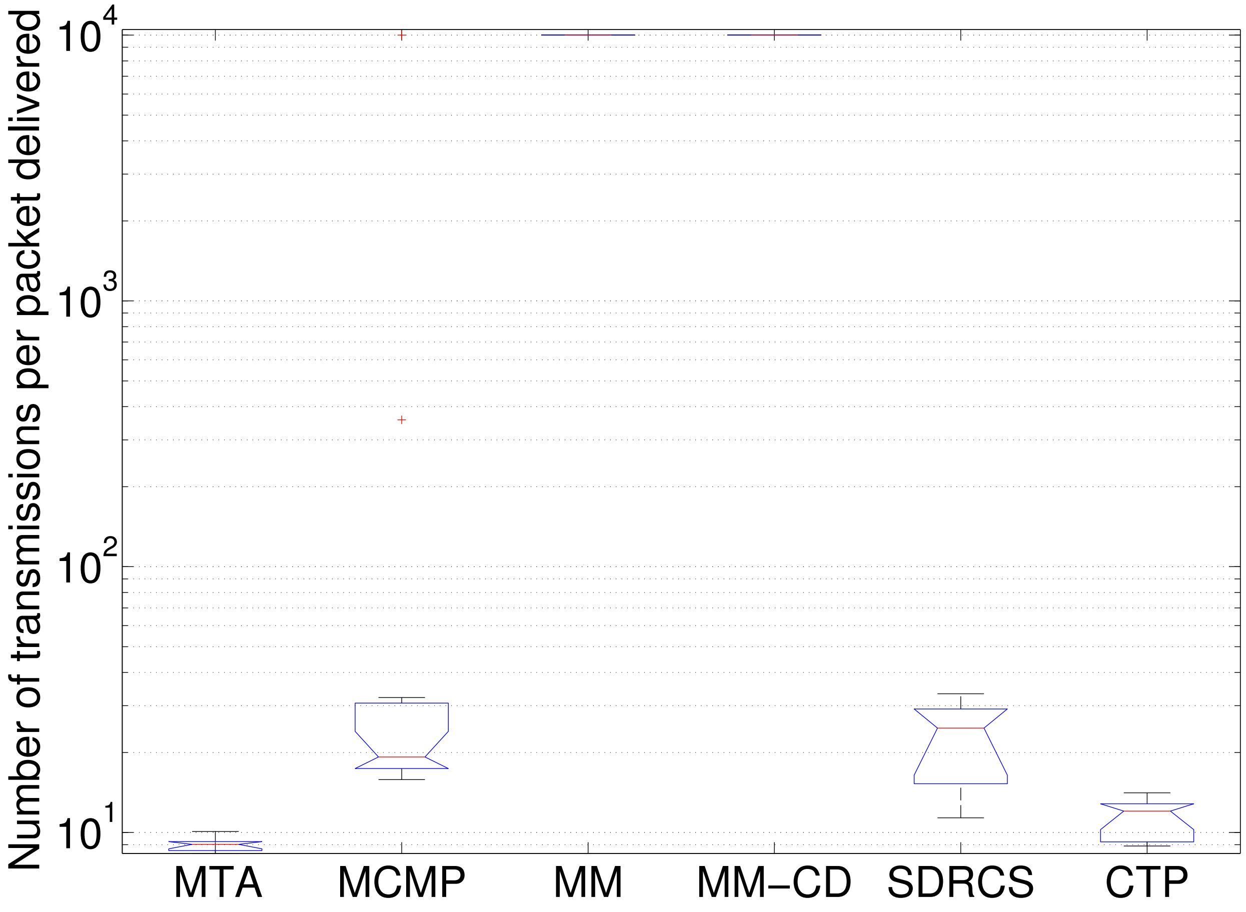


Figure 58: Number of transmissions per packet delivered: MTA and existing protocols in Indriya

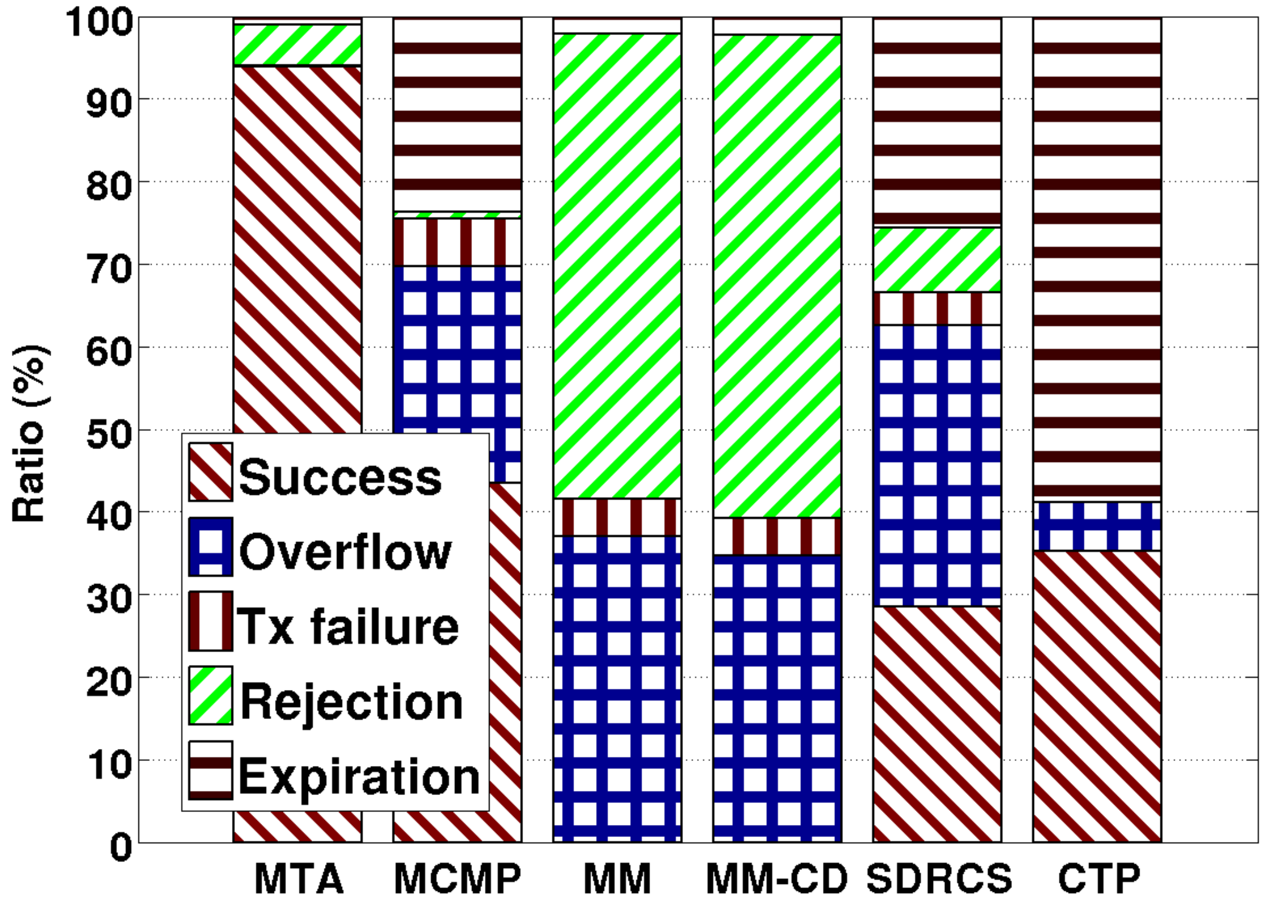


Figure 59: Packet delivery status: MTA and existing protocols in Indriya

overall relative behavior between protocols is similar to that in NetEye, but the performance of MM, MM-CD, and SDRCS become much worse compared with MTA. MTA still ensures real-time data delivery, but MM and MM-CD can hardly deliver any packet to the sink, let alone delivering packets in time; the deadline success ratio in SDRCS is also more than 72% less than that in MTA. One major cause for this is that MM, MM-CD, and SDRCS implicitly assume uniform network conditions while the degree of heterogeneity in Indriya is significant and higher than that in NetEye. As a result of the uniformity assumption, for instance, about 60% and 50% of packets are rejected in MM and MM-CD respectively as shown in Figure 59. Given a packet, more specifically, its deadline and the distance from its source to the sink determines the required forwarding speed for this packet. For the packet to reach the sink, every single hop it traverses has to provide a speed no less than the required speed. That is, if any intermediate hop cannot meet the required speed, a packet is rejected. For MM and MM-CD, it only takes about 3 or 4 hops to reach the sink from the sources in NetEye, while it takes about 7 or 8 hops in Indriya, making packets in Indriya more likely to be rejected. Note that there is significant queue overflow in SDRCS because SDRCS happen to use low reliability links in Indriya which reduce the network throughput and thus increase the queueing and queue overflow.

Figures 60, 61, 62, and 63 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status for event traffic in Indriya respectively. The results are similar to those of periodic traffic with some minor differences. Firstly, MM and MM-CD can deliver some packets in time instead of almost none as in periodic traffic, though their deadline catch ratio is less than 10%. Secondly, the ratio of overflows tend to decrease significantly compared to periodic traffic except for MTA and CTP. Both can be primarily attributed to the reduction of traffic load.

6 Discussion: sparse networks

To evaluate the effect of network connectivity on the performance of MTA, we also run it and other routing protocols in sparse networks in NetEye and Indriya.

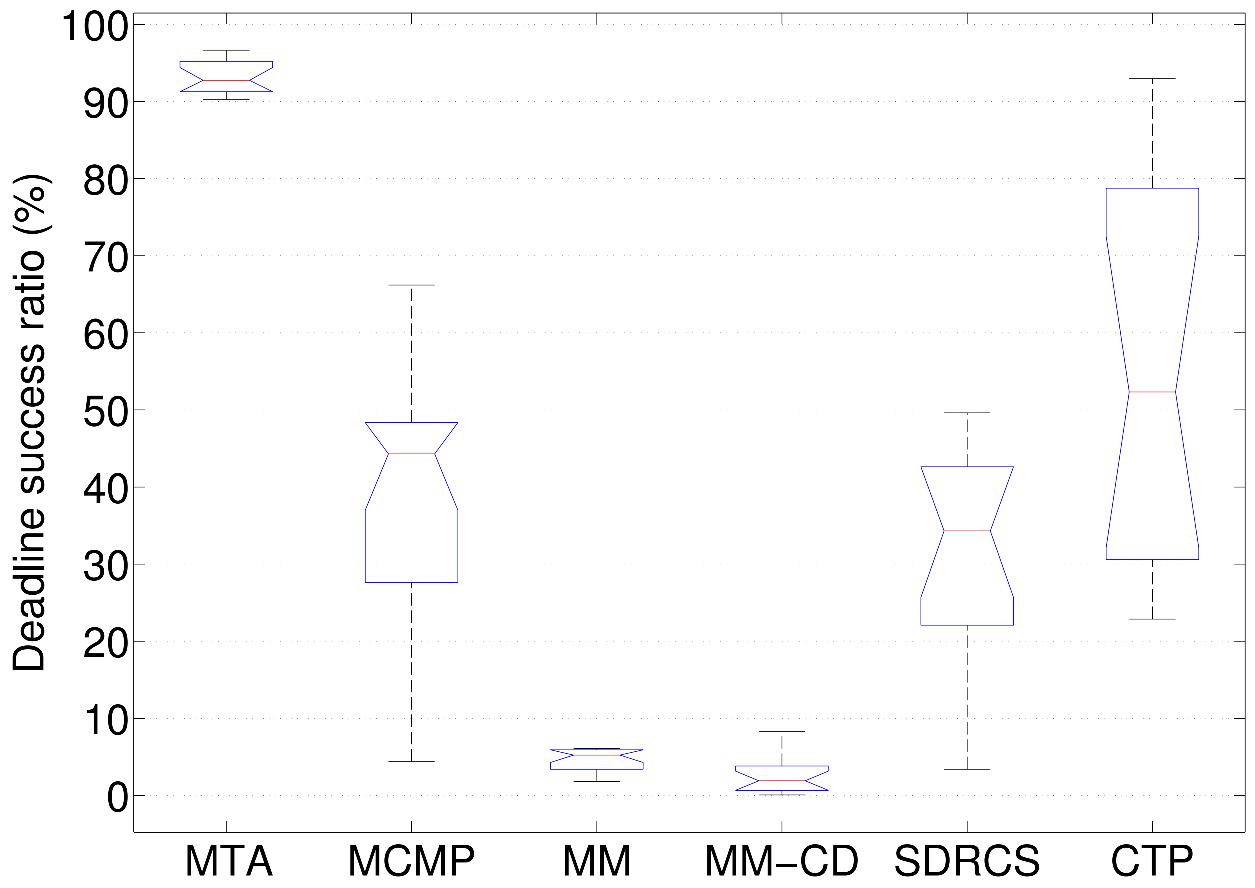


Figure 60: Deadline success ratio: MTA and existing protocols under event traffic in Indriya

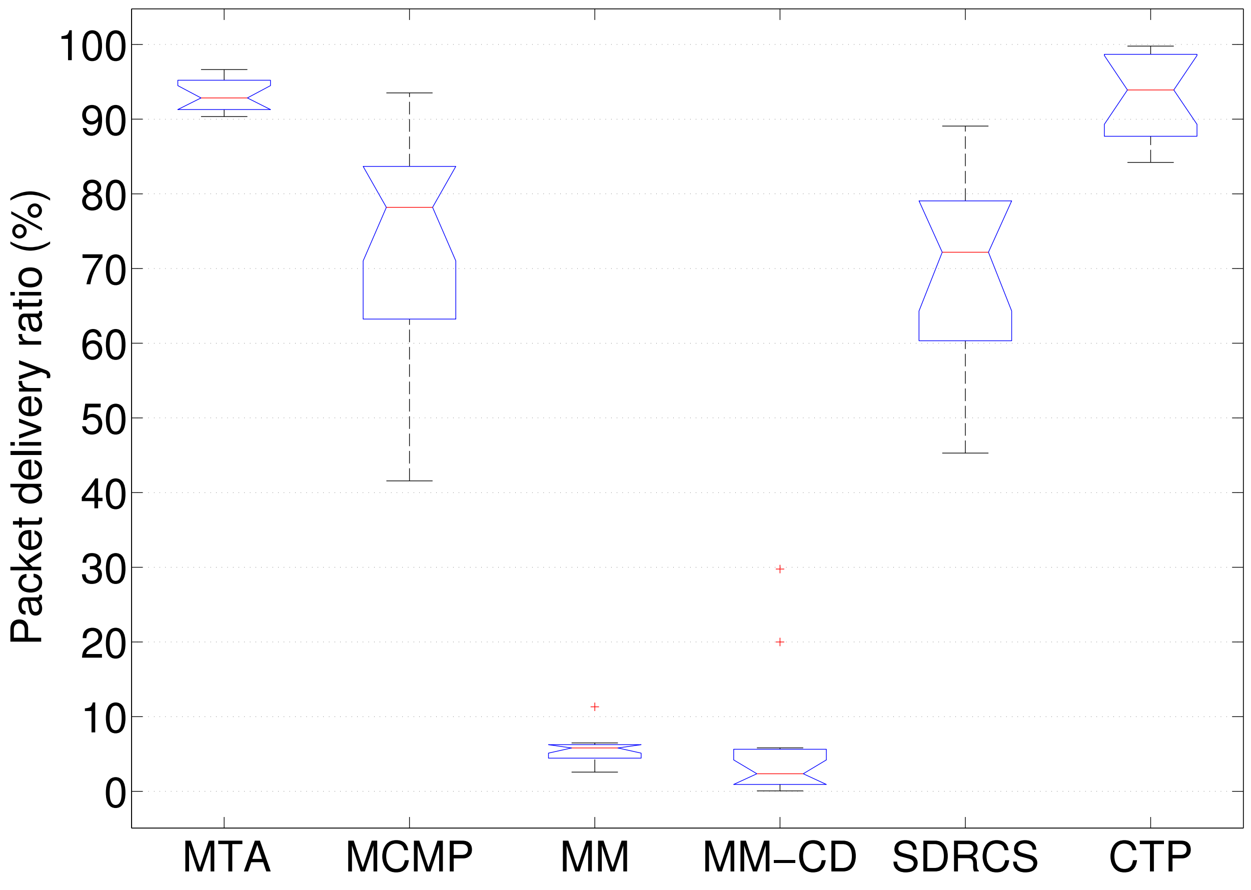


Figure 61: Packet delivery ratio: MTA and existing protocols under event traffic in Indriya

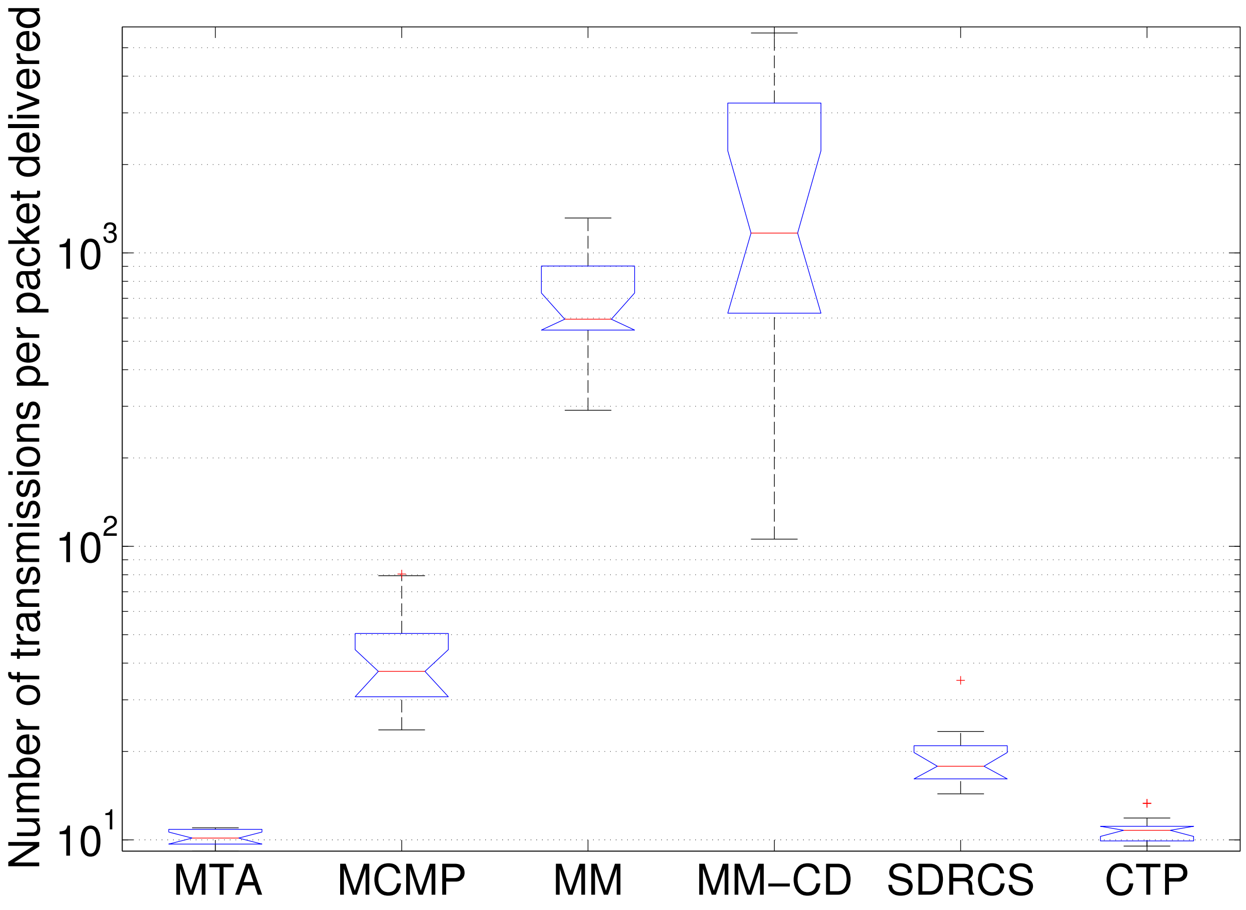


Figure 62: Number of transmissions per packet delivered: MTA and existing protocols under event traffic in Indriya

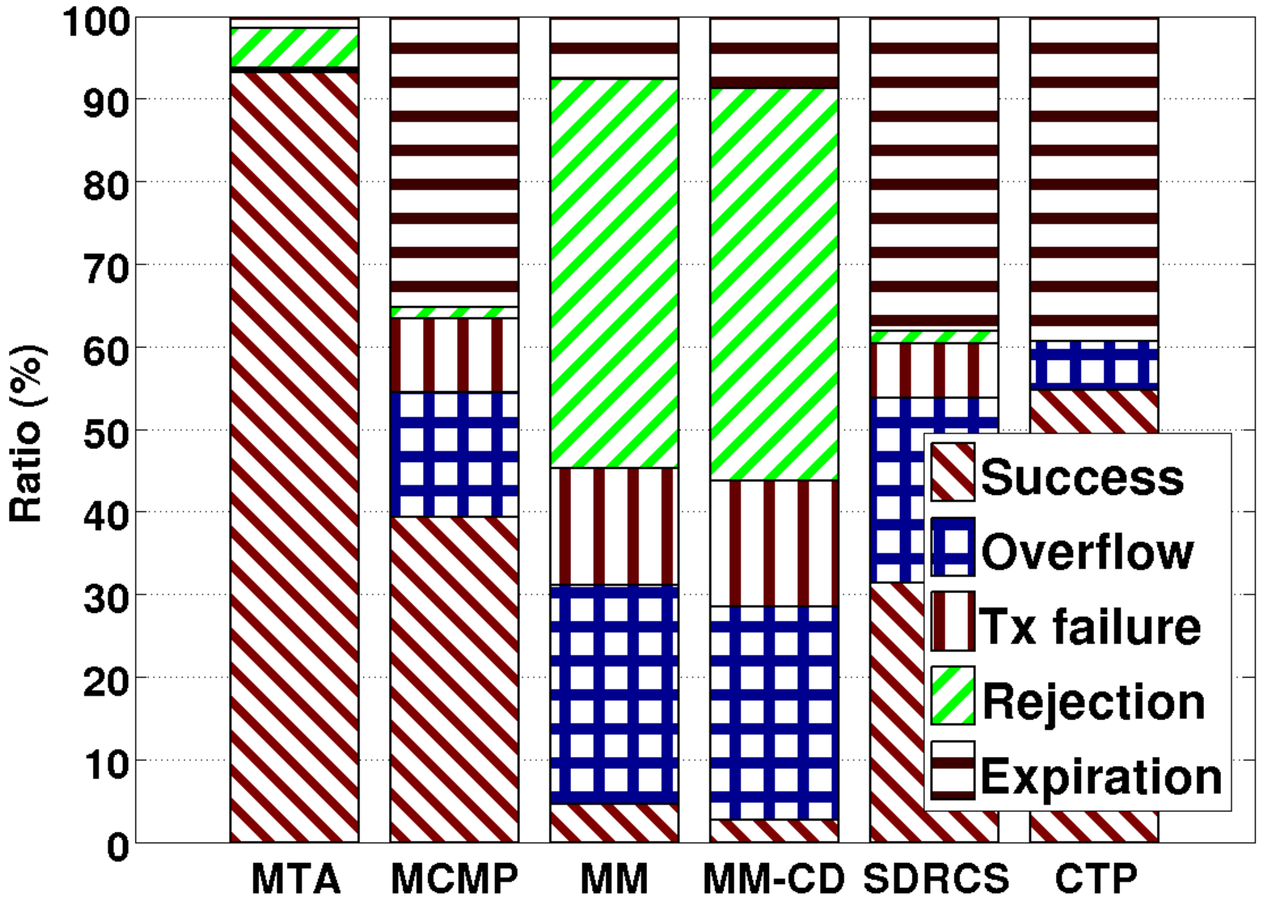


Figure 63: Packet delivery status: MTA and existing protocols under event traffic in Indriya)

- *NetEye*: In the 15 by 7 grid, we randomly select 50% of the nodes. Figure 65 shows the number of neighbors each node has in the resulting topology, contrasting Figure 64 for randomly selecting 80% of the grid. Obviously, the former with median of 9 has a lower density than the latter with median of 12. We choose the farthest four nodes, namely node 61, 62, 79, and 80 as sources, and still choose node 15 as the sink. Every source generates a packet every 100 ms with deadline 4 seconds and required probability 90%.

  Figures 68 shows the deadline success ratio in the sparse network. Again, MTA significantly outperforms other protocols by constantly meeting deadline catch ratio requirement. It also maintains better packet delivery ratio and transmission cost than other protocols except CTP, whose performance is comparable to MTA's as shown in Figure 69 and 70. Figure 71 shows the packet delivery status. MCMP, MMSPEED, MMSPEED and SDRCS experience more queueing overflows because the network's capacity is reduced when the node density decreases. The fact that MTA and CTP do not suffer from queueing overflow can be attributed to their ability to identify good paths to compensate for reduced capacity.

- *Indriya*: We reduce the transmission power level from 11 to 6. Figure 66 and 67 show the number of neighbors each node has in the resulting topology for power level 11 and 6, respectively. The latter with median of 5 is clearly sparser than the former with median of 8.5. We choose node 1, 2, 3, and 4 as sources and still node 100 as the sink. Every source generates a packet every 500 ms with deadline 5 seconds and required probability 90%.

  Figures 72, 73, 74, and 75 show the deadline success ratio, packet delivery ratio, transmission cost, and packet delivery status respectively. All the relative performance comparisons are similar to those of power level 11.

7 Related work

QoS routing has been well studied for the Internet [24, 28] and wireless networks [42, 25, 27]. But most did not consider uncertainties in link/path properties (e.g., delay). Link property uncertainties

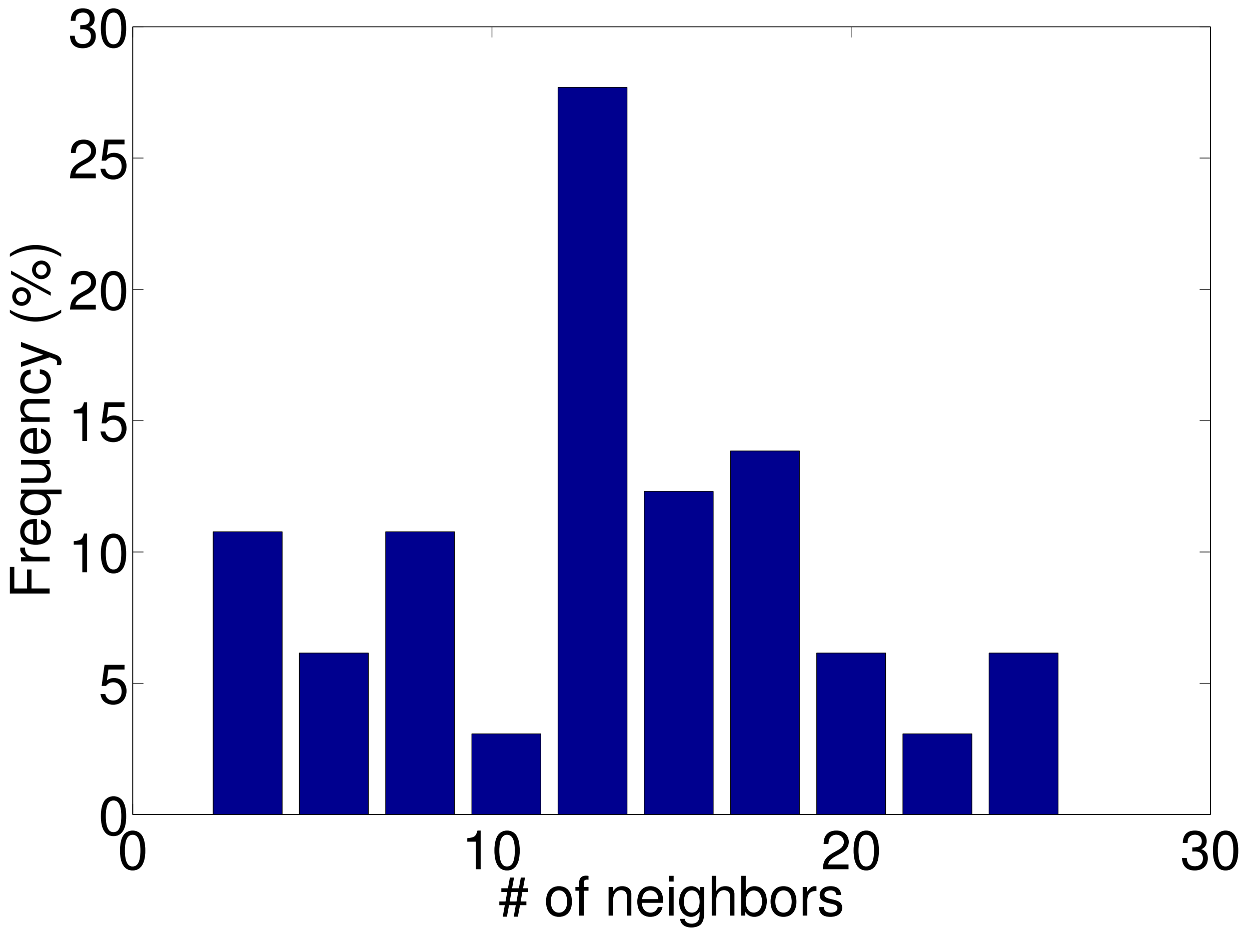


Figure 64: Histogram of the number of neighbors for each node in random 80% NetEye

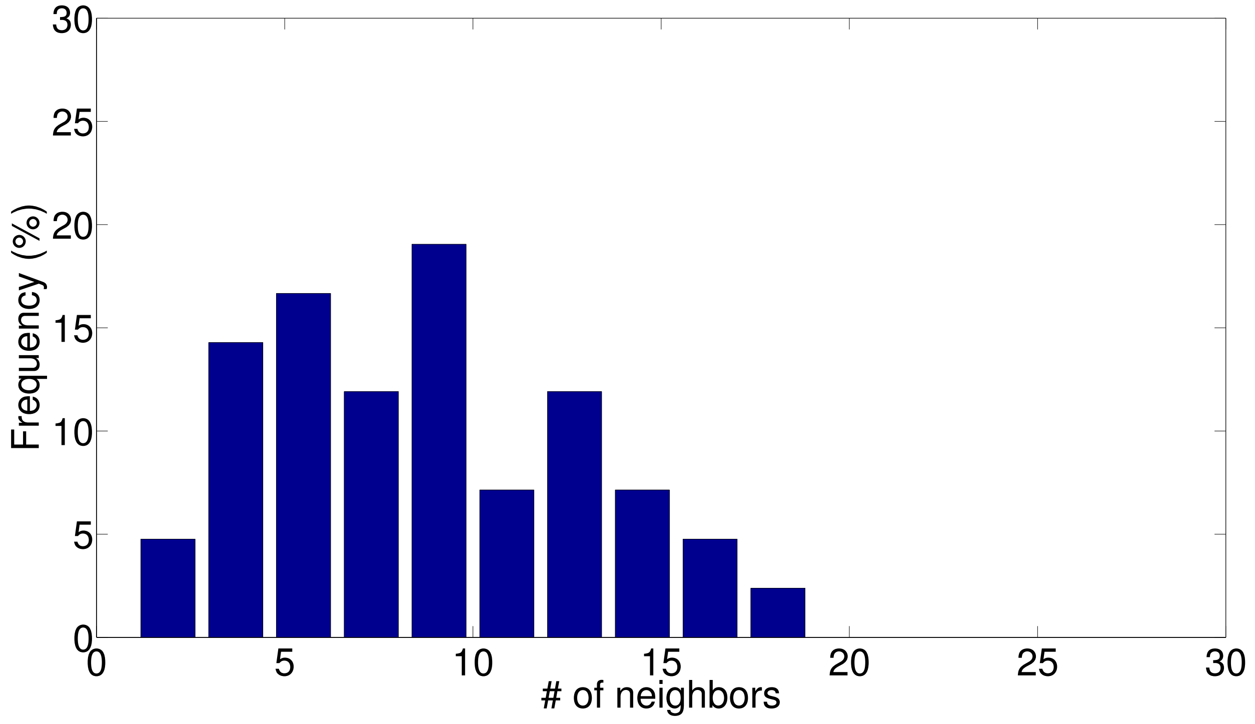


Figure 65: Histogram of the number of neighbors for each node in random 50% NetEye

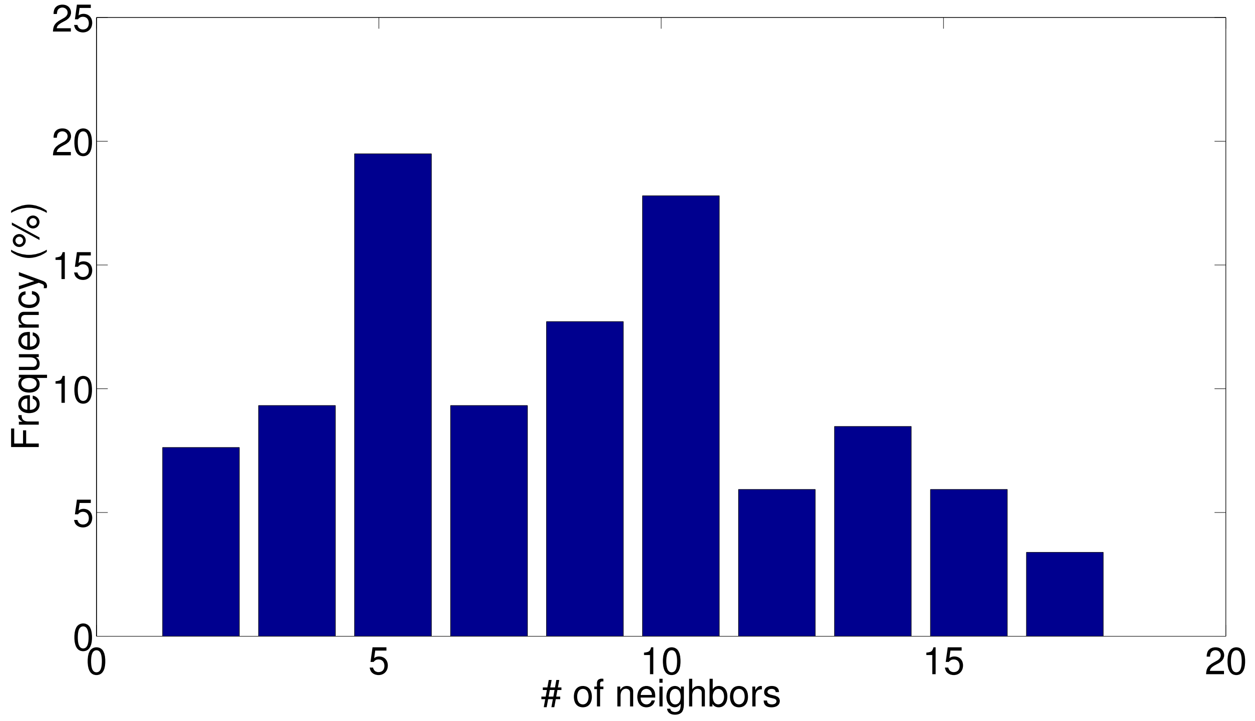


Figure 66: Histogram of the number of neighbors for each node in Indriya using power level 11

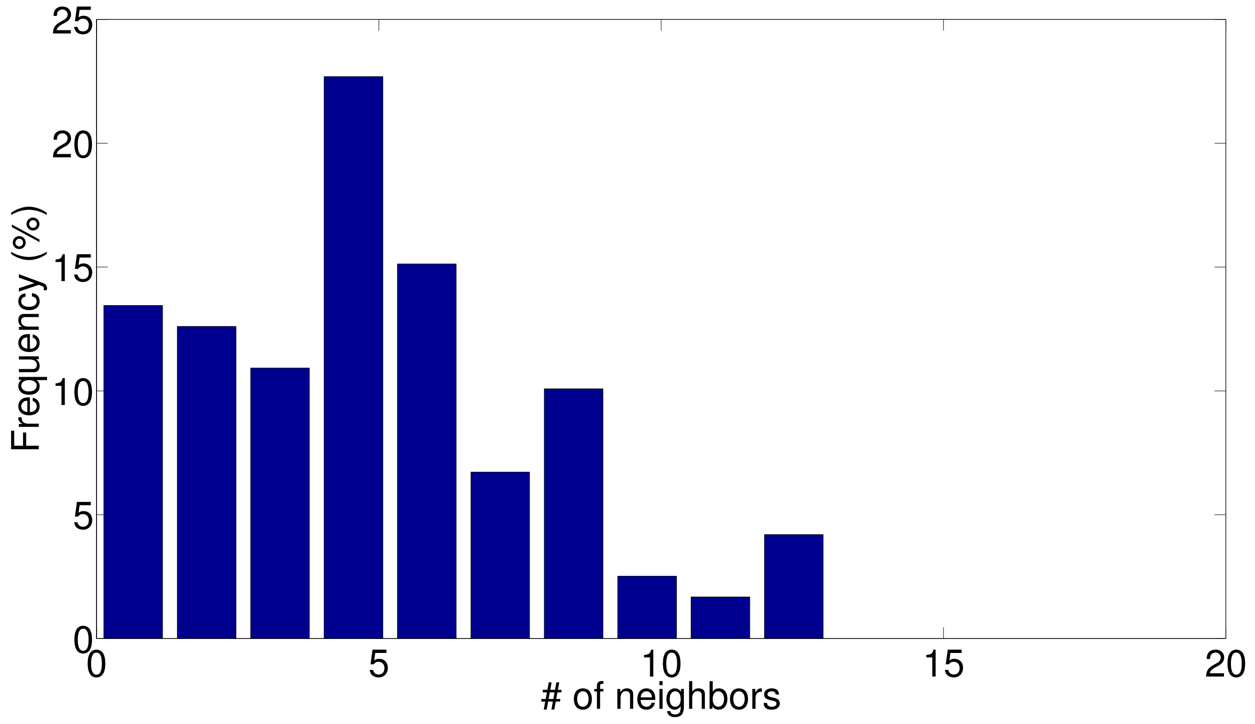


Figure 67: Histogram of the number of neighbors for each node in Indriya using power level 6

were considered in [51] and [69], and it was shown that the problem of checking probabilistically guaranteed path delays is NP-hard [51]. Focusing on Internet QoS routing, these work assumed link-state routing, and their solutions were not amenable to light-weight, distance-vector-type implementation. Since link-state routing is usually not suitable for dynamic, resource constrained WSC networks where reliable network-wide link-state update itself is a challenging issue and where nodes may only have very limited memory space (e.g., up to 4KB of RAM), the approaches of [51] and [69] are not applicable to WSC networks.

Data delivery delay was also considered in wireless and sensor network routing [43, 54, 60, 74, 129]; but they only tried to minimize average path delay without ensuring probabilistic delay bounds [126, 43, 54, 74, 129], they did not consider the probabilistic nature of link/path delays [91], they were based on geographic forwarding without addressing network non-uniformity and wireless communication irregularity [54, 43], or they uniformly partitioned multi-hop QoS requirements (e.g., reliability and timeliness) along the links of a path without considering network

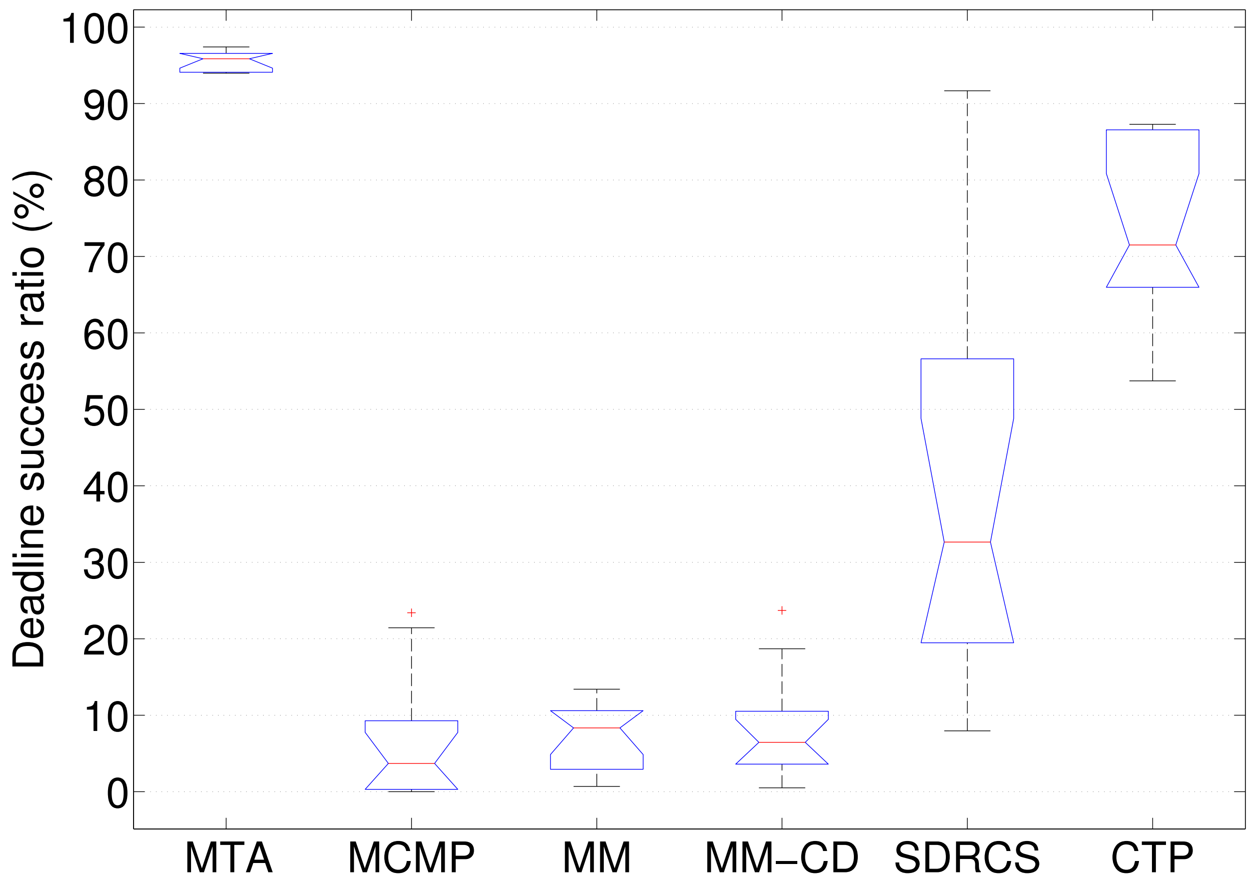


Figure 68: Deadline success ratio of different protocols in sparse NetEye

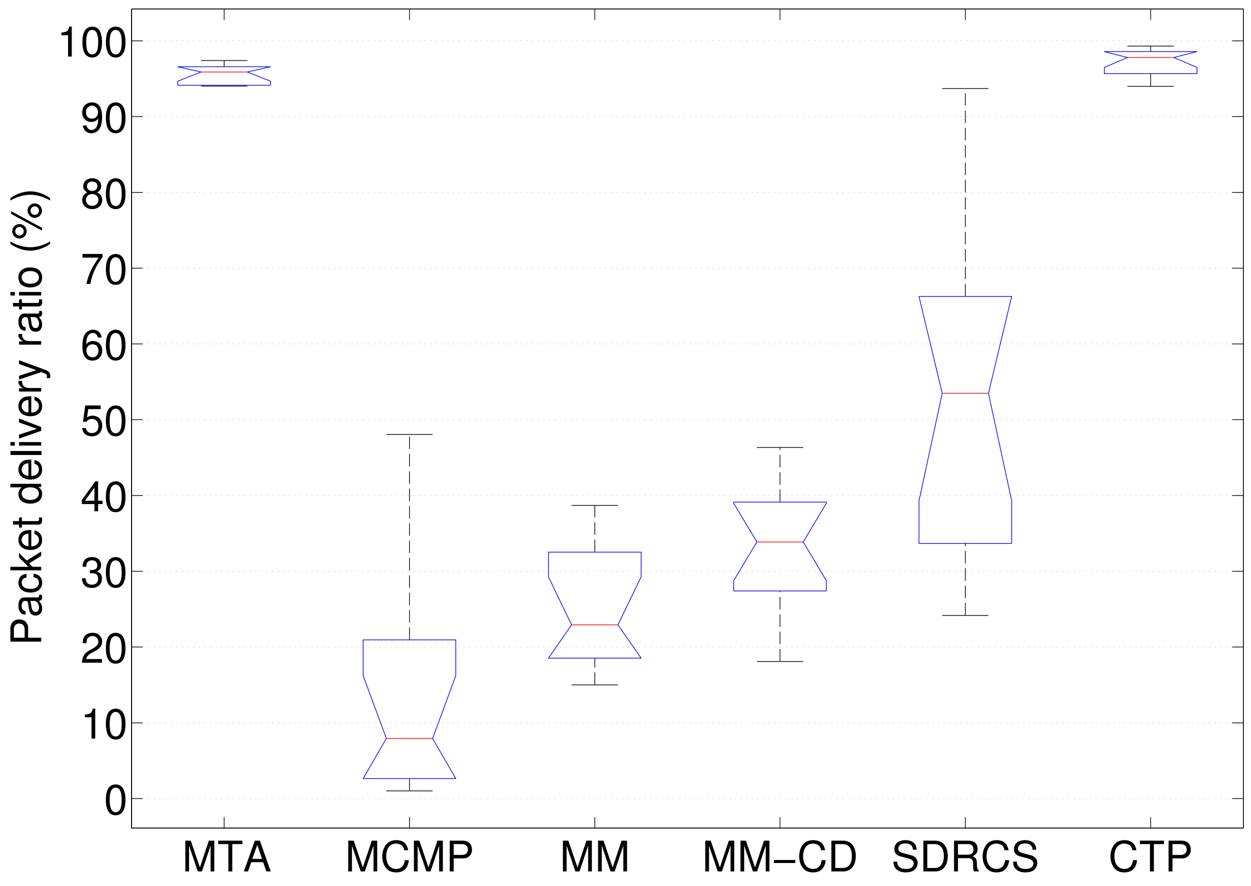


Figure 69: Packet delivery ratio of different protocols in sparse NetEye

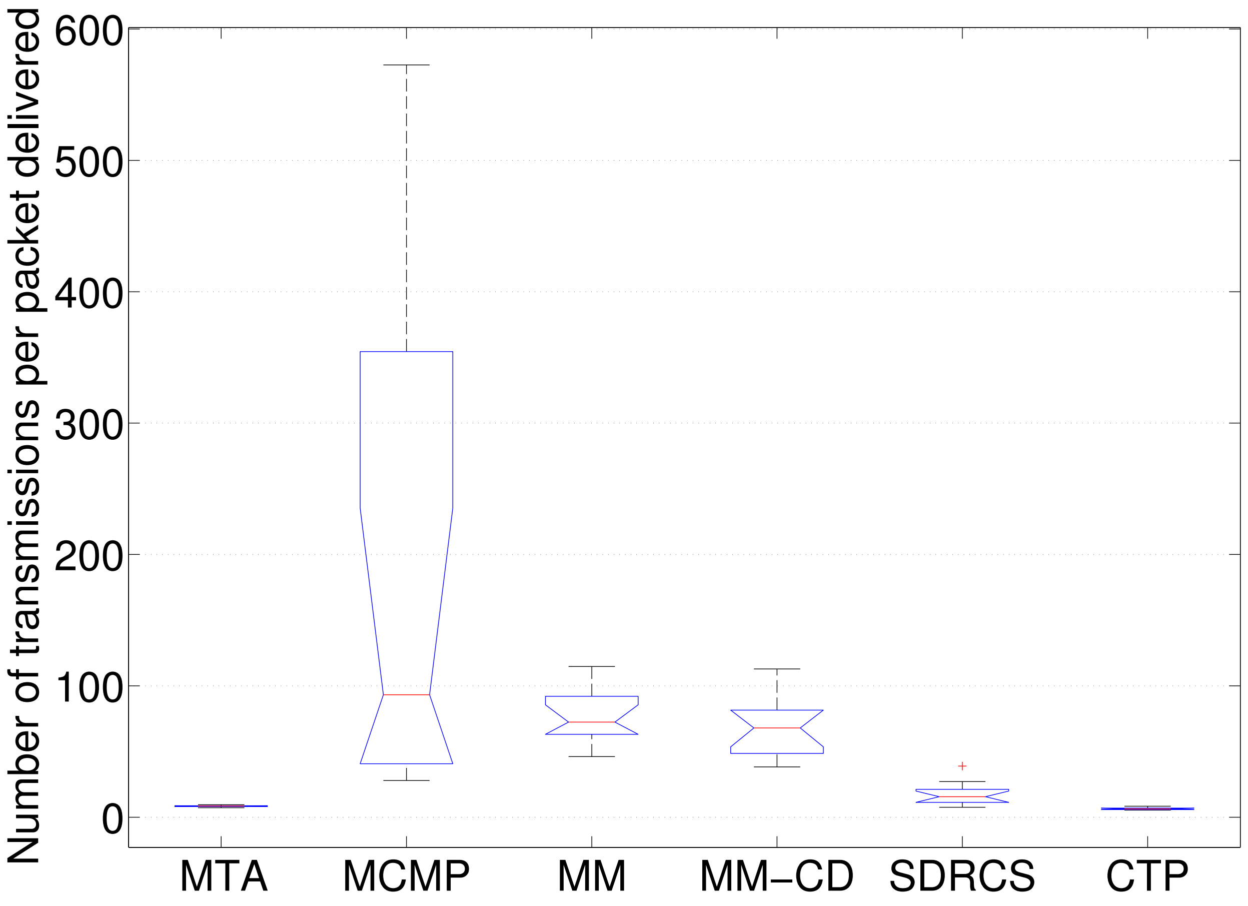


Figure 70: NTX of different protocols in sparse NetEye

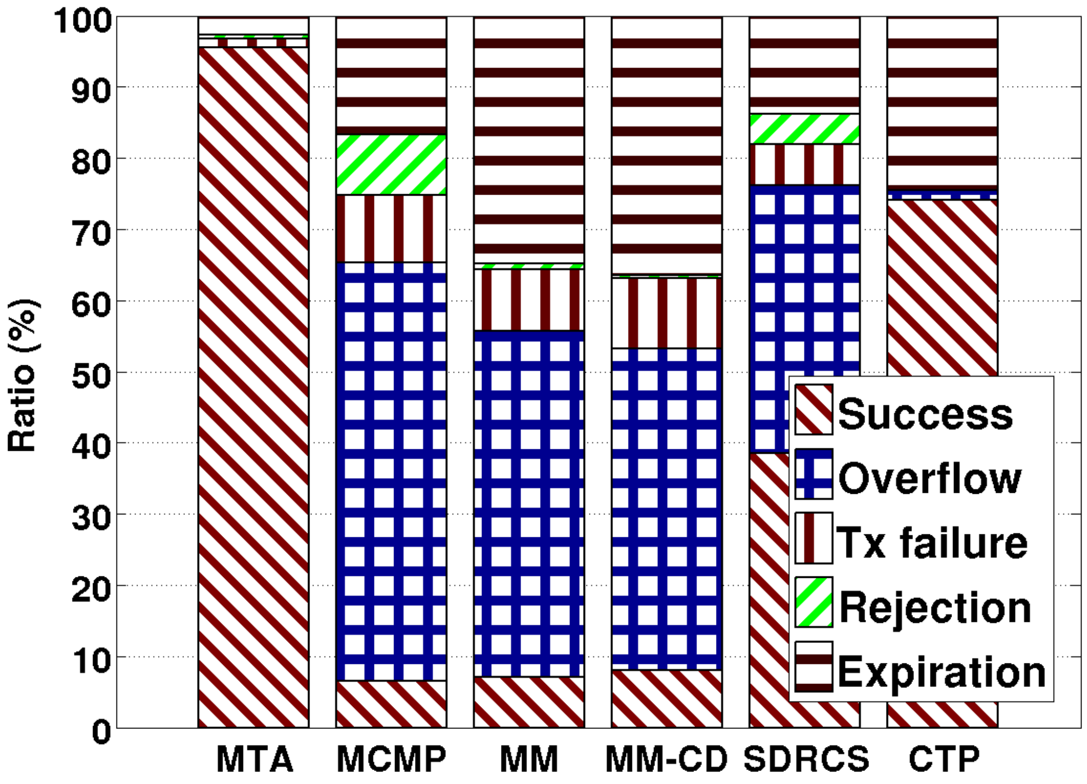


Figure 71: Packet outcomes of different protocols in sparse NetEye

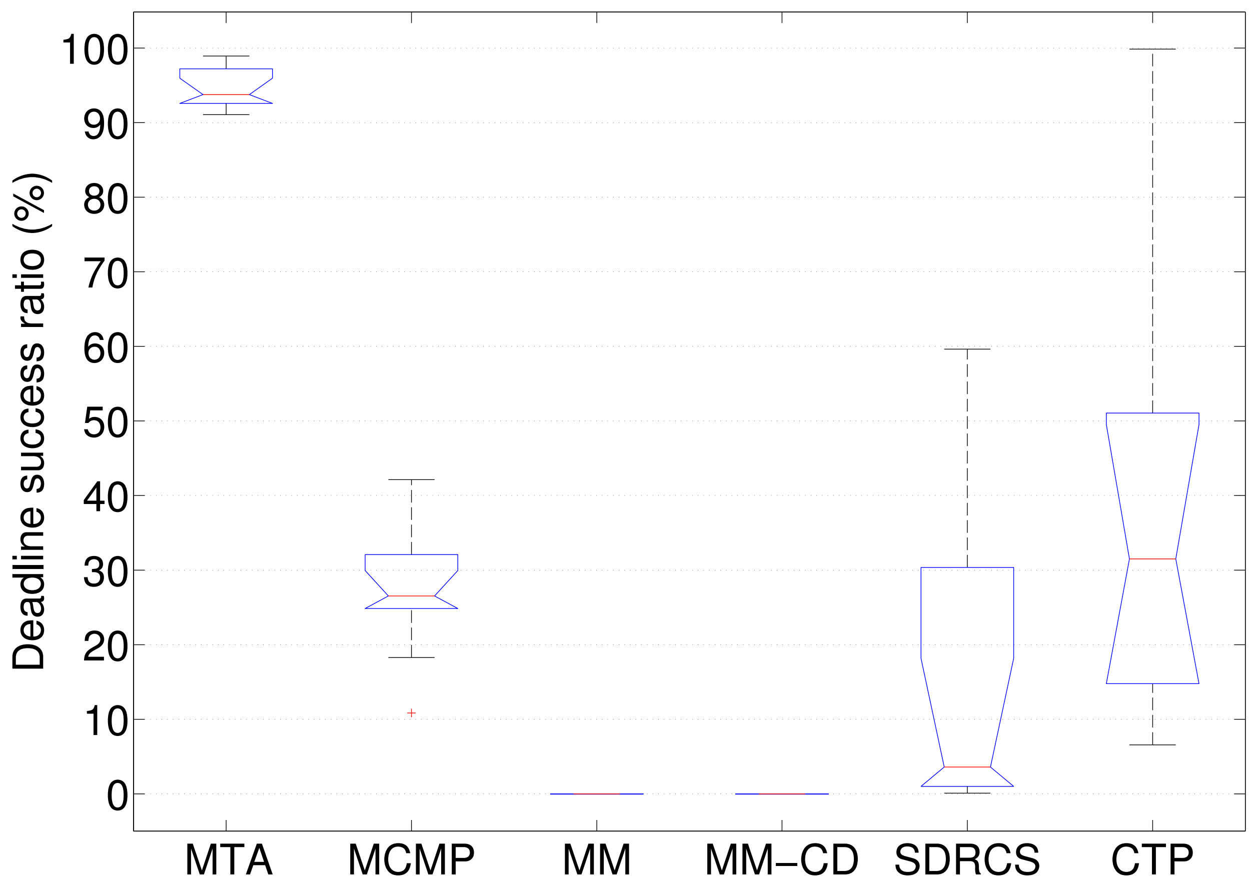


Figure 72: Deadline success ratio of different protocols in sparse Indriya

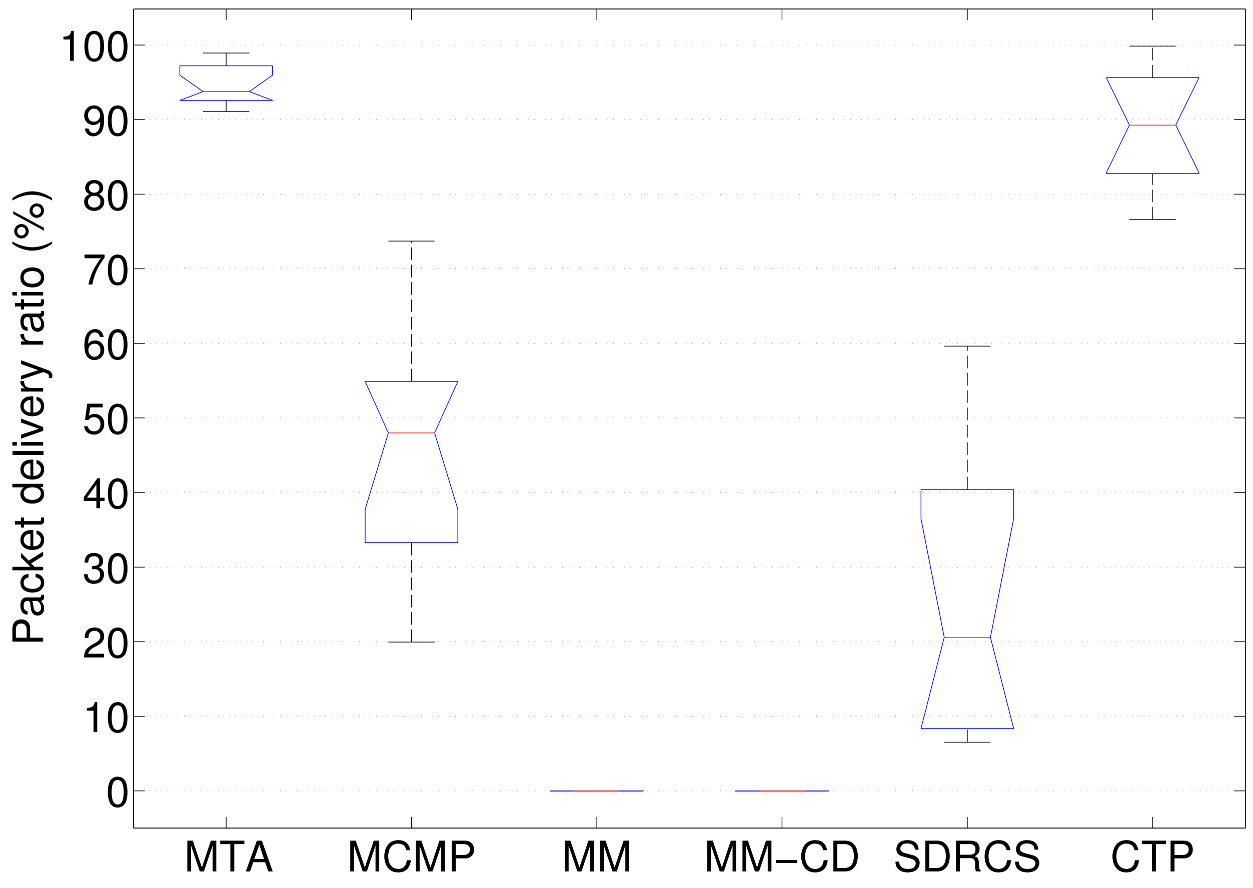


Figure 73: Packet delivery ratio of different protocols in sparse Indriya

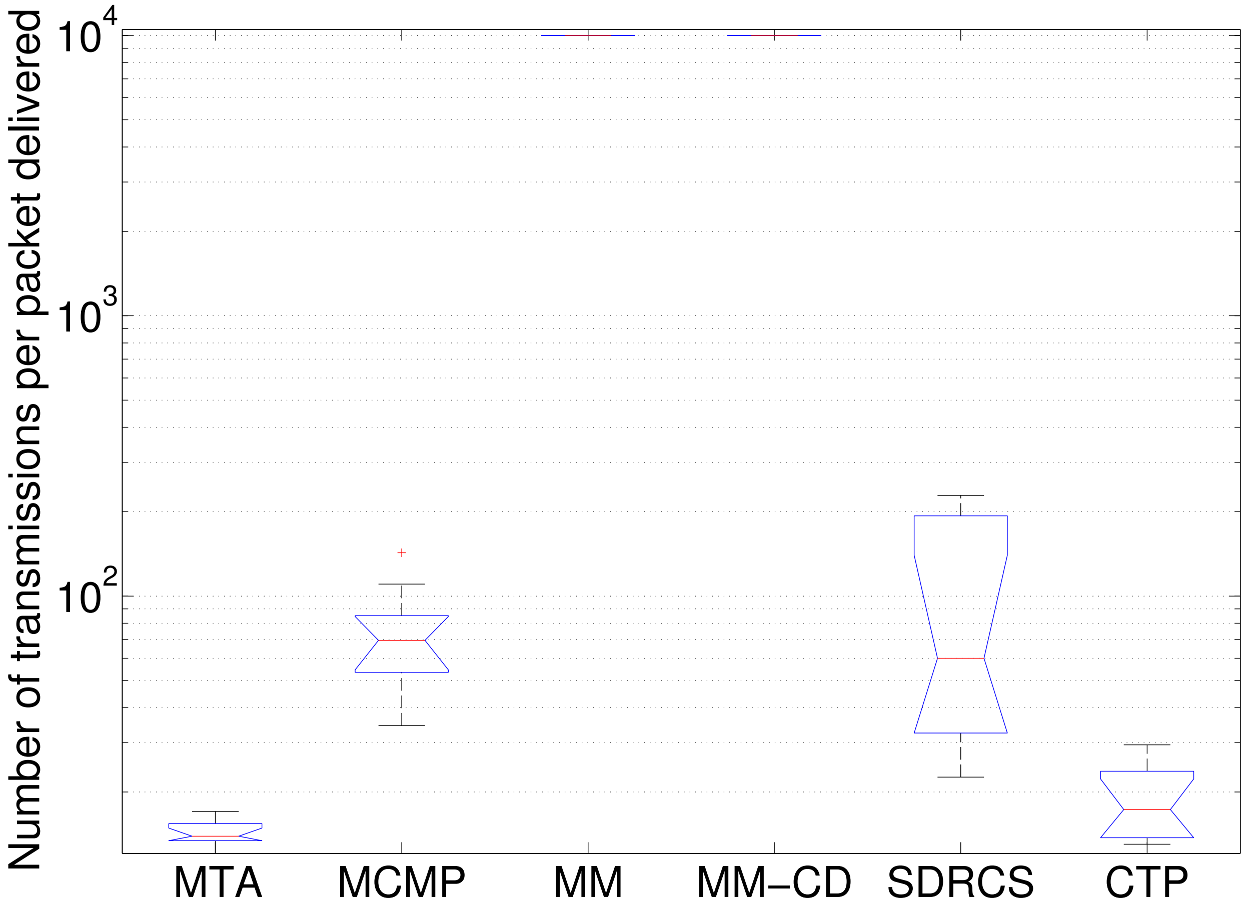


Figure 74: NTX of different protocols under periodic traffic in sparse Indriya

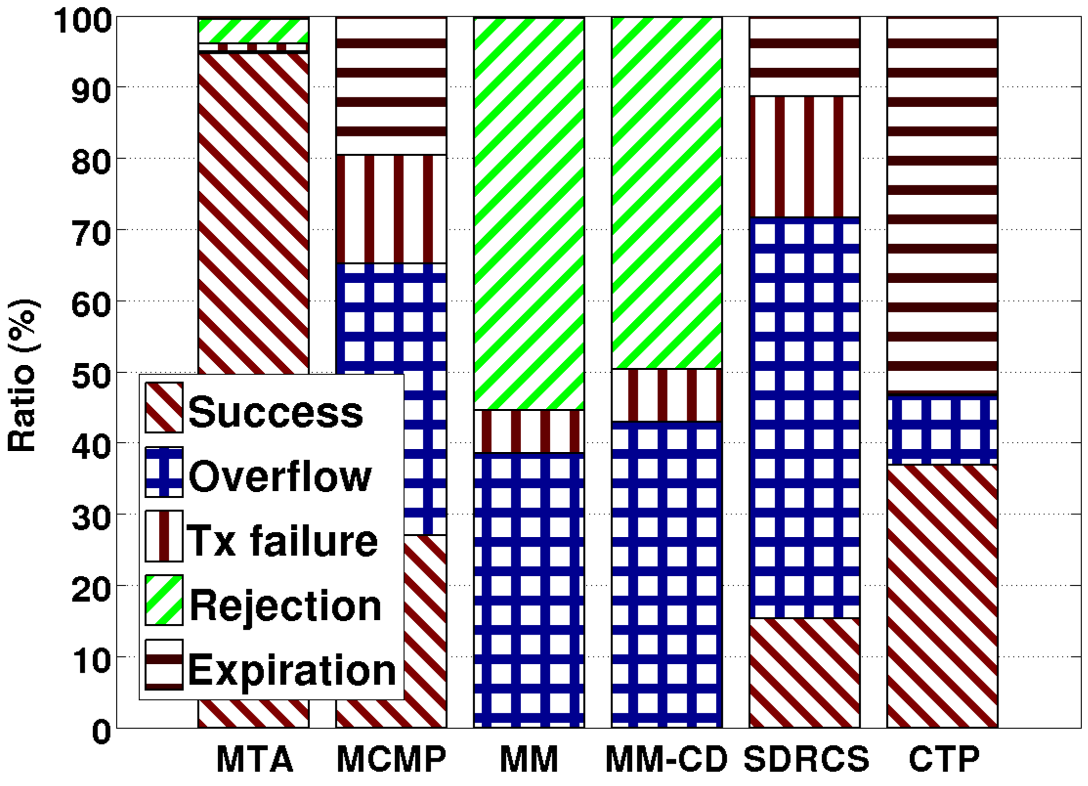


Figure 75: Packet outcomes of different protocols in sparse Indriya

non-uniformity [126, 60]. Huang et al. [60] used Chebyshev Inequality in single-hop real-time satisfiability testing, but they did not address the challenges of accurate, agile estimation of multi-hop probabilistic delay bounds, and they did not comparatively study Chebyshev Inequality with other well-known probability inequalities. Liu et al. [82] proposed the pseudo-polynomial time algorithm optimal-partition-minimum-delay (OPMD) for upper-bounding probabilistic path delays, but, as we have shown in Section 3, the bound of the OPMD algorithm is quite loose for multi-hop paths.

Multi-timescale adaptation has been considered in Internet traffic engineering [21, 66]. Focusing on load balancing, these work did not consider QoS assurance. Liu et al. [82] also studied multi-timescale adaptation in routing, but they used the OPMD method to estimate probabilistic delay bounds which are significantly looser than the bound identified through our MTE method, thus leading to real-time capacity loss. The IETF ROLL working group [61] considered building routing trees based on directed-acyclic-graphs (DAG) for low-power wireless networks. Serving as a general reference framework, the ROLL routing proposal did not consider specific optimization methods (e.g., for real-time guarantees). For stable data delivery reliability, Lin et al. [80] proposed to route data based on long-term link properties and to address transient perturbations using power and retransmission control; they focused on data delivery reliability instead of real-time, thus they did not consider the challenges of dynamic, uncertain link/path delays in real-time routing.

The WirelessHART [122] and the ISA SP100.11a [112] standards have been recently proposed for wireless networking in industrial process measurement and control. They mostly focus on high-level system frameworks instead of specific algorithms in real-time routing. In the literature of real-time wireless networking, techniques such as power control [29] as well as joint routing and scheduling [91] have been studied; energy-efficiency [39] has also been considered too. Orthogonal to these studies, our study here has focused on addressing the challenges that dynamic, uncertain link/path delays pose to two basic elements of real-time routing, i.e., determining probabilistic path delays and addressing instability in delay-adaptive routing. Integrating our results with those work will be an interesting research avenue to pursue, but detailed study of it is beyond the scope of this paper.

8 Concluding remarks

For addressing the challenges of highly-varying path delays to distributed estimation of path delay quantiles, we have proposed the MTE method that leverages the stability of packet-time distribution and the quick diffusion of path delay statistics (i.e., mean and variance) to accurately estimate probabilistic path delay bounds in an agile manner. Based on accurate, agile characterization of path delays using MTE, our MTA routing framework enables the stability and optimality of data forwarding while adapting to fast-changing network queueing and delay. Through extensive measurement study in both the NetEye and the Indriya wireless sensor network testbeds, we have shown that MTE/MTA-based routing ensures efficient, real-time data delivery, and it significantly outperforms existing real-time routing protocols. We have mainly focused on real-time spatial flow control in this study, even though we have experimentally analyzed the benefits of using EDF instead of FCFS in intra-node scheduling; how to control temporal packet flow between neighbors and across the network and how to jointly optimize the spatial and temporal packet flow will be an important area to explore, where the MTE method and the MTA framework are expected to serve as basic systems building-blocks. The technique of leveraging different timescales of dynamics in protocol design may well be of generic interest to wireless networking in dynamic, uncertain environments too.

CHAPTER 3

PRK-BASED SCHEDULING FOR PREDICTABLE LINK RELIABILITY IN WIRELESS NETWORKED SENSING AND CONTROL

1 Introduction

Given the varying impact that the reliability, delay, and throughput of wireless messaging have on networked control and the inherent tradeoff between messaging reliability, delay, and throughput, the optimal operation of WSC systems also requires controlling the tradeoff between reliability, delay, and throughput in messaging, and controlling link reliability in a predictable manner is a basis for such system-level optimization [89, 90, 110, 133]. Causing collisions of concurrent transmissions, co-channel interference is a major source of unpredictability in link reliability [133, 131, 132]. Thus scheduling transmissions for co-channel interference control is a basic element of wireless messaging in WSC systems.

Distributed scheduling & interference models. In WSC systems, not only does wireless link dynamics introduce uncertainty as in traditional wireless sensor networks, dynamic control strategies also introduce dynamic network traffic patterns and pose different requirements on messaging reliability [90]. For agile adaptation to uncertainties and for avoiding information inconsistency in centralized scheduling, distributed scheduling becomes desirable for interference control in WSC networks. Despite decades of research on interference-oriented channel access control, most existing literature are either based on the physical interference model or the protocol interference model, neither of which is a good foundation for distributed interference control in the presence of uncertainties [133]. In the physical model, a set of concurrent transmissions $(S_i, R_i), i = 1 \ldots N$, are regarded as not interfering with one another if the following conditions hold:

$$\frac{P(S_i, R_i)}{N_i + \sum_{j=1 \ldots N, j \neq i} P(S_j, R_i)} \geq \gamma, i = 1 \ldots N \tag{3.1}$$

where $P(S_i, R_i)$ and $P(S_j, R_i)$ is the strength of signals reaching the receiver R_i from the transmitter S_i and S_j respectively, N_i is the background noise power at receiver R_i, and γ is the signal-to-interference-plus-noise-ratio (SINR) threshold required to ensure a certain link reliability. In the protocol model, a transmission from a node S to its receiver R is regarded as not being interfered by a concurrent transmitter C if

$$D(C, R) \geq K \times D(S, R) \tag{3.2}$$

where $D(C, R)$ is the geographic distance between C and R, $D(S, R)$ is the geographic distance between S and R, and K is a constant number.

The physical model is a high-fidelity interference model in general, but interference relations defined by the physical model are non-local and combinatorial; this is because, as can be seen from (3.1), whether one transmission interferes with another explicitly depends on all the other transmissions in the network. Even though many centralized TDMA scheduling algorithms have been proposed based on the physical model [17, 50], distributed physical-model-based scheduling still has various drawbacks: it converges slowly due to explicit network-wide coordination [19, 88], it has to employ strong assumptions such as the reliable detection of a node's busy-tone signal by the whole network [102] or the knowledge of node locations and wireless channel path loss bound [128], it does not control cumulative interference which introduces uncertainties in communication [139, 124], it is not suitable for dynamic network settings due to the need for centrally computing the interference set of each link (i.e., the set of links interfering with the link) [100] or the interference neighborhood of each link (i.e., the set of links causing non-negligible interference to the link) [71], or it does not address the challenge of designing scheduling protocols when interfering links are beyond the communication range of one another [71, 102]. Many of the SINR-based MAC protocols are also throughput-oriented, and they do not control multi-hop interference to ensure predictable link reliability [105].

Unlike the physical model, the protocol model defines local, pairwise interference relations; that is, according to (3.2), interference is regarded as existent only between nodes in a local neighborhood, and whether one transmission interferes with another only depends their own spatial distribution irrespective of other transmissions in the network. The locality of the protocol model can enable agile protocol adaptation in the presence of uncertainties. *However, the protocol model is usually inaccurate* [86], thus scheduling based on the protocol model [65, 106, 119] or its variants [72, 77, 111] does not ensure link reliability and also tends to reduce network throughput. Choi et al. [32] recently proposed grant-to-send (GTS) as a new mechanism for collision avoidance. Only focusing on intra-flow interference, GTS does not address inter-flow interference and cannot ensure data delivery reliability; for instance, GTS may only enable a data delivery reliability of 47.4% in event-detection sensor networks [32].

Besides scheduling based on the physical and protocol interference models, distributed scheduling algorithms using general pairwise interference models have also been proposed [52, 92]. Theoretical in nature, however, these algorithms did not address the important question of how to identify the interference set of each link, and their implementation usually assumes a model similar to the protocol model [92]. These algorithms also did not address important systems issues such as how to design scheduling protocols when interfering links are beyond the communication range of one another.

Without field-deployable solutions to predictable co-channel interference control, current systems practice, such as the WirelessHART standard for industrial sensing and control [122], avoids co-channel interference by allowing only one node in the whole network to transmit in a wireless channel at any moment in time. Without spatial channel reuse, however, this approach does not fully utilize wireless network capacity, which is undesirable for high data-rate sensing and control applications and for new networked control paradigms that involve only communications between close-by nodes [95].

Physical-ratio-K (PRK) interference model. The gap between the existing interference models and the design of distributed, field-deployable scheduling protocols with predictable data delivery reliability calls for an interference model that is both local and of high-fidelity, which are important for the agility and predictability of interference control respectively. We have recently identified the *physical-ratio-K (PRK) interference model that integrates the protocol model's locality with the physical model's high-fidelity* [133]. In the PRK model, a node C' is regarded as not interfering and thus can transmit concurrently with the transmission from another node S to its receiver R if and only if the following holds:

$$P(C', R) < \frac{P(S, R)}{K_{S,R,T_{S,R}}} \tag{3.3}$$

where $P(C', R)$ and $P(S, R)$ is the average strength of signals reaching R from C' and S respectively, and $K_{S,R,T_{S,R}}$ is the minimum real number chosen such that, in the presence of interference from all concurrent transmitters, the probability for R to successfully receive packets from S is no less than the minimum link reliability $T_{S,R}$ required by applications (e.g., control algorithms). As shown in Figure 76, the PRK model defines, for each link (S, R), an exclusion region $\mathbb{E}_{S,R,T_{S,R}}$

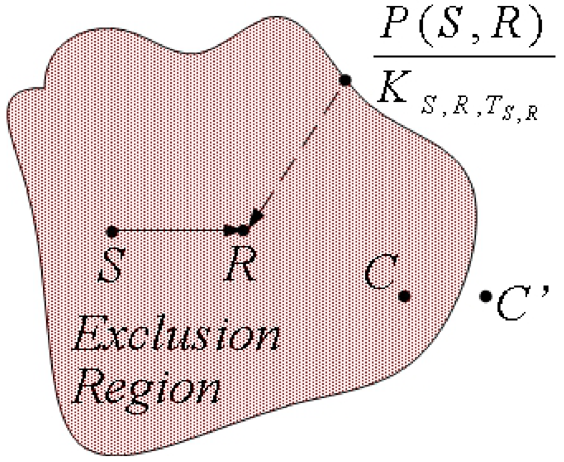


Figure 76: The physical-ratio-K (PRK) interference model

around the receiver R such that a node $C \in \mathbb{E}_{S,R,T_{S,R}}$ if and only if $P(C, R) \geq \frac{P(S,R)}{K_{S,R,T_{S,R}}}$. Accordingly, every node $C \in \mathbb{E}_{S,R,T_{S,R}}$ is regarded as interfering with and thus shall not transmit concurrently with the transmission from S to R.

Unlike the physical model, the PRK model is *local* and suitable for distributed protocol design and implementation: 1) The parameters of the PRK model are either locally measurable (i.e., for signal strength and link reliability between close-by nodes) or locally controllable (i.e., for $K_{S,R,T_{S,R}}$ of each link (S, R)), thus PRK-based scheduling does not need to rely on parameters such as nodes' locations or channel path loss between far-away nodes which are often used in physical-model-based scheduling [128] but are difficult to obtain precisely, especially in a distributed manner; 2) Only pairwise interference relations between close-by nodes need to be defined in the PRK model, thus PRK-based scheduling does not require explicit global coordination which is often used in physical-model-based scheduling [19, 88]. Unlike the protocol model, the PRK model is *of high-fidelity* because it captures the properties of wireless communication (including cumulative interference, anisotropy, and asymmetry) by ensuring the required link reliability in scheduling and by using signal strength instead of geographic distance in model formulation. Through comprehensive analysis, simulation, and measurement, we have observed that PRK-based scheduling can enable a channel spatial reuse very close to (e.g., >95%) what is feasible in physical-model-based scheduling while ensuring application-required reliability [133].

Focusing on formulating the PRK interference model and understanding the theoretically achievable performance of PRK-based scheduling, we left the design of distributed protocols for PRK-based scheduling as an open problem in [133]. Yet realizing distributed PRK-based scheduling in real-world settings poses the following major challenges:

- The parameter $K_{S,R,T_{S,R}}$ of the PRK model (3.3) depends on the specific link (S, R), the application requirement on the link reliability (i.e., $T_{S,R}$), as well as the network and environmental conditions such as traffic pattern and wireless path loss which may well be dynamic and unpredictable. So the challenge is how to instantiate the PRK model parameter $K_{S,R,T_{S,R}}$ on the fly depending on in-situ application requirements as well as network and environmental conditions.

- Given a link (S, R) and a specific instantiation of the PRK model, every node in the exclusion region $\mathbb{E}_{S,R,T_{S,R}}$ should be prevented from transmitting concurrently with the transmission from S to R. As we will discuss in detail in Sections 2.2 and 2.3, however, it is difficult to ensure this property due to large interference range, anisotropy and asymmetry in wireless communication, as well as the delay in protocol signaling.

Contributions of this paper. To enable predictable link reliability in distributed scheduling, we address the aforementioned challenges by designing the PRK-based scheduling protocol *PRKS*. In PRKS, we formulate the problem of identifying the PRK model parameter $K_{S,R,T_{S,R}}$ as a minimum-variance regulation control problem, and we design distributed controllers that allow each link to adapt its PRK model parameter for ensuring the desired link reliability through purely local coordination. For ensuring that nodes interfering with one another do not transmit concurrently, we propose the concept of local signal map that allows nodes close-by to maintain the wireless path loss among themselves; together with the PRK model and transmission power control in protocol signaling, local signal maps enable nodes to precisely identify the interference relations among themselves despite anisotropic, asymmetric wireless communication and large interference range. To address the inherent delay in protocol signaling and to avoid interference between protocol signaling and data transmissions, PRKS decouples protocol signaling from data transmissions by leveraging the different timescales of PRK model adaptation and data transmission. With the above mechanisms, PRKS precisely identifies and then avoids interfering concurrent transmissions in scheduling; accordingly, PRKS eliminates the hidden terminal issue which has been a basic challenge in interference-oriented channel access control since 1975 [113].

We have implemented PRKS in TinyOS [9]. Through measurement study in the NetEye [11] and Indriya [2] sensor network testbeds, we observe the following: 1) The distributed controllers enable network-wide convergence to a state where the desired link reliabilities are ensured; 2) Unlike existing scheduling protocols where link reliability is unpredictable and the reliability requirement satisfaction ratio can be as low as 0%, PRKS enables predictably high link reliability (e.g., 95%) in different network and environmental conditions without a priori knowledge of these conditions; 3) With local, distributed coordination alone, PRKS achieves a channel spatial reuse very close to what is enabled by the state-of-the-art centralized physical-model-based scheduler iOrder [26] while ensuring the required link reliability; 4) By ensuring the required link reliability, PRKS also reduces communication delay and improves network throughput.

Organization of the chapter. We elaborate on the design of PRKS in Section 2, and we evaluate the performance of PRKS in Section 3. We discuss node mobility and related work in Sections 4 and 5 respectively. We make concluding remarks in Section 6.

2 PRKS: PRK-based scheduling

In what follows, we first present our control-theoretic approach to instantiating the PRK model, then we present the local signal maps for protocol signaling and the protocol PRKS for distributed PRK-based scheduling. For convenience, Table 3.1 summarizes the major notations used in this section.

2.1 A control-theoretic approach to PRK model instantiation

Minimum-variance regulation control. Given a link (S, R), the task of instantiating the PRK interference model is to identify the parameter $K_{S,R,T_{S,R}}$ such that the resulting scheduling can ensure the required minimum link reliability $T_{S,R}$.[1] It is, however, difficult to characterize the relation between $K_{S,R,T_{S,R}}$ and the packet delivery reliability along (S, R) in closed form, and the relation is complex and dependent on network and environmental conditions which may well be unpredictable at design time [133]. To address the challenge, we observe that the PRK model instantiation problem can be formulated as an online *regulation control* problem [55], where the "plant" is the link (S, R), the "reference input" is the required link reliability $T_{S,R}$, the "plant output" is the actual link reliability $Y_{S,R}$ from S to R, and the "control input" is the PRK model parameter $K_{S,R,T_{S,R}}$. To address the difficulty in characterizing the "plant model" on the relation between the control input $K_{S,R,T_{S,R}}$ and the plant output $Y_{S,R}$, we observe that changing the PRK

[1]Focusing on interference-oriented scheduling, we only consider the links whose packet delivery reliabilities are above the required ones in the absence of interference.

$Y_{S,R}(t)$	Measured packet delivery rate for link (S, R) at time t.
$P(S, R, t)$	Expected power, in units of mW, of data packet signals reaching R from S at time t; assumed to be mostly static at short timescales.
$P_{S,R}(t)$	Expected power, in units of dBm, of data packet signals reaching R from S at time t;
$I_R(t)$	Sum of background noise power and interference power at receiver R at time t, in units of dBm.
$f(.)$	The function modeling the relation between packet delivery rate and SINR.
$a(t)$	$f'(P_{S,R}(t) - I_R(t))$.
$b(t)$	$f(P_{S,R}(t) - I_R(t)) - (P_{S,R}(t) - I_R(t))f'(P_{S,R}(t) - I_R)(t)$.
$y(t)$	Smoothed link reliability measurement, i.e., $y(t) = cy(t-1) + (1-c)Y_{S,R}(t)$.
c	Parameter of the EWMA filter in feedback loop.
$\Delta I_R(t)$	Computed control input at time instant t.
$\Delta I_U(t)$	Change of interference from outside the exclusion region of R from time t to $t+1$.
$\mu_U(t)$	Mean of $\Delta I_U(t)$.
$\sigma_U^2(t)$	Variance of $\Delta I_U(t)$.
$K_{S,R,T_{S,R}}(t)$	PRK model parameter for link (S, R) at time t.
$\mathbb{E}_{S,R,T_{S,R}}(t)$	Exclusion region around receiver R at time t; a node $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$ iff. $P_{C,R}(t) \geq \frac{P_{S,R}(t)}{K_{S,R,T_{S,R}}(t)}$.
$P'(C, R)$	Average signal power attenuation from a node C to another node R; maintained in nodes' local signal maps.

Table 3.1: Major notations used in Section 2

model parameter $K_{S,R,T_{S,R}}$ changes the exclusion region around the receiver R and thus the concurrent transmissions along with the transmission from S to R, which in turn leads to the change in the interference power at receiver R. Accordingly, we propose to regard this change in interference power, denoted by ΔI_R, as the actual control input. This way, we can leverage the existing communication theory to derive the plant model on the relation between $Y_{S,R}$ and ΔI_R as follows.

For conciseness, we use $I_R(t)$ to denote, in units of dBm, the sum of the average background noise power and the average power of all interfering signals at the receiver R at time t ($t = 1, 2, \ldots$); we also use $P_{S,R}(t)$ to denote the average received data signal power $P(S, R)$ in units of dBm at time t. Given a modulation and coding scheme, communication theory gives us the following [133]:

$$Y_{S,R}(t) = f(P_{S,R}(t) - I_R(t)), \tag{3.4}$$

where f is a non-decreasing function, and $P_{S,R}(t) - I_R(t)$ represents the SINR in dB at time t.[2] For IEEE 802.15.4-compatible radios such as Chipcon CC2420, for instance,

$$Y_{S,R}(t) = (1 - \frac{8}{15} \times \frac{1}{16} \times \sum_{k=2}^{16} (-1)^k \binom{16}{k} e^{(20\times(P_{S,R}(t)-I_R(t))\times(\frac{1}{k}-1))})^{8\ell},$$

where ℓ is the packet length in units of bytes [133]. Given that the function f is usually non-linear and to address this challenge of non-linear control, we propose to approximate function f through linearization and use self-tuning regulators [55] to adapt controller behavior depending on the current operating point of the system. Given the SINR $P_{S,R}(t) - I_R(t)$ at time t ($t = 1, 2, \ldots$), more specifically, we linearize function f with the following linear function:

$$\begin{aligned} & Y_{S,R}(t) = a(t)(P_{S,R}(t) - I_R(t)) + b(t), \\ & \text{where} \\ & a(t) = f'(P_{S,R}(t) - I_R(t)), \\ & b(t) = f(P_{S,R}(t) - I_R(t)) - (P_{S,R}(t) - I_R(t))a(t). \end{aligned} \tag{3.5}$$

As a first-step towards ensuring predictable link reliability in distributed scheduling, we consider in this study mostly-static wireless networks where the average background noise power and average wireless path loss are mostly static and do not change at very short timescales (e.g., a few milliseconds for transmitting a few packets) [23]; focusing on predictable co-channel inter-

[2]Note that feedback control tends to be robust to modeling errors such that it is robust to minor model deviations from the theoretical model (3.4) in practice [55, 81].

ference control, we also only consider the cases when the data transmission power along a link is fixed even though different links may use different transmission powers; mobile networks and data transmission power control are relegated as future research. Assuming a discrete-time model where system properties such as the average background noise power and the average wireless path loss remain constant between time instants t and $t+1$,[3] $I_R(t+1)$ may differ from $I_R(t)$ for two possible reasons:

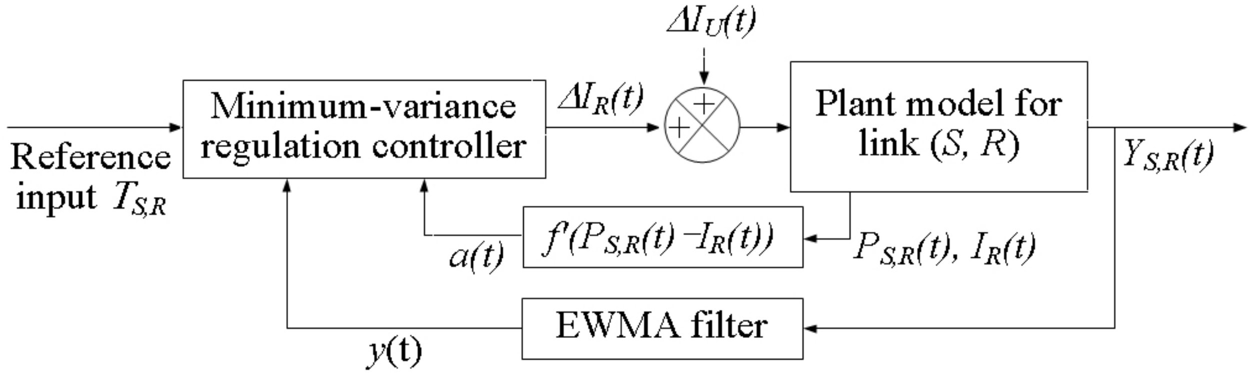
- From time t to $t+1$, the PRK model parameter may change from $K_{S,R,T_{S,R}}(t)$ to $K_{S,R,T_{S,R}}(t+1)$. Accordingly, the exclusion region around the receiver R changes from $\mathbb{E}_{S,R,T_{S,R}}(t)$ to $\mathbb{E}_{S,R,T_{S,R}}(t+1)$. If $K_{S,R,T_{S,R}}(t+1) > K_{S,R,T_{S,R}}(t)$, nodes in $\mathbb{E}_{S,R,T_{S,R}}(t+1) \setminus \mathbb{E}_{S,R,T_{S,R}}(t)$ may transmit concurrently with the transmission from S to R and thus introduce interference to R at time t but not at time $t+1$; similarly, if $K_{S,R,T_{S,R}}(t+1) < K_{S,R,T_{S,R}}(t)$, nodes in $\mathbb{E}_{S,R,T_{S,R}}(t) \setminus \mathbb{E}_{S,R,T_{S,R}}(t+1)$ may introduce interference to R at time $t+1$ but not at time t. We use $\Delta I_R(t)$ to denote the average interference change at receiver R due to the change of the PRK model parameter from t to $t+1$. Since the receiver R can control the changes of the PRK model parameter as we will discuss shortly, $\Delta I_R(t)$ can be controlled by the receiver R and is thus treated as the "control input".

- The set of nodes that are not in the exclusion region around the receiver R but transmit concurrently with the link (S, R) may change from time t to $t+1$. Accordingly, the average interference introduced by nodes outside the exclusion region around R changes from t to $t+1$, and we use $\Delta I_U(t)$ to denote this change. Since $\Delta I_U(t)$ is beyond the local control of link (S, R), we treat $\Delta I_U(t)$ as a "disturbance" to the system and denote the mean of $\Delta I_U(t)$ as $\mu_U(t)$.

Therefore,

$$I_R(t+1) = I_R(t) + \Delta I_R(t) + \Delta I_U(t),$$

[3]In protocol implementation, the actual time interval between t and $t+1$ can be chosen to be the small interval required for computing a sample of link reliability.



$\Delta I_U(t)$
Minimum-variance regulation controller
$\Delta I_R(t)$
Plant model for link (S, R)
Reference input $T_{S,R}$
$Y_{S,R}(t)$
$f'(P_{S,R}(t) - I_R(t))$
$a(t)$
$P_{S,R}(t), I_R(t)$
EWMA filter
$y(t)$

Figure 77: PRK model instantiation: minimum-variance regulation control architecture

where $\Delta I_R(t)$ and $\Delta I_U(t)$ are in units of dB. Using the linear approximation of function f as shown by Equation (3.5) at time t, the predicted link reliability for time $t+1$ calculates as follows:

$$Y_{S,R}(t+1) = a(t)(P_{S,R}(t+1) - I_R(t+1)) + b(t).$$

Therefore, the "plant model" for link (S, R) at time t is

$$\begin{aligned} I_R(t+1) &= I_R(t) + \Delta I_R(t) + \Delta I_U(t) \\ P_{S,R}(t+1) &= P_{S,R}(t) \\ Y_{S,R}(t+1) &= a(t)(P_{S,R}(t+1) - I_R(t+1)) + b(t) \end{aligned} \tag{3.6}$$

where $(I_R(.), P_{S,R}(.))$ and $Y_{S,R}(.)$ are the "state" and the "output" of the plant respectively. To deal with the noise in measuring $Y_{S,R}(.)$, we use an exponentially-weighted-moving-average (EWMA) filter with a weight factor c ($0 \leq c < 1$) in the feedback loop [55].[4] Thus, the system model is as shown in Figure 77, where

$$y(t) = cy(t-1) + (1-c)Y_{S,R}(t) \tag{3.7}$$

[4] c determines the tradeoff between the stability and agility of the EWMA filter. We use a relatively large c (e.g., $\frac{15}{16}$) in our implementation so that the filter is relatively stable in the presence of noise.

Given the probabilistic nature of wireless communication, the measured link reliability $y(t)$ is expected to be inherently random. Thus the goal is to minimize the variance of $y(t)$ while making sure that its mean value is the required link reliability. More formally, the objective of the control design at time t is to choose the control input $\Delta I_R(t)$ that minimizes the variance of $y(t+1)$ while ensuring $E[y(t+1)] = T_{S,R}$, where $T_{S,R}$ is the required link reliability.[5] For this minimum-variance regulation control problem, we have

Theorem 1. *The control input that minimizes $var[y(t+1)]$ while ensuring $E[y(t+1)] = T_{S,R}$ is*

$$\Delta I_R(t) = \frac{(1+c)y(t) - cy(t-1) - T_{S,R}}{(1-c)a(t)} - \mu_U(t). \tag{3.8}$$

Proof. In what follows, we first derive the minimum-variance control input $\Delta I_R(t)$ by assuming $E[y(t+1)] = T_{S,R}$, then we show that $E[y(t+1)] = T_{S,R}$ actually holds with the derived $\Delta I_R(t)$.

If $E[y(t+1)] = T_{S,R}$, then

$$\begin{aligned} var[y(t+1)] &= E[y(t+1) - E[y(t+1)]]^2 \\ &= E[y(t+1) - E[y(t+1)]- \\ &\quad (1-c)a(t)(\mu_U(t) - \mu_U(t))]^2 \\ &= E[cy(t) + (1-c)[a(t)(P_{S,R}(t+1)- \\ &\quad I_R(t+1)) + b(t)] - T_{S,R}- \\ &\quad (1-c)a(t)(\mu_U(t) - \mu_U(t))]^2 \\ &= E[X - (1-c)a(t)(\Delta I_U(t) - \mu_U(t))]^2 \end{aligned} \tag{3.9}$$

where

$$\begin{aligned} X &= cy(t) + (1-c)[a(t)(P_{S,R}(t+1) - I_R(t)) + b(t)] \\ &\quad -T_{S,R} - (1-c)a(t)\mu_U(t) - (1-c)a(t)\Delta I_R(t). \end{aligned}$$

Since $E[(1-c)a(t)(\Delta I_U(t) - \mu_U(t))] = 0$, we need $X = 0$ to minimize $var[y(t+1)]$, and the

[5]Note that $E[y(t+1)] = E[Y_{S,R}(t+1)]$ at steady state.

corresponding control input is as follows:

$$\Delta I_R(t) = \frac{cy(t)+(1-c)[a(t)[P_{S,R}(t+1)-I_R(t)]+b(t)]-T_{S,R}}{(1-c)a(t)} - \mu_U(t).$$

With a constant data transmission power, we have $P_{S,R}(t+1) = P_{S,R}(t)$. Thus

$$\begin{aligned}
\Delta I_R(t) &= \frac{cy(t)+(1-c)[a(t)[P_{S,R}(t)-I_R(t)]+b(t)]-T_{S,R}}{(1-c)a(t)} - \mu_U(t) \\
&= \frac{cy(t)+(1-c)Y_{S,R}(t)-T_{S,R}}{(1-c)a(t)} - \mu_U(t) \\
&= \frac{cy(t)+y(t)-cy(t-1)-T_{S,R}}{(1-c)a(t)} - \mu_U(t) \\
&= \frac{(1+c)y(t)-cy(t-1)-T_{S,R}}{(1-c)a(t)} - \mu_U(t).
\end{aligned}$$

Given the above control input $\Delta I_R(t)$,

$$\begin{aligned}
y(t+1) &= cy(t) + (1-c)[a(t)(P_{S,R}(t+1) - I_R(t+1)) + b(t)] \\
&= cy(t) + (1-c)[a(t)(P_{S,R}(t+1) - I_R(t) - \Delta I_R(t) - \\
&\qquad \Delta I_U(t)) + b(t)] \\
&= cy(t) + (1-c)[a(t)(P_{S,R}(t+1) - I_R(t) - \\
&\quad \frac{cy(t)+(1-c)Y_{S,R}(t)-T_{S,R}}{(1-c)a(t)} - \mu_U(t) - \Delta I_U(t)) + b(t)] \\
&= cy(t) + (1-c)[a(t)(P_{S,R}(t+1) - I_R(t) - \\
&\quad \frac{cy(t)+(1-c)[a(t)[P_{S,R}(t+1)-I_R(t)]+b(t)]-T_{S,R}}{(1-c)a(t)} - \\
&\quad \mu_U(t) - \Delta I_U(t)) + b(t)] \\
&= T_{S,R} + (1-c)a(t)(\Delta I_U(t) - \mu_U(t))
\end{aligned} \tag{3.10}$$

Since $E[(1-c)a(t)(\Delta I_U(t) - \mu_U(t))] = 0$, $E[y(t+1)] = T_{S,R}$ indeed holds. □

As a first milestone towards predictable link reliability in interference-oriented scheduling, here we focus on ensuring the expected link reliability. Besides being important for mission-critical sensing and control by itself, ensuring the expected link reliability also reduces communication delay and improves network throughput as we will show in Section 3, which are all important for real-time sensing and control. Instantaneous link reliability at a time instant can also be controlled by integrating PRK-based scheduling with techniques such as power control [79] and rate control [12], but detailed study of this is beyond the scope of this paper.

From $\Delta I_R(t)$ to $K_{S,R,T_{S,R}}(t+1)$. Given that it is convenient for the receiver R to measure link reliability $y(t)$ [132], we propose to execute the minimum-variance controller (3.8) at R. Using similar techniques as what we will discuss in Section 2.2, R can also measure $P_{S,R}(t)$ and $I_R(t)$, thus R can compute $a(t)$ using Equation (3.5). For each time instant t, R can also derive $\Delta I_U(t-1)$ based on $I_R(t)$, $I_R(t-1)$, $\Delta I_R(t-1)$, and Equation (3.6); using these derived samples of $I_U(.)$ and an EWMA filter, R can then estimate $\mu_U(.)$. Therefore, R can execute the controller (3.8) using information that is either locally measured (e.g., for $y(t)$ and $y(t-1)$) or locally derived (e.g., for $a(t)$ and $\mu_U(t)$).

After R computes the control input $\Delta I_R(t)$ at time t, R needs to compute $K_{S,R,T_{S,R}}(t+1)$ so that

$$\begin{cases} K_{S,R,T_{S,R}}(t+1) = K_{S,R,T_{S,R}}(t), & \text{if } \Delta I_R(t) = 0 \\ K_{S,R,T_{S,R}}(t+1) > K_{S,R,T_{S,R}}(t), & \text{if } \Delta I_R(t) < 0 \\ K_{S,R,T_{S,R}}(t+1) < K_{S,R,T_{S,R}}(t), & \text{if } \Delta I_R(t) > 0 \end{cases} \tag{3.11}$$

and that, when the PRK model parameter is $\min\{K_{S,R,T_{S,R}}(t), K_{S,R,T_{S,R}}(t+1)\}$, the expected interference introduced to R by the nodes in either $\mathbb{E}_{S,R,T_{S,R}}(t)$ or $\mathbb{E}_{S,R,T_{S,R}}(t+1)$ but not in both is as close to $|\Delta I_R(t)|$ as possible while ensuring that the expected link reliability is no less than $T_{S,R}$ when the PRK model parameter is $K_{S,R,T_{S,R}}(t+1)$.[6] To realize this, we define, for each node C in the local region around R, the expected interference $I(C,R,t)$ that C introduces to R when C is not in the exclusion region of R. Then $I(C,R,t) = \beta_C(t)P(C,R,t)$, where $\beta_C(t)$ is the probability for C to transmit data packets at time t and $P(C,R,t)$ is the power strength of the data signals reaching R from C.[7] Considering the discrete nature of node distribution in space and the requirement on satisfying the minimum link reliability $T_{S,R}$, we propose the following rules for

[6]Due to the discrete nature of node distribution, the resulting link reliability may be slightly higher than the required reliability $T_{S,R}$ instead of being exactly equal to $T_{S,R}$.

[7]$P(C,R,t)$ and $\beta_C(t)$ can be estimated through purely local coordination between R and C using the protocol signaling mechanism of Section 2.2.

computing $K_{S,R,T_{S,R}}(t+1)$:

- When $\Delta I_R(t) = 0$, let $K_{S,R,T_{S,R}}(t+1) = K_{S,R,T_{S,R}}(t)$.
- When $\Delta I_R(t) < 0$ (i.e., need to expand the exclusion region), let $\mathbb{E}_{S,R,T_{S,R}}(t+1) = \mathbb{E}_{S,R,T_{S,R}}(t)$, then keep adding nodes not already in $\mathbb{E}_{S,R,T_{S,R}}(t+1)$, in the non-increasing order of their data signal power at R, into $\mathbb{E}_{S,R,T_{S,R}}(t+1)$ until the node B such that adding B into $\mathbb{E}_{S,R,T_{S,R}}(t+1)$ makes $\sum_{C \in \mathbb{E}_{S,R,T_{S,R}}(t+1) \setminus \mathbb{E}_{S,R,T_{S,R}}(t)} I(C,R,t) \geq |\Delta I_R(t)|$ for the first time. Then let $K_{S,R,T_{S,R}}(t+1) = \frac{P(S,R,t)}{P(B,R,t)}$.
- When $\Delta I_R(t) > 0$ (i.e., need to shrink the exclusion region), let $\mathbb{E}_{S,R,T_{S,R}}(t+1) = \mathbb{E}_{S,R,T_{S,R}}(t)$, then keep removing nodes out of $\mathbb{E}_{S,R,T_{S,R}}(t+1)$, in the non-decreasing order of their data signal power at R, until the node B such that removing any more node after removing B makes $\sum_{C \in \mathbb{E}_{S,R,T_{S,R}}(t) \setminus \mathbb{E}_{S,R,T_{S,R}}(t+1)} I(C,R,t) > |\Delta I_R(t)|$ for the first time. Then let $K_{S,R,T_{S,R}}(t+1) = \frac{P(S,R,t)}{P(B,R,t)}$.

Figure 78 demonstrates the above idea for cases when $\Delta I_R(t) \neq 0$. In our study, we set the initial

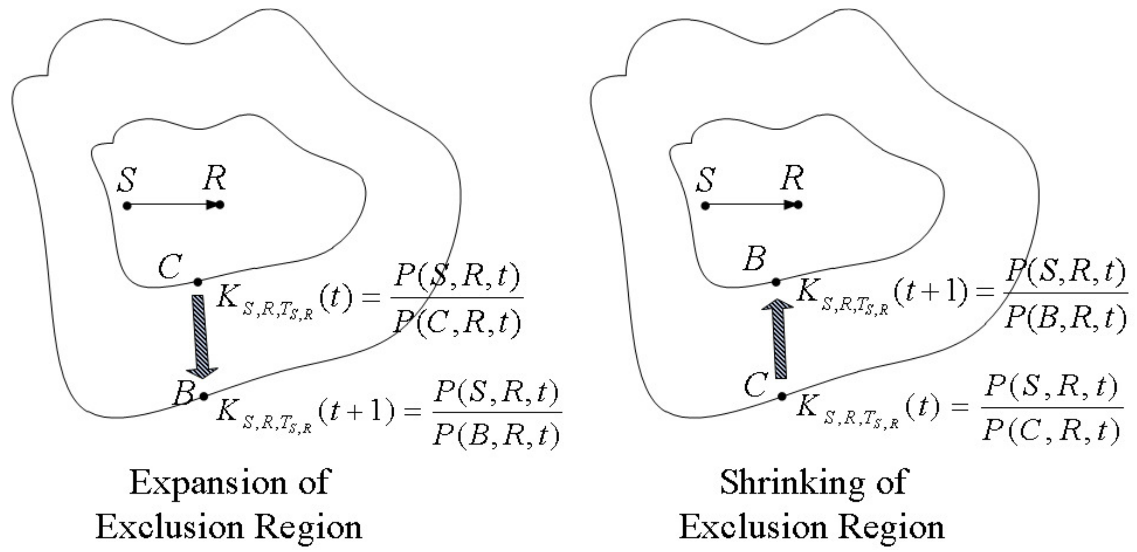

S
R
C
$K_{S,R,T_{S,R}}(t) = \frac{P(S,R,t)}{P(C,R,t)}$
B
$K_{S,R,T_{S,R}}(t+1) = \frac{P(S,R,t)}{P(B,R,t)}$
Expansion of Exclusion Region
S
R
B
$K_{S,R,T_{S,R}}(t+1) = \frac{P(S,R,t)}{P(B,R,t)}$
C
$K_{S,R,T_{S,R}}(t) = \frac{P(S,R,t)}{P(C,R,t)}$
Shrinking of Exclusion Region

Figure 78: Computing $K_{S,R,T_{S,R}}(t+1)$

value of the PRK model parameter such that the initial exclusion region around R includes every node whose transmission alone, concurrent with the transmission from S to R, can make the link reliability drop below $T_{S,R}$.

Stability of self-tuning adaptive control. The controller design and analysis based on the linear model (3.5) tend to be more accurate when $y(t)$ is closer to $T_{S,R}$. When $y(t)$ is far away from $T_{S,R}$, directly using the linear model (3.5) may lead to significant undershoot or overshoot in feedback control. Assuming the target operating point is A where the link reliability is $T_{S,R}$ in Figure 79, for instance, applying the linear model (3.5) and the control input (3.8) when the operating point

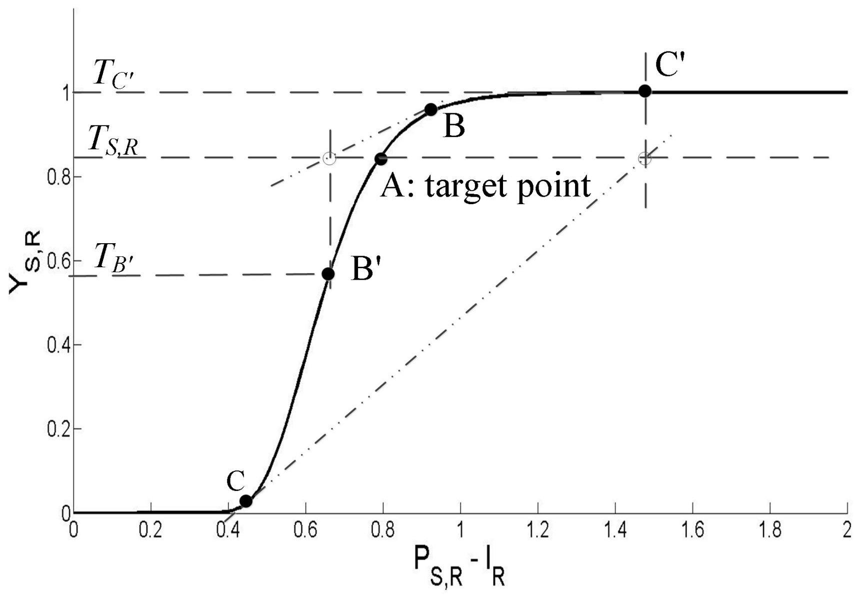


Figure 79: Stability of adaptive control (3.8)

is B at time t will lead to $E[y(t + 1)] = T_{B'}$, which is significantly lower than $T_{S,R}$ and thus lead to significant undershoot; similarly, applying the linear model (3.5) and the control input (3.8) when the operating point is C at time t will lead to $E[y(t + 1)] = T_{C'}$, which is significantly higher than $T_{S,R}$ and thus lead to significant overshoot. Significant undershoot or overshoot is not only undesirable from a single-link's point of view, it may also lead to network-wide instability in feedback control due to the coupling between individual links via $\Delta I_U(t)$. For stability and for avoiding significant undershoot and overshoot in control, we propose to replace $a(t)$ with its refined version $a_r(t)$ in the controller implementation:

$$a_r(t) = \begin{cases} a(t), & \text{if } |y(t) - T_{S,R}| \le e_0 \\ a_0, & \text{if } |y(t) - T_{S,R}| > e_0 \end{cases} \tag{3.12}$$

where e_0 is a threshold value for the linear model (3.5) to be accurate around the neighborhood of $T_{S,R}$, and $a_0 = \frac{T_{S,R}-y(t)}{f^{-1}(T_{S,R})-f^{-1}(y(t))}$ is the gradient of the line connecting the current operating point $y(t)$ and the target point $T_{S,R}$ on function f. Letting $a(t) = a_0$ when $|y(t) - T_{S,R}| > e_0$ avoids overshoot and undershoot in the feedback control of $K_{S,R,T_{S,R}}(.)$ at link (S, R), thus preventing $Y_{S,R}(.)$ from oscillating around $T_{S,R}$ for a given disturbance $\Delta I_U(.)$ and helping enable network-wide convergence in the regulation control. Note that, according to Huang et al. [59], the functional form of f in Equation (3.4) and thus its gradient are much more stable than the specific realization of f (e.g., specific mapping between $Y_{S,R}$ and $P_{S,R} - I_R$) across different network and environmental conditions; hence letting $a_r(t)$ be $a(t)$ instead of a_0 when $|y(t) - T_{S,R}| \le e_0$ helps address the inaccuracy of the theoretical model (3.4) in practice. In our implementation, we use an e_0 of 5%.

2.2 Local signal maps for real-world use of the PRK model

Given a link (S, R) and a specific instantiation of the PRK model, the parameter $K_{S,R,T_{S,R}}(t)$ defines an exclusion region $\mathbb{E}_{S,R,T_{S,R}}(t)$ around the receiver R such that a node $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$ if and only if $P(C, R, t) \ge \frac{P(S,R,t)}{K_{S,R,T_{S,R}}(t)}$. In PRK-based scheduling, every node $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$ should be aware of its existence in $\mathbb{E}_{S,R,T_{S,R}}(t)$ and should not transmit concurrently with the reception at R; yet it is difficult to ensure this property for the following real-world complexities in wireless communication: 1) node C may be located beyond the communication range of R such that R cannot inform C about its state (e.g., the value of $\frac{P(S,R,t)}{K_{S,R,T_{S,R}}(t)}$) with the regular data transmission power; 2) wireless communication may be anisotropic such that it is difficult for R to transmit protocol signaling messages that reaches and only reaches nodes in $\mathbb{E}_{S,R,T_{S,R}}(t)$; 3) wireless communication may be asymmetric such that nodes interfering with one another may not know one another's state (e.g., $\frac{P(S,R,t)}{K_{S,R,T_{S,R}}(t)}$).

Local signal maps. To address these challenges, we propose that every node R maintains a *local signal map* that contains the average signal power attenuation between R and every node C close-by. To measure the signal power attenuation $P'(C,R)$ from a node C to another node R, we let C inform R of its transmission power P_C by piggybacking the information onto its packets to R, and then R can derive the power attenuation as long as R can estimate the power of the received signals from C, denoted by $P(C,R)$.[8] To this end, R samples the RSSI value P_{total} at an instant right before finishing receiving a packet from C, and, immediately after receiving the packet, R samples the RSSI value P_I again. As shown in Figure 80, P_I is the sum of the

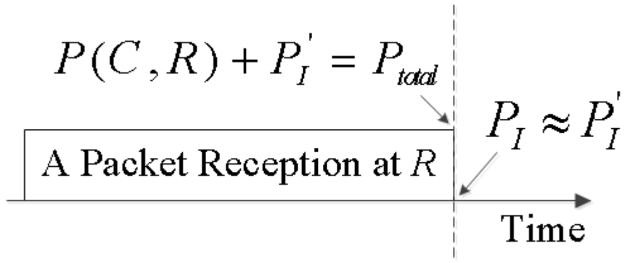


Figure 80: Estimation of signal power attenuation

background noise power and the interference power at R right after the packet reception, and $P_{total} = P(C,R) + P_I'$ where P_I' is the sum of the background noise power and the interference power at R right before the packet reception. As we will discuss in Section 2.3, signal maps are maintained in the control plane of the protocol PRKS where wireless channel access is based on the traditional random access method CSMA/CA as used in IEEE 802.15.4 and 802.11. Given that P_{total} and P_I can be sampled at very short interval (e.g., less than 0.01 milliseconds for TelosB motes [8]) and that the background noise power as well as the interference power do not change much in such short intervals in CSMA/CA-based wireless networks, the sum of the background noise power and the interference power do not change much immediately before and immediately

[8]To address the challenge of large interference range, P_C may potentially be higher than the regular data transmission power as we discuss later in this section.

after a packet reception, i.e., $P'_I \approx P_I$. Thus,

$$P(C, R) = P_{total} - P'_I \approx P_{total} - P_I. \tag{3.13}$$

Once R gets a sample of $P(C, R)$, it can compute a sample of $P'(C, R)$ as

$$P'(C, R) = P_C - P(C, R). \tag{3.14}$$

This way, R can get a series of samples of $P'(C, R)$ and then use these samples to derive the average signal power loss from C to itself.

Using the above method of sampling signal power attenuation, nodes close-by can establish their local signal maps through purely local sampling of their packet receptions without any global coordination in the network, and the local signal maps generated in this manner tend to be very accurate as we show next. For mostly-static WSC networks which we consider in this study, the local signal maps can be maintained in a passive manner after their initial establishment. That is, signal map maintenance does not introduce extra communication overhead other than having nodes piggyback their transmission powers onto the packets they transmit anyway. Note that the local signal map maintains power attenuation from a node C to R instead of simply the reception power of signals from C to R so that the signal map can be used to estimate the reception power of signals that are transmitted at different powers (e.g., for the control signals of protocol PRKS to be discussed in Section 2.3). For protocol signaling in PRK-based scheduling, the local signal maps also maintain bi-directional power attenuation between a pair of close-by nodes. After estimating $P'(C, R)$, for instance, R informs C of $P'(C, R)$ so that C is aware of the power attenuation from itself to R.

To corroborate the effectiveness of the aforementioned method of estimating wireless signal power attenuation, we apply it to estimate power attenuation across links in both the NetEye [11] and the Indriya [2] sensor network testbeds where nodes transmit at a power of -25dBm and -10dBm respectively. We first collect ground-truth data about power attenuation across links when there is only one transmitter at a time and no concurrent transmissions in the network; then we use the aforementioned method and Equations (3.13) and (3.14) to estimate power attenuation across links when all the nodes transmit packets using the CSMA/CA-based B-MAC [99] and at an average inter-packet interval of 25 seconds, 2.5 seconds, and 0.1 seconds respectively, which we denote as light traffic, medium traffic, and heavy traffic respectively.

For the NetEye testbed, Figures 81 and 82 show the CDFs of the absolute and relative errors

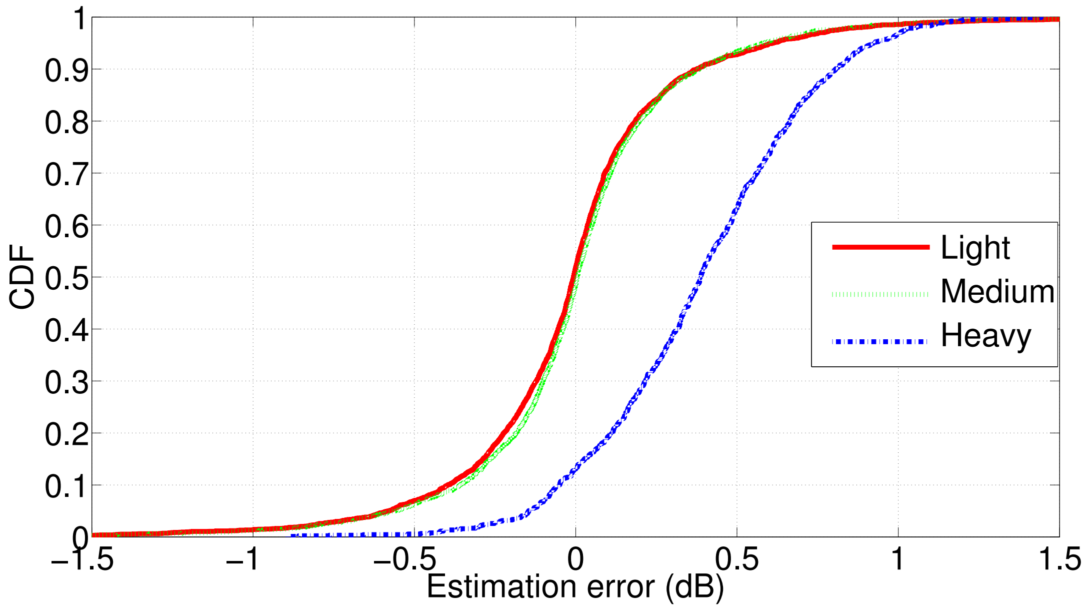


Figure 81: Absolute errors in estimating link signal power attenuation in NetEye

in estimating power attenuation across links in different traffic conditions, where the absolute error for a link is defined as the estimated attenuation minus the ground-truth attenuation for the link and the relative error is defined as the absolute error divided by the ground-truth attenuation. We see that the estimation is quite accurate. For instance, the relative estimation errors are all very close to 0 and almost always within $[-2\%, 2\%]$; in addition, the 95% confidence interval for the median relative error is $[-0.0508\%, 0.0535\%]$, $[-0.0152\%, 0.0280\%]$, and $[-0.0087\%, 0.0245\%]$ for the light, medium, and heavy traffic condition respectively, thus the median estimation error is 0 at the 95% confidence level for all traffic conditions.

For details of the estimation behavior, Figures 83 and 84 show the time series of signal power

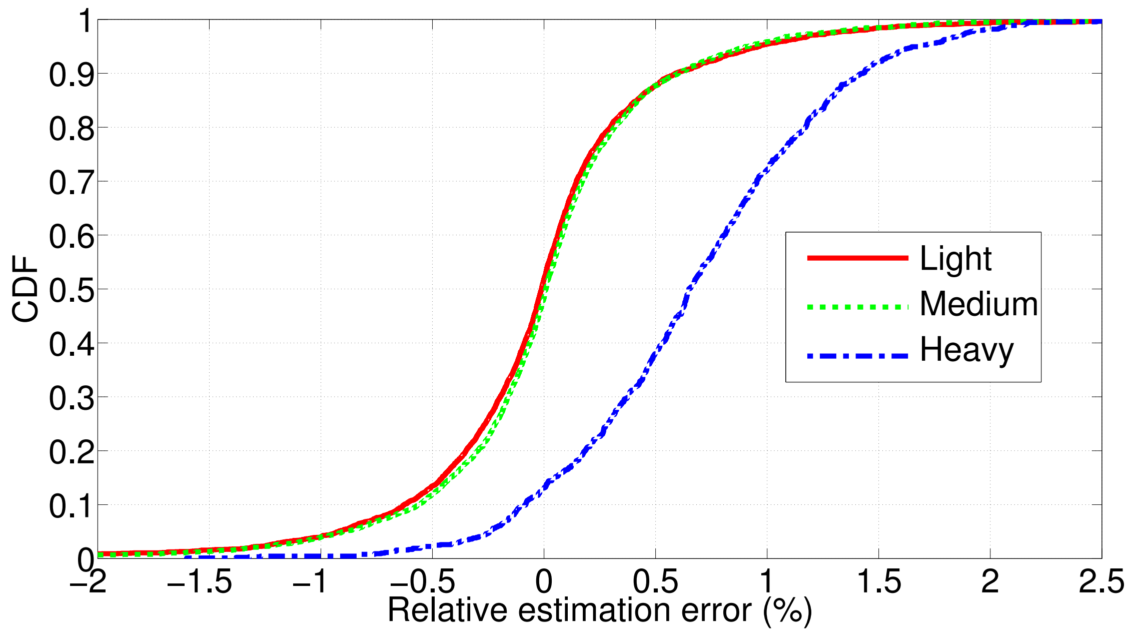


Figure 82: Relative errors in estimating link signal power attenuation in NetEye

attenuation for a typical link in NetEye where the power attenuation is estimated in heavy traffic condition and without concurrent transmission (i.e., ground-truth) respectively. We see that the power attenuation has small variation (e.g., mostly less than 1dB); in addition, even though the estimated power attenuation in heavy traffic seems to exhibit greater fluctuation, its mean value is nearly identical to that of the ground-truth data, thus further verifying the validity of the aforementioned method for estimating average signal power attenuation.

For the Indriya testbed, Figures 85 and 86 show the CDFs of the absolute and relative errors in estimating power attenuation across links in different traffic conditions, and Figures 87 and 88 show the time series of the estimated and ground-truth signal power attenuation for a typical link respectively. The observations are similar to those for the NetEye testbed, showing the effectiveness of our method of signal power attenuation estimation in different network and traffic conditions.

Protocol signaling based on signal maps. The local signal map at R records signal power attenuation between R and the nodes close-by. Using these information and the transmission power control algorithms proposed by Leung et al. [73], node R can broadcast signaling packets at an

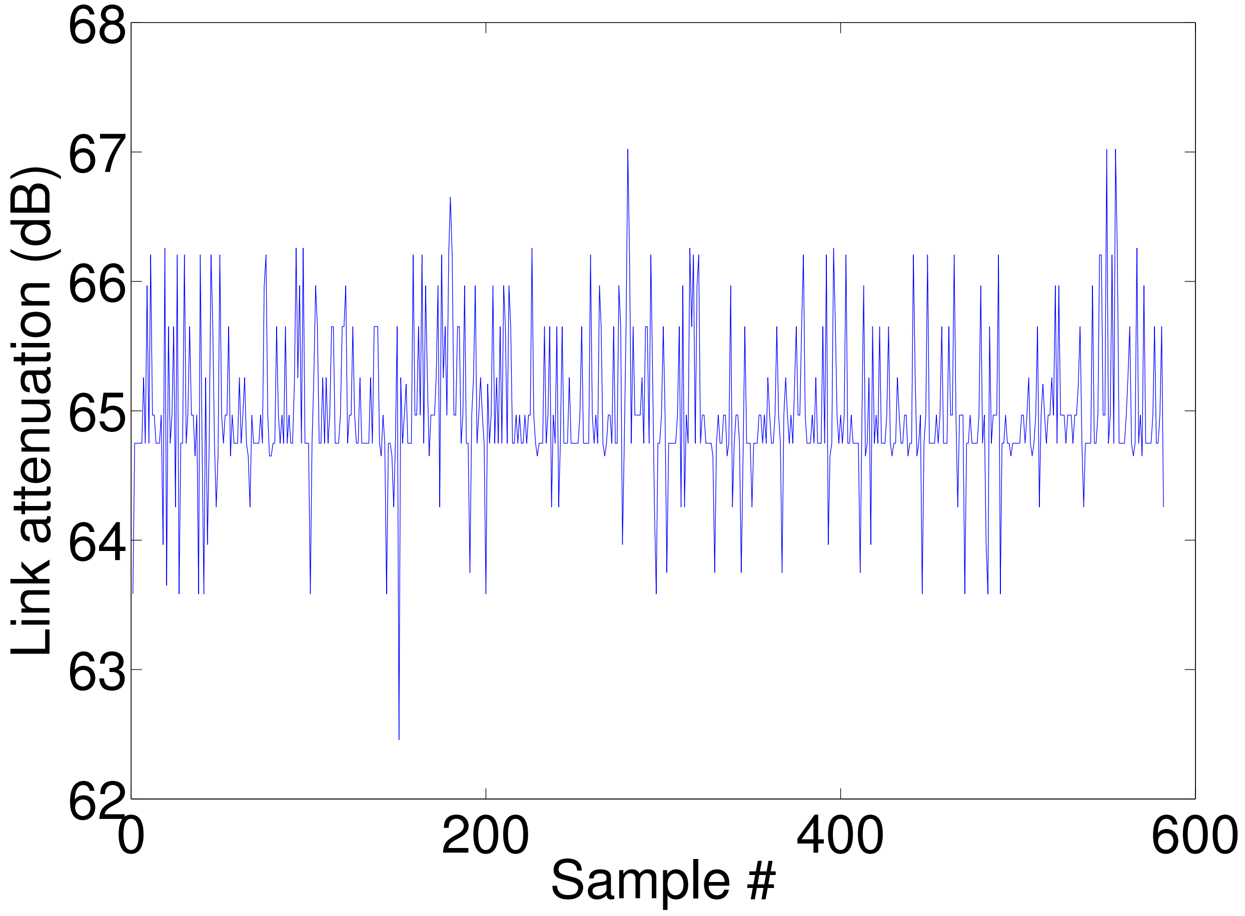


Figure 83: Time series for a link’s signal power attenuation in NetEye: heavy traffic

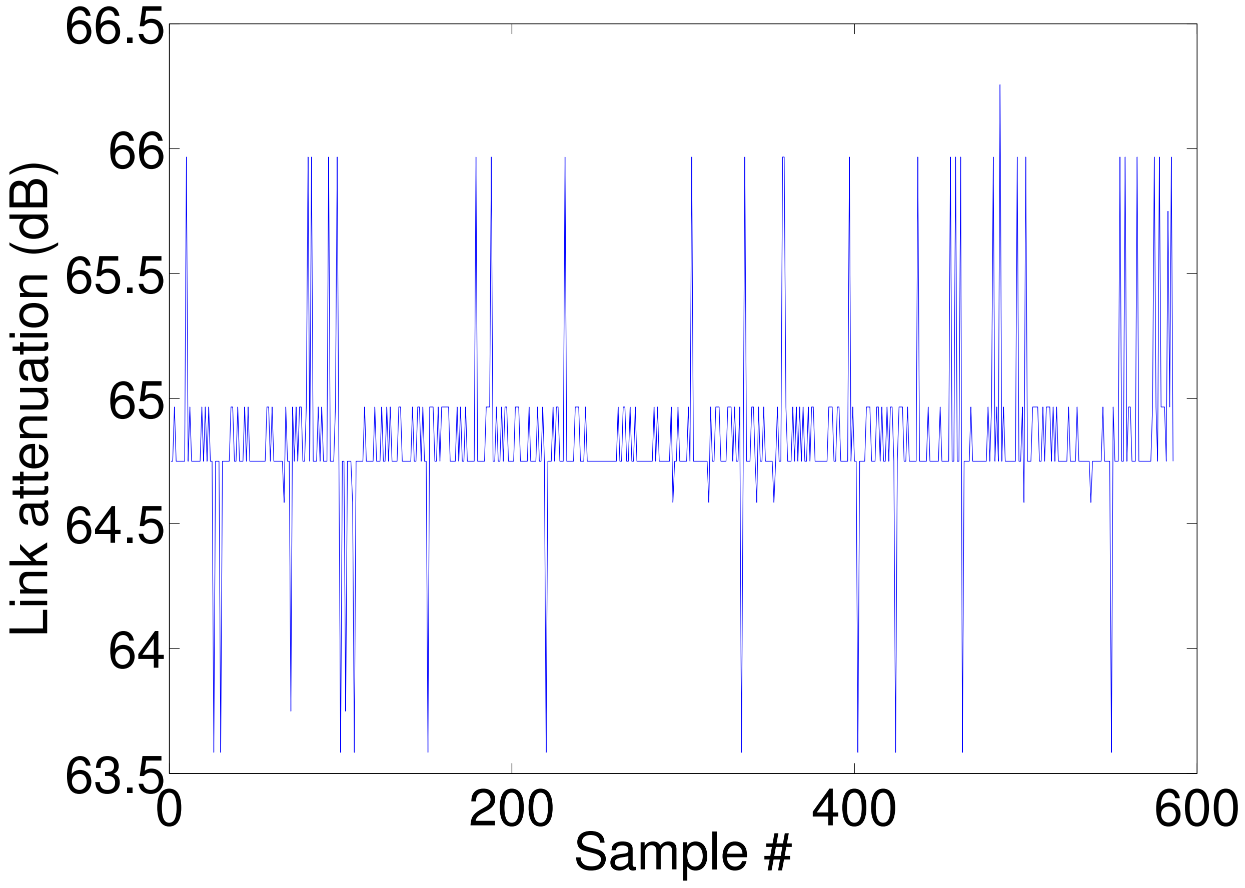


Figure 84: Time series for a link's signal power attenuation in NetEye: ground-truth

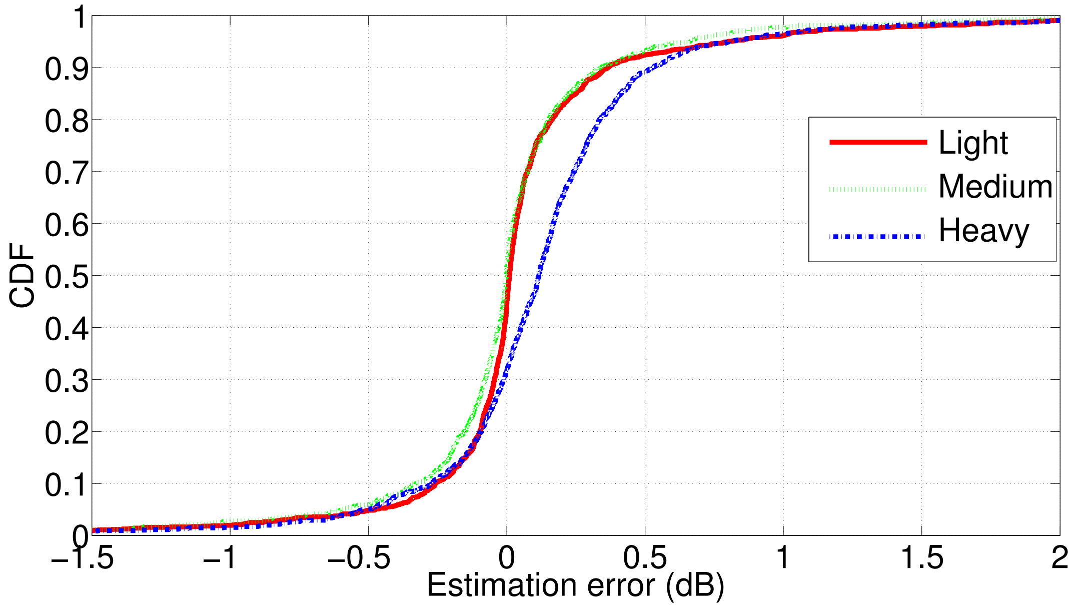


Figure 85: Absolute errors in estimating link signal power attenuation in Indriya

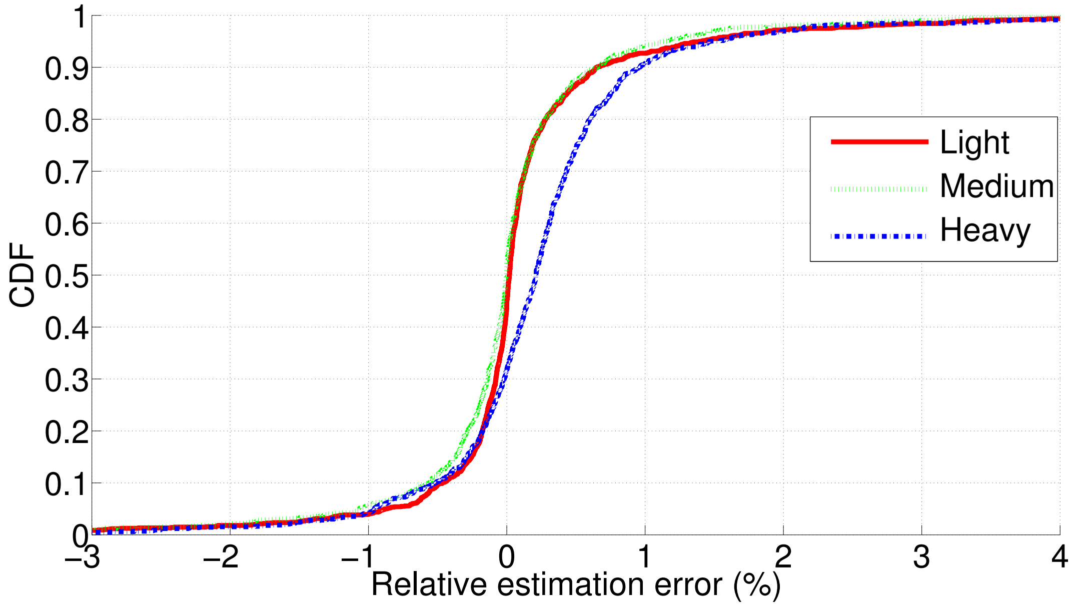


Figure 86: Relative errors in estimating link signal power attenuation in Indriya

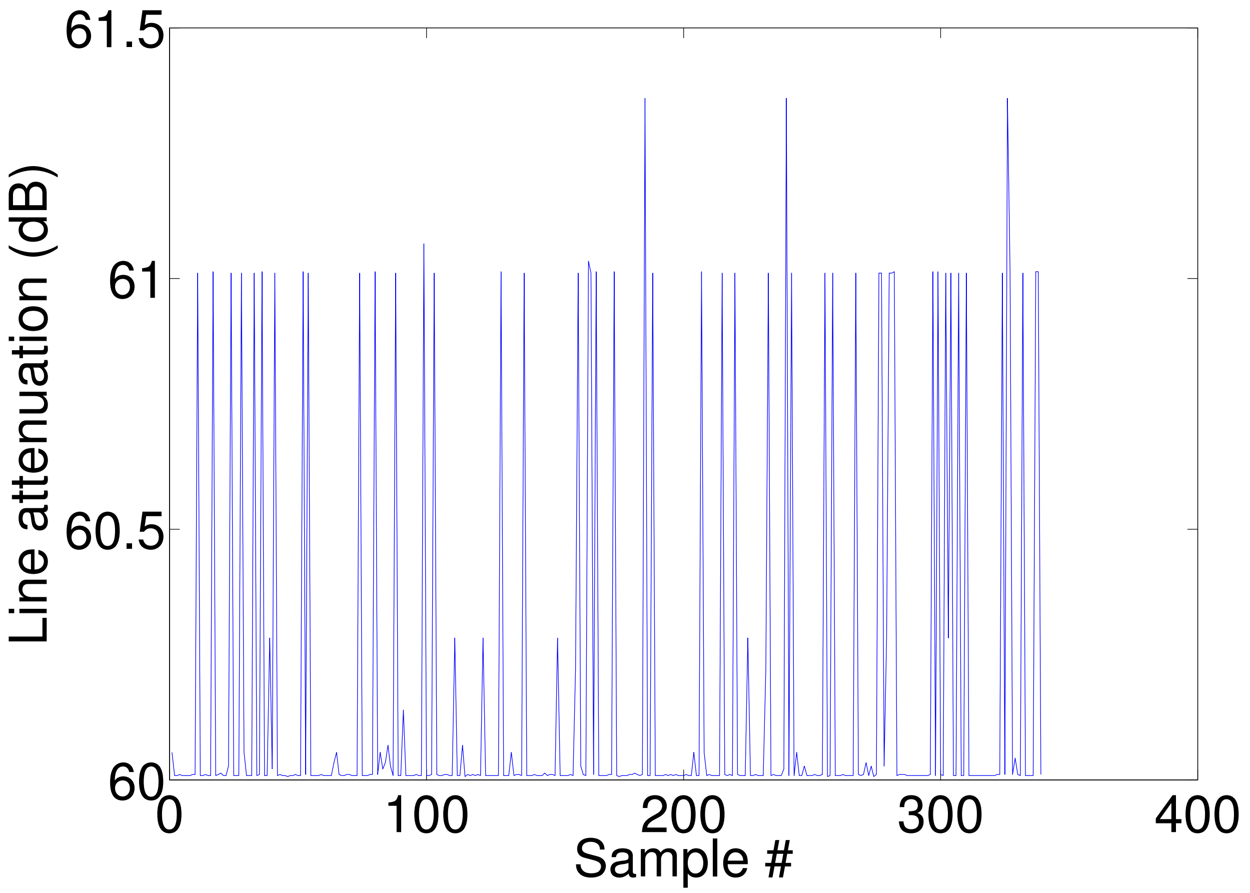


Figure 87: Time series for a link's signal power attenuation in Indriya: heavy traffic

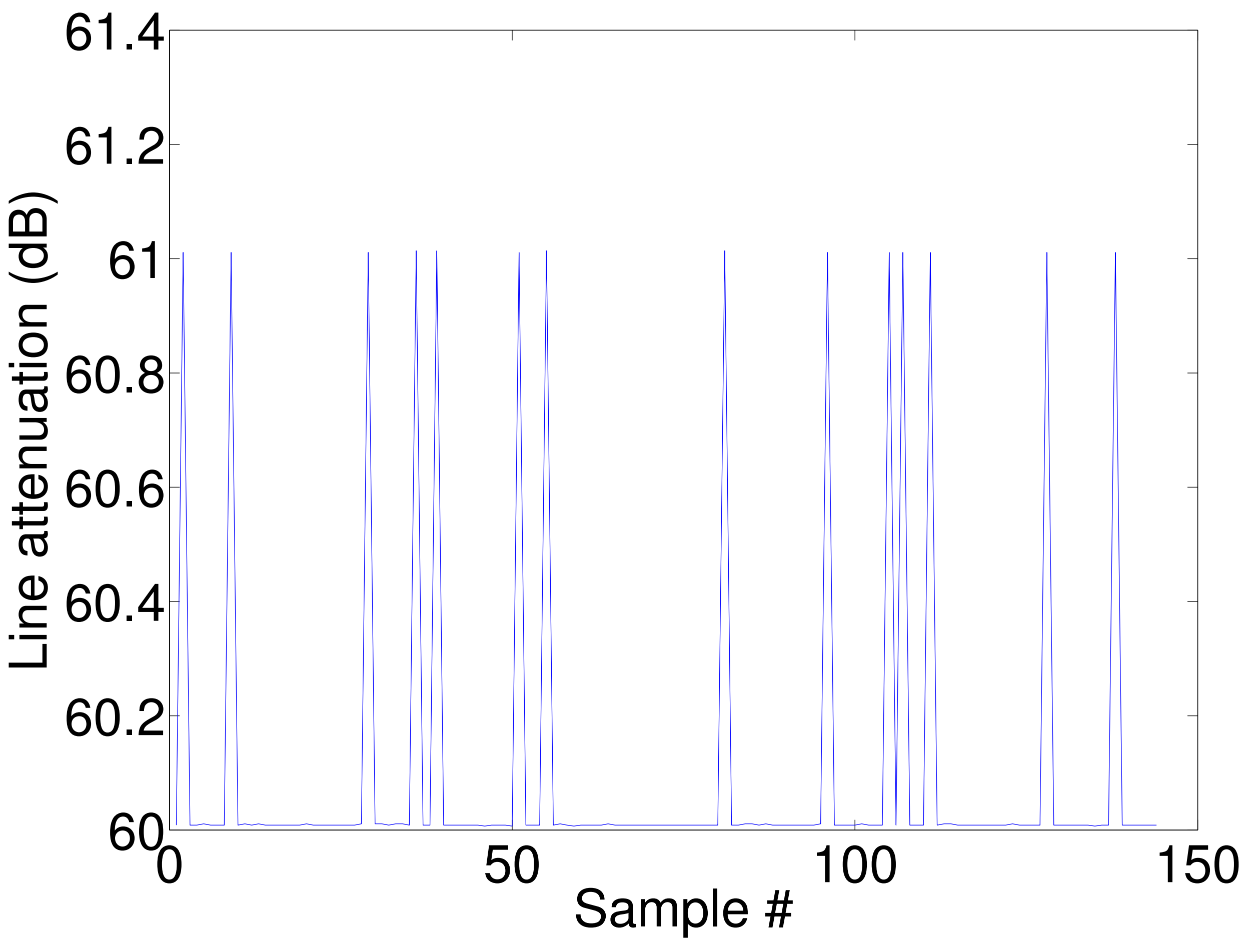


Figure 88: Time series for a link's signal power attenuation in Indriya: ground-truth

appropriate power level such that these packets with the value of $\frac{P(S,R,t)}{K_{S,R,T_{S,R}}(t)}$ can be received with high probability by all the nodes in the exclusion region $\mathbb{E}_{S,R,T_{S,R}}(t)$ around R; this can be accomplished even if a node $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$ is beyond the regular data communication range of R, in which case the broadcast packets are transmitted at a power higher than the regular data transmission power. Therefore, the local signal map enables addressing the challenge of large interference range through transmission power control; as we will show in Section 3, the required control signaling power is usually no more than 22dB over the data transmission power, and the power control scheme can be implemented using today's commodity radios such as Chipcon CC2420. To further increase the reliability of protocol signaling, node R can broadcast each signaling packet multiple times, and nodes in the exclusion region of R can re-broadcast the signaling packet they hear from R. To reduce the delay in information sharing, signaling packets with fresher information (i.e., information that has been transmitted for fewer number of times) also have higher priorities in channel access by using smaller contention windows in CSMA/CA.

When a node C receives the signaling packet from R, C can use its local signal map to decide whether his transmission may interfere with the transmission from S to R (i.e., whether $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$) by checking whether $P(C,R,t) \geq \frac{P(S,R,t)}{K_{S,R,T_{S,R}}(t)}$. Therefore, the signaling packets can reach nodes not in $\mathbb{E}_{S,R,T_{S,R}}(t)$ without falsely including those nodes into $\mathbb{E}_{S,R,T_{S,R}}(t)$, thus addressing the challenge of anisotropic wireless communication. Similarly, using power control algorithms and local signal maps, a pair of nodes C and R can inform each other of their respective states (e.g., the PRK model parameter and the data transmission probability) using different transmission powers for signaling packets, thus addressing the challenge of asymmetric wireless communication in protocol signaling.

For the correctness of the above protocol signaling method, the signal map of a node R should include the set $\mathbb{E}'$ of nodes whose transmission may interfere with the reception at R or whose reception may be interfered by the transmission by R (e.g., the transmission of ACK packets by R). Since the set $\mathbb{E}'$ may well be dynamic and uncertain depending network and environmental conditions, a node R dynamically adjusts the set of nodes in its local signal map through local coordination with nodes close-by, and R may also maintain a relative large signal map to include the nodes that may be in $\mathbb{E}'$ over time. Together with the PRK model instantiation method discussed in Section 2.1, the above field-deployable signaling mechanisms enable agile, high-fidelity identification of interference relations among nodes, thus serving as a foundation for predictable interference control.

2.3 Protocol PRKS: putting things together

Decoupling of protocol signaling & data transmissions. Based on the methods of PRK model instantiation and protocol signaling presented in Sections 2.1 and 2.2 respectively, two basic tasks of interference control are 1) enabling nodes to be accurately aware of the mutual interference relations among themselves and 2) controlling channel access so that no two interfering links use the same wireless channel at the same time. These tasks make the commonly-used single-channel contention-based approach unsuitable for the following reasons:

- In contention-based channel access control, each data transmission is usually preceded by a protocol signaling phase either implicitly through carrier sensing or explicitly through RTS-CTS handshake such as in IEEE 802.11. Due to the probabilistic nature of wireless communication and the potentially large interference range, it is difficult to make such per-transmission protocol signaling perfectly reliable even with the mechanisms discussed in Section 2.2. Accordingly, it is difficult for nodes to be accurately aware of their mutual interference relations, thus it is difficult to control interference in a predictable manner.

- Even if we can make the per-transmission protocol signaling more reliable through mechanisms such as retransmission of signaling packets, this introduces significant delay and overhead for each data transmission. Even worse, the signaling packets may well be transmitted at relatively higher power to ensure coverage of the potentially large exclusion regions, and the high-power transmissions of signaling packets introduce significant interference to the data transmissions themselves; in trying to ensure the required data delivery reliability in the presence of strong interference from protocol signaling, nodes will adapt their PRK model parameters to expand their individual exclusion regions, which in turn requires the signaling packets to be transmitted at even higher power and thus leads to system instability (as we have seen in our earlier trials of contention-based approaches).

To address the aforementioned challenges, we propose to decouple protocol signaling from data transmission by leveraging the different timescales of PRK model adaptation and data transmission. Given a link (S, R), highly accurate estimation of its reliability usually requires the knowledge of the transmission status of several (e.g., 20) data transmissions along (S, R) [132]. Accordingly, it takes time to get a new link reliability feedback, and the timescale of PRK model adaptation as well as the resulting change in interference relations between (S, R) and close-by nodes/links is longer than the timescale of individual data transmissions along (S, R). Using the protocol signaling mechanisms discussed in Section 2.2, the receiver R can inform, after each PRK model adaptation, the relevant nodes of the new value of parameter $K_{S,R,T_{S,R}}$ and thus the corresponding change in interference relations; we have observed, in our experimental analysis in Section 3, that each new value of $K_{S,R,T_{S,R}}$ can be reliably and quickly signaled within 1.4 transmissions of the signaling packet on average. Therefore, instead of requiring perfectly reliable signaling for each data transmission as in contention-based channel access control, we propose to treat protocol signaling as an independent process which ensures timely awareness of the mutual interference between nodes/links. Based on the latest information on mutual interference relations, data transmissions can be scheduled in a TDMA fashion without being coupled with protocol signaling.[9]

Besides enabling precise awareness of mutual interference relations, the decoupling of protocol signaling and data transmission also enables the transmission of signaling packets and data packets in different wireless channels, thus avoiding the interference between protocol signaling and data transmission as well as the corresponding system instability. For convenience, we regard the wireless channels used for protocol signaling and data transmission as the control channel and the data

[9]Note that the periodic sampling of physical processes in WSC networks also makes TDMA an efficient scheduling mechanism as compared with contention-based approaches.

channel respectively. Using a control channel is necessary for avoiding system instability while addressing the challenges of protocol signaling at the same time; since protocol signaling does not introduce high traffic load, it may well be able to reuse the control channel that has been set aside in industry standards such as IEEE 1609.4 [7] as well as in research proposals [108, 123, 97].

Protocol PRKS. Based on the above design principles, we propose the PRK-based scheduling protocol *PRKS* that separates the functionalities of PRK-based channel access control into control plane functions and data plane functions as shown in Figure 89. In the control plane, the sender

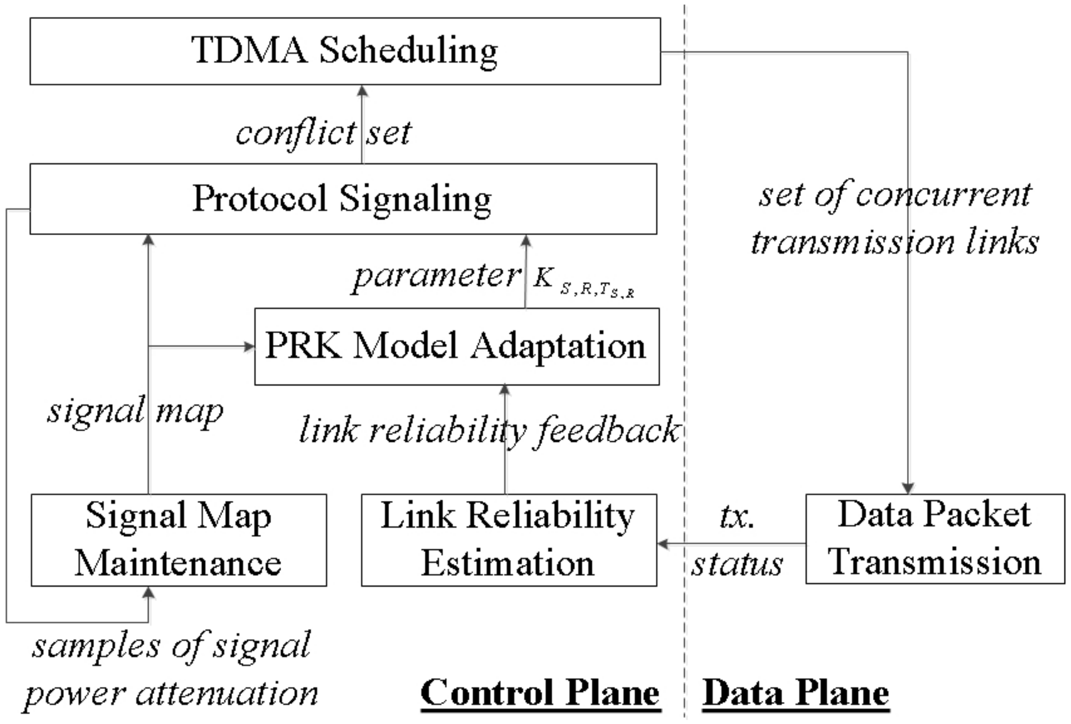


Figure 89: Architecture of PRKS

S and the receiver R of a given link (S, R) get to know the set of links whose transmissions cannot take place concurrently with the transmission from S to R through the protocol signaling mechanisms presented in Section 2.2, and we define this set of links as the *conflict set* of link (S, R). More specifically, a link (C, D) is in the conflict set of (S, R) and thus conflicting with (S, R) at a time instant t if links (C, D) and (S, R) share a common end-node, $C \in \mathbb{E}_{S,R,T_{S,R}}(t)$, or $S \in \mathbb{E}_{C,D,T_{C,D}}(t)$, where $T_{S,R}$ and $T_{C,D}$ are the required packet delivery reliability across (S, R) and (C, D) respectively. Based on the conflict sets of links, data transmissions along individual links can be scheduled in a distributed, TDMA manner according to the methodology of the Link-Activation-Multiple-Access (LAMA) algorithm [14]. With the LAMA algorithm, a link (S, R) is regarded as active in a time slot if S transmits to R in the slot. Given a time slot, the sender S and the receiver R of link (S, R) first compute the priorities for the link (S, R) and the links in the conflict set of (S, R) to be active in the time slot, then S decides to transmit to R and R decides to receive data from S if and only if, for this time slot, (S, R) has higher priority to be active than every conflicting link. Every node in the network computes link activation priorities in the same manner such that no two conflicting links will be active in the same time slot as long as links are accurately aware of their mutual interference relations. If a link (S, R) is active in a time slot, S will transmit data packet(s) to R in this time slot. The status (i.e., successes or failures) of data transmissions in the data plane are fed back into the control plane for estimating the in-situ link reliabilities, which in turn triggers PRK model adaptation and then the adaptation of the TDMA transmission scheduling accordingly. In the control plane, nodes also leverage the transmissions and receptions of protocol signaling packets to maintain their local signal maps as we have presented in Section 2.2.[10] Given that the instantiated PRK models precisely identify the conflict sets of individual links, the TDMA scheduling in PRKS also eliminates the hidden terminal issue which has been a basic challenge in interference-oriented channel access control since 1975 [113].

To avoid interference between protocol signaling transmissions and data transmissions, protocol signaling packets and data packets are transmitted in the control channel and the data channel respectively. By default, a node stays in the control channel. At the beginning of a time slot, every node executes the LAMA scheduling algorithm to decide whether any of its associated links will be active in this time slot. If one of its associated links is active in this time slot, the node switches to the data channel for data transmission or reception depending on whether the node is the transmitter or receiver of the associated active link; after the data transmission/reception, the node switches back to the control channel, and, if the node is a receiver of a link in this time slot,

[10]For a given link, the path losses in the control channel and the data channel are approximately the same given the close proximity of the two channels in spectrum [125].

it feeds back the status (i.e., success or failure) of this transmission to the control plane for link reliability estimation and the corresponding PRK model adaptation. On the other hand, if the node is not involved in any data transmission/reception in the time slot, it stays in the control channel, and it tries to access the control channel via CSMA/CA: if it wins channel access (e.g., sensing the channel as idle), it transmits a signaling packet including information on the PRK model parameters for all of its associated links and their conflicting links; if it does not win channel access, it stays in the control channel receiving signaling packets from other nodes and perform functions related to signal map maintenance and protocol signaling as discussed in Section 2.2. The length of a time slot is chosen such that the aforementioned actions can be completed in a single time slot whether the node is involved in control plane functions alone or it is also involved in a data transmission/reception.

With the above approach to PRK-based scheduling, the TDMA scheduling of data transmissions happens at the beginning of each time slot based on the PRK model information that is readily available in the control plane, hence there is no need for ensuring perfectly reliable protocol signaling on a per-transmission basis and thus no delay introduced on a per-transmission basis just for protocol signaling either. Given that the timescales of PRKS model adaptation at a link (S, R) is longer than the timescales of individual data transmissions along (S, R), in particular, the time instants t_a and t_b for two consecutive PRK model adaptations at (S, R) tend to be well separated such that, within the early part of the time window $[t_a, t_b]$, the PRK model parameter of link (S, R) generated at time t_a can be reliably delivered to the relevant nodes and then be used for the TDMA scheduling of data transmissions.

One premise for the correct operation of PRKS is that, for every link (S, R), the sender S and the receiver R always use the same PRK model parameters of the relevant links when deciding whether (S, R) should be active in a time slot; otherwise, S and R may well derive different conflicting relations between links, and S may think (S, R) shall be active for this time slot and switch to the data channel to transmit, but R thinks (S, R) shall be inactive and stays in the control channel, which makes R unable to receive the transmitted data from S and leads to data packet loss. Since protocol signaling takes time, however, there are transient periods when S and R may have inconsistent information about the PRK model parameters in the network. To avoid the receiver R being in the control channel when the sender S transmits in the data channel, one approach is to have S and R only use the same information about PRK model parameters and delay the use of newly updated PRK model parameters that are not known to both S and R, as in our preliminary study [135]. This approach relies on perfect consistency between S and R in the use of PRK model parameters; due to the probabilistic nature of wireless communication, however, the requirement on perfect consistency introduces non-negligible delay in the use of the latest PRK model parameters, which is undesirable as we will discuss shortly. To allow for the immediate use of the latest PRK model parameters as they become available while ensuring that the receiver R is in the data channel whenever the sender S transmits data packets, we introduce the following sender-receiver coordination mechanism:

- If the link (S, R) shall be active in a time slot t_0, the sender S computes the time slot t_1 when the link (S, R) will be active again the next time; then S piggybacks the value of t_1 onto the data packet, if any, to be transmitted to R at t_0 as well as onto every protocol signaling packet that S may transmit during t_0 and t_1. If the receiver R receives a data or a protocol signaling packet from S showing that S will transmit in a future time slot t_1, R will stay in the data channel at t_1 even if the local execution of the LAMA algorithm at R may show link (S, R) as inactive at t_1.

- After computing at time t_0 the next time slot t_1 to transmit to R, S will not transmit to R at any time slot $t'_1 \in (t_0, t_1)$ unless the receiver R tells S to transmit at t'_1 as we discuss next. This rule applies even if the local PRK model parameters at S shows at slot t'_1 that the link (S, R) shall be active at this time slot; this rule implicitly introduces delay in using the latest information on PRK model parameters, but we have observed that this delay is significantly less than that in the approach based on perfect consistency between S and R as we have discussed.

- After R learns that S will transmit in a future time slot t_1, if the execution of the LAMA algorithm at R at a time slot t'_0 shows that the link (S, R) shall be active at a time slot $t'_1 < t_1$ and if the time window of $[t'_0, t'_1]$ is long enough for R to successfully inform S of the value of t'_1 with high probability, R changes its local value of t_1 to t'_1 and piggybacks the value of t'_1 onto packets (e.g., protocol signaling packets) that R may transmit during $[t'_0, t'_1)$. If S receives a packet from R showing that link (S, R) shall be active at a future time slot $t'_1 < t_1$, S changes the value of t_1 to t'_1. This rule is to ameliorate the implicit delay in using the latest PRK parameter values that the previous rule may introduce.

- If the receiver R does not receive any data packet from S at the time slot t_1 (e.g., due to data packet loss), R enters and stays in a "conservative" state until it receives a packet from S again showing that (S, R) shall be active at another future time slot. While in the "conservative" state, the receiver R stays in the data channel for a time slot t_2 as long as, at t_2, the link (S, R) has higher priority to be active than other links associated with R; the conservative state ensures that R is in the data channel whenever S transmits a data packet to R, and it enables R to be synchronized with S on the data-transmission schedule again.

- In the system boot-up phase, the sender S executes the basic LAMA algorithm and the receiver R remains in the conservative state when deciding whether to stay in the data or control channel for a time slot, until S transmits for the first time and R receives a first packet from S respectively.

With the above coordination mechanism, the receiver R is guaranteed to be in the data channel whenever the sender S transmits data packets; in the mean time, R stays in the control channel often enough to be updated with the latest PRK model parameters of close-by links.

The aforementioned coordination between a sender and its receiver is the only inter-node coordination needed to address the potential inconsistency on the PRK model parameters during transient periods. In particular, we do not need perfect information consistency that requires the same PRK model parameter of a link (S, R) to be used by link (S, R) and all the links whose transmitters are in the exclusion region around receiver R. That is, as long as a receiver is in the data channel when its sender transmits data packets, a node can use the new PRK model parameter of a link the moment the node learns of the parameter. The intuition of this design is that the earliest use of new PRK model parameters helps improve data delivery reliability when the corresponding exclusion regions expand, or it helps improve the channel spatial reuse and the concurrency of data transmissions when the corresponding exclusion regions shrink. As shown in Figure 90, for

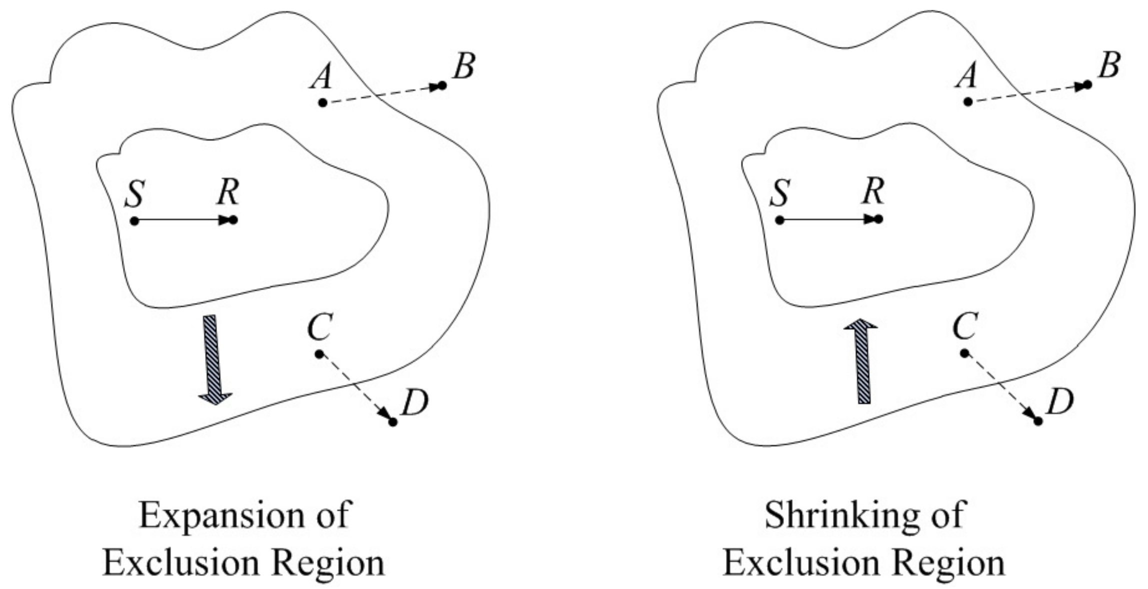


Figure 90: Coordination-free use of the latest PRK model parameters

instance, assuming that node A learns the latest PRK model parameter $K_{S,R,T_{S,R}}(.)$ of link (S, R) earlier than node C does when the exclusion region around R expands or shrinks, then it is desirable for A to use the latest value of $K_{S,R,T_{S,R}}(.)$ without waiting for C to learn it: this prevents A from transmitting concurrently with the transmission along (S, R) when the exclusion region expands, and this allows A to transmit concurrently with the transmission along (S, R) when the exclusion region shrinks.

Our discussions in this paper focus on ensuring data delivery reliability across links, thus we have focused on the exclusion regions around receivers alone. If it is important to ensure ACK reliability at the link layer (e.g., for avoiding unnecessary retransmissions), similar approaches to protecting data receptions can be applied to protect ACK receptions by maintaining an exclusion region around the transmitter of each link. For conciseness of presentation, however, we only focus on ensuring data delivery reliability in this paper.

3 Experimental evaluation

We have implemented PRKS in TinyOS [9]. In what follows, we experimentally evaluate PRKS through measurement in the NetEye [11] and Indriya [2] sensor network testbeds as we do in Chapter 2 .

3.1 Methodology

Protocols. To understand the design decisions of PRKS, we comparatively study PRKS with its following variants:

- *PRKS-R*: same as PRKS but formulates the PRK model instantiation problem as a dead-beat PID regulation control problem instead of as a minimum-variance regulation control problem.
- *PRKS-L*: same as PRKS but directly use the linear model (3.5) instead of its refined model (3.12).

Towards understanding the benefits of PRKS, we also comparatively study PRKS with the following distributed scheduling protocols:

- *CSMA*: a contention-based MAC protocol that uses the basic CSMA/CA mechanism to ameliorate the impact of co-channel interference; this represents the interference control mechanism used by protocols such as B-MAC [99];
- *RTS-CTS*: a contention-based MAC protocol that uses CSMA/CA and RTS-CTS to ameliorate the impact of co-channel interference and hidden terminals; this represents the interference control mechanism used by protocols such as S-MAC [127];

- *RIDB*: a TDMA scheduling protocol that uses a TDMA protocol similar to the one used in PRKS and that uses the physical interference model to derive interference relations between nodes but ignores cumulative interference in networks [139].
- *CMAC*: a contention-based MAC protocol where a node transmits at a time instant only if the SINR of this transmission and the SINRs of other concurrent transmissions overheard by the node are above a certain threshold (e.g., for ensuring a certain link reliability); this represents the interference control mechanism used by protocols such as C-MAC [105];
- *SCREAM*: a TDMA scheduling protocol using the SCREAM primitive [19] to schedule concurrent transmissions according to the physical interference model; this represents the interference control mechanism used by protocols such as FDD [19] and DSS [102];

Among these protocols, CSMA and RTS-CTS represent the protocol-model-based techniques in existing industry standards such as IEEE 802.15.4 and 802.11p; RIDB, CMAC, and SCREAM represent the techniques used in existing physical-model-based scheduling. Focusing on predictable co-channel interference control, we do not compare PRKS with protocols such as WirelessHART [122] that do not consider channel spatial reuse.

Network and environmental settings. We use a subset of the 130 TelosB motes in NetEye. The subset of motes forms a random network, and it is generated by using each mote of NetEye with probability 0.8; each mote uses a data transmission power of -25dBm (i.e., power level 3 in TinyOS). On average, a node can reach 10 nodes with a data delivery reliability of at least 95% in the absence of interference.

Focusing on interference-oriented scheduling of single-hop transmissions at the link layer, we consider single-hop data transmissions in this study.[11] We choose data transmission links such that each mote transmits data packets to a receiver to whom the average SNR is the closest to 15dB in the absence of interference. Unless mentioned otherwise, every node transmits a data packet to

[11]Predictable reliability in single-hop transmissions is important by itself for new networked control paradigms that involve only communications between close-by nodes [95], while predictably reliable single-hop transmission is also a basis for reliable multi-hop transmission in general.

its receiver every 20ms. For reflecting different application scenarios, we consider the cases when the required mean data delivery reliability (PDR) is 70%, 80%, 90%, or 95% for all the links and the "mixed PDR requirement" case when the required mean reliability for each link is randomly chosen as 70%, 80%, 90%, or 95% with equal probability.

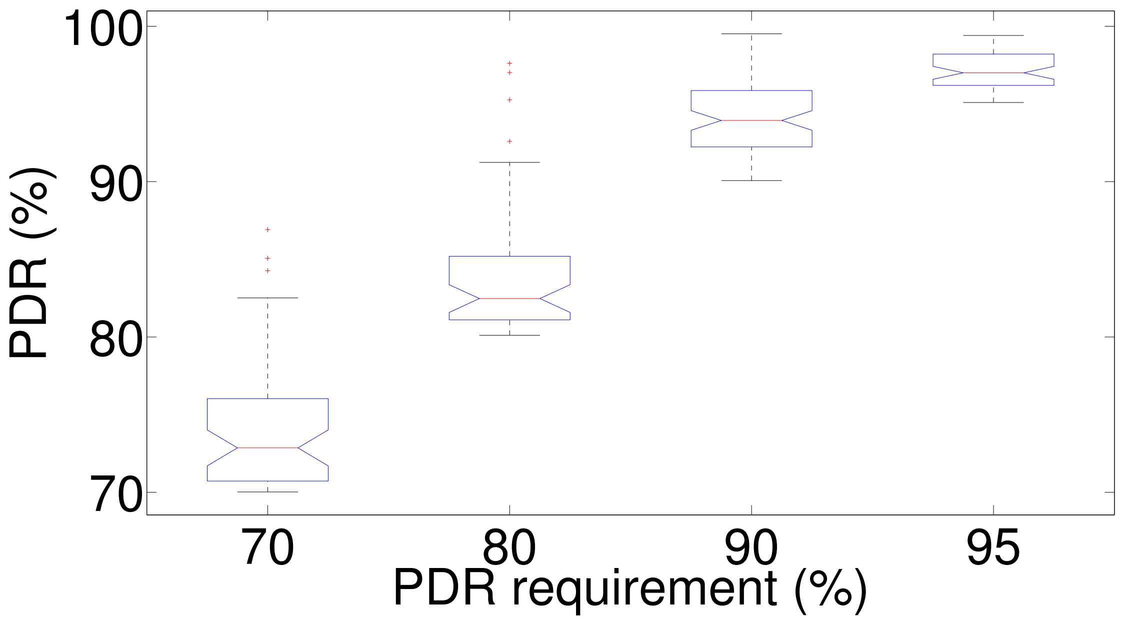


Figure 91: Packet delivery reliability (PDR) in PRKS

3.2 Measurement results

Behavior of PRKS. For different PDR requirements, Figures 91 and 92 show the boxplots of link packet delivery reliability (PDR) and PRK model parameter in PRKS respectively. We see that PRKS adapts the PRK model parameter according to different PDR requirements, and that the required minimum mean PDR is always guaranteed in PRKS through predictable interference control.[12] In particular, the PRK model parameter increases with the PDR requirement so that more close-by nodes are prevented from transmitting concurrently with a link's transmission. Given that

[12]Due to the discrete nature of the spatial distribution of concurrent transmitters, the actual PDR tends to be slightly higher than (instead of being strictly equal to) the required mean PDR.

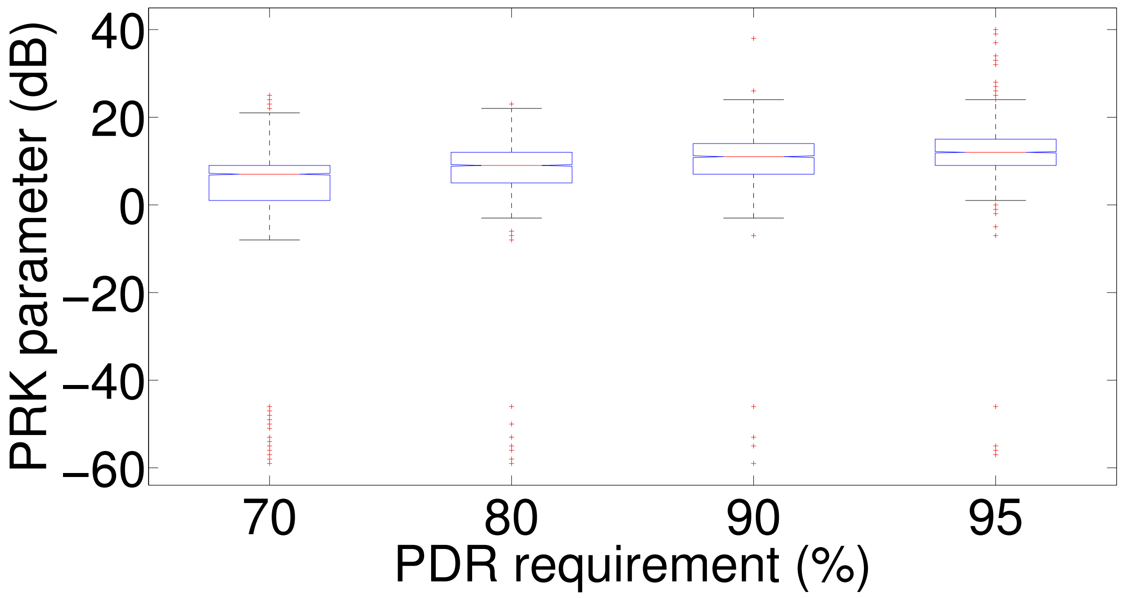


Figure 92: PRK model parameter in PRKS

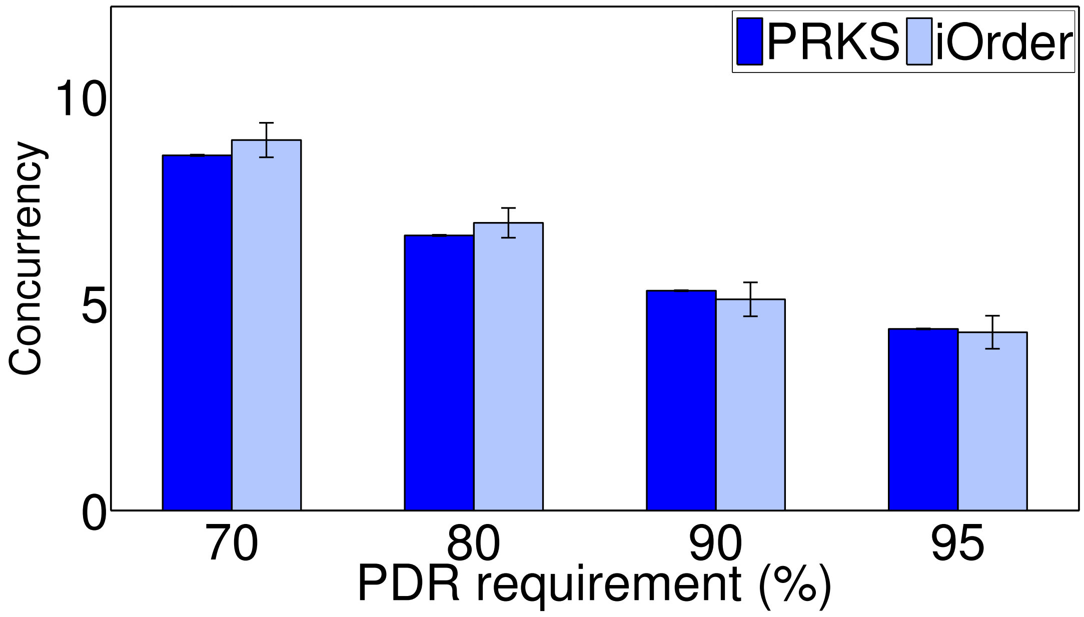


Figure 93: Mean concurrency in PRKS and iOrder

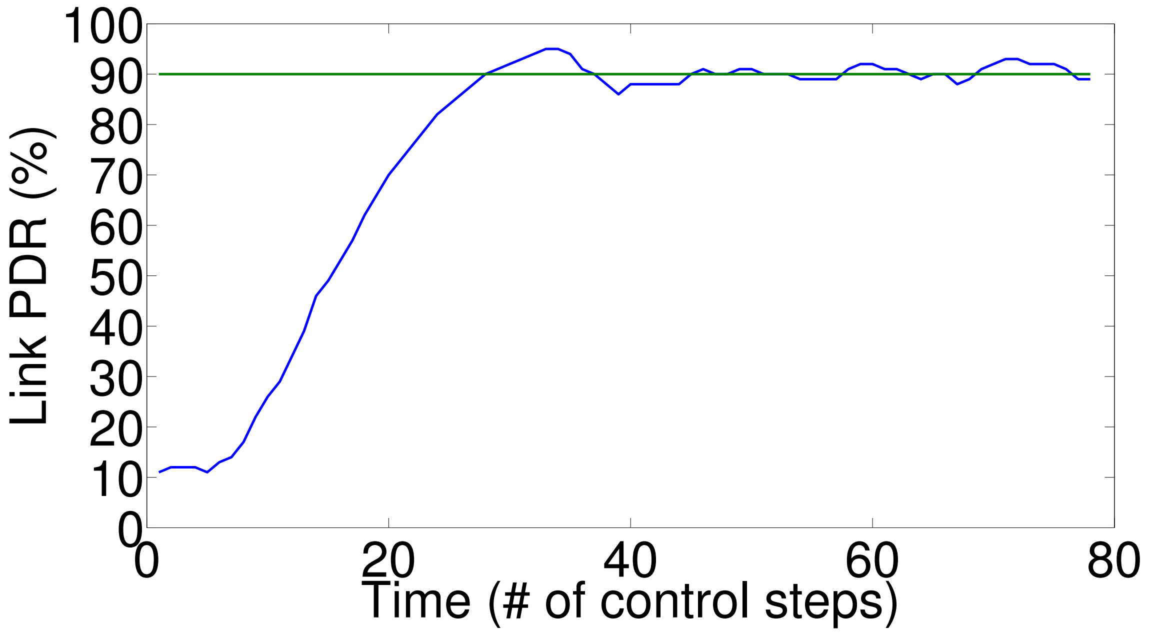


Figure 94: Temporal link PDR: PDR req. = 90%

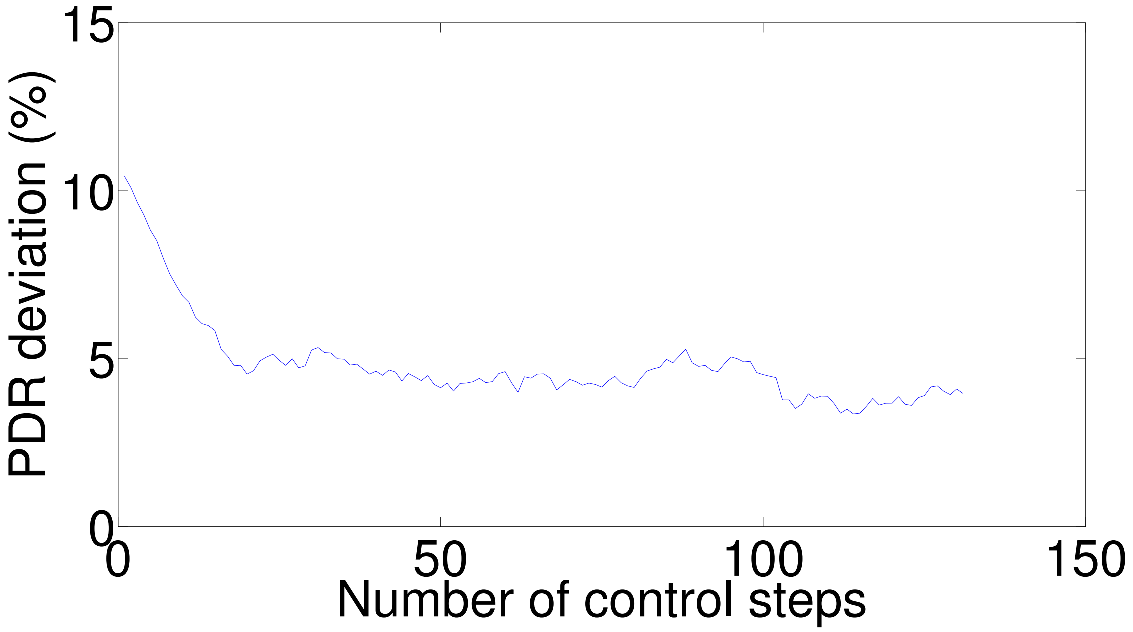


Figure 95: Network-wide convergence in PRKS

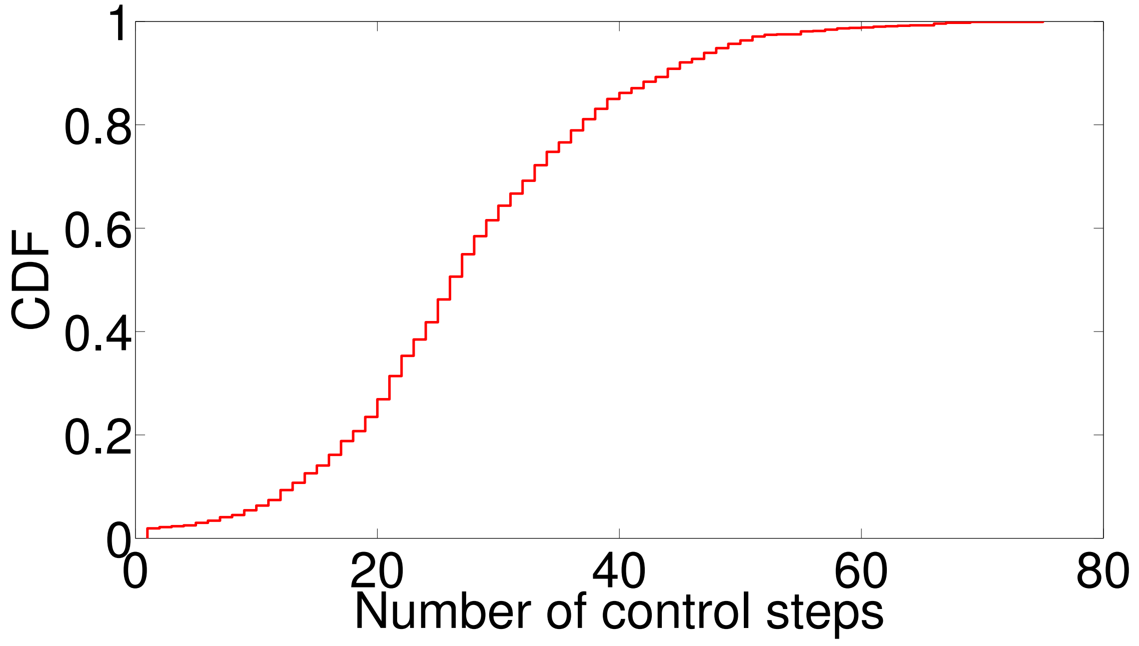


Figure 96: CDF for the settling time of link PDR

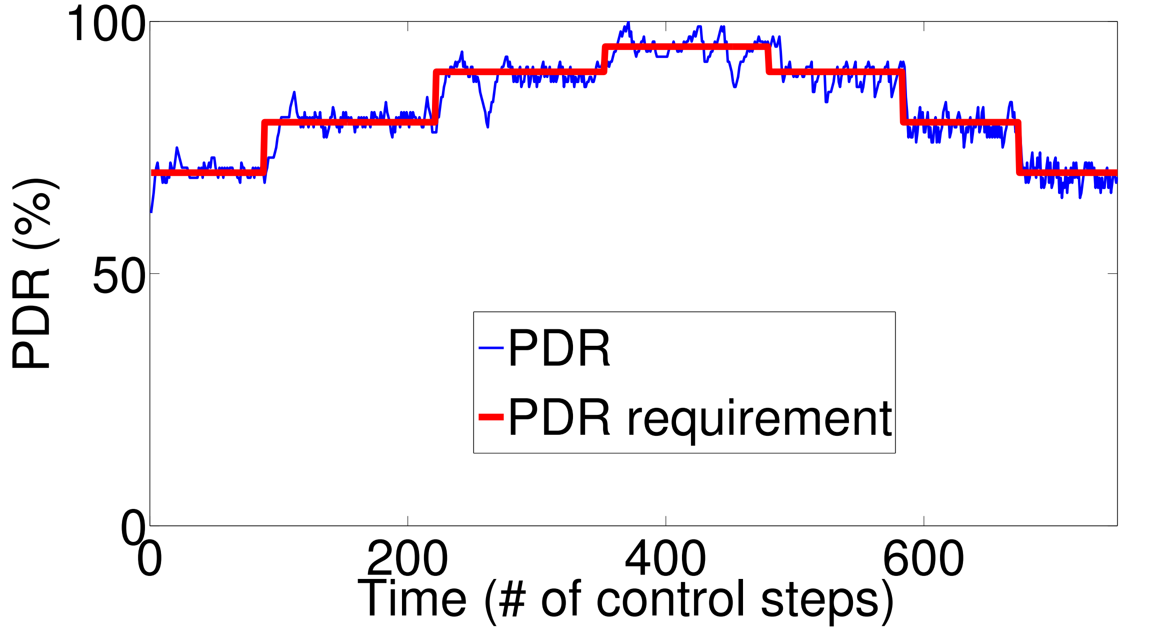


Figure 97: PDR in PRKS: temporally varying PDR requirements

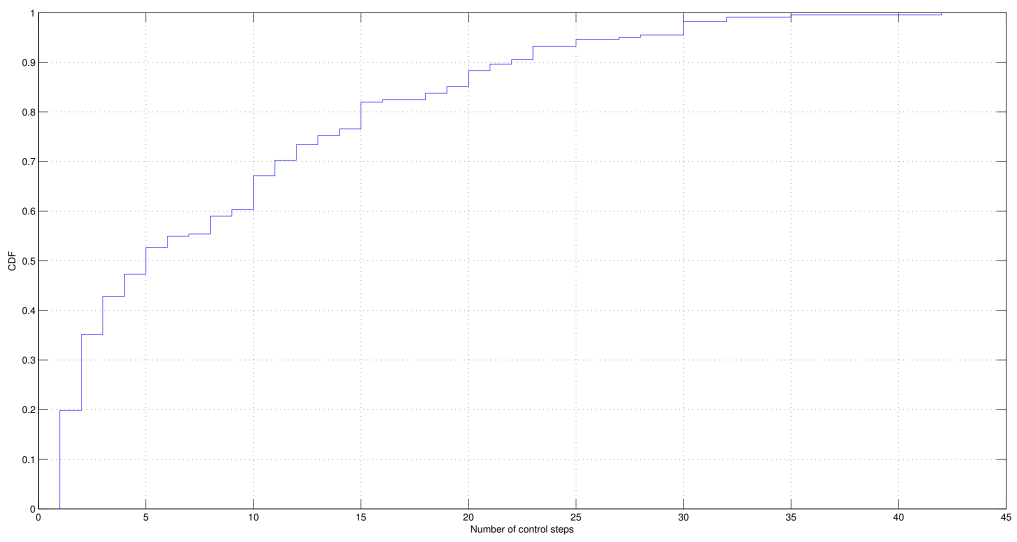


Figure 98: Settling time: temporally varying PDR requirements

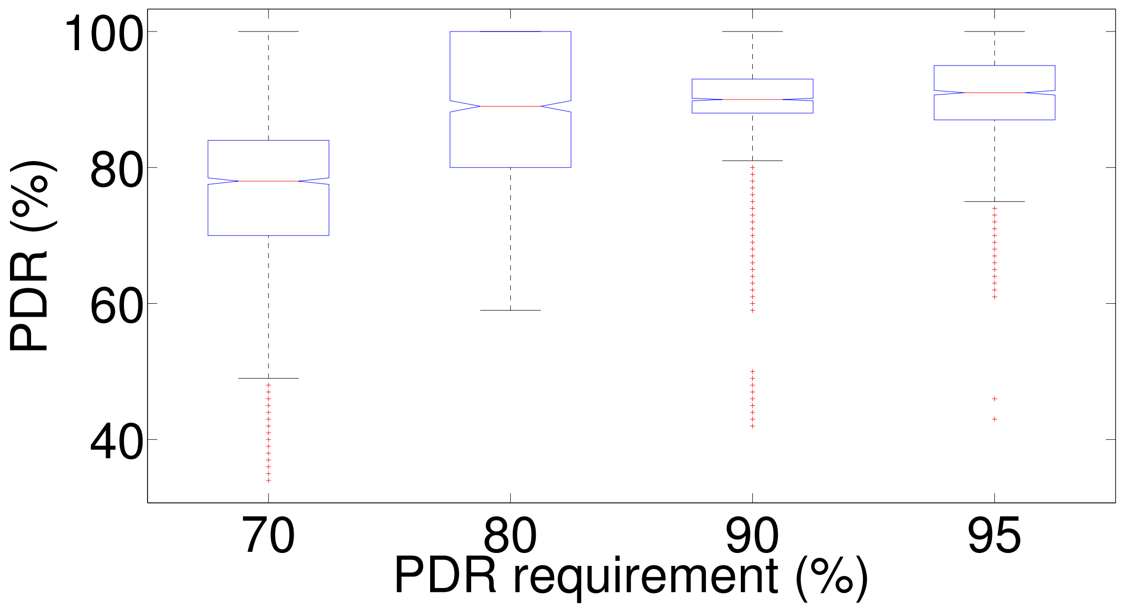


Figure 99: Packet delivery reliability in PRKS: mixed PDR requirement

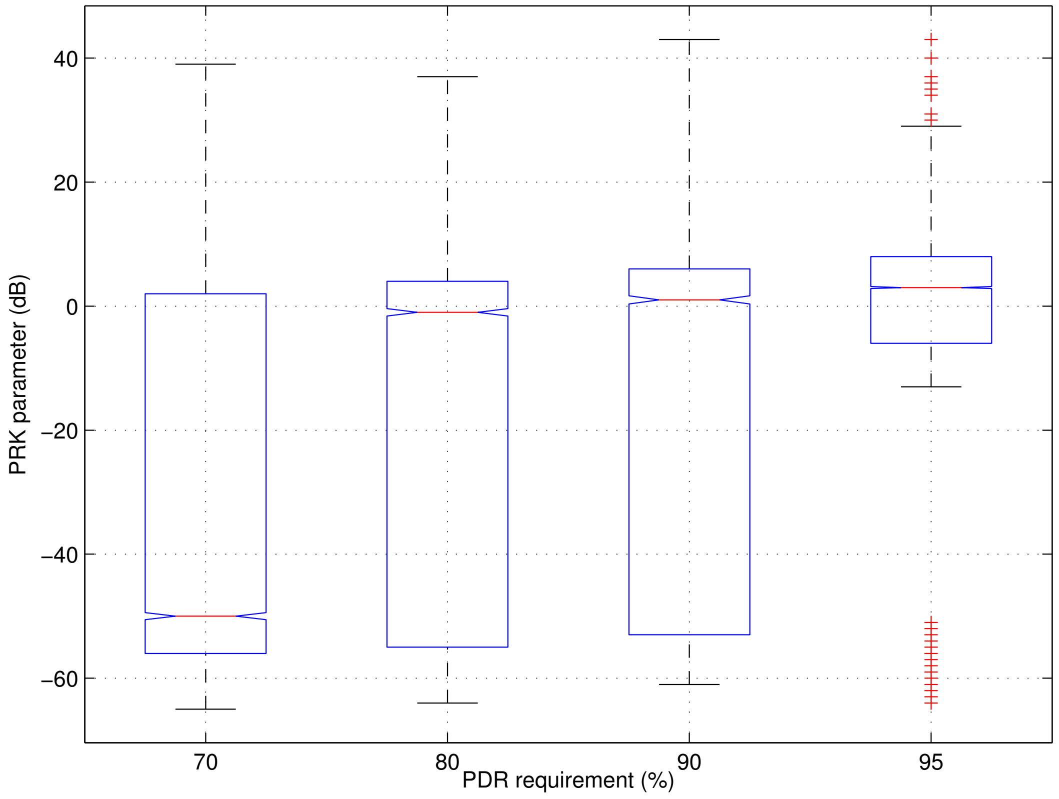


Figure 100: PRK model parameter in PRKS: mixed PDR requirement

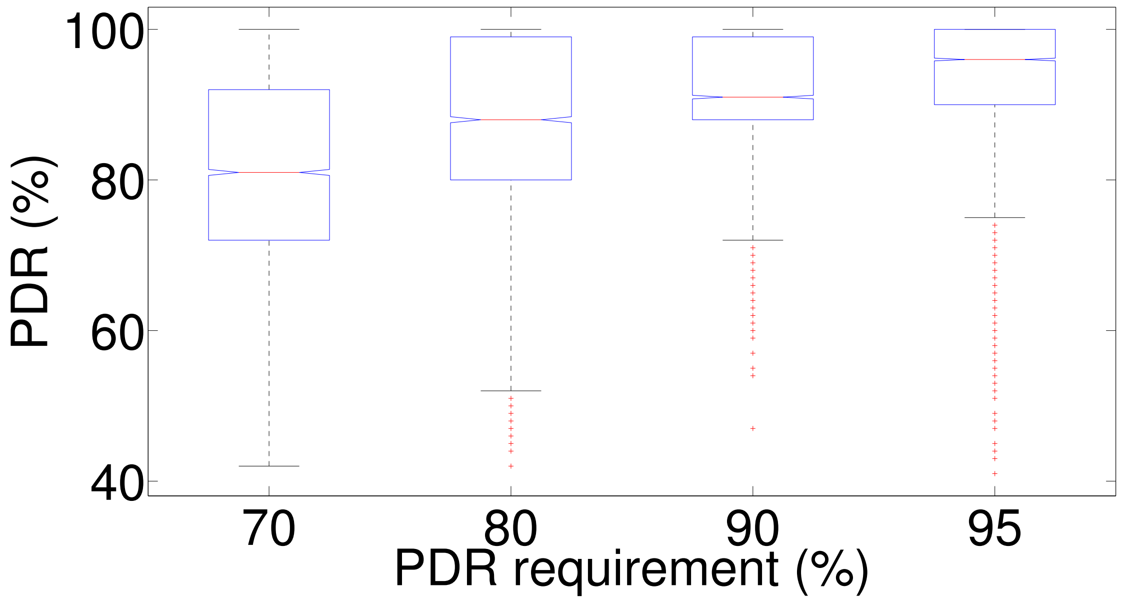


Figure 101: Packet delivery reliability in PRKS: different transmission powers

the PRK model parameter K reflects the ratio of data signal power to the interference power from a weakest interferer (see Inequality (3.3)), Figure 92 shows that the required control signaling power is usually no more than 22dB over the data transmission power, thus the power control scheme of control signaling can be implemented even with today's commodity radios such as Chipcon CC2420.

To understand the spatial reuse in PRKS, Figure 93 shows the mean concurrency (i.e., number of concurrent transmissions at a time instant) and its 95% confidence interval[13] in PRKS as well as in a state-of-the-art, centralized scheduling protocol *iOrder* [26] which maximizes channel spatial reuse in interference-oriented scheduling.[14] We see that, despite its nature of local and distributed control, PRKS enables a concurrency and spatial reuse statistically equal or close to what is enabled by the centralized algorithm iOrder while ensuring the required PDR at the same time.

[13]For the figures of this section that present performance statistics (e.g., mean concurrency or PDR), we also show the 95% confidence intervals of the statistics, but some of the confidence intervals may be too narrow to be noticeable in the figures.

[14]In terms of maximizing spatial reuse, iOrder has been shown to outperform well-known existing scheduling protocols such as Longest-Queue-First [71], GreedyPhysical [18], and LengthDiversity [50].

Despite the distributed nature of the minimum-variance regulation controller in PRKS, the individual controllers converge to a state where the required PDR is satisfied. For a typical link in the network, for instance, Figure 94 shows the temporal behavior of link PDR when the minimum application PDR requirement is 90%. We see that the link PDR converges to its steady state after around 25 control steps. As a way of reflecting the network-wide convergence, Figure 95 shows the temporal, convergence behavior of the network-wide average deviation of link reliability from the required mean PDR (i.e., $\frac{\sum_{\text{Every link } (S,\, R)} |Y_{S,R}(t) - T_{S,R}|}{\text{Total number of links}}$). In general, link PDRs converge quickly, as shown by Figure 96 where the settling time is defined as the number of control steps taken for a link to reach its steady state PDR distribution. In addition to convergence to a state where the required PDRs are satisfied, the collective behavior of the distributed controllers in PRKS also enables a spatial reuse close to what is feasible with the state-of-the-art, centralized scheduler iOrder as we have shown in Figure 93.

To understand the adaption of PRKS to online dynamics, we also run experiments where the mean PDR requirements change over time. Figure 98 shows, for a typical link in the network, the time series of link PDR when the application PDR requirements are set to 70%, 80%, 90%, 95%, 90%, 80%, and 70% over time. We see that PRKS adapts to application-specific PDR requirements in a predictable manner.

For the "mixed PDR requirement" scenario where different links of the same network have different PDR requirements, Figures 99 and 100 show the boxplots of link PDR and PRK model parameter for the links grouped by their PDR requirements. We see that PRKS adaptively ensures the required PDR in a predictable manner even when different links of the same network have different PDR requirements. We have also studied the scenario where different links use different data transmission powers to ensure a desired SNR of 6dB at receivers in the absence of interference, and have observed similar behavior. For instance, Figure 101 shows that the desired PDRs are ensured when different links use different transmission powers.

Variants of PRKS. To corroborate the design decisions of PRKS, Figures 102, 103, and 104 show the mean PDR, mean concurrency, and mean standard deviation of PDR as well as

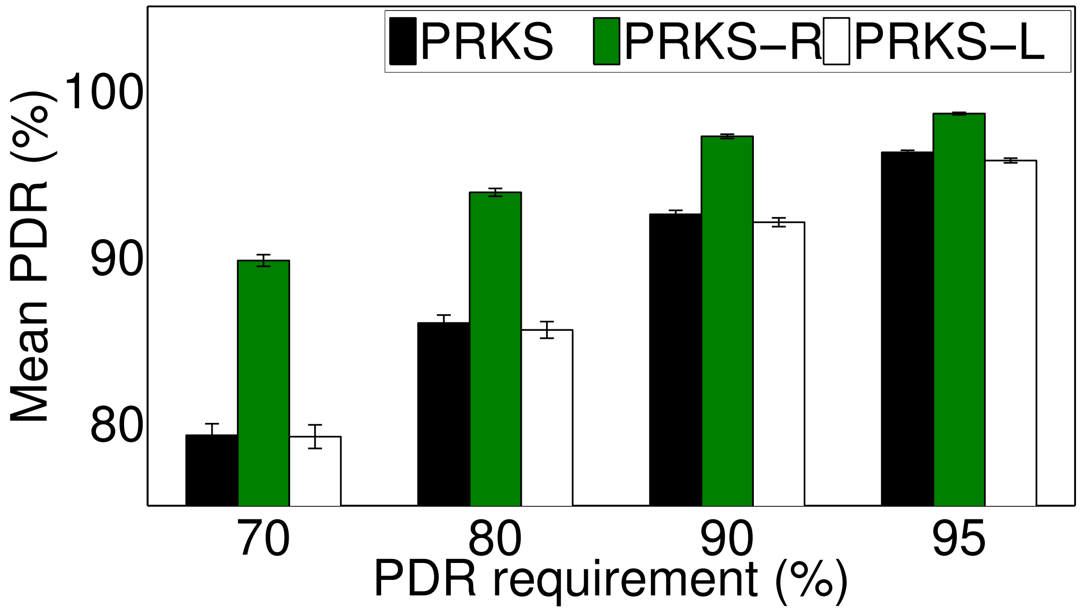


Figure 102: Mean PDR in variants of PRKS

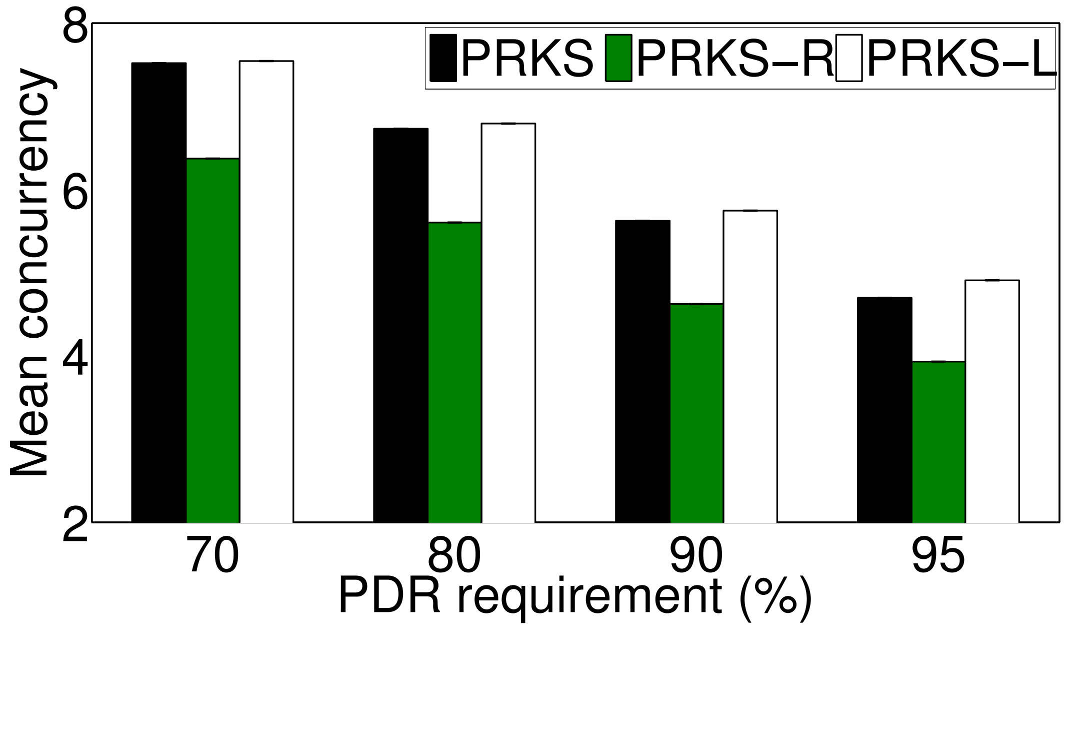


Figure 103: Mean concurrency in variants of PRKS

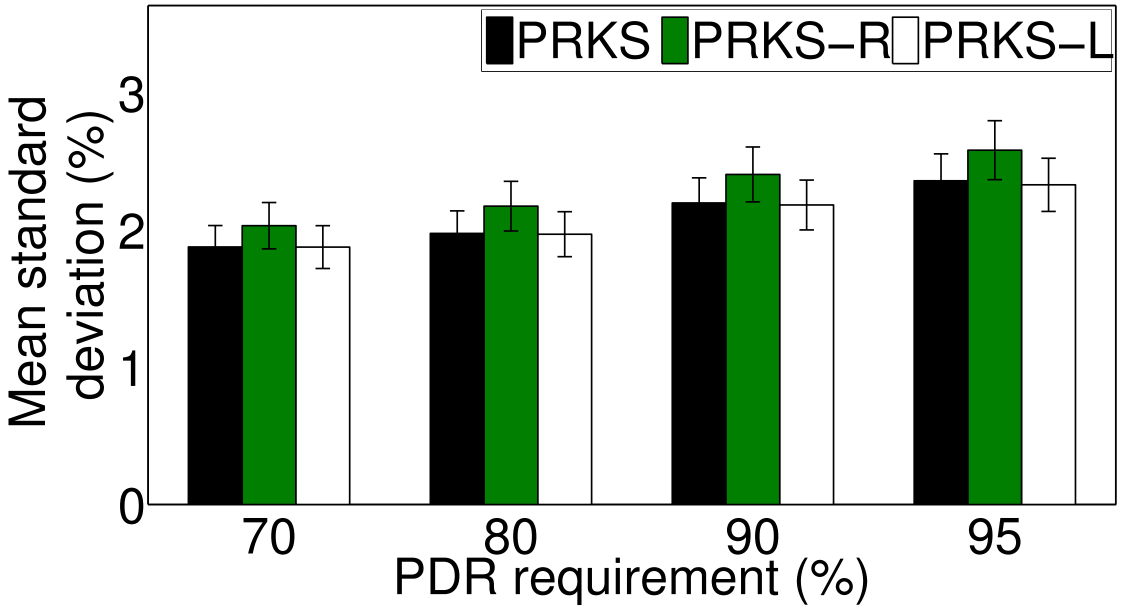


Figure 104: Standard deviation of PDR in variants of PRKS

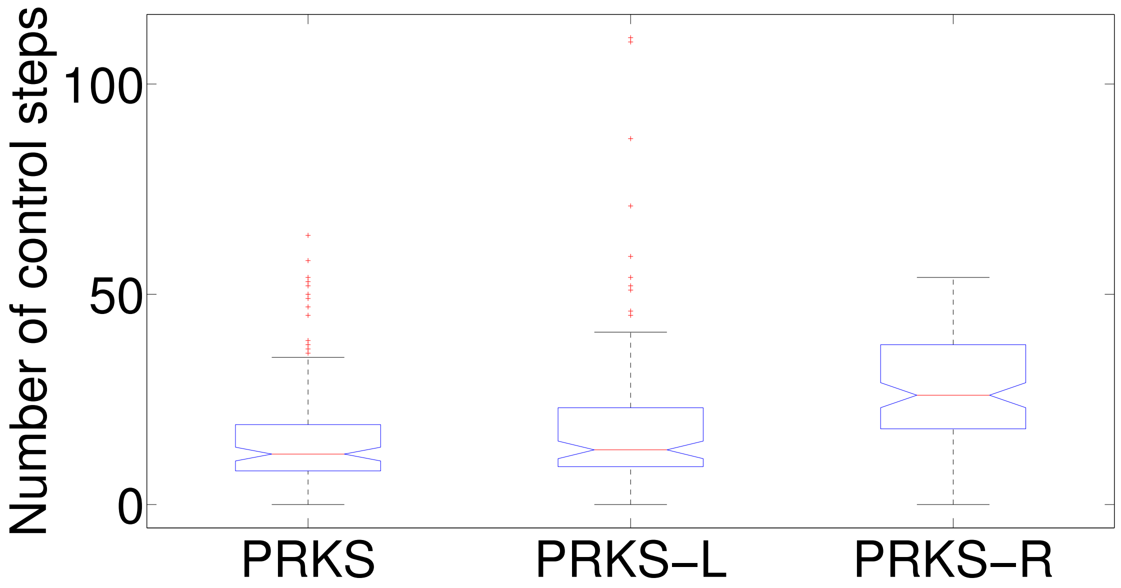


Figure 105: Settling time of link PDR in variants of PRKS

their 95% confidence intervals in different variants of PRKS respectively. We see that, by using minimum-variance regulation control instead of the traditional PID regulation control, PRKS and PRKS-L outperform PRKS-R by enabling higher concurrency and reducing PDR variation while satisfying the PDR requirement. Without using minimum-variance control, there also exist significant undershoot and overshoot during convergence in PRKS-R, thus the settling time in PRKS-R is also significantly longer than that in PRKS and PRKS-L as can be seen from Figure 105. Compared with PRKS, PRKS-L uses the minimum-variance control but does not use the refined linear plant model (3.12), thus there exist more undershoot and overshoot during convergence. Accordingly, the settling time in PRKS-L is slightly longer than that in PRKS as shown in Figure 105.

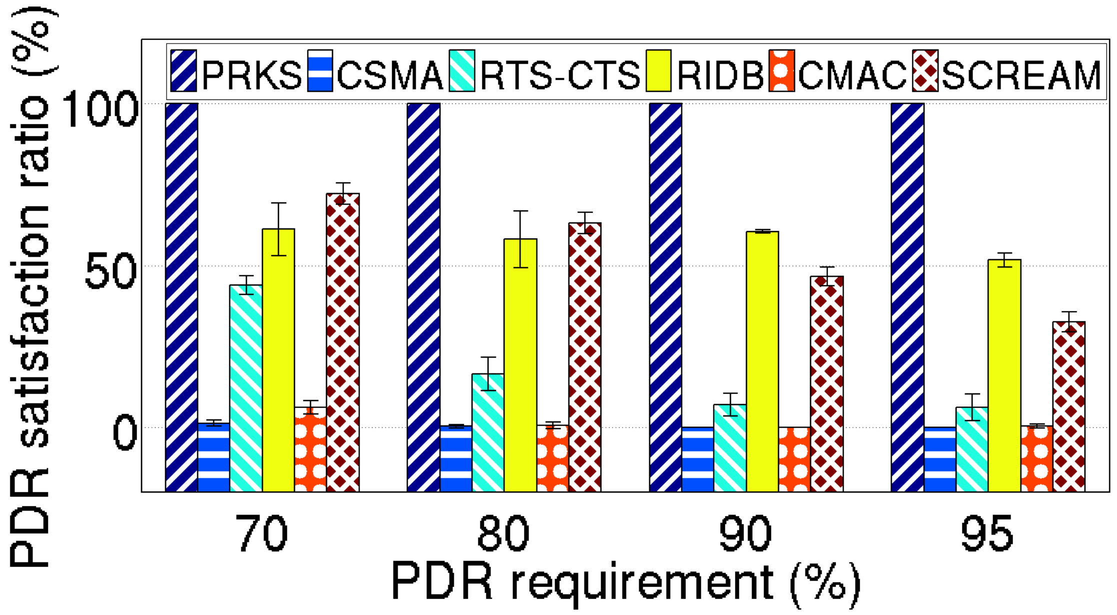


Figure 106: PDR requirement satisfaction ratios in different protocols

Comparison with existing protocols. Figure 106 shows the ratio of links whose PDRs are no less than the application required PDRs in PRKS and other existing protocols. We see that, unlike PRKS that always ensures application required PDRs in a predictable manner, existing protocols do not ensure the required PDRs due to co-channel interference that is not well

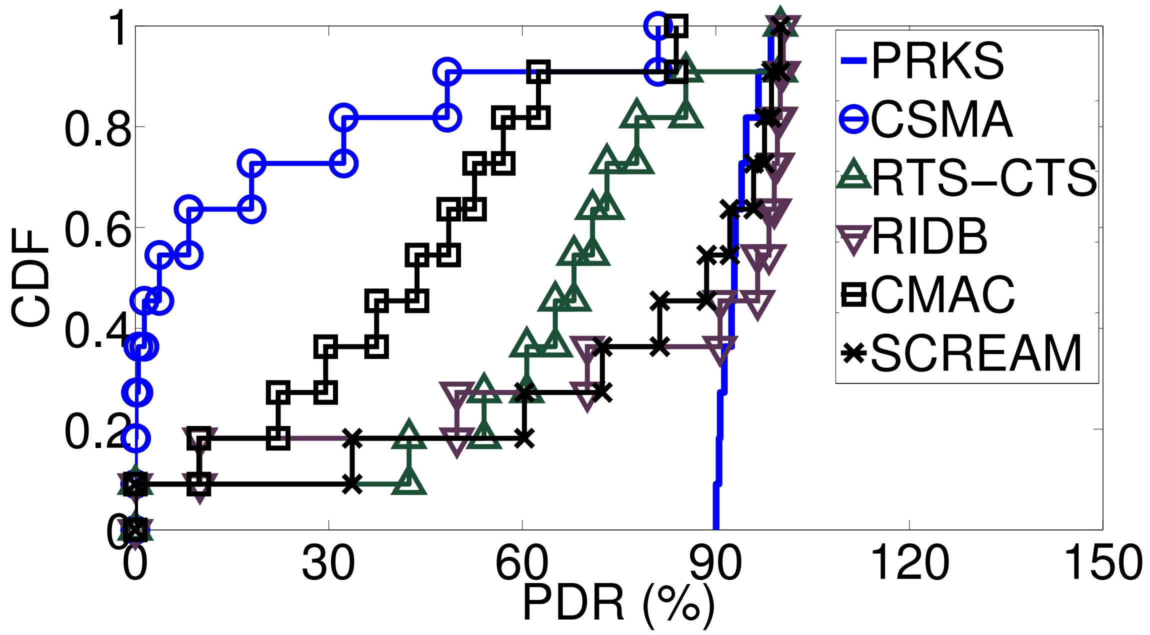


Figure 107: CDF of link PDRs in different protocols: PDR req. = 90%

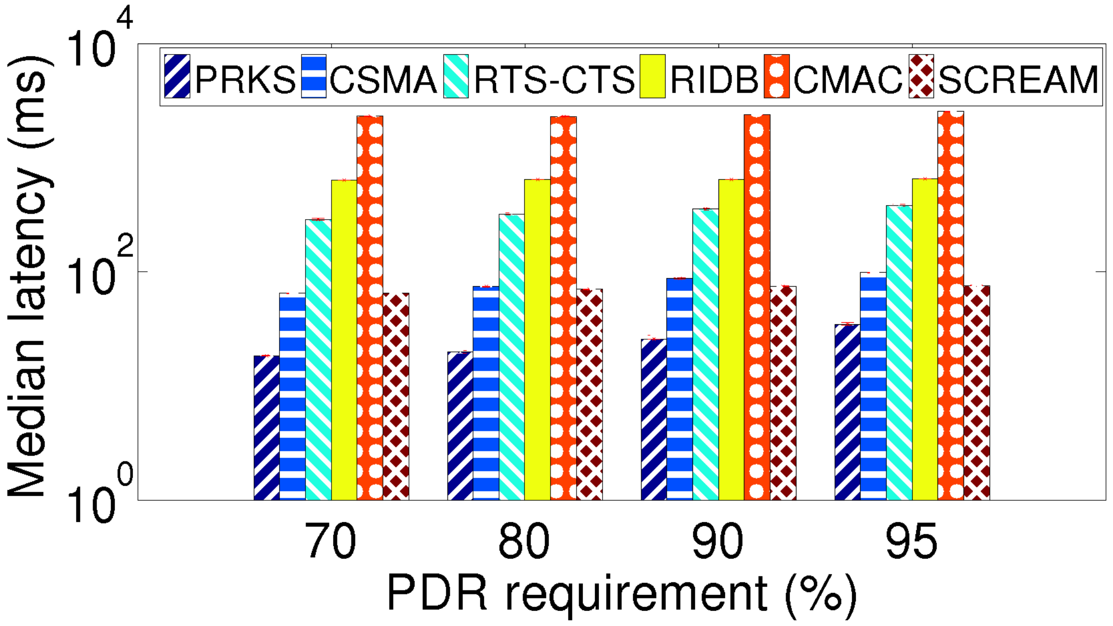


Figure 108: Median delay: reliability via packet retransmission

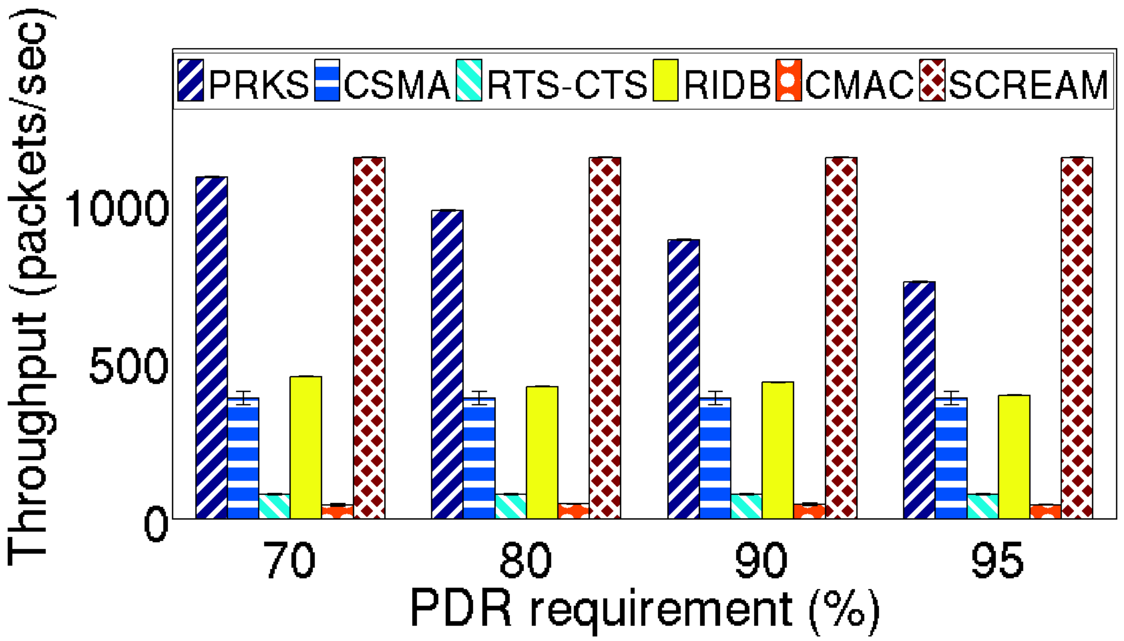


Figure 109: Mean network spatial throughput

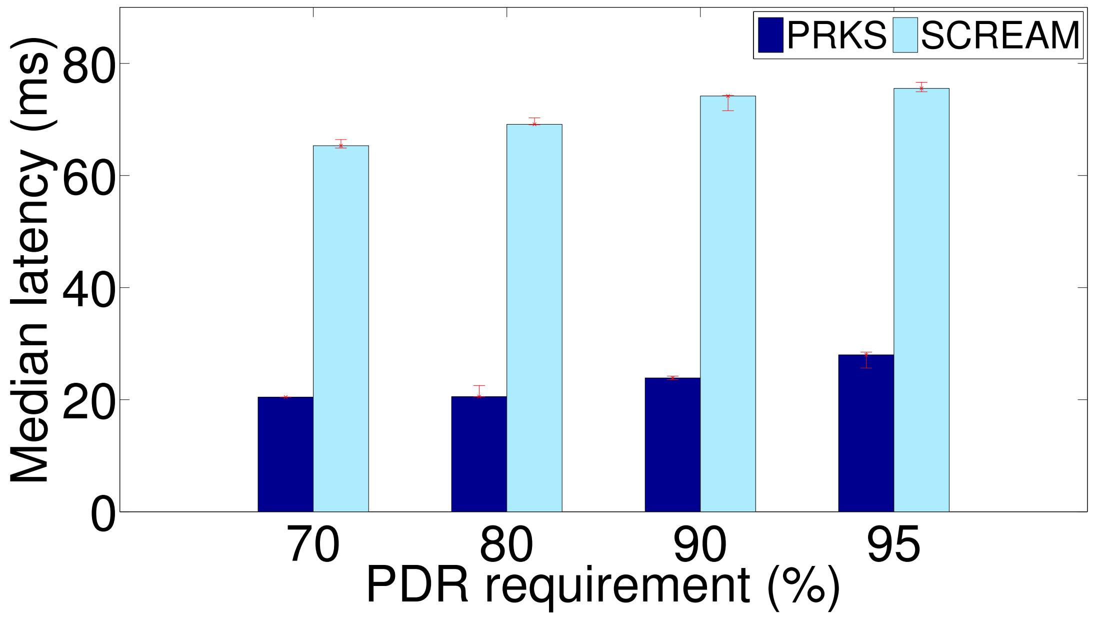


Figure 110: Median delay: PRKS with SCREAM's link PDRs

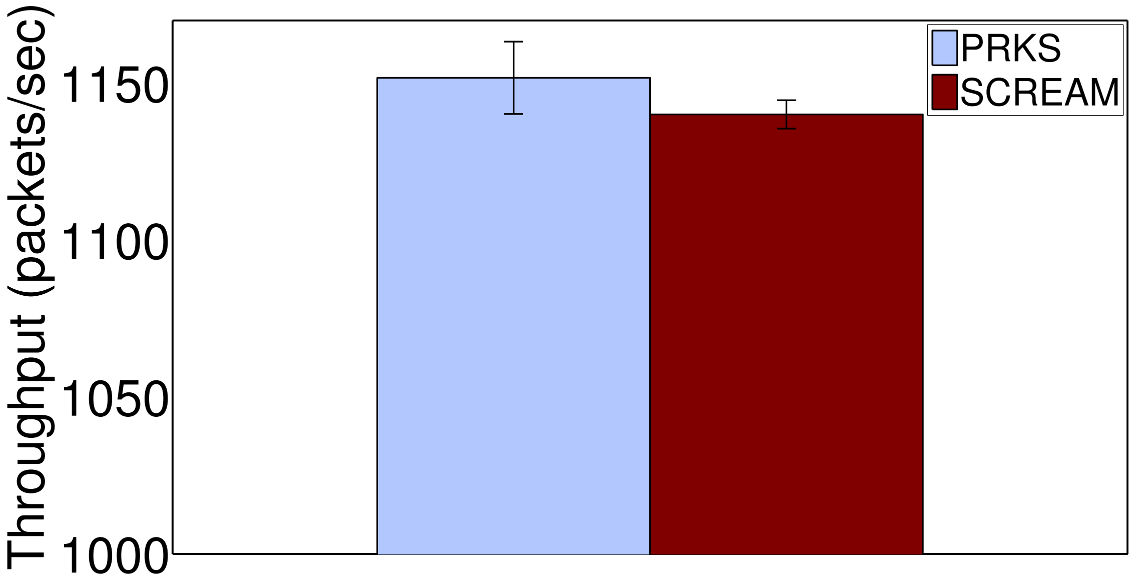


Figure 111: Mean network throughput: PRKS with SCREAM's link PDRs

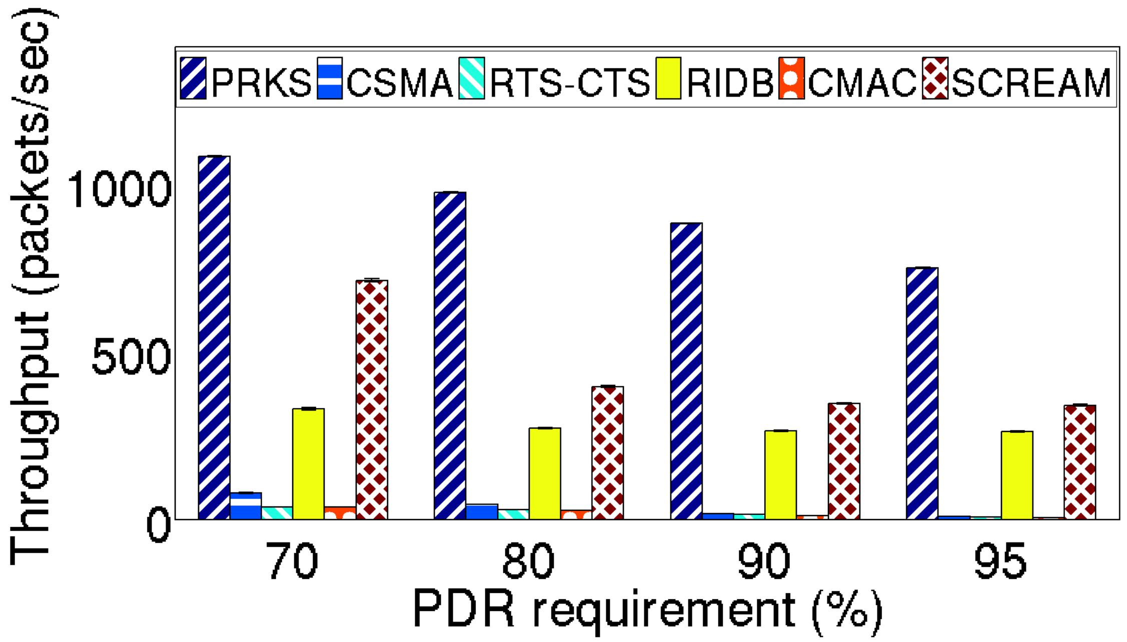


Figure 112: Mean network throughput: reliability via traffic load reduction

controlled. We also see that the PDR satisfaction ratios in the existing protocols tend to decrease with increased PDR requirements, thus the existing protocols cannot control link reliability in a predictable manner.

Among the existing protocols, RIDB enables higher PDR satisfaction ratios than RTS-CTS and CSMA do because RIDB considers the physical interference model and application PDR requirements in defining pairwise interference relations between nodes; nonetheless, due to its lack of consideration of cumulative interference from multiple concurrent interferers, RIDB does not ensure predictable interference control and thus does not ensure predictable link reliability. When the application PDR requirement is 95%, for instance, RIDB can only enable a PDR satisfaction ratio 50.72%. RTS-CTS ensures higher PDR satisfaction ratio than CSMA does due to its use of RTS-CTS handshake, but the PDR satisfaction ratios are quite low in both protocols (e.g., as low as 8.5% and 0% in RTS-CTS and CSMA respectively) since neither protocols are based on high-fidelity interference models.

Among the existing protocols that explicitly use the physical interference model, CMAC and SCREAM consider cumulative interference. Nonetheless, the PDR satisfaction ratio is quite low in CMAC, and the PDR satisfaction ratio in SCREAM can also be lower than that in RIDB which does not even consider cumulative interference. CMAC cannot ensure the required PDRs since CMAC cannot ensure predictable interference control when the interference range is greater than the communication range, which is usually the case in practice (especially when the required PDR is high). Since CMAC does not decouple control signaling from data transmissions as in PRKS, interference control in CMAC is also negatively affected by any unreliability in the per-transmission-based control signaling (e.g., observing neighboring nodes' SINRs). In SCREAM, the collision among a set of concurrent transmitters is detected through network-wide coordination. The detection is based on a sample of the status (i.e., success or failure) of concurrent data transmissions and cannot ensure accurate collision detection, thus SCREAM cannot accurately control interference to ensure predictable PDR. Additionally, it takes $\Theta(nk)$ time slots for the network to find the schedule of a single time slot in SCREAM, where n is the number of links in a network and k is the interference diameter of the network (which is approximately the ratio of the geometric diameter of a network to the carrier sensing range) [19]. Thus SCREAM is not suitable for dynamic, large-scale networks where schedules need to adapt to changing network and environmental conditions frequently.

Incapable of ensuring predictable link reliability in scheduling, existing protocols can try to improve link reliability by packet retransmission. Nonetheless, packet retransmission increases data delivery delay; this can be seen from Figure 108 which shows the median packet delivery delay when packets are retransmitted to ensure a certain required PDR. The network capacity analysis by Zhang et al. [133] has shown that the inherent tradeoff between PDR, delay, and throughput is such that higher PDR and lower delay usually comes at the cost of reduced network spatial throughput (i.e., number of packets successfully delivered to its one-hop receivers per second). The higher PDR and high spatial reuse in PRKS, however, makes PRKS still achieve higher throughput than existing protocols except for SCREAM, as shown by Figure 109. The inherent PDR-throughput tradeoff and the higher PDR in PRKS make the throughput in PRKS lower than that in SCREAM; if we set the PDR requirements in PRKS according to the actual link PDRs in SCREAM, however, PRKS enables lower delay and higher throughput, as shown by Figures 110 and 111 respectively. This shows that PRKS enables controlling a network's operating point and thus the optimization of the PDR-delay-throughput tradeoff, which is important for the joint-design of networking and control systems in real-time wireless networked sensing and control [89].

Existing protocols can also try to improve link reliability by reducing the application traffic load such that interference becomes negligible. Nonetheless, reducing traffic load decreases network throughput; this can be seen from Figure 112 which shows the mean network spatial throughput when packet arrival rates are limited from above to ensure a certain required PDR.

Indriya testbed.

We have observed similar behavior in the Indriya testbed as those in NetEye. For the Indriya testbed, Figures 113 and 114 show the packet delivery reliability (PDR) and PRK model parameter in PRKS respectively. We see that PRKS ensures required PDRs in an adaptive manner by adjusting the PRK model parameter according to application requirements on PDRs. Figure 115

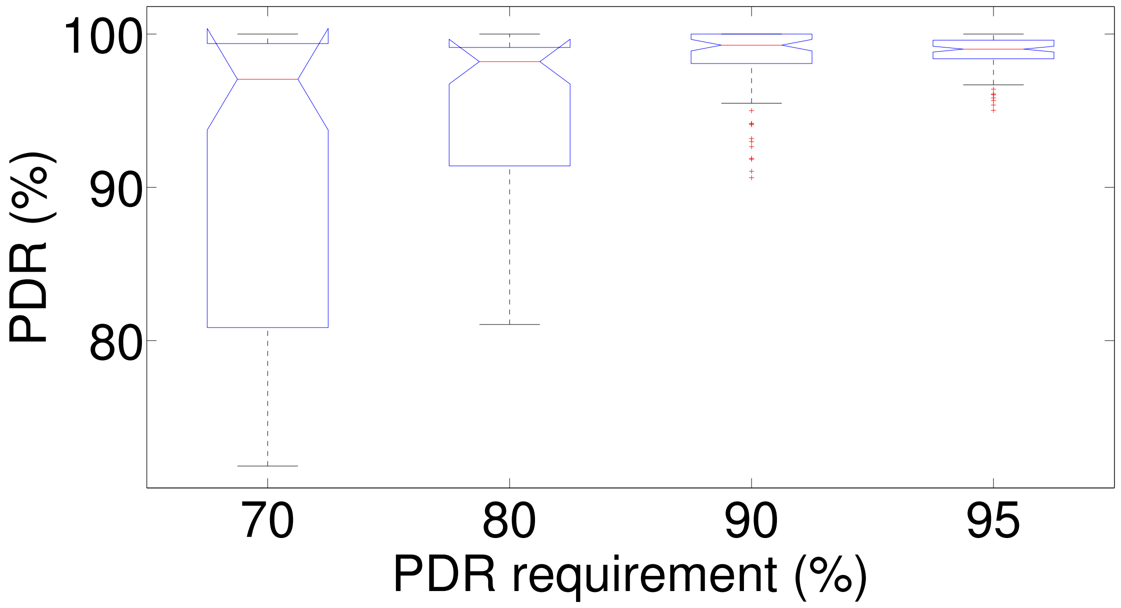


Figure 113: Packet delivery reliability in PRKS: in Indriya

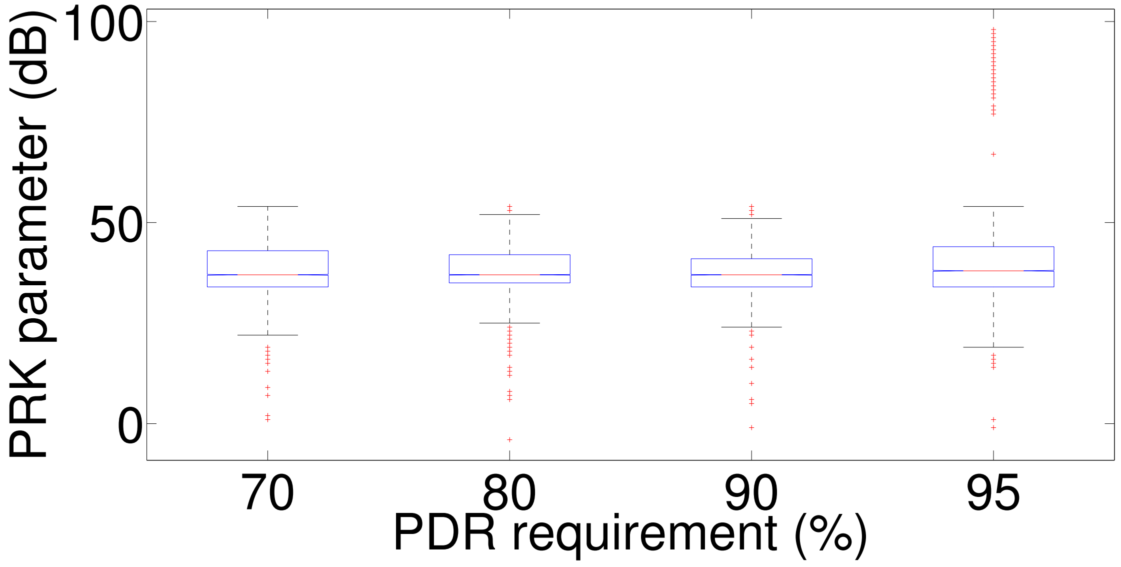


Figure 114: PRK model parameter in PRKS: in Indriya

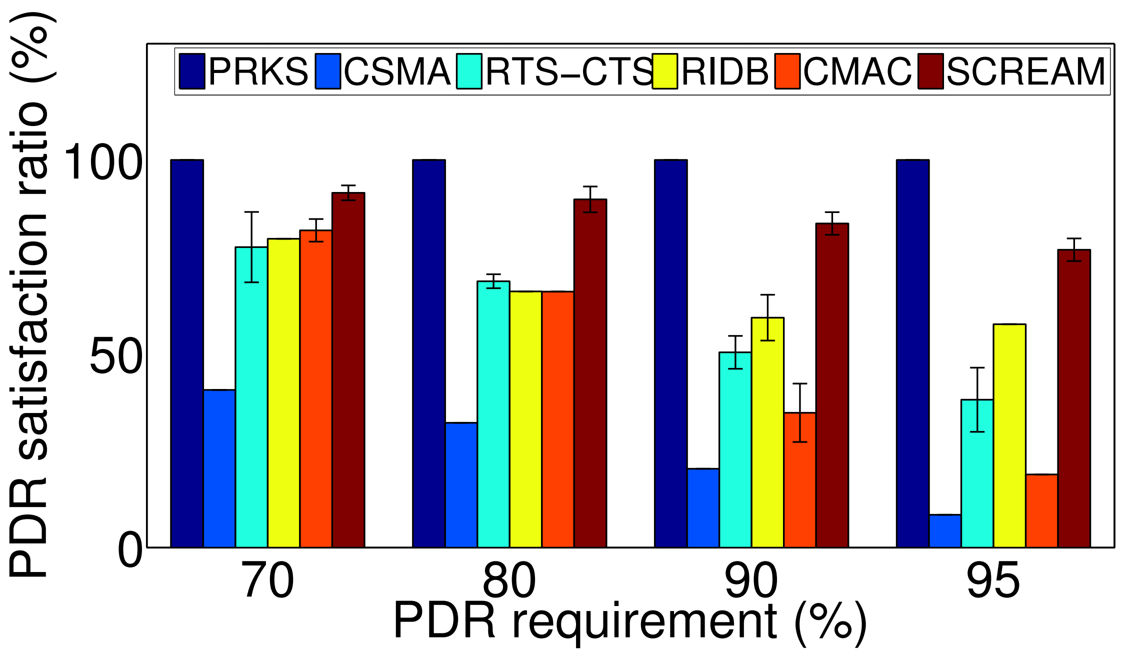


Figure 115: PDR requirement satisfaction ratios in different protocols: in Indriya

shows the PDR requirement satisfaction ratios in different protocols. We see that, unlike PRKS, other protocols cannot ensure application required PDR in a predictable manner.

4 Discussion

Mobility. As a first step in developing field-deployable solutions to distributed, predictable interference control, we have focused on mostly-static WSC networks such as those in smart power grid and industrial automation. In vehicular WSC networks such as those for inter-vehicle active safety control, however, nodes are mostly-mobile as vehicles move. Vehicle mobility introduces dynamics in vehicle spatial distribution and thus dynamics in wireless communication. In particular, dynamics in vehicle spatial distribution increases the dynamics in signal power attenuation between nodes, which challenges the maintenance of local signal maps and the adaptation of the PRK model. While detailed study of PRK-based scheduling in vehicular WSC networks is our future work, we observe that the following facts may well help address the challenges of vehicle mobility: 1) The timescale of non-negligible vehicle movement is in seconds, while the timescale of wireless communication is in milliseconds or microseconds; the significantly longer timescale of the physical movement of vehicles enables vehicles to exchange control signals for PRK-based scheduling adaptation on the fly; 2) There are well-established, microscopic mobility models for vehicle movement [33]; these models can help estimate dynamics in vehicle spatial distribution and thus dynamics in signal power attenuation between vehicles; therefore, these models can help enable predictive adaptation of the signal map and the PRK model. We will explore the above opportunities in our future work.

5 Related work

Similar to PRKS, existing physical-model-based scheduling algorithms also try to control concurrent transmissions so that link reliabilities or receiver-side SINRs are above a certain threshold. As we have discussed in Sections 1 and 3, however, due to the non-local, combinatorial nature of the physical interference model, distributed physical-model-based scheduling algorithms have various drawbacks such as requiring network-wide coordination and employing strong systems assumptions which make it difficult to deploy these algorithms in real-world settings. In addition, many of these algorithms do not address the challenge of designing scheduling protocols when interfering links are beyond the communication range of one another [71, 102].

The concepts of guard-zone or exclusion-region around receivers have also been exercised in distributed scheduling algorithms [22, 53], but these algorithms assumed uniform traffic load or uniform wireless signal power attenuation across the whole network, which are unrealistic in general. They did not address the challenge of designing scheduling protocols when interfering links are beyond the communication range of one another either.

Adaptive physical carrier sensing has been proposed to enhance network throughput [64, 85], but cumulative interference is not considered. We have also observed in [133] that throughput-optimal scheduling usually leads to low link reliability, which is not desirable in wireless sensing and control (WSC) networks. Park et al. [96] considered link reliability when adapting carrier sensing range, but their solution did not guarantee link reliability due to the price function involved. Fu et al. [47] proposed to control carrier sensing range to ensure a certain SINR at receivers, but the derivation of safe-carrier-sensing-range was based on the unrealistic assumption of homogeneous signal power attenuation across the whole network.

Given that mission-critical WSC applications (e.g., those in transportation [3] and medicine [4]) may well use licensed spectrum, we expect co-channel interference to be a major source of interference in those settings and thus PRKS to be a basic element of those systems. For application domains without licensed spectrum, channel hopping has been leveraged to address external interference [98]; in these setting, PRKS can be integrated with channel hopping to address external interference, for instance, by dynamically changing the wireless channels used for transmitting the signaling packets and data packets in PRKS, but detailed study of this is beyond the scope of this paper. Glossy [45] leverages non-destructive interference between synchronized concurrent transmissions of the same packet to enable efficient flooding of packets across a network, upon which LWB [44] develops algorithms for scheduling many-to-one, one-to-many, and many-to-many communications. Glossy and LWB does not allow for spatial reuse of wireless channels to transmit different packets at the same, thus they can lead to capacity loss in large scale networks. The network-wide flooding of the same packet in Glossy and LWB also leads to waste of channel resources in emerging networked control paradigms that involve only communications between close-by nodes [95].

Focusing on addressing the decades old open problem of predictable co-channel interference control, our study in this paper does not consider duty-cycling nodes for energy efficiency. In WSC networks where energy efficiency may be an important issue, the basic design of PRKS can be integrated with duty-cycling mechanisms in ways similar to those in Glossy [45] and LWB [44]; for instance, the communication of data packets and signaling packets in PRKS can be done in the active periods of a network, allowing the network to sleep in its inactive periods. Detailed study of this is, however, beyond the scope of this paper.

Focusing on distributed control of co-channel interference based on the PRK interference model, our study in this paper does not consider other interference management techniques such as interference cancellation and multi-channel scheduling, and we do not consider other link-reliability control techniques such as rate adaptation and power control. Nonetheless, we expect this work to be relevant in the context of these techniques too, since co-channel interference still needs to be managed even with interference cancellation [76], multi-channel scheduling [108], rate control [12], and power control [79]. We will explore this synergy in our future work. Having nodes transmit busy tones in control channels to share their transmission or reception status has also been explored for channel access control [113], but these work did not study the fundamental problem of identifying interference relations between links, thus they could not ensure predictable interference control.

Real-time channel access scheduling algorithms [13, 35, 78, 107] have been proposed for wireless networks, but most were based on the inaccurate protocol interference model [133].Venkataramanan et al. [116] and Jaramillo et al. [63] also considered delay in distributed scheduling, but they did not address the question of how to identify the interference set of each link. Wang et al. [118] studied delay-constrained link scheduling using the SINR model, but the distributed implementation of their scheduling algorithms requires multiple rounds of network-wide coordination and thus not suitable for dynamic traffic patterns and network conditions. Munir et al. [91] considered using link burst length as the basis of routing and real-time scheduling. But they did not consider cumulative interference in scheduling, and the scheduling algorithm was centralized and not suitable for dynamic, large scale networks either; the algorithm also assumed prior knowledge of traffic patterns and was based on long-term (e.g., 21 days), offline measurement data, thus not suitable for dynamic traffic patterns and network conditions. Saifullah et al. [104] considered real-time scheduling for WirelessHART networks; the work assumed application scenarios where the scheduling is centralized and no channel spatial reuse is allowed, thus the scheduling algorithm is not suitable for dynamic, large-scale networks with limited number of channels available. Even though we do not focus on real-time scheduling in this paper, we expect the predictable link reliability enabled by PRKS to serve as a foundation for developing real-time scheduling protocols in the presence of co-channel interference.

6 Concluding remarks

To enable predictable reliability in data delivery for wireless networked sensing and control, we have proposed the wireless transmission scheduling protocol PRKS that ensures predictable interference control in the presence of non-local interference as well as network and environmental uncertainties. Extensive experimental analysis of PRKS show that it enables predictable link reliability while achieving a high degree of channel spatial reuse in data transmissions. Besides being important by itself, the predictable link reliability enabled by PRKS also serves as a basis for predictable real-time data delivery and for predictable tradeoff between reliability, delay, and throughput in wireless sensing and control networks; we will explore this direction of research in our future work. Building upon the insight from PRKS for mostly-static networks, we will also explore mechanisms of extending PRKS to enable predictable link reliability in mobile networks such as vehicular sensing and control networks.

CHAPTER 4

REAL-TIME CONVERGECAST SCHEDULING IN LOSSY MULTI-HOP WIRELESS SENSOR NETWORKS

1 Introduction

As wireless sensor networks (WSNs) mature, more and more of them are being deployed for real-time applications such as intruder detection, structure monitoring, and patient monitoring. In many such applications, it is imperative that sensed data distributed across a large region is reported to a common sink (a traffic pattern called convergecast) by a deadline, so some actions can be taken promptly, for example, sounding an alarm when an intruder is detected. Even if data reaches the sink after the deadline, it is useless and discarded. In the previous example, the intruder may be beyond range of pursuit.

Existing work in real-time WSNs usually assumes that interference-free transmissions are error-free based on some interference models. Nevertheless, WSNs are often deployed in harsh, even hazardous and hostile, environments such as forests, volcanoes, and battlefields. Besides, low-power wireless communication of sensors is susceptible to external interference, shadowing, and fading. It is thus extremely difficult, if possible at all, to ensure every transmission is successful. Despite much work in real-time WSNs, the effects of unreliable wireless links on real-time data delivery are not well studied, especially in multi-hop scenarios.

In this paper, we aim to fill the gap of delivering real-time data in lossy multi-hop WSNs. More specifically, given a collection tree rooted at the sink and the number of packets queued at each node, we study the problem of finding an optimal scheduling algorithm that maximizes the expected total number of packets delivered to the sink by a deadline, subject to interference constraints. Given the packets queued in the network initially, this is equivalent to maximizing the overall deadline catch ratio (DCR).

Recently, there is some work studying real-time scheduling over unreliable links in multi-hop WSNs [58]. However, it focuses on meeting real-time performance requirement for each individual flow/node. By contrast, we are concerned with the aggregate data gathered at the sink by a deadline. This is motivated by the observation that nodes in a WSN are usually designed to collectively accomplish a common task, e.g., detecting and tracking an intruder. The end results depend on the aggregate data gathered, not necessarily the specific amount of data from each individual node.

Our problem definition and solutions encompass two major types of real-time data collection applications: event detection and periodic sampling. In the former, an event of interest occurs. Some nodes in the vicinity detect it and generate data to notify the sink quickly to respond to the event. In the latter, nodes sense the environment in each period and generate several packets to be delivered to the sink, which processes it immediately to take certain actions depending on application requirements. In periodic sampling applications, we assume the relative deadline is no larger than the period. Thus, it suffices to run our scheduling in every period to maximize the long-term deadline catch ratio.

Contributions of this paper. Towards the objective, we make the following contributions.

1. We formulate the problem into a finite horizon Markov decision process in Section 3, which enables us to compute an optimal scheduling policy based on value iteration. However, plagued by the curse of dimensionality, solving the Markov decision problem becomes computationally prohibitive when the network grows large.

2. We propose an efficient greedy scheduling algorithm called Best Link First Scheduling (BLF) to circumvent the intractability of the Markov decision process formulation in Section 4. After proving a simple greedy scheduling policy is optimal for single-hop WSNs, we extend it to multi-hop settings. The basic idea of BLF is simple: always pick the best quality

links first in a top-down manner. It has the salient feature of directly taking link unreliability into consideration. We also propose one approach for distributed implementation.

3. We carry out extensive simulations to demonstrate BLF's considerable performance improvements over existing solutions, in a wide range of network and traffic settings in Section 5. BLF is consistently better and its advantage is more obvious when real-time requirements are stringent.

2 Preliminaries

2.1 Network and Traffic Models

We consider a multi-hop wireless network of n nodes, where node 0 is the sink. Each node is equipped with a half-duplex radio, which can either transmit or receive a packet at a time. A tree rooted at the sink is constructed by some routing protocols. When node i transmits a packet to its parent, denoted by $h(i)$, $h(i)$ receives it with probability $p(i)$, which we refer to as the quality of the link from i to $h(i)$ hereafter. Figure 116 illustrates an example. The number on each link from node i to $h(i)$ represents its quality $p(i)$. We assume that transmissions along each link and across different links are independent. We also assume transmissions along links that do not share nodes do not interfere with each other as in [58], i.e., we use the primary interference model. This can be achieved by, for instance, separating them in different channels. The absence of secondary interference enables us to isolate the effects of link unreliability on real-time packet delivery.

Time is slotted and a node can transmit one packet within a slot. At time 0, node i has v_i ($v_i \geq 0$) packets to be delivered to the sink by deadline D. Note a node can generate multiple packets because it is equipped with multiple sensors for different parameters, each generating one packet. Or its sampling frequency is different from others'. Also, node i can act as pure relay and does not generate any packet itself, i.e., $v_i = 0$.

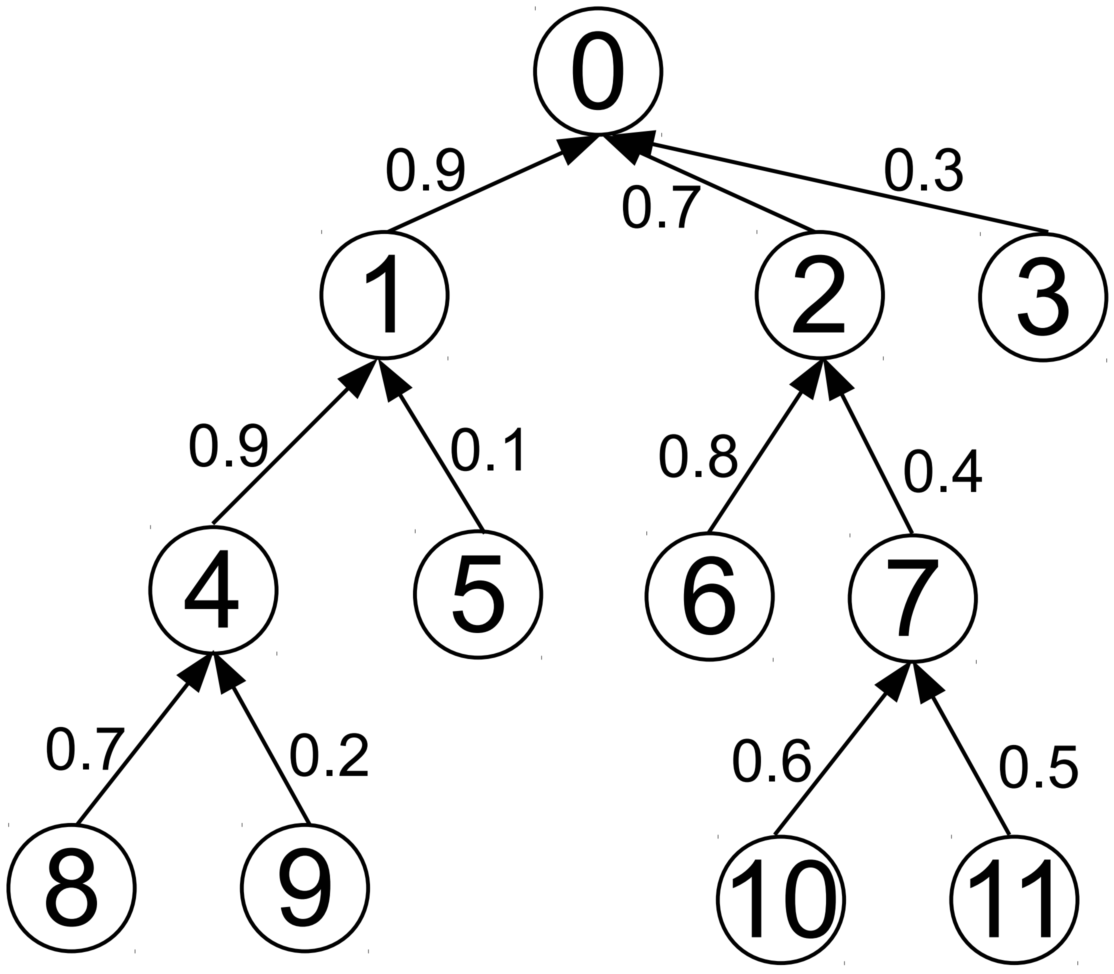


Figure 116: An example of collection tree consisting of 12 nodes with node 0 as the root

2.2 Problem Definition

Given $\mathbf{h} = (h_1, h_2, ..., h_{n-1})$, $\mathbf{p} = (p_1, p_2, ..., p_{n-1})$, and $\mathbf{v} = (v_1, v_2, ..., v_{n-1})$, our objective is to design a scheduling algorithm that maximizes the expected total number of packets reaching the root by deadline D, respecting the primary interference model. Note this optimality is different from the feasibility optimality in [58]. The feasibility optimality means fulfilling the DCR requirement of each individual flow if there exists a scheduling policy that can do so, while our optimality is concerned with maximizing the aggregate DCR of all flows. We do not consider in-network aggregation of packets in this paper.

3 A Mathematical Framework for Optimal Scheduling

Finding an optimal schedule involves sequential decision making in every slot, whose outcomes are stochastic in nature due to link losses. This naturally leads to formulating the aforementioned problem as a Markov decision process (MDP). The MDP's 4-tuple (S, A, P, R) are instantiated as follows.

1. State space S: a state $s \in S$ is the number of packets queued at each node.

2. Action space A: an action $a \in A$ is to schedule a set of nodes to transmit.

3. Transition probability P: P_{ij}^a is the probability state i changes to j when action a is taken. When node m is scheduled to transmit, a packet goes from m to its parent h_m with probability p_m; it stays with probability $1 - p_m$. Since transmissions across different links are independent, we can calculate P_{ij}^a as the product of the probability of each individual transmission success/failure, given the set of transmissions in action a.

4. Reward R: $R(i, a)$ is the expected reward obtained in state i when action a is taken. In each slot, a reward of 1 is gained if a packet reaches root; 0 otherwise.

A policy π is a sequence of decisions that determine which action to take at a given state. Based on this formulation, our objective translates into finding an optimal scheduling policy maximizing the expected total rewards over finite horizon D. An optimal policy can be found by the value iteration algorithm as follows:

$$V_k(i) = \max_a \{R(i,a) + \sum_j P_{ij}^a V_{(k-1)}(j)\} \tag{4.1}$$

$$V_0(i) = 0 \tag{4.2}$$

$V_k(i)$ is the k-step-to-go value of state i, i.e., the maximal reward one expects to get if one starts in state i and proceeds for k steps. The optimal policy π^* is $\{a_1^*, a_2^*, ..., a_D^*\}$ with initial state $\mathbf{v}$, in which a_k^* is the a maximizing the right hand side of Equation 4.1. The MDP formulation enables us to solve the original problem optimally. Unfortunately, it suffers from the curse of dimensionality, where solving for the MDP becomes computationally prohibitive when the network grows large. To see why, consider a simple case of line topology where each node has one packet, the computational complexity is $\mathcal{O}((n!)^2 2^n D)$.

4 Best Link First Scheduling

To overcome the shortcoming of MDP, we propose an efficient greedy scheduling algorithm, Best Link First Scheduling. We demonstrate its underlying intuition in a simplified single-hop setting. After detailing the centralized BLF, we discuss one approach for distributed implementation.

4.1 Single-hop Scenario

Consider a one-hop network in Figure 117. Without loss of generality, let us assume that $p_1 \geq p_2$. Suppose that nodes 1 and 2 each has one packet and it takes them X_1 and X_2 transmissions to successfully deliver their packet to root 0, respectively. X_1 and X_2 are geometric random variables with parameters p_1 and p_2, respectively. Nodes 1 and 2 cannot be active in the same slot. It

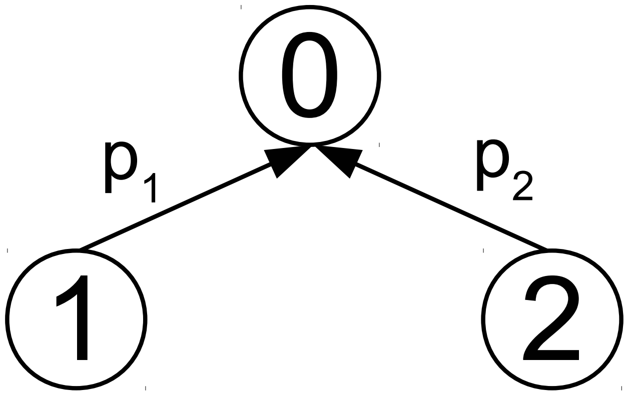


Figure 117: A single-hop tree with 3 nodes

is obvious that an optimal scheduling policy must be work-conserving, i.e., always transmitting whenever there are packets left. Thus, we restrict our discussion on work-conserving policies. Let us consider the following two policies.

- π_1: node 1 transmits first. After node 1's packet is delivered, node 2 transmits.
- π_2: node 2 transmits first. After node 2's packet is delivered, node 1 transmits.

Under π_1, root 0 receives the first packet at slot X_1, the second $X_1 + X_2$. Under π_2, root 0 receives the first packet at slot X_2, the second $X_2 + X_1$. The second packet arrives at the same time in two policies. We focus on the first packet. Given the same deadline D, let $Pr\{X_1 \leq D\}$ and $Pr\{X_2 \leq D\}$ denote the probability that the first packet arrives by D under π_1 and π_2, respectively. Since $p_1 \geq p_2$, $Pr\{X_1 \leq D\} = 1 - Pr\{X_1 > D\} = 1-(1-p_1)^D \geq 1-(1-p_2)^D = 1-Pr\{X_2 > D\} = Pr\{X_2 \leq D\}$. The expected number of packets collected by D under π_1 is no less than that under π_2.

Similarly, it can be shown that other policies where the transmissions of node 1 and node 2 interleave are no better than π_1. This is because any transmission of node 2 before node 1's successful packet delivery delays it. Thus, π_1 is optimal. In fact, this argument holds even when there are more than two children and some of them have multiple packets as Lemma 1 shows.

Lemma 1. *If $h_i = 0$ for every $i > 0$, the policy π' that always schedules the non-empty node with the best quality link to the root is optimal.*

Proof. This result follows from Theorem 1 in [68]. It is also based on translating the problem into an MDP as in Section 3. The difference is that packets are weighted and a reward of α_i is obtained when a packet from flow i is delivered. We let root collect data from its children, instead of transmitting data to them. After this reversal of traffic, our problem in single-hop settings turns out to be a special case of the general real-time multicast problem studied in [68], where each multicast flow degenerates into a unicast flow and is assigned the same weight (i.e., $\alpha_i = 1$ for all i). The optimal GreedyU policy that serves the user with the maximum expected weighted throughput $\alpha_i p_i$ among all unsatisfied users becomes π' above. □

4.2 Multi-hop Scenario

Inspired by the above result in single-hop networks, we propose Best Link First Scheduling for multi-hop networks. The core idea is to still schedule the best link first, as we do in single-hop networks, and then add more links as long as interference constraints are not violated, taking advantage of spatial reuse in a multi-hop network. When multiple links can be added, we add links closer to the root first since their packets are closer and are more likely to arrive before deadline expires. Algorithm 1 details BLF. Ties are broken arbitrarily.

Algorithm 1 Best Link First Scheduling

```
Input: a collection tree $\mathbf{h}$, its link qualities $\mathbf{p}$, and node queue levels $\mathbf{v}$
Output: a slot schedule $\mathcal{S}$
 1: $\mathcal{S} = \emptyset$
 2: $k =$ height of $\mathbf{h}$
 3: for $i$ = 0 to $k - 1$ do
 4:     $\mathcal{A} = \{a \mid a$ is at level $i$ and $h_a \notin \mathcal{S}\}$
 5:     for every node $a \in \mathcal{A}$ do
 6:         $\mathcal{B} = \{b \mid h_b = a$ and $v_b > 0\}$
 7:         $b^* = \underset{b \in \mathcal{B}}{\operatorname{argmax}}\, p_b$
 8:         $\mathcal{S} = \mathcal{S} \cup \{b^*\}$
 9:     end for
10: end for
11: return $\mathcal{S}$
```

Another perspective to look at BLF is through recursion. Starting from the root, we include the non-empty child with the best quality link to it. For all other children, we repeat the same process on the subtree rooted at them recursively, using them as the new roots. For the chosen one, since its children cannot be scheduled, we instead repeat the above process on subtrees rooted at its children. BLF is greedy in the sense that in every slot, it schedules a set of nodes that maximizes the expected number of packets deliverable in that slot. BLF runs in $\mathcal{O}(nD)$ since in every slot, each link is visited once to identify the best one among those sharing a common receiver in the worst case.

As an example, let us consider the tree in Figure 116. Assume that every node has at least one packet. At level 1, nodes 1, 2, and 3 share common parent node 0. Node 1 is scheduled since it has the best quality link to node 0, i.e., $p_1 > p_2 > p_3$. At level 2, nodes 4 and 5 cannot be scheduled since their parent, node 1, has been scheduled. Out of the two children of node 2, which is not scheduled, node 6 is selected since $p_6 > p_7$. Similarly, nodes 8 and 10 are chosen at level 3. Together, nodes 1, 6, 8, and 10 are scheduled to transmit as Figure 118 shows.

4.3 Distributed Implementation

Algorithm 1 is a centralized algorithm. It is amenable for distributed implementation. One straightforward way to implement the distributed algorithm is as follows: precede from top to bottom in rounds to determine the set of nodes to transmit in a slot. In the first round, each child of the root tells the root whether it has any packet. This can be attained by, for example, letting each child transmit in round-robin. The root, upon receiving information from all its children, decides the non-empty child with the best link to it. It announces the decision to all its children. In the second round, nodes at level 1 and not chosen become roots and repeat the same process. This goes on till the last level but one. In the end, all chosen nodes transmit in that slot. We assume that in the bootstrap phase, each node obtains the qualities of the links to its parent and from its children through certain link estimation process. Measures can be taken to ensure reliable control signaling between a parent and a child, such as retransmissions based on known link quality. We delegate more efficient implementations to future work.

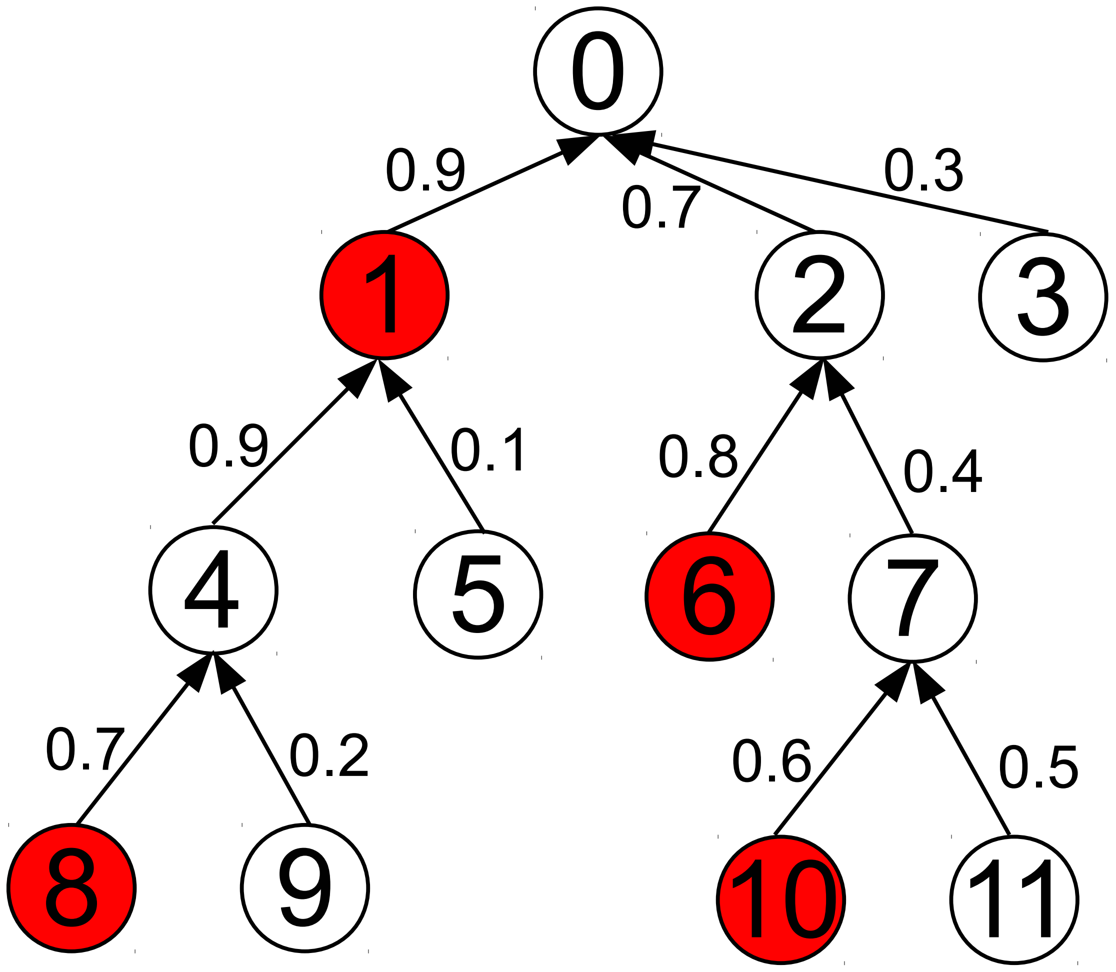


Figure 118: Highlighted nodes are scheduled

5 Evaluation

5.1 Experimental Settings

Protocols. We compare BLF with state of the art through MATLAB simulations in a wide range of network and traffic settings. The performance metric is deadline catch ratio, defined as the percentage of packets reaching the root by deadline out of all the packets in the network initially.

- LDF: we use the Closest Sensor First scheduling in [58] to represent Largest Debt First (LDF) scheduling. LDF schedules the largest debt flow first. The debt of a flow is defined as the number of timely deliveries the flow is lagging behind to satisfy its DCR requirement. For fair comparison, we set the DCR requirement of each flow as the aggregate DCR in BLF.

- CR-SLF: the Channel Reuse-based Smallest latest-start-time (LST) First (CR-SLF) protocol [75] applies techniques from traditional real-time systems theory to real-time scheduling in multi-hop wireless networks. The LSF of a flow can be regarded as its laxity, and CR-SLF is essentially least laxity first scheduling. We set a link's delay as its expected transmission count (ETX) [36] in the implementation.

- LBF: the Largest Branch First scheduling delivers packets in the largest branch/subtree first. The size of a branch is the total number of packets in the branch initially. It is an optimal minimal-length scheduling protocol when transmissions are error-free.

In general, schedule- or TDMA-based protocols are better than contention- or CSMA-based ones for providing real-time (as demonstrated in, for example, [109] and [75]), since the latter are prone to repeated collisions that result in excessive and varying exponential backoff, especially under heavy traffic load. Thus we do not compare against CSMA-based protocols in this study.

Generating networks and traffic. We randomly place n nodes in a unit square. A random node is selected as the sink. We denote transmission range as ρ. That is, two nodes within the distance of ρ can communicate with each other. We construct a collection tree rooted at the sink using the shortest-path algorithm based on hop count. We assign each link in the tree a random quality in the range of $[\alpha, 1]$. Except the root, each node generates a number of packets randomly chosen from 0 to γ.

In the following experiments, we set the default value of n, ρ, α, and γ as 100, 0.2, 0, and 20, respectively, unless otherwise stated. Figure 119 shows the distribution of hop counts to the sink in such a network. When varying a parameter to study its effect on real-time data delivery, we run 30 experiments and show the average result.

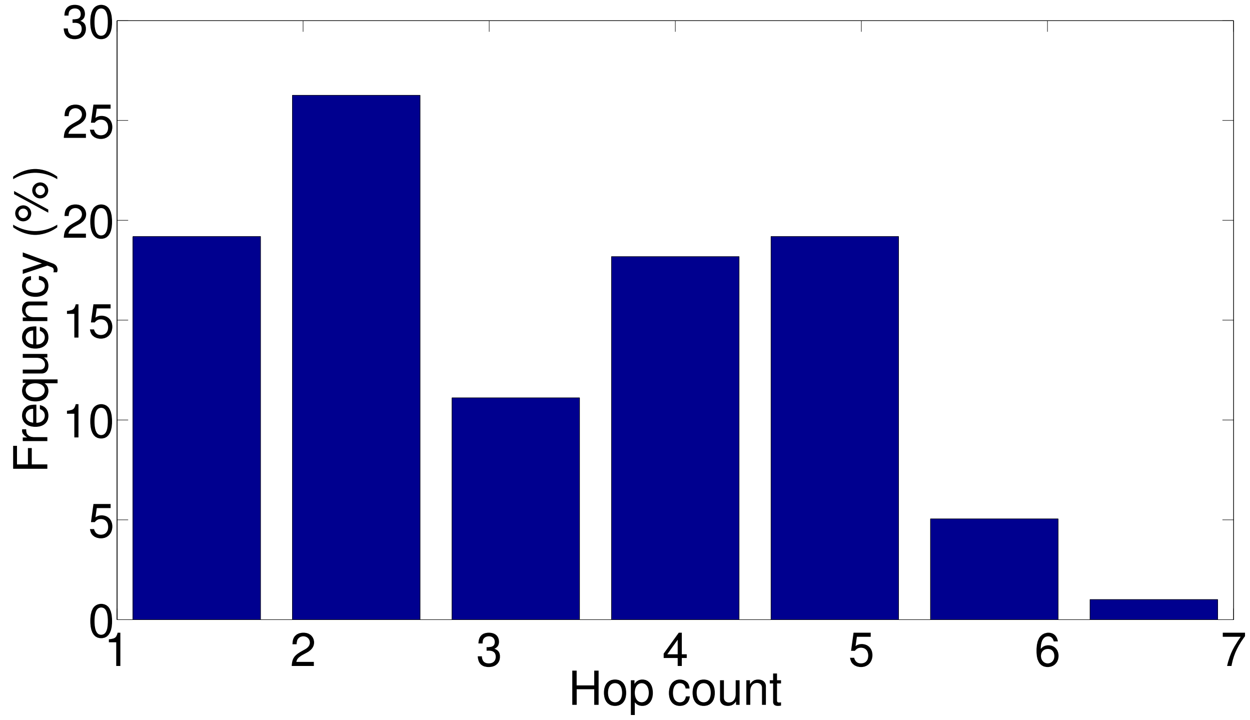


Figure 119: Histogram of hop counts to the root in the random network

5.2 Impact of Deadline

In this experiment, we vary D while keeping all other parameters constant. Figure 120 shows the DCRs of different protocols. We can see BLF outperforms others consistently, especially when D is small. For instance, the DCR of BLF is 52% when D is 1000, versus 16%, 2%, and 15% of LDF, CR-SLF, and LBF, respectively. Also, the DCRs of all protocols increase as deadline increases since more packets reach the root by deadline.

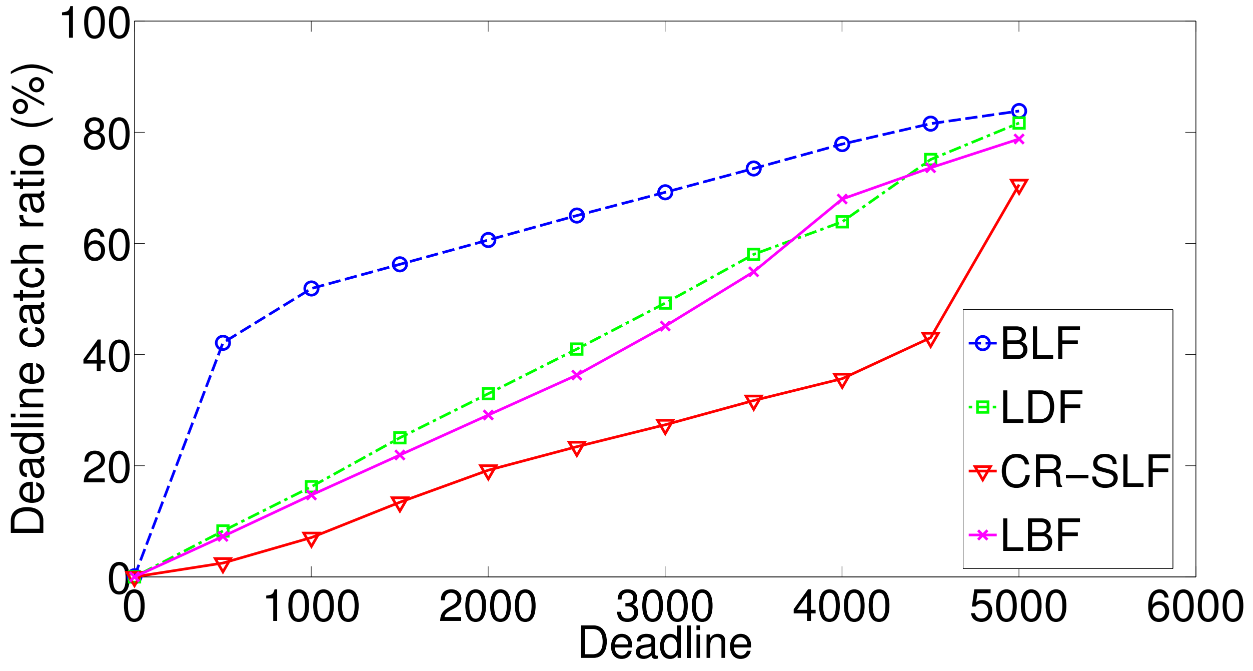


Figure 120: Deadline catch ratios of different protocols for various deadlines

5.3 Impact of Link Quality

In this experiment, we set D as 1000 and use the default value of all parameters except α. Figure 121 shows the DCRs of different protocols. When α is small, i.e., links are unreliable, BLF is significantly better than other protocols. This clearly shows the benefit of explicitly accounting for link loss in real-time scheduling. As α grows, links tend to become more reliable, and more packets are delivered by the same deadline. BLF is still the best among all protocols, even though its advantage gradually diminishes as links become reliable. In fact, when α is no less than 80%, BLF becomes indistinguishable from the rest.

5.4 Impact of Traffic Load

In this experiment, we set D as 1000 and keep all parameters intact except γ. A larger γ amounts to heavier traffic on average by the same deadline. Figure 122 shows the DCRs of different protocols.

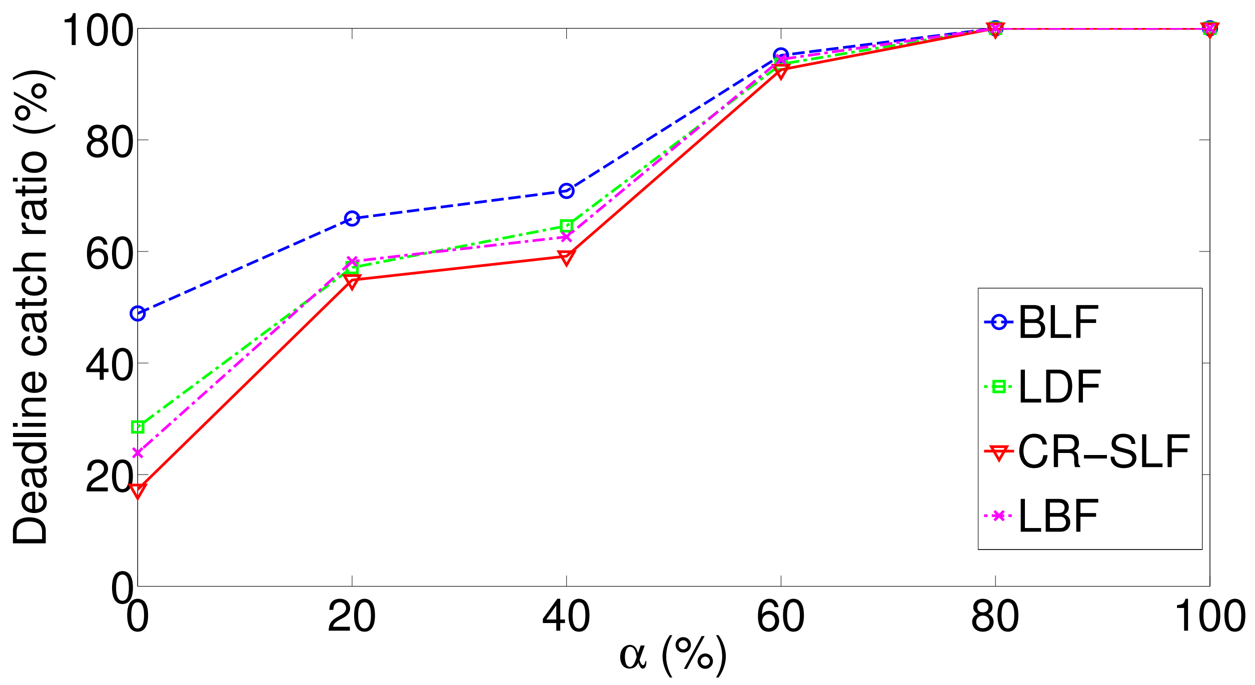


Figure 121: Deadline catch ratios of different protocols for various link qualities

When traffic is light, i.e., γ is small, all protocols enjoy high DCRs. As traffic grows heavier, BLF still maintains high DCRs relative to other protocols, despite that all protocols' DCRs decrease. When γ is 40, BLF still obtains a DCR of 40%, while others only deliver 9% or less of packets in time.

6 Related work

Much work has studied scheduling real-time traffic in wireless networks, which we categorize as follows.

Work in the first category is based on the concept of debt. Each flow has a DCR requirement. The debt of a flow reflects how many timely deliveries the flow is lagging behind to satisfy its DCR requirement. Larger debt flow is favored upon smaller debt one in scheduling. [57] proposes a theory of real-time quality of service (QoS) for wireless networks, factoring in deadline, DCR requirement, and unreliable nature of wireless communication. It derives schedulability test for

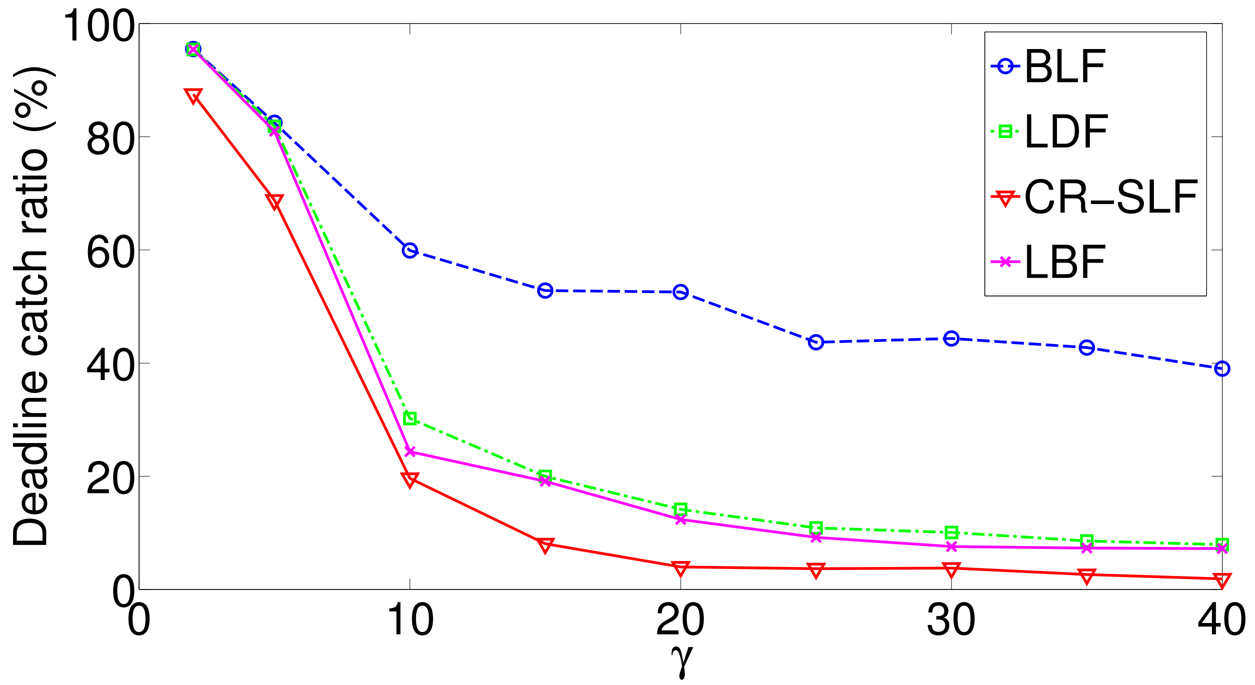


Figure 122: Deadline catch ratios of different protocols for various traffic

admission control that is both sufficient and necessary. Accordingly, largest debt first scheduling is devised and proved optimal. Though the flexible formulation allows for tractable analytical results, it only deals with single-hop topologies. [58] extends real-time QoS to multi-hop WSNs. It derives a sufficient condition for optimal scheduling. For full-duplex radios, Greedy Forwarder scheduling that transmits a packet belonging to be largest debt flow at each node is proved optimal. For half-duplex radios, only a heuristic Closest Sensor First scheduling is proposed, which is only optimal for a restrictive path-topology case. [68] considers scheduling real-time multicast traffic and shows that a policy that schedules the multicast flow whose users' sum of debt is maximal is optimal. However, it works only for one hop scenario. All of the above work provides real-time QoS for individual flow/user, not for their aggregate as we do.

The second category leverages results from real-time systems theory. Generally, these scheduling algorithms quantify the urgency of a packet/flow based on its temporal parameters, borrowing notations from real-time systems. They schedule the most urgent (e.g., smallest latest-start time, least laxity, highest static priority) packet/flow first and then add other transmissions as long as interference and/or deadline constraints permit. [75] proposes the Channel Reuse-based Smallest latest-start-time (LST) First (CR-SLF) protocol to schedule real-time packets in multi-hop robotic sensor networks. The LST of a packet at a hop is defined as the latest time the packet must be transmitted at that hop to reach the destination in time. CR-SLF chooses the packet transmission with the smallest LST and adds other transmissions as many as possible if they do not interfere with any existing transmission nor cause any to miss deadline. [93] schedules periodic real-time flows with given routes and interference matrix using the Interference Aware Anticipatory (IAA) algorithm. It heuristically schedules flow with the smallest ratio of spatial overlap over laxity first. The spatial overlap of a flow is defined as the number of different links of other flows it interferes with, and the laxity of a flow is the tolerable waiting time during its ensuing downstream transmissions. [31] proposes Real-time Flow Scheduling (RFS), an enhanced extension of Real-Time Query Scheduling (RTQS) [30], for periodic real-time flows in WSNs under generic traffic patterns and interference models. Each flow is assigned a static priority. In the core of RFS, it schedules multiple flows concurrently if they are conflict free; when there is a conflict, it preferentially schedules higher priority flows over lower priority ones. [103] proposes an efficient heuristic algorithm, Conflict-aware Least Laxity First (C-LLF), to schedule real-time traffic in WirelessHART networks.

Work in the third category is based on Carrier Sensing Multiple Access/Collision Avoidance (CSMA/CA). The basic idea is to prioritize urgent packets by manipulating channel contention so that higher priority packets are more likely to win the channel. In [67], the authors propose multi-hop coordination, where a priority is assigned to each packet as it traverses across various broadcast regions. A packet's priority is increased at downstream nodes if it experiences excessive delay at upstream nodes and is likely to miss its end-to-end deadline, and vice versa. Within a broadcast region, higher priority packets randomly backoff in smaller contention windows in an extension of IEEE 802.11 and thus are more likely to capture the channel early on. [20] provides priority differentiation among soft real-time flows. It uses Black Burst (BB) for channel contention, which is a jamming signal that a node transmits after hearing the channel idle for a certain duration. The length of a BB is proportional to its packet priority. This ensures a packet with higher priority is favored over one with lower priority when contending the shared channel.

Another closely related line of study is to find a shortest interference-free TDMA schedule, given the number of packets at each source and conflict graph in a multi-hop collection tree. Examples include [109] and [46]. A good survey is [94]. They do not address real-time data collection directly and assume error-free transmissions.

None of the above aims to solve our problem and thus cannot be directly applied.

7 Conclusion

In this paper, we study how to schedule packet transmissions to maximize the expected number of packets evacuated during a given time period in a lossy multi-hop wireless sensor network. We formulate it as a finite horizon Markov decision process and solve it optimally. Unfortunately, the solution becomes computationally intractable as the network and traffic grow. Thus, we propose the efficient Best Link First Scheduling protocol. BLF greedily schedules the best links first in a top-down fashion, complying with interference constraints. We prove its optimality for single-hop networks and develop an approach for distributed implementation. Through simulations in a plethora of network and traffic settings, BLF is shown to consistently outperform existing work, mainly by explicitly accounting for link losses in scheduling. Its gains are especially significant when deadline is small, links are highly unreliable, and traffic load is heavy, making it appealing to increasingly stringent real-time applications.

Considerable work is required to make BLF ultimately decentralized and field-deployable with provable properties. Interesting extensions abound and examples include, but are not limited to, the following: (1) consider more realistic interference models such as the Physical Ratio-K [134] and SINR models; (2) consider heterogeneous traffic where packets can have different arrival times and deadlines and also online traffic where packet arrivals are not known a priori; (3) consider link models other than Bernoulli such as the Gilbert-Elliott or general Markov chain model to account for burst errors; (4) develop efficient distributed implementation with low control overhead, quick convergence, and small memory footprint; (5) consider the problem when in-network aggregation of raw packets is allowed; (6) consider joint scheduling and routing to maximize expected deadline catch ratio.

CHAPTER 5

SUMMARY

My Ph.D. thesis is on predictable and real-time wireless networking for sensing and control. Applications include, but are not limited to, connected vehicles, microgrids, and industrial automation. Toward this goal, my previous work has generated interesting results, including a real-time routing protocol that predictably delivers data by their deadlines, a scheduling protocol to ensure a certain link reliability based on the Physical-ratio-K (PRK) model, which is both realistic and amenable for distributed implementation, and a greedy scheduling algorithm to deliver as many packets as possible to the sink by a deadline in lossy multi-hop wireless sensor networks. Together with the above projects, I attempt to develop a holistic communication stack and solid theoretical foundation for reliable, predictable, real-time communication for wireless networked sensing and control.

REFERENCES

[1] An event traffic trace for sensor networks. http://www.cs.wayne.edu/~hzhang/group/publications/Lites-trace.txt.

[2] Indriya testbed. http://indriya.comp.nus.edu.sg/.

[3] FCC ruling on dedicated short range communications (DSRC). http://wireless.fcc.gov/services/index.htm?job=service_home&id=dedicated_src.

[4] FCC ruling on medical body area networks (MBAN). http://www.fcc.gov/document/medical-body-area-networks-first-report-and-order.

[5] IEEE 802.15 smart utility networks task group 4g. http://www.ieee802.org/15/pub/TG4g.html.

[6] IEEE 802.15.4e Working Group. http://www.ieee802.org/15/pub/TG4e.html.

[7] IEEE standard for wireless access in vehicular environments (WAVE) — multi-channel operation. IEEE Std 1609.4.

[8] TelosB sensor node. http://www.memsic.com.

[9] TinyOS. http://www.tinyos.net/.

[10] TinyOS TEP 133: Packet-level time synchronization. http://www.tinyos.net/tinyos-2.x/doc/html/tep133.html.

[11] NetEye testbed. http://neteye.cs.wayne.edu/neteye/home.php, 2008.

[12] R. Al-Zubi and M. Krunz. Interference management and rate adaptation in OFDM-based UWB networks. *IEEE Transactions on Mobile Computing*, 9(9), 2010.

[13] B. O. Baldwin, N. J. D. IV, and S. F. Midkiff. A real-time medium access control protocol for ad hoc wireless local area networks. *ACM Mobile Computing and Communictions Review*, 3(2), April 1999.

[14] L. Bao and J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *ACM MobiCom*, 2001.

[15] S. Bapat, V. Kulathumani, and A. Arora. Analyzing the yield of exscal, a large-scale wireless sensor network experiment. In *IEEE ICNP*, pages 53–62, 2005.

[16] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM*, 2005.

[17] D. M. Blough, S. Das, G. Resta, and P. Santi. A framework for joint scheduling and diversity exploitation under physical interference in wireless mesh networks. In *IEEE MASS*, 2008.

[18] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *ACM MobiCom*, 2006.

[19] G. Brar, D. M. Blough, and P. Santi. The SCREAM approach for efficient distributed scheduling with physical interference in wireless mesh networks. In *ICDCS*, 2008.

[20] B. D. Bui, R. Pellizzoni, M. Caccamo, C. F. Cheah, and A. Tzakis. Soft Real-Time Chains for Multi-Hop Wireless Ad-Hoc Networks. In *13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS'07)*, pages 69–80. IEEE, Apr. 2007.

[21] M. Caesar, M. Casado, T. Koponen, J. Rexford, and S. Shenker. Dynamic route computation considered harmful. *ACM SIGCOMM Computer Communication Review*, 40(2):66–71, 2010.

[22] L. X. Cai, L. Cai, X. Shen, J. W. Mark, and Q. Zhang. MAC protocol design and optimization for multi-hop ultra-wideband networks. *IEEE TWC*, 8(8), 2009.

[23] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. In *ACM MobiHoc*, 2005.

[24] A. Chakrabarti and G. Manimaran. Reliability constrained routing in QoS networks. *IEEE/ACM Transactions on Networking*, 13(3):662–675, 2005.

[25] S. Chakrabarti and A. Mishra. QoS issues in ad hoc wireless networks. *IEEE Communications Magazine*, 39(2):142–148, February 2001.

[26] X. Che, X. Ju, and H. Zhang. The case for addressing the limiting impact of interference on wireless scheduling. In *IEEE ICNP*, 2011.

[27] L. Chen and W. B. Heinzelman. QoS-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE JSAC*, 23(3):561–571, 2005.

[28] S. Chen, M. Song, and S. Sahni. Two techniques for fast computation of constrained shortest paths. *IEEE/ACM Transactions on Networking*, 16(1):105–115, February 2008.

[29] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. In *IWQoS*, 2006.

[30] O. Chipara, C. Lu, and G. Roman. Real-time query scheduling for wireless sensor networks. In *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*, pages 389–399, 2007.

[31] O. Chipara, C. Wu, C. Lu, and W. Griswold. Interference-aware real-time flow scheduling for wireless sensor networks. In *Proceedings of the 2011 23rd Euromicro Conference on Real-Time Systems*, ECRTS '11, pages 67–77, Washington, DC, USA, 2011. IEEE Computer Society.

[32] J. I. Choi, M. Jain, M. A. Kazandjieva, and P. Levis. Granting silence to avoid wireless collisions. In *IEEE ICNP*, 2010.

[33] D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4-6), 2000.

[34] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM MobiCom*, 2003.

[35] T. L. Crenshaw, A. T. Hoke, and M. Caccamo. A robust implicit access protocol for real-time wireless collaboration. In *ECRTS*, 2005.

[36] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom '03*, page 134, New York, New York, USA, Sept. 2003. ACM Press.

[37] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *ACM SIGCOMM*, 2004.

[38] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MobiCom*, 2004.

[39] S. C. Ergen and P. Varaiya. Energy efficient routing with delay guarantee for sensor networks. *Wireless Networks*, 13(5):679–690, 2007.

[40] M. Erol-Kantarci and H. T. Mouftah. Wireless sensor networks for cost-efficient residential energy management in the smart grid. *IEEE Transactions on Smart Grid*, 2(2):314–325, 2011.

[41] Y. P. Fallah, C. Huang, R. Sengupta, and H. Krishnan. Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics. In *ACM/IEEE ICCPS*, 2010.

[42] X. Fang, D. Yang, P. Gundecha, and G. Xue. Multi-constrained anypath routing in wireless mesh networks. In *IEEE SECON*, 2010.

[43] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In *IEEE INFOCOM*, 2005.

[44] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *ACM SenSys*, 2012.

[45] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *ACM/IEEE IPSN*, 2011.

[46] C. Florens, M. Franceschetti, and R. McEliece. Lower Bounds on Data Collection Time in Sensory Networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1110–1120, Aug. 2004.

[47] L. Fu, S. C. Liew, and J. Huang. Effective carrier sensing in CSMA networks under cumulative interference. In *IEEE INFOCOM*, 2010.

[48] H. Gharavi and B. Hu. Multigate communication network for smart grid. *Proceedings of the IEEE*, 99(6):1028–1045, 2011.

[49] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, 2009.

[50] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric SINR. In *ACM MobiHoc*, 2007.

[51] R. A. Guerin and A. Orda. QoS routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350–364, 1999.

[52] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. *IEEE/ACM Transactions on Networking*, 17(6), 2009.

[53] A. Hasan and J. G. Andrews. The guard zone in wireless ad hoc networks. *IEEE TWC*, 6(3), 2007.

[54] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In *IEEE ICDCS*, 2003.

[55] J. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. Wiley-IEEE Press, 2004.

[56] B. Hobijn, P. H. Franses, and M. Ooms. Generalizations of the KPSS-test for stationarity. *Statistica Neerlandica*, 58(4):483–502, 2004.

[57] I. hong Hou, V. Borkar, and P. R. Kumar. A theory of qos for wireless. In *In Proc. of IEEE INFOCOM*, 2009.

[58] I.-H. Hou. Providing End-to-End Delay Guarantees for Multi-hop Wireless Sensor Networks over Unreliable Channels. Apr. 2012.

[59] J. Huang, S. Liu, G. Xing, H. Zhang, J. Wang, and L. Huang. Accuracy-aware interference modeling and measurement in wireless sensor networks. In *IEEE ICDCS*, 2011.

[60] X. Huang and Y. Fang. Multiconstrained QoS multipath routing in wireless sensor networks. *Wireless Networks*, 14:465–478, 2008.

[61] IETF. Routing over low power and lossy networks (ROLL) working group. `http://www.ietf.org/html.charters/roll-charter.html`.

[62] R. Jain and I. Chlamtac. The P^2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM*, 28(10):1076 – 1085, 1985.

[63] J. J. Jaramillo and R. Srikant. Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic. In *IEEE INFOCOM*, 2010.

[64] L. B. Jiang and S. C. Liew. Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks. *IEEE Transactions on Mobile Computing*, 7(1), 2006.

[65] C. Joo and N. B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. In *IEEE INFOCOM*, 2007.

[66] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *ACM SIGCOMM*, 2005.

[67] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. *Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom 01*, pages 200–209, 2001.

[68] K. S. Kim, C.-p. Li, and E. Modiano. Scheduling multicast traffic with deadlines in wireless networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2193–2201. IEEE, Apr. 2014.

[69] T. Korkmaz and M. Krunz. Bandwidth-delay constrained path selection under inaccurate state information. *IEEE/ACM Transactions on Networking*, 11(3):384 – 398, 2003.

[70] A. LaJoie. Wireless sensors enjoy timely role in factory automation. *Industrial Ethernet Book*, April 2010.

[71] L. B. Le, E. Modiano, C. Joo, and N. B. Shroff. Longest-queue-first scheduling under SINR interference model. In *ACM MobiHoc*, 2010.

[72] S. Leng, Y. Zhang, H.-H. Chen, J. Xiang, and M. Guizani. Power-fixed and power-aware MAC protocols for multihop wireless networks with a large interference area. *IEEE Transactions on Vehicular Technology*, 58(6), July 2009.

[73] K. K. Leung, C. W. Sung, W. S. Wong, and T. M. Lok. Convergence theorem for a general class of power-control algorithms. *IEEE Transactions on Communications*, 52(9):1566–1574, 2004.

[74] H. Li, Y. Cheng, and C. Zhou. Minimizing end-to-end delay: A novel routing metric for multi-radio wireless mesh networks. In *IEEE INFOCOM*, 2009.

[75] H. Li, P. Shenoy, and K. Ramamritham. Scheduling messages with deadlines in multi-hop real-time sensor networks. *11th IEEE Real Time and Embedded Technology and Applications Symposium*, pages 415–425, 2005.

[76] L. E. Li, R. Alimi, D. Shen, H. Viswanathan, and Y. R. Yang. A general algorithm for interference alignment and cancellation in wireless networks. In *IEEE INFOCOM*, 2010.

[77] Q. Li and R. Negi. Scheduling in multi-hop wireless networks with priorities. In *IEEE SECON*, 2009.

[78] C. R. Lin and M. Gerla. Real-time support in multihop wireless networks. *Wireless Networks*, 5:125–135, 1999.

[79] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. ATPC: Adaptive transmission power control for wireless sensor networks. In *ACM SenSys*, 2006.

[80] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. A. Stankovic, and T. He. Towards stable network performance in wireless sensor networks. In *IEEE RTSS*, 2009.

[81] S. Liu, G. Xing, H. Zhang, J. Wang, M. Sha, and L. Huang. Passive interference measurement in wireless sensor networks. In *IEEE ICNP*, 2010.

[82] X. Liu, H. Zhang, and Q. Xiang. Towards predictable real-time routing for wireless networked sensing and control. In *CPS Week RealWin workshop*, 2011.

[83] X. Liu, H. Zhang, Q. Xiang, X. Che, and X. Ju. Taming uncertainties in real-time routing for wireless networked sensing and control. Technical Report DNC-TR-11-04 (`https://sites.google.com/site/dnctrs/DNC-TR-11-04.pdf`), Wayne State University, 2011.

[84] M. Loeve. *Probability Theory*. Springer-Verlag, 1977.

[85] H. Ma, H. M. Alazemi, and S. Roy. A stochastic model for optimizing physical carrier sensing and spatial reuse in wireless ad hoc networks. In *IEEE MASS*, 2005.

[86] R. Maheshwari, S. Jain, and S. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *ACM SenSys*, 2008.

[87] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *ACM SenSys*, 2004.

[88] A. H. Mohsenian-Rad, V. W. Wong, and R. Schober. Optimal SINR-based random access. In *IEEE INFOCOM*, 2010.

[89] J. R. Moyne and D. M. Tilbury. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9–28, 2007.

[90] J. R. Moyne and D. M. Tilbury. The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proceedings of the IEEE*, 95(1), 2007.

[91] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *IEEE/ACM IPSN*, 2010.

[92] J. Ni, B. Tan, and R. Srikant. Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks. In *IEEE INFOCOM*, 2010.

[93] S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse. IAA: Interference aware anticipatory algorithm for scheduling and routing periodic real-time streams in wireless sensor networks. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 14–21. IEEE, June 2010.

[94] Ozlem Durmaz Incel and Amitabha Ghosh and Bhaskar Krishnamachari. *Scheduling Algorithms for Tree-Based Data Collection in Wireless Sensor Networks*.

[95] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam. The wireless control network: A new approach for control over networks. *IEEE Transactions on Automatic Control*, 56(10):2305–2318, 2011.

[96] K.-J. Park, L. Kim, and J. C. Hou. Adaptive physical carrier sense in topology-controlled wireless networks. *IEEE Transactions on Mobile Computing*, 9(1), 2009.

[97] S. Park and S.-M. Yoo. An efficient reliable one-hop broadcast in mobile ad hoc networks. *Ad Hoc Networks*, 11(1):19–28, 2013.

[98] K. S. J. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *IASTED DSN*, 2008.

[99] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *ACM SenSys*, 2004.

[100] D. Qian, D. Zheng, J. Zhang, and N. Shroff. CSMA-based distributed scheduling in multihop MIMO networks under SINR model. In *IEEE INFOCOM*, 2010.

[101] S. M. Ross. *Introduction to Probability Models*. Academic Press, 2006.

[102] J. Ryu, C. Joo, T. Kwon, N. Shroff, and Y. Choi. DSS : Distributed SINR-based scheduling algorithm for multihop wireless networks. *IEEE Transactions on Mobile Computing*, 12(6):1120–1132, 2013.

[103] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Real-time scheduling for wirelesshart networks. In *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*, pages 150–159, 2010.

[104] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Real-time scheduling in WirelessHART networks. In *IEEE RTSS*, 2010.

[105] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-MAC: Model-driven concurrent medium access control for wireless sensor networks. In *IEEE INFOCOM*, 2009.

[106] G. Sharma, R. R. Mazumdar, and N. B. Shroff. On the complexity of scheduling in wireless networks. In *ACM MobiCom*, 2006.

[107] J.-P. Sheu, C.-H. Liu, S.-L. Wu, and Y.-C. Tseng. A priority MAC protocol to support real-time traffic in ad hoc networks. *Wireless Networks*, 10, 2004.

[108] J. So and N. Vaidya. Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *ACM MobiHoc*, 2004.

[109] W.-z. Song, F. Yuan, and R. LaHusen. Time-Optimum Packet Scheduling for Many-to-One Routing in Wireless Sensor Networks. In *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems*, pages 81–90. IEEE, Oct. 2006.

[110] S. Srinivasa and M. Haenggi. Throughput-delay-reliability tradeoffs in multihop networks with random access. In *Allerton*, 2010.

[111] S. Tang, X. Wu, X. Mao, Y. Wu, P. Xu, G. Chen, and X.-Y. Li. Low complexity stable link scheduling for maximizing throughput in wireless networks. In *IEEE SECON*, 2009.

[112] ISA SP100.11a. `http://www.isa.org//MSTemplate.cfm?MicrositeID=1134&CommitteeID=6891`.

[113] F. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, 1975.

[114] H.-M. Tsai, C. Saraydar, T. Talty, M. Ames, A. Macdonald, and O. K. Tonguz. Zigbee-based intra-car wireless sensor network. In *IEEE ICC*, 2007.

[115] J. V. Uspensky. *Introduction to Mathematical Probability*. McGraw-Hill Book Company Inc., 1937.

[116] V. Venkataranmanan, X. Lin, L. Ying, and S. Shakkottai. On scheduling for minimizing end-to-end buffer usage over multihop wireless networks. In *IEEE INFOCOM*, 2010.

[117] L. Wang and G. Yin. Quantized identification under dependent noise and fisher information ratio for communication channels. *IEEE Transactions on Automatic Control*, 55:674–690, 2010.

[118] Q. Wang, D. Wu, and P. Fan. Delay-constrained optimal link scheduling in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 59(9), 2010.

[119] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder. Efficient interference-aware TDMA link scheduling for static wireless networks. In *AMC MobiCom*, 2006.

[120] W. Wang, Y. Xu, and M. Khanna. A survey on the communication architectures in smart grid. *Computer Networks*, 55:3604–3629, 2011.

[121] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, May 2008.

[122] WirelessHART. `http://www.hartcomm.org/protocol/wihart/wireless_technology.html`.

[123] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-L. Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *IEEE ISPAN*, 2000.

[124] S. S. T. R. J. L. R. L. X. Wu, S. Tavildar and A. Jovicic. FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks. In *Allerton*, 2010.

[125] G. Xing, M. Sha, J. Huang, G. Zhou, X. Wang, and S. Liu. Multi-channel interference measurement and modeling in low-power wireless networks. In *IEEE RTSS*, 2009.

[126] Y. Xue, B. Ramamurthy, and M. C. Vuran. SDRCS: A service-differentiated real-time communication scheme for event sensing in wireless sensor networks. *Computer Networks*, 55:3287–3302, 2011.

[127] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *IEEE InfoCom*, pages 1567–1576, 2002.

[128] Y. Yi, G. D. Veciana, and S. Shakkottai. On optimal MAC scheduling with physical interference model. In *IEEE INFOCOM*, 2007.

[129] S. Yin, Y. Xiong, Q. Zhang, and X. Lin. Traffic-aware routing for real-time communications in wireless multi-hop networks. *Wireless Communications and Mobile Computing*, 6:825–843, 2006.

[130] F. R. Yu, P. Zhang, W. Xiao, and P. Choudhury. Communication systems for grid integration of renewable energy resources. *IEEE Network*, 25(5):22–29, September/October 2011.

[131] H. Zhang, A. Arora, Y. ri Choi, and M. Gouda. Reliable bursty convergecast in wireless sensor networks. *Computer Communications (Elsevier)*, 30(13), 2007.

[132] H. Zhang, A. Arora, and P. Sinha. Link estimation and routing in sensor network backbones: Beacon-based or data-driven? *IEEE Transactions on Mobile Computing*, 8(5):653 – 667, May 2009.

[133] H. Zhang, X. Che, X. Liu, and X. Ju. Adaptive instantiation of the protocol interference model in wireless networked sensing and control. *ACM Transactions on Sensor Networks (TOSN)*, 10(2), 2014.

[134] H. Zhang, X. Che, X. Liu, and X. Ju. Adaptive instantiation of the protocol interference model in wireless networked sensing and control. *ACM Trans. Sen. Netw.*, 10(2):28:1–28:48, Jan. 2014.

[135] H. Zhang, C. Li, X. Liu, Y. Chen, X. Che, F. Lin, L. Y. Wang, and G. Yin. PRK-based scheduling for predictable link reliability in wireless networked sensing and control. Technical Report DNC-TR-12-02 (`https://sites.google.com/site/dnctrs/DNC-TR-12-02.pdf`), Wayne State University, 2012.

[136] H. Zhang, L. Sang, and A. Arora. On the convergence and stability of data-driven link estimation and routing in sensor networks. *ACM Transactions on Autonomous and Adaptive Systems*, 4(3):18:1–29, July 2009.

[137] H. Zhang, L. Sang, and A. Arora. Comparison of data-drive link estimation methods in low-power wireless networks. *IEEE Transactions on Mobile Computing*, 9(11):1634–1648, November 2010.

[138] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, 2003.

[139] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher. RID: Radio interference detection in wireless sensor networks. In *IEEE INFOCOM*, 2005.

ABSTRACT

PREDICTABLE REAL-TIME WIRELESS NETWORKING FOR SENSING AND CONTROL

by

XIAOHUI LIU

May 2015

Advisor: Dr. Hongwei Zhang

Major: Computer Science

Degree: Doctor of Philosophy

Towards the end goal of providing predictable real-time wireless networking for sensing and control, we have developed a real-time routing protocol MTA that predictably delivers data by their deadlines, and a scheduling protocol PRKS to ensure a certain link reliability based on the Physical-ratio-K (PRK) model, which is both realistic and amenable for distributed implementation, and a greedy scheduling algorithm to deliver as many packets as possible to the sink by a deadline in lossy multi-hop wireless sensor networks.

1. *MTA:* Real-time routing is a basic element of closed-loop, real-time sensing and control, but it is challenging due to dynamic, uncertain link/path delays. The probabilistic nature of link/path delays makes the basic problem of computing the probabilistic distribution of path delays NP-hard, yet quantifying probabilistic path delays is a basic element of real-time routing and may well have to be executed by resource-constrained devices in a distributed manner; the highly-varying nature of link/path delays makes it necessary to adapt to in-situ delay conditions in real-time routing, but it has been observed that delay-based routing can lead to instability, estimation error, and low data delivery performance in general. To address these challenges, we propose the *Multi-Timescale Estimation (MTE)* method; by accurately estimating the mean and variance of per-packet transmission time and by adapting to fast-varying queueing in an accurate, agile manner, MTE enables accurate, agile, and efficient

estimation of probabilistic path delay bounds in a distributed manner. Based on MTE, we propose the *Multi-Timescale Adaptation (MTA)* routing protocol; MTA integrates the stability of an ETX-based directed-acyclic-graph (DAG) with the agility of spatiotemporal data flow control within the DAG to ensure real-time data delivery in the presence of dynamics and uncertainties. We also address the challenges of implementing MTE and MTA in resource-constrained devices such as TelosB motes. We evaluate the performance of MTA using the NetEye and Indriya sensor network testbeds. We find that MTA significantly outperforms existing protocols, e.g., improving deadline success ratio by 89% and reducing transmission cost by a factor of 9.7.

2. *PRKS:* Predictable wireless communication is another basic enabler for networked sensing and control in many cyber-physical systems, yet co-channel interference remains a major source of uncertainty in wireless communication. Integrating the protocol model's locality and the physical model's high fidelity, the physical-ratio-K (PRK) interference model bridges the gap between the suitability for distributed implementation and the enabled scheduling performance, and it is expected to serve as a foundation for distributed, predictable interference control. To realize the potential of the PRK model and to address the challenges of distributed PRK-based scheduling, we design protocol PRKS. PRKS uses a control-theoretic approach to instantiating the PRK model according to in-situ network and environmental conditions, and, through purely local coordination, the distributed controllers converge to a state where the desired link reliability is guaranteed. PRKS uses local signal maps to address the challenges of anisotropic, asymmetric wireless communication and large interference range, and PRKS leverages the different timescales of PRK model adaptation and data transmission to decouple protocol signaling from data transmission. Through sensor network testbed-based measurement study, we observe that, unlike existing scheduling protocols where link reliability is unpredictable and the reliability requirement satisfaction ratio can be as low as 0%, PRKS enables predictably high link reliability (e.g., 95%) in different network and environmental conditions without a priori knowledge of these conditions, and,

through local distributed coordination, PRKS achieves a channel spatial reuse very close to what is enabled by the state-of-the-art centralized scheduler while ensuring the required link reliability. Ensuring the required link reliability in PRKS also reduces communication delay and improves network throughput.

3. *BLF:* We study the problem of scheduling packet transmissions to maximize the expected number of packets collected at the sink by a deadline in a multi-hop wireless sensor network with lossy links. Most existing work assumes error-free transmissions when interference constraints are complied, yet links can be unreliable due to external interference, shadowing, and fading in harsh environments in practice. We formulate the problem as a Markov decision process, yielding an optimal solution. However, the problem is computationally intractable due to the curse of dimensionality. Thus, we propose the efficient and greedy Best Link First Scheduling (BLF) protocol. We prove it is optimal for the single-hop case and provide an approach for distributed implementation. Extensive simulations show it greatly enhances real-time data delivery performance, increasing deadline catch ratio by up to 50%, compared with existing scheduling protocols in a wide range of network and traffic settings.

AUTOBIOGRAPHICAL STATEMENT

Xiaohui Liu is a Ph. D. student in the Computer Science Department of Wayne State University. He is a member of the Dependable Networking and Computing Group, led by Dr. Hongwei Zhang. His primary interest lies in predictable and reliable communication for wireless and sensor networks, especially with real-time networking for sensing and control. He works at the confluence of theory and systems, designing MAC, routing, and cross-layer protocols with theoretical foundation as well as empirical considerations. Prior to joining Wayne State University, he received his bachelor degree in Computer Science from Wuhan University, China. Xiaohui has published in premier journals and conferences such as IEEE Transactions on Smart Grid, the ACM Transactions on Sensor Networks, IEEE Transactions on Mobile Computing, IEEE Real-Time Systems Symposium, The ACM International Symposium on Mobile Ad Hoc Networking and Computing, and IEEE International Conference on Sensing, Communication, and Networking. In addition, he has led a student team to become one of the 15 US finalists (out of several thousand teams nationwide) in the 2009 Microsoft Imagine Cup software design competition. He is a recipient of the Outstanding Graduate Research Assistant Award in Computer Science Department at Wayne State University. He is a student member of ACM. More detailed information about Xiaohui can be found from his website at http://www.cs.wayne.edu/xliu.