Traffic-aware Routing for Real Time Communications in Wireless Multi-hop Networks

Shouyi Yin*, Yongqiang Xiong†, Qian Zhang‡, Xiaokang Lin*

*Department of Electronic Engineering, Tsinghua University

†Microsoft Research Asia

‡Department of Computer Science, Hongkong University of Science and Technology

Abstract

In this paper, we propose a novel traffic-aware routing metric for Real Time Communications (RTC) in wireless multi-hop networks. Our routing metric, Path Predicted Transmission Time (PPTT), is designed to choose a high-quality path for RTC flow between a source and a destination. PPTT can serve as both single-radio and multi-radio routing metric for RTC flow.

Real Time Communications has critical quality of service (QoS) requirements in terms of delay, bandwidth and so on. Traditional measurement-based routing schemes ignore the interference from the coming RTC flow itself (i.e. self-traffic), so they may choose the inefficient path to serve the coming RTC flow due to the inaccurate quality estimation of the transmission path. PPTT takes explicit consideration of both self-traffic and neighboring traffic interfering with the RTC flow, and thus offers an accurate estimation of path transmission delay. Through differentiating the links by the wireless channel/radio they are using, PPTT has the capability to choose a high-quality path for the coming RTC flow in both single-radio and mutli-radio networks.

To evaluate the performance, we implement PPTT scheme and study its performance in a wireless multi-hop testbed consisting of 32 nodes equipped with two IEEE 802.11 a/b/g combo cards, and we also conduct extensive simulations with different random topologies in network simulator NS2 for a more comprehensive comparison. The results of simulation and experiment show that this routing metric outperforms other non traffic-aware one such as ETX (Expected Transmission Count) and WCETT (Weighted Cumulative Expected Transmission Time) in terms of delay and goodput in both single radio and multi-radio wireless networks.

Index Terms

Wireless multi-hop networks, IEEE 802.11 MAC, ad-hoc routing, multi-radio routing, interference, quality of service (QoS), real time communication (RTC).

This work has been done when the first author was visiting wireless and networking group, Microsoft Research Asia.

I. INTRODUCTION

Real time communication (RTC), as one of the most popular applications, has drawn great attention in recent years, especially in wireless multi-hop network [1] which is emerging due to its decentralized nature and the popularity of wireless devices such as Wi-Fi (IEEE 802.11[27]) cards[1]. Routing is the key issue for RTC over wireless multi-hop networks, since it determines whether the coming RTC traffic[2] can be served on a high quality path or not. There are two major challenges for routing RTC traffic over wireless multi-hop networks: strong QoS provision and severe interference in wireless networks. On one hand, RTC applications have critical delay and bandwidth requirements, to ensure providing high quality support for the coming RTC traffic, the system needs to accurately predict the path quality in advance before the traffic really being accepted in order to select the best candidate path. On the other hand, in wireless multi-hop networks, interference is the key factor impacting the path performance. To serve the coming RTC traffic, besides the interference coming from physical environment, two types of traffics will also interfere with this RTC flow, one is neighboring traffic including the traffic cross the same node and adjacent nodes, the other is this RTC traffic itself (we call it self-traffic) along the path. Both these two types of interference should be considered to estimate path quality.

We use Fig.1 to illustrate the impact of the self-traffic. Suppose the RTC traffic is sending along node A to node D via node B and C sequentially. We focus on analyzing the performance of the link from A to B. Firstly, this RTC flow sending from node B to node C will contend with the transmission on node A in the same channel, thus affect the traffic on link A-B. Moreover, when node C begins forwarding this RTC traffic to D, node B can not receive packets from A, so transmission on link A-B is also affected by the same flow traffic on link C-D. Thus we can see that besides the neighboring traffic, self-traffic also affects the path quality, so we need take the self-traffic effect into account to estimate the path quality for the coming RTC traffic. However traditional measurement-based routing schemes can not get an accurate estimation, especially for large volume RTC traffic, because when such a scheme performs probing and measurement in routing selection stage, the coming traffic is not injected into the network yet so that the measurement results can not account the self-traffic interference. This weakness will cause wrong routing selection for large volume RTC traffic. The simulation and experiment results in Section III and Section VI illustrate this phenomenon.

Recently, people try to equip multiple radios for wireless nodes to improve the capacity of wireless network[5] [2]. Routing in multi-radio network is different from that in traditional single-radio network. In multi-radio network, two adjacent nodes or links can choose two non-interfering radios or channels. This

[1]In this paper, we only focus on wireless multi-hop networks consisting of IEEE 802.11 devices.

[2]In this paper, the terms “traffic” and “flow” are used inter-exchangeable.

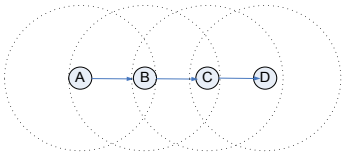


Fig. 1. RTC traffic from node A to D via B and C. Traffic on link A-B will be interfered with the same flow on link B-C and link C-D, respectively.

means a node can send and receive packets on two non-overlapping radios or channels simultaneously, and two adjacent nodes or links can send packet at the same time without mutual interference. The interference caused by self-traffic and neighboring traffics will be affected by this channel diversity in multi-radio networks. As we will explain in detail in Section III.B, the self-traffic interference is more severe in multi-radio scenario.

The key problem of routing RTC flow is to take both neighboring traffic and self-traffic into account in wireless single/multi-radio network. For neighboring traffic, we can use some probing techniques, such as "Round Trip Time" [23], "Packet Pair Delay" [24], "Expected Transmission Count" [22], to get its traffic information and measure its impact. However, since self-traffic will not appear until the RTC traffic is injected into the path, we can not use any measurement-based method to quantify the impact of self-traffic interference. Thus we have to construct a mathematical model to predict the path quality when the RTC flow is serving on that path. To address this issue, we analyze how neighboring traffic and self-traffic affect the path quality. We find that both interference affect packet processing time (We call it *packet service time* in the following.) of MAC layer in essence. And we also derive an analytical model from IEEE 802.11 MAC standard, which can calculate the packet service time according to the neighboring traffic and self-traffic information. Considering the rate of RTC flow is well-controlled, *e.g.* a video stream delivered with 256kbps bitrate, we can get self-traffic information from application layer. Thus we can get an accurate prediction of the RTC packet service time over IEEE 802.11 networks, according to both self-traffic information and neighboring traffic information which can be measured. Based on the aforementioned idea, we propose a novel traffic-aware routing metric to address quality prediction issue for RTC traffic. We use expected end-to-end transmission time on the path as the quality metric which can also reflect the bandwidth [2]. The proposed prediction-based metric, PPTT, is the sum of delay estimation on each link along the routing path, which further consists of packet service time and queueing delay. In the PPTT scheme, when we model the impact of the interfering traffics and link condition, we differentiate them by the channel/radio they are using. In this way, we can avoid selecting same channel for adjacent nodes which may lead to bad link quality. By explicitly considering the radio characteristic, PPTT is also a unified metric for both single-radio and multi-radio networks. PPTT routing scheme works

in a source routing manner, the source node calculates each route's PPTT and select a route with minimal PPTT for RTC flow.

The remainder of this paper is organized as follows. Section II introduces related schemes for RTC in wireless multi-hop networks. In Section III, we illustrate why we need a new routing metric for RTC. The details of our proposed new traffic-aware routing metric and the PPTT routing scheme are described in Section IV, as well as the prediction of the packet service time. In Section V, we discuss the issues of implementation of PPTT. Section VI demonstrates the effectiveness of our proposed scheme with experiment results both in real test bed and in network simulator. Finally, we draw some conclusions and future perspective in Section VII.

II. Related Work

To support RTC traffic in wireless multi-hop networks, in the literature there are numerous related proposals working on different layers. We mainly focus on the solutions in network layers.

Several QoS routing schemes [6], [7] have been proposed for wireless multi-hop networks. They can be classified into reservation-oriented and reservation-less approaches. Reservation-less approaches [8], [9] adopt similar idea with DiffServ framework for Internet to offer a soft-QoS guarantee by serving different flows with different service classes on each node. These approaches, however, are in fact another type of higher-level scheduling mechanisms similar to MAC-based schemes, so they can not provide QoS guidance such as delay or bandwidth budget to the underlying QoS-aware MAC in multi-hop networks. On the other hand, reservation-oriented approaches intend to offer a hard QoS guarantee by reserving resources for each flow on every node, such as [10], [11], [12], [13] and [14]. To calculate the available resource for reservation, [15] proposed a formula to estimate the available bandwidth, and [16] estimates link according to the prediction of existing traffics and location of mobile nodes. In summary, all these reservation-oriented protocols use a metric, such as available bandwidth and end-to-end delay to estimate the path quality. However, none of them takes self-traffic interference into account. Without considering self-traffic effect, all the above mentioned solutions can not offer an accurate prediction of path quality for the coming RTC traffic.

As mentioned above, routing metric is one of the key issues for QoS routing, there are some studies focusing on routing metric. Most traditional ad-hoc routing schemes [17], [18], [19], [20] adopt a simple metric named *hop-count* to find a shortest path from sender to receiver, but recent research results [21], [22] show that this metric may work well in mobile scenarios where topology changes dynamically, it results in poor performance in stationary mesh networks because it does not consider the link quality which may lead to select long (in terms of distance) but error-prone routes. Recently, researchers proposed some link quality metrics such as "Per-hop Round Trip Time" (RTT) [23], "Per-hop Packet Pair Delay" (PktPair)

[24], "Expected Transmission Count" (ETX) [22], and "Weighted Cumulative Expected Transmission Time" (WCETT) [2], signal strength or signal-to-noise-ratio (SNR) [25], [26] to select path with good quality such as high bandwidth, low loss ratio, and short transmission time, in order to achieve higher network capacity.

Multi-radio networks have drawn a great portion of people's attention in recent years. [30] proposed a multi-interface supported routing protocol, which can used in multi-radio networks. It uses *hop-count* as routing metric. In [2], Richard Draves *et al.* proposed a link-quality path metric called Weighted Cumulative Expected Transmission Time (WCETT) for multi-radio wireless network, in which the channel bandwidth and channel diversity are considered. WCETT combines of each link's Expected Transmission Time (ETT) that explicityly accounts for interference among links that use the same channel, thus having good performance in multi-radio networks.

All these link-quality routing metrics, however, use a measurement-based scheme to probe the link conditions based on the current wireless condition and the existing traffics in the network, and so they do not take an explicit consideration on the especially self-traffic, and therefore, they can not obtain accurate prediction of link quality due to combination of interference from different traffics after the self-traffic is injected into system.

III. Why a New Routing Metric for RTC?

Much prior research work has been conducted to propose different routing metrics, such as *hop-count*, ETX, RTT, PktPair, WCETT, for wireless multi-hop networks. However, all these existing link-quality routing metrics can not guarantee always to select good path for RTC.

A. Single-radio Scenario

Let us consider a single-radio scenario which is shown in Fig.2. In this scenario each node has one 802.11b radio with fixed 2Mbps link rate. Suppose there is a flow from node 6 to 0 with 256kbps transmission rate (flow-1). Now we inject another 384kbps RTC flow from node 3 to 4 (flow-2). There are two possible paths for flow-2. More specifically, path-1 is along nodes set (3, 5, 1, 4), where node 5 and node 1 are in the interference range of node 7 and node 8. path-2 is along nodes set (3, 10, 2, 9, 4), where no node along this path is interfered by flow-1.

We use ETX metric to illustrate why we need a new routing metric since recent report pointed out that ETX can achieve rather good performance in single-radio scenario. The ETX metric measures the expected number of transmissions, including retransmissions, needed to send a unicast packet across a link. To derive ETX, each node broadcasts one probe packet every second. Its neighbors then calculate

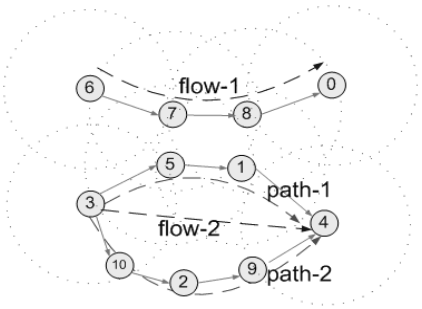


Fig. 2. A scenario to show how self-traffic affects the routing selection. The dotted circles stand for the interference range.

the loss rate of the probes on the links. Let p denote the probability that packet transmission fails. ETX of this link then can be calculated with the following formula.

$$ETX = \frac{1}{1-p} \tag{1}$$

The path metric is the sum of the ETX value of each link in the path. The routing protocol then selects the path with minimal ETX. Specifically in this case, the measurement shows that the packet loss rate of the three links of path-1 is 13%, 23%, 14%, respectively, due to neighboring traffic 6 to 0, and the packet loss rate of all links of path-2 is 0. Hence the ETX of path-1 is 3.61 (= 1/(1-13%) + 1/ (1-23%) + 1/(1-13%)), while ETX of path-2 is 4 (=1/(1-0) + 1/(1-0) + 1/(1-0) + 1/(1-0)). Therefore, ETX will select path-1 rather than path-2 for new coming RTC flow. However, if this RTC flow is delivered through path-1, it will encounter higher contention and loss probability due to the joint inference from flow-1 and flow-2 itself. So contrary to ETX's selection, the performance of path-1 is worse than that of path-2 in terms of goodput to serve the RTC traffic: its goodput along path-1 is 318kbps, while the goodput along path-2 is 384kbps. This phenomenon is just because that ETX does not consider self-traffic interference. ETX selects path-1 according to the probing results. However, the self-traffic interference can not be neglected, after flow-2 is injected into path-1, the joint impact cause that performance of path-1 is worse than path-2. This weakness is universal for measurement based routing scheme, since they can not measure the self-traffic interference before the coming traffic is injected. Thus the measurement based routing schemes can not reflect path quality accurately.

B. Multi-radio Scenario

In multi-radio network, the impact of self-traffic is even more severe than it in single-radio network. In single-radio networks, the self-traffic will affect routing selection with neighboring traffic jointly. If there is no neighboring traffic, although self-traffic affect the path performance, it will not cause mistake of routing selection. However, in multi-radio networks, even there is no neighboring traffic, it may cause wrong routing selection if self-traffic is not taken into account. Consider the scenario shown in Fig.3,

there are two candidate multi-radio path for a flow from node 1 to node 10. We vary the traffic rate of the flow and deliver it through these two paths respectivley. The delay and goodput of these two paths are compared in Fig.4. When offered load is light, the self-traffic impact is not distinct so that path-1 and path-2 achieve similar performance. When offered load is heavy, the impact of self-traffic emerges. Since there is a hidden terminal in path-1 while no hidden terminal in path-2, the performance of path-1 is worse than path-2. This simple example shows that the self-traffic should be considered to select high performance path for RTC in multi-radio network.

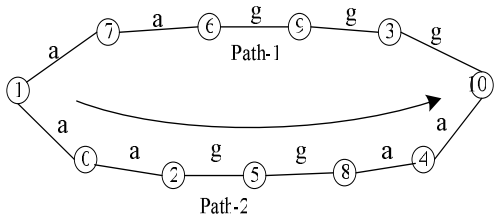


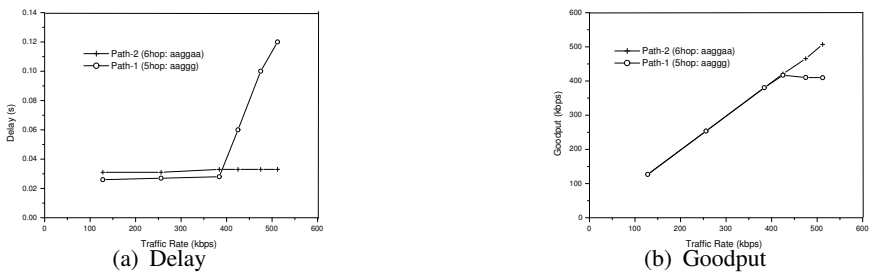
Fig. 3. Multi-radio scenario. Each node has two radio (802.11a and 802.11g).

From the above analysis for the single-radio and multi-radio scenarios, we can conclude: a prediction-based routing metric which considers self-traffic interference explicitly and can predict the whole path performance is required.

IV. Traffic-aware routing metric: PPTT

A. Basic idea

In this section, we present a new prediction-based routing metric to explicitly consider different types of interfering traffics, i.e. neighboring traffic and self-traffic, that interfere with the requested flow. We can classify these interfering traffics (both neighboring traffic and self-traffic) as Carrier Sensing (CS) and Hidden Terminal (HT) traffic according to their relative positions. CS traffic is the cumulative traffic of all nodes that are in the carrier sensing range of link's sender (traffic sender on this link). In 802.11 wireless networks, when sender wants to transmit a packet across the link, it will compete with the nodes in CS range for channel access. Thus larger volume of CS traffic leads to longer channel access time. HT



(a) Delay (b) Goodput

Fig. 4. Delay and goodput comparison of path-1 and path-2.

traffic is the cumulative traffic of all nodes that are in the carrier sensing range of link's receiver (traffic receiver on this link) but not in the carrier sensing range of link's sender. A packet transmitted across the link may collide with the packets from hidden terminals. Thus larger volume of HT traffic causes more packet collisions, which results in longer retransmission time. Considering the different impact of CS and HT traffic to link quality, we need differentiate them by their locations. We use a 25-node grid-topology network (Fig.5) to illustrate the differentiation of interfering traffic (in this topology, we set the interference range and transmission range of all nodes equal). There are four flows in the network. Flow a to e is delivered via link (a, b), (b, c), (c, d) and (d, e). It has two neighboring flows, f→g and j→m, since nodes f and j are in nodes b's and c's carrier sensing range, respectively. Considering link (b, c), CS traffic of this link includes flows a→b, c→d, and f→g, since they are all in the carrier sensing range of b; while its HT traffic includes flows d→e and j→k, since they are in receiver c's carrier sensing range but out of sender b's carrier sensing range.

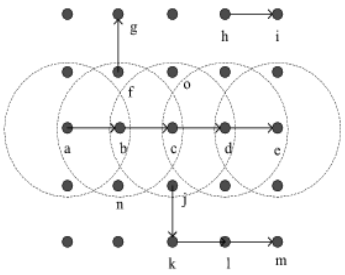


Fig. 5. 25-node grid topology. Each node contributes CS traffic or HT traffic according to its relative position in the network.

With HT traffic and CS traffic concepts, we can further study the impact of self-traffic interference which is in two fold. Firstly, the self-traffic will enlarge CS traffic volume of each link in its delivery path. For instance, traffic of link (a, b) and (c, d) are the increment of CS traffic of link (b, c). As a result, self-traffic interference elongates the channel access time. Secondly, self-traffic will enlarge HT traffic volume of some links in this path. For instance, traffic of link (d, e) is the increment of HT traffic of link (b, c). As a result, self-traffic interference causes more packet collisions, which in turn elongates packet retransmission time.

In summary, the interfering traffic, no matter neighboring traffic or self-traffic, act as either CS traffic or HT traffic to impact the link quality, which results in the increment of packet transmission time. Therefore, by considering traffic interference, we propose a new time-based routing metric, named Path Predicted Transmission Time (PPTT), to explicitly account for different types of traffic interference. PPTT tries to predict the end-to-end delay after the traffic started to be delivered along the path.

To calculate PPTT, we predict packet transmission time link by link, which we name as Link Predicted Transmission Time (LPTT). Then summing LPTT of each link, we get PPTT of whole path. LPTT is

defined as the time from the instant the packet enters the queue of link's sender to the instant it successfully reach receiver or be dropped, which comprises queueing delay and packet service time. The packet service time is the time span that MAC layer costs to send out the packet and we calculate it based on the analysis of the 802.11 MAC behavior.

In the following, we will describe how to calculate LPTT and PPTT in detail, as well as the packet service time for RTC traffic. Like other link-quality routing metrics, our proposed path metric is also the sum of individual link quality, so any routing protocols which use PPTT metric will select the path with minimum path metric and in return they can get highest-quality path.

B. Path Predicted Transmission Time (PPTT) for 802.11

The Path Predicted Transmission Time (PPTT) can be calculated by summing the packet transmission time link by link (LPTT). The packet transmission time on the certain link is related to the existing carrier-sense (CS) traffic, hidden-terminal (HT) traffic and its self-traffic. Thus LPTT can be represented as $LPTT(\lambda_{cs}, \lambda_{ht}, \lambda)$, where λ is the traffic rate of this RTC flow, λ_{cs} and λ_{ht} denote the average CS traffic rate and HT traffic rate, respectively. The existing CS traffic and HT traffic of neighboring interfering traffic can be obtained by exchanging traffic information between neighboring nodes (We will discuss it in Section V.). However, to calculate LPTT, the CS and HT traffic obtained by exchanging the existing traffic information is not sufficient. The potential CS and HT traffic caused by self-traffic interference must be added with the current CS/HT traffic together, therefore we can use $LPTT(\lambda_{cs}, \lambda_{ht}, \lambda)$ to compute the accurate PPTT. We'll describe how to estimate LPTT with given $(\lambda_{cs}, \lambda_{ht}, \lambda)$ in Subsection IV.C. In this subsection, we focus on calculating $(\lambda_{cs}, \lambda_{ht}, \lambda)$, namely the potential CS traffic and HT traffic by considering the impact of self-traffic in order to estimate the PPTT.

As mentioned above, CS traffic and HT traffic consist of two parts, namely neighboring traffic and self-traffic, so we need measure these two kind of traffic. The traffic from neighbors can be measured along the corresponding routing path, but it is impossible to measure the self-traffic. To estimate the impact of self-traffic, we begin by characterizing the effect of self-traffic to each link of the path because the self-traffic also act as CS or HT traffics to other links according to the link positions. We use two parameters to represent the self-traffic interference, namely Carrier Sensing Factor and Hidden Terminal Factor. Carrier Sensing Factor (CSF) of a link indicates the effect that the self-traffic act as CS traffic, and it can be estimated by the number of links in the path, which are on the same channel and in sender's CS range. Hidden Terminal Factor(HTF) of a link denotes the effect that the self-traffic act as HT traffic, and it can be measured by the number of links which are on the same channel along the path and in receiver's CS range but not in sender's range.

(a) Single-radio (b) Multi-radio

Fig. 6. Illustration of CSF and HTF.

For a link from node i to j, the increased CS traffic due to self-traffic interference is $CSF_{ij} \cdot \lambda$; and the increased HT traffic due to self-traffic interference is $HTF_{ij} \cdot \lambda$, where λ is the traffic rate along the path. Thus for this link (i, j), after considering the self-traffic, the CS traffic becomes $\lambda_{cs} + CSF_{ij} \cdot \lambda$, and HT traffic becomes $\lambda_{ht} + HTF_{ij} \cdot \lambda$. Consequently, the LPTT of link (i, j) can be denoted as $LPTT(\lambda_{cs} + CSF_{ij} \cdot \lambda, \lambda_{ht} + HTF_{ij} \cdot \lambda, \lambda)$. So the next problem is how to calculate the CSF and HTF.

The CSF and HTF of a link are determined by link position and channel distribution of the path. Let's illustrate it with an example. Considering a 4-hop path in Fig.6. We first consider single-radio scenario (see Fig.6(a)). Since all node use the same channel, i.e., 802.11a, two adjacent links interfere each other and two-hop away node becomes hidden terminal. Taking link (b, c) as an example, it has two CS links (a, b) and (c, d) and one HT link (d, e) in the path, thus CSF_{bc} is 2 and HTF_{bc} is 1. CSF and HTF of other links can be calculated similarly.

Next we consider multi-radio scenario (see Fig. 6(b)). Each node has two radios, which can operate in 802.11a or 802.11g mode. Link (a, b) and (b, c) use 802.11a radio, and link (c, d) and (d, e) use 802.11g radio. Since links on 802.11a mode and 802.11g do not interfere with each other, link (b, c) is only interfered by the self-traffic of link (a, b). Thus we get value of CSF_{bc} (1) and HTF_{bc} (0) for link (b, c), which are smaller than that in single-radio case. Therefore, we can see that the channel diversity can be reflected by CSF and HTF of each link. So our proposed PPTT metric offers a unified way to take channel diversity into account. It can select path with larger channel diversity which has smaller CSF and HTF, resulting in better performance.

In summary, for a n-hop path, we can obtain CSF and HTF of each link according to its position in the path in order to calculate the LPTT. Summing the LPTT of each link, we get the predicted transmission time of the whole path as follows

$$PPTT(\lambda) = \sum_{i=1}^{n} LPTT_i(\lambda_{cs}(i, i+1) + CSF_{i,i+1} \cdot \lambda, \lambda_{ht}(i, i+1) + HTF_{i,i+1} \cdot \lambda, \lambda) \tag{2}$$

where λ is the average traffic rate of this RTC traffic.

C. Link Predicted Transmission Time (LPTT) for 802.11

Link Predicted Transmission Time (LPTT) is the estimation of packet transmission time over the certain link. As aforementioned, LPTT is related to CS and HT traffics which has been estimated in Subsection

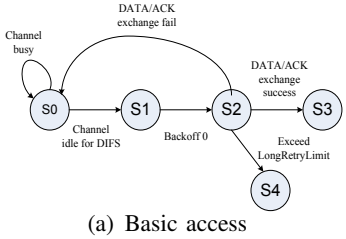
IV.B, so in this subsection, we'll focus how to calculate the LPTT with given CS and HT traffics.

LPTT consists of two parts, namely queueing delay and MAC layer processing time. As for queueing delay, it denotes the average wait time with which the packet enters the sender's sending queue and wait to be sent out when this packet needs to be transmit over a link. As for MAC layer processing time, it denotes the average service time at the MAC layer when the packet departs the queue and is handled by MAC protocol. We also name it *packet service time*. Considering the traffic pattern of the RTC traffic, the queueing delay can be calculated according to a certain queueing model, e.g. M/M/1 model. The challenge of calculating LPTT is then how to calculate packet service time.

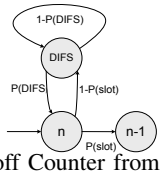
To get an accurate estimation of MAC layer processing time for the coming RTC flow and take self-traffic interference into account, we introduce a new MAC model for RTC derived from IEEE 802.11 protocol, in which we estimate the packet collision probability according to the given and self-traffic. This is an enhanced model comparing with prior work [28][29] which also studied the MAC layer processing time of IEEE 802.11 protocol. They all assume that every node in the network has the same packet collision probability. This is indeed the case in saturated networks. However, RTC traffic with proper rate control mechanism will not saturate the networks. Thus each node will have different packet collision probability which is not considered in previous works. A packet may be collided by hidden terminal (HT) traffic, which includes both existing neighboring traffic and coming self-traffic, which means the packet collision probability can reflect the impact of both neighboring traffic and self-traffic.

Packet service time is calculated according to the specific MAC (802.11) behavior and neighboring traffic condition. There are two access methods that are used under 802.11 Distributed Coordination Function (DCF) mode, namely basic access method and RTS/CTS access method. Basic access method uses only DATA and ACK packets. In the RTS/CTS access method, two more packets, RTS/CTS, are exchanged before transmitting DATA packets. In this subsection we show how to calculate the packet service time for the basic access method. The calculation of for RTS/CTS access method is similar.

Before proceeding further we first introduce some necessary notations and assumptions. Considering a link from host i to j, we define *Contention Link* and *Hidden Link* as follows. We call a link as *Contention Link* of link (i, j) if the link's sender is in the carrier sensing range of node i. And we call a link as *Hidden Link* if the link' sender is the hidden terminal of node i. Let CL_{ij} be the set of contention links. If node i wants to send a packet, it will compete with links in CL_{ij} to access the channel. Similarly, let HL_{ij} denote the set of hidden links. The packet sent from i to j may be collided with the packets sent along the links in HL_{ij}. According to 802.11 specification, we use $DIFS$, $SIFS$, $EIFS$ and $slot$ to denote the time interval of DCF Inter-Frame Space, Short Inter-Frame Space, Extended Inter-Frame Space and a slot[27], respectively. And we use S_{D}, S_{A}, S_{R} and S_{C} to denote the size of DATA, ACK,



(a) Basic access



(b) Backoff Counter from *n* to *n-1*

Fig. 7. State transition of IEEE 802.11 MAC in basci access method

RTS and CTS packet. Let B_{ij} and CH_{ij} to denote the bandwidth and channel number of link (i, j). In 802.11 network, RTS, CTS and ACK packet are transmitted at basic rate, we use B_{basic} to denote it. And we use notations ACK, RTS and CTS for the required time periods of transmitting ACK, RTC and CTS packets, respectively. In addition, we use $DATA_{ij}$ to denote DATA packet transmission time along link (i, j). We assume that all the nodes in the network send packets exponentially. It should be noticed that in fact, our scheme is independent of traffic model. Here we use Poisson traffic model just for the simplicity of implementation. We can use other traffic model, e.g. MMPP (Markov Modulated Poisson Process) model in the scheme and we give the derivation of LPTT based on MMPP model in appendix. Moreover, despite the potential inaccuracy of Poisson approximation, as presented in the Section VI, the estimation of PPTT is reasonably accurate and we can get rather good performance in both test bed experiments and NS-2 simulations. We use λ_{kl} to denote packet transmitting rate along a certain link (k, l).

Fig.7(a) shows the simplified state transmit diagram of basic access method for transmitting a DATA packet from node i to node j. When node i wants to send packet to node j, it first enters to state S0. In this state, node i senses the channel, if the channel is idle for DIFS period, it enters state S1, after delaying a random backoff time interval until its backoff counter becomes 0, then enters state S2. Otherwise, if the channel is busy in DIFS period, it goes back to state S0. In state S2, node i sends a DATA packet to node j, if the DATA/ACK pair exchange successfully, it will enter state S3. If the exchange is failed, it will return to state S0. If the retransmission times exceed LongRetryLimit (LRL), it transits to state S4, and drop this packet. Thus the average transition time from S0 to S3 and S4 is the service time of each packet.

For convenience, let P^i_{DIFS} and P^i_{slot} denote the probabilities of node i sensing channel idle for time interval DIFS and slot, respectively. And P^i_{DATA} is used to denote the probability of node i successfully sending DATA packet.

There are two possible cases that node i senses channel busy in DIFS period which are shown in Fig.8.

Assume node i starts to sense channel at time t. First, if a neighbor node k sends packet in $DATA$ period before t, the transmission will not be finished at t, thus node i will sense channel busy. Second, if a neighbor node m starts to send packet in $DIFS$ period after t, node i also senses channel busy.

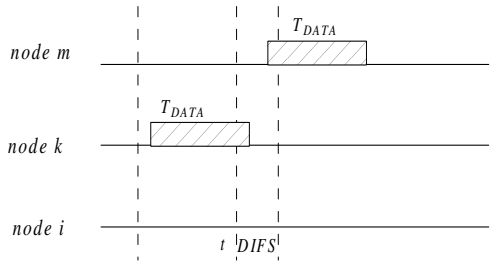


Fig. 8. Two possible cases cause channel busy.

Therefore P^i_{DIFS} is equivalent to the probability that no link of CL_{ij} transmitting packets in the time interval of $DATA + DIFS$. For a certain link (k, l), the probability of no packet transmitting along it in $DATA + DIFS$ is $\exp[-(\frac{S_{\mathrm{D}}}{B_{kl}} + DIFS) \cdot \lambda_{kl}]$.

Thus we get the following equations.

$$P^i_{\mathrm{DIFS}} = \prod_{(k,l)\in CL_{ij}} \exp[-(\frac{S_{\mathrm{D}}}{B_{kl}} + DIFS) \cdot \lambda_{kl}] = \exp[-(S_{\mathrm{D}} \cdot \sum_{(k,l)\in CL_{ij}} \frac{\lambda_{kl}}{B_{kl}} + DIFS \cdot \sum_{(k,l)\in CL_{ij}} \lambda_{kl})] \quad (3)$$

Here $\sum_{(k,l)\in CL_{ij}} \lambda_{kl}$ represents the CS traffic of link (i, j), namely,

$$\lambda_{\mathrm{cs}}(i, j) = \sum_{(k,l)\in CL_{ij}} \lambda_{kl} \quad (4)$$

Then we normalize the CS traffic with the data rate of the link as follows.

$$\lambda^{\mathrm{norm}}_{\mathrm{cs}}(i, j) = \sum_{(k,l)\in CL_{ij}} \frac{\lambda_{kl}}{B_{kl}} \quad (5)$$

We name $\lambda^{\mathrm{norm}}_{\mathrm{cs}}(i, j)$ as *nomalized CS traffic* of link (i, j).

Similarly, we get the following equation for P^i_{slot}.

$$P^i_{\mathrm{slot}} = \exp[-slot \cdot \lambda_{\mathrm{cs}}(i, j)] \quad (6)$$

Two possible cases causing DATA packet collision are shown in Fig.9. The probability P^i_{DATA} of DATA packet successful transmission is equivalent to the probability that no link of HL_{ij} transmits packet in the same time interval of packets transmission which can be denoted as follows.

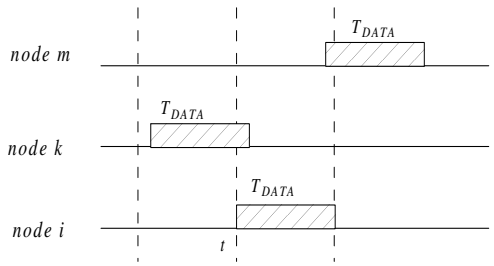


Fig. 9. Two possible cases cause DATA collision.

$$P^i_{\mathrm{DATA}} = \prod_{(k,l)\in HL_{ij}} \exp[-(\frac{S_{\mathrm{D}}}{B_{kl}} + \frac{S_{\mathrm{D}}}{B_{ij}}) \cdot \lambda_{kl})] = \exp[-(S_{\mathrm{D}} \cdot \sum_{(k,l)\in HL_{ij}} \frac{\lambda_{kl}}{B_{kl}} + \frac{S_{\mathrm{D}}}{B_{ij}} \cdot \sum_{(k,l)\in HL_{ij}} \lambda_{kl})] \quad (7)$$

Here $\sum_{(k,l)\in HL_{ij}} \lambda_{kl}$ represents the HT traffic of link (i,j), namely,

$$\lambda_{\mathrm{ht}}(i,j) = \sum_{(k,l)\in HL_{ij}} \lambda_{kl} \quad (8)$$

Similarly, we normalize it with the data rate of the link as follows.

$$\lambda^{\mathrm{norm}}_{\mathrm{ht}}(i,j) = \sum_{(k,l)\in HL_{ij}} \frac{\lambda_{kl}}{B_{kl}} \quad (9)$$

We name $\lambda^{\mathrm{norm}}_{\mathrm{ht}}(i,j)$ as *nomalized HT traffic* of link (i,j).

To calculate the packet service time, we need consider the kth retransmission of the packet from node i to j. First, the node waits to ensure that the medium is idle for DIFS period of time. This costs $\frac{DIFS}{P^i_{DIFS}}$ period of time. The backoff counter then selects a random number of backoff slots. Fig.7(b) shows the state transit diagram of backoff counter.

At the state in which the backoff counter is n, if the channel is idle in the slot, it moves to next state in which the backoff counter is decreased by 1. If there are transmissions by other stations during the slot, the station freezes its backoff counter and resumes the count where it leaves off, after DIFS interval in which channel is idle. Thus, considering the freezing of backoff counter, the expected time duration of one backoff slot is given by the following equation.

$$\tau = \frac{slot}{P^i_{slot}} + \frac{1 - P^i_{slot}}{P^i_{slot}} \cdot \frac{DIFS}{P^i_{DIFS}} \quad (10)$$

In 802.11 MAC protocol, the value of backoff counter is chosen between 0 and contention window CW randomly. CW is an integer between CW_{min} and CW_{max} with typical values being 31 and 1023, respectively. Initially, CW is equal to CW_{min}. Upon an unsuccessful transmission, CW is doubled, until it reaches CW_{max}. After a successful transmission, CW is again set to CW_{min}. Thus the average number of backoff slots at the kth retransmission is $\frac{CW_{min}}{2} \cdot 2^{k-1}$.

If the transmission of DATA packet fails after the kth attempt, the time cost is denoted as follows.

$$t^f_k = \frac{DIFS}{P^i_{DIFS}} + \frac{CW_{min}}{2} \cdot 2^{k-1} \cdot \tau + DATA_{ij} + EIFS \quad (11)$$

If DATA packet is transmitted successfully at the kth attempt, the total time t^s_k is as follows.

$$t^s_k = \frac{DIFS}{P^i_{DIFS}} + \frac{CW_{min}}{2} \cdot 2^{k-1} \cdot \tau + DATA_{ij} + SIFS + ACK \quad (12)$$

The probability that the kth retransmission is successful can be denoted as: $P^i_{DATA} \cdot (1 - P^i_{DATA})^{k-1}$.

Then the following equation gives the average packet service time of basic access method.

$$T_{MAC} = \sum_{k=1}^{LRL} P_{DATA}^{i}(1 - P_{DATA}^{i})^{k-1}(\sum_{i=1}^{k-1} t_i^f + t_k^s) + (1 - P_{DATA}^{i})^{LRL} \sum_{k=1}^{LRL} t_k^f \tag{13}$$

Now we turn to obtain the queueing delay, where we use a simple M/M/1 queueing model to illustrate the procedure. Let λ denote the packet transmission rate over link (i, j). The packet service rate is μ, where $\mu = 1/T_{MAC}$.

The queueing delay is denoted by the following equation.

$$T_{queue} = \frac{\lambda/\mu}{\mu - \lambda} \tag{14}$$

Thus, substituting μ into equation (14), we get the following equation.

$$T_{queue} = \frac{\lambda T_{MAC}^2}{1 - \lambda T_{MAC}} \tag{15}$$

Finally, summing the queueing delay and packet service time, we get the formula for LPTT as follows.

$$LPTT(\lambda_{cs}, \lambda_{ht}, \lambda) = T_{queue} + T_{MAC} = \frac{T_{MAC}}{1 - \lambda T_{MAC}} \tag{16}$$

We can see that with equation (2), (13) and (16), we can calculate the transmission time along the path with the estimated effects from neighboring traffic and self-traffic.

V. IMPLEMENTATION

In this section, we discuss the implementation of routing scheme with the new traffic-aware routing metric, Path Predicted Transmission Time (PPTT).

We have implemented our PPTT metric in an ad-hoc routing framework called Mesh Connectivity Layer (MCL)[3]. MCL is a loadable Windows driver and it is implemented as 2.5 layer protocol. We introduced a link Info exchange scheme to exchange neighboring- and self-traffic information. The data structure of Link Info Message is shown in Fig.10(a). There are five fields in Link Info Message. “Link Target Address” is a network address of the node where the link ends. In other words, a Link Info Message contains some informations of the link, which is from the node sending the Link Info Message to the “Link Target Address” node. In our implementation, we have considered the compability for both single-radio and multi-radio network. In multi-radio network, different link may use different channel and have different link bandwidth. We use “Channel Number” and “Link BW” field to deliver the channel number which the link used and link bandwidth respectively. “Tx Pkt Rate” is the packet sending rate along this link, and “Rx Pkt Rate” is the packet receiving rate along this link.

Each node maintains two tables, “Local Link Info Table” (LLI Table) and “Neighboring Link Info Table” (NLI Table). We use a sample network shown in Fig.11 to illustrate these two tables. LLI Table

(a) Link Info Msg (b) RREP Msg

Fig. 10. Message format in PPTT routing scheme

contains each link starting from the local node. For example, in the transmission range of node i, there are node h, j, a_1, a_2, a_3, thus LLI Table of node i looks like table I. Each node periodically broadcast a Link Info Message which contains all of its local links.

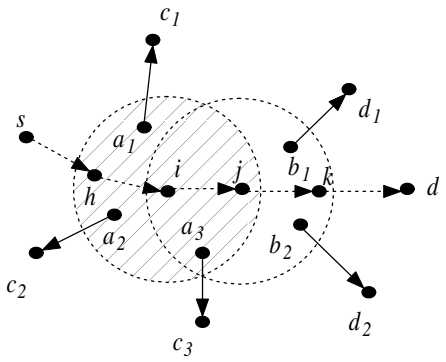


Fig. 11. Sample Network to illustrate LLI and NLI table

TABLE I

LOCAL LINK INFO TABLE

Link	Bandwidth	Channel	Tx Pkt Rate	Rx Pkt Rate
(i,j)	B_{ij}	CH_{ij}	$\lambda_{TX}(i,j)$	$\lambda_{RX}(i,j)$
(i,h)	B_{ih}	CH_{ih}	$\lambda_{TX}(i,h)$	$\lambda_{RX}(i,h)$
(i,a_1)	B_{ia_1}	CH_{ia_1}	$\lambda_{TX}(i,a_1)$	$\lambda_{RX}(i,a_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Each node can receive its neighbors' Link Info message, by gathering these Link Info Messages, NLI table can be constructed. For node i, it can receive Link Info of node a_1, a_2, a_3, h, j. Thus, NLI table of node i looks like table II. Based on this table, we can calculate CS traffic and HT traffic of a link.

When a source node wants to send packets to destination node, it generates and broadcasts RREQ. When destination node receives RREQ, it generates and send RREP to source node. The RREP contains link info of each hop in the path. The data structure of RREP is shown in Fig.10(b). "Address n" is the address of nth node in the path, other fields are the traffic information of link $(n, n+1)$. When RREP arrives at node n, node n calculates $\lambda_{\rm cs}(n,n+1)$, $\lambda_{\rm cs}^{\rm norm}(n,n+1)$, $\lambda_{\rm ht}(n,n+1)$ and $\lambda_{\rm ht}^{\rm norm}(n,n+1)$, then fill these parameters into RREP.

TABLE II

NEIGHBORING LINK INFO TABLE

Link	Bandwidth	Channel	Tx Pkt Rate	Rx Pkt Rate
(j,k)	B_{jk}	CH_{jk}	$\lambda_{TX}(j,k)$	$\lambda_{RX}(j,k)$
(h,s)	B_{hs}	CH_{hs}	$\lambda_{TX}(h,s)$	$\lambda_{RX}(h,s)$
(a_1,c_1)	$B_{a_1c_1}$	$CH_{a_1c_1}$	$\lambda_{TX}(a_1,c_1)$	$\lambda_{RX}(a_1,c_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\lambda_{\text{cs}}(n,n+1)$ and $\lambda_{\text{cs}}^{\text{norm}}(n,n+1)$ are calculated according to NLI tables. Since all links in NLI table are contention links of node n, they can be calculate as following

$$\lambda_{\text{cs}}(n,n+1) = \sum_{\substack{(i,j)\in NLI_n,\\ CH_{ij}=CH_{n,n+1}}} \lambda_{TX}(i,j) \tag{17}$$

$$\lambda_{\text{cs}}^{\text{norm}}(n,n+1) = \sum_{\substack{(i,j)\in NLI_n,\\ CH_{ij}=CH_{n,n+1}}} \frac{\lambda_{TX}(i,j)}{B_{ij}} \tag{18}$$

where NLI_n is the set of links in NLI table of node n.

$\lambda_{ht}(n,n+1)$ is the HT traffic rate of link $(n,n+1)$. It is the sum of traffic rate of nodes which are neighbor of $n+1$ but not n. Since when RREP arrives at node n, it already contains $\lambda_{cs}(n+1,n+2)$. Therefore, by eliminating the traffic rate of nodes which are neighbor of both n and $n+1$, we can get $\lambda_{ht}(n,n+1)$. By lookup NLI table of node n, we can know which node is neighbor of node $n+1$, since NLI table contains the link info of node $n+1$. Thus, we can know which node is neighbor of both n and $n+1$. We use A_n and A_{n+1} to denote the set of neighbor nodes of node n and $n+1$ respectively. Hence we can calculate $\lambda_{ht}(n,n+1)$ as follows

$$\lambda_{ht}(n,n+1) = \lambda_{cs}(n+1,n+2) - \sum_{\substack{i\in A_n\cap A_{n+1},\\ CH_{ij}=CHn,n+1}} \lambda_{TX}(i,j) \tag{19}$$

Similarly, $\lambda_{\text{ht}}^{\text{norm}}(n,n+1)$ can be calculated as following

$$\lambda_{ht}^{\text{norm}}(n,n+1) = \lambda_{cs}^{\text{norm}}(n+1,n+2) - \sum_{\substack{i\in A_n\cap A_{n+1},\\ CH_{ij}=CHn,n+1}} \frac{\lambda_{TX}(i,j)}{B_{ij}} \tag{20}$$

When source node receives RREP, it get the CS and HT traffic information of each link in the path. In a multi-radio network, we also get the channel distribution of the whole path. We can calculate CSF and HTF of each link according to link position and channel distribution of the path. In the calculation of λ_{cs} and λ_{ht}, we only calculate those links which are on the same channel, since the links on different channel

do not intefere each others. Thus PPTT can be calculated based on this traffic information. Source node selects min-PPTT path for the coming traffic flow.

Since PPTT needs nodes exchange traffic information between each other, it causes some overhead. In our implementation, each node broadcast Link Info message periodically with the same frequency as that in WCETT. The size of each item in the Link Info message is 12 bytes and the complexity of broadcast is $O(n)$. Thus we can estimate PPTT without incurring too much overhead.

VI. Performance Evaluation

We evaluate the performance of PPTT scheme in both single-radio and multi-radio scenarios. In this section, we describe the results of our experiments on a real testbed and simulations by *ns-2*. First, we illustrate the accuracy of PPTT by simulation. Then, we present experiment results of PPTT scheme in our testbed and compare the performance of PPTT to ETX in single-radio scenario. We evaluate PPTT in random topology through extensive simulations. Finally, we study the performance of PPTT in multi-radio scenario by both experiments in testbed and simulations.

A. Accuracy of PPTT

To verify the accuracy of PPTT, we conducted the following simulations in fixed topology and random topology. The fixed topology is shown in Fig.12. There are two paths connected 1 to 6 and 7 to 11, respectively. The transmission range and interference range are equal. Node 8, 9 and 10 are interfered by the path 1 to 6. Each node has an 802.11b radio with 2Mbps bandwidth.

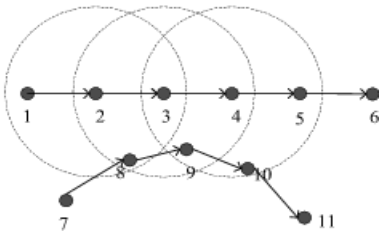


Fig. 12. Two path interfered each other. Flow from node 1 to node 6 is background traffic, and we study the end-to-end delay of traffic from node 7 to node 11 by prediction and measurement, respectively.

We first only consider the self-traffic interference. We inject a flow from 7 to 11. There is no other neighboring traffic. Thus only self-traffic will impact the end-to-end path delay. We vary the traffic sending rate, and compare the end-to-end path delay with the computed PPTT. Fig.13 shows the comparison. It is shown that PPTT is well matched with the real end-to-end delay. We can also observe that as the traffic rate increases, the end-to-end delay increases exponentially. This is because that as traffic rate is increased, the packet collision probability is increased, which leads to backoff time increased exponentially.

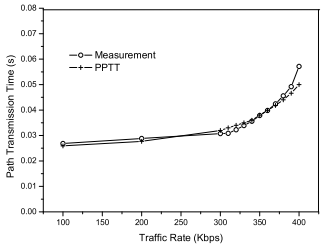


Fig. 13. PPTT without nbr traffic

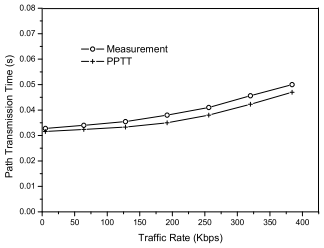


Fig. 14. PPTT with nbr traffic

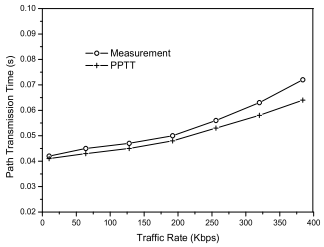


Fig. 15. PPTT in random topology

Secondly, we set the interfering traffic from 1 to 6 as 128kbps, and we also vary the traffic rate from 7 to 11 and compare the prediction of path transmission time with the end-to-end path delay measured from the test bed. Fig.14 shows that PPTT is also well matched with real path delay.

Finally, we generate a random topology with 30 nodes in 1000m$\times$1000m rectangular field. We establish 10 CBR connections randomly and choose one of these connections to measure the end-to-end delay. We vary the traffic rate of each connection. Fig.15 shows the comparison of measured end-to-end delay and PPTT. The plot shows that PPTT estimate is accurate for most of data rates, while at very high data rates there is a gap. Despite these inaccuracies, note that in the plot, we are able to unambiguously distinguish between various path qualities.

Thus, the overall conclusion from this experiments is that PPTT produces reasonably accurate estimates of path quality.

B. Testbed

We built a 32-node wireless testbed. Our testbed is located on the same floor of a fairly typical office building. All nodes are placed in fixed locations and did not move during testing. The nodes are all DELL PCs equipped with LinkSys Dual-Band Wireless A+G Adapter and ORiNOCO 802.11abg ComboCard Gold card. In our experiments, each card runs on 802.11a and/or 802.11g radios. We installed MCL driver in which PPTT metric is implemented on each PC in our testbed.

C. Single-radio Scenario

We first conduct the some experiments to illustrate the self-traffic effect and PPTT performance in our testbed. We add two flows to our testbed which are shown in Fig.16, one is from 26 to 10 and the other is from 11 to 21. The flow from 11 to 21 starts earlier than flow from 26 to 10 and acts as background traffic. From node 26 to node 10, there are two paths, path1 (26, 30, 12, 10) and path2 (26, 8, 9, 13, 4, 10). Path1 is interfered by the background traffic. We begin by setting flow 11 to 21 as a fixed traffic rate, 2Mbps and varying the traffic rate from 26 to 10. The goodput comparison of PPTT to ETX is shown

in Fig.17(a). We can see that when traffic rate from 11 to 21 is between 2.2Mbps and 5Mbps, PPTT can achieve larger goodput than ETX; and at other traffic rates, PPTT has the comparable goodput with ETX.

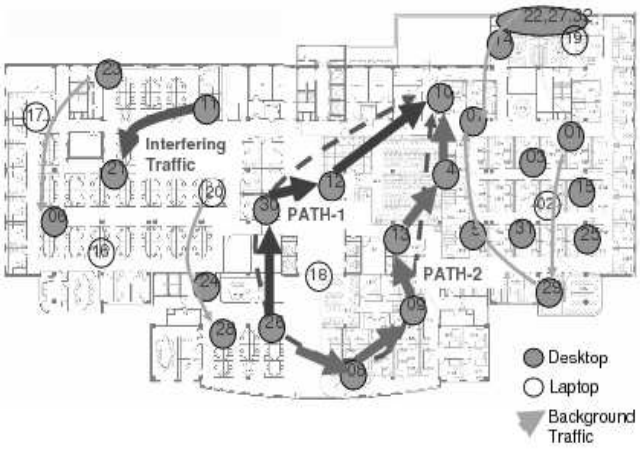


Fig. 16. Our testbed consisting of 32 nodes placed in fixed locations inside an office building.

The better performance of PPTT comes from the consideration of both neighboring traffic interference and self-traffic interference. When traffic rate is less than 2.2Mbps, the self-interference is not serious. Although path1 is interfered by flow 11 to 21, it performs better than path1, since it is shorter. When traffic rate is more than 2.2Mbps, the self-interference can not be ignored. The performance of path2 is better than path1. Since ETX does not take self-interference into account, nor does it attempt to select path2. This is reflected in the fact that the goodput using ETX is lower than PPTT.

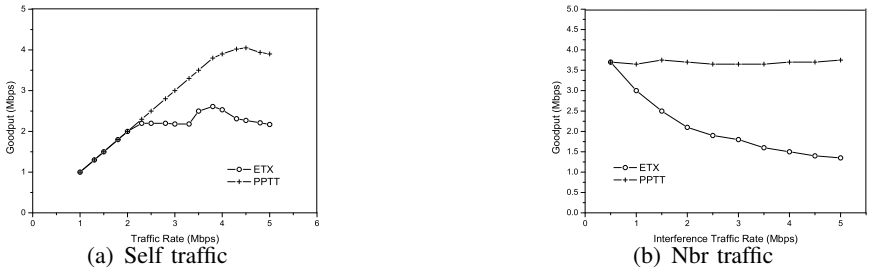
Next, we set traffic rate from 26 to 10 as fixed 3.8Mbps and vary the traffic rate from 11 to 21. The comparison of goodput are shown in Fig.17(b).

The plots show that PPTT outperforms ETX. When background traffic exceeds 512kbps, the cumulative interference of both neighboring traffic and self-traffic leads to worse performance of path 1 than path 2. Since PPTT takes both types of traffic interference into account, it can select better one. This is reflected in the fact that the delay using PPTT is lower than ETX and goodput is better.

The main conclusion from the experiments is that PPTT performs better than ETX. The increase in performance is a result of the fact that PPTT takes both neighboring interference and self-interference into account. This sometimes leads it to select longer path than ETX, however, these longer paths result in better performance.

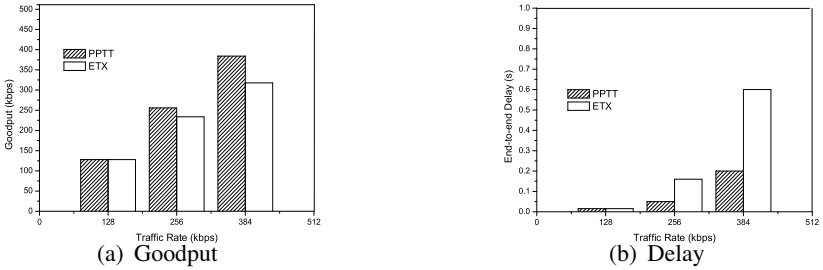
Then we use ns-2 simulator to evaluate PPTT in random topology wireless network. The DCF of IEEE 802.11 is used as the MAC layer protocol. RTS/CTS are disabled. The bandwidth of each node is 2Mbps. The carrier sensing range is 300m and transmission range is 250m.

We generate a random topology with 30 nodes in 1000m $\times$ 1000m rectangular field. All nodes are static.



(a) Self traffic (b) Nbr traffic

Fig. 17. Goodput comparison of PPTT to ETX with varied neighboring and self traffics



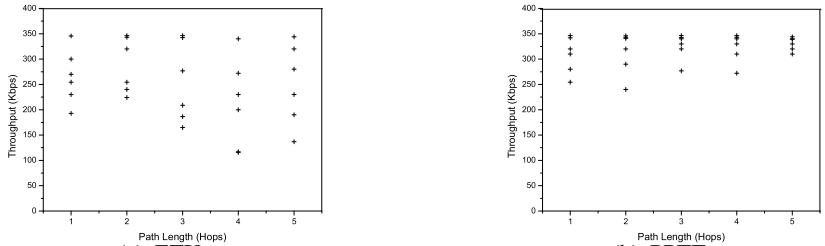
(a) Goodput (b) Delay

Fig. 18. Comparison of goodput and delay of ETX to PPTT

The source-destination pairs are spread randomly over the network. We establish 20 CBR connections by choosing source and destination randomly and ensure that there are at least two connections active at the same time. Each connection lasts for 20 seconds. The order in which the connections are established is randomized. The waiting time between the start of two successive connections is 5 seconds. Simulations last for 500 simulated seconds. To compare the performance under different circumstance of traffic load, we set traffic rate of each CBR connection as 128kbps, 256kbps and 384kbps respectively.

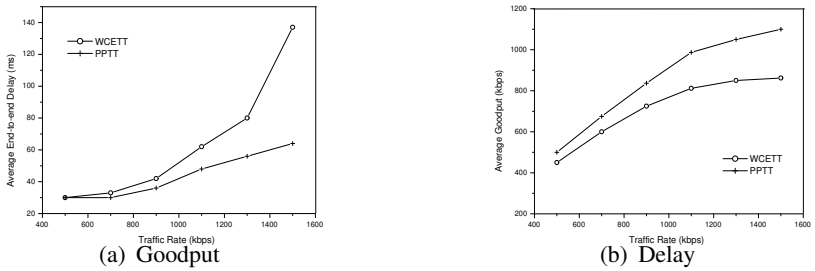
For each simulation, we calculate the average goodput and delay of all CBR connections. In Fig.18(a) and 18(b), we compare goodput and end-to-end delay of ETX to PPTT. From these figures, we can see that in 128kbps case, ETX can achieve the same performance as PPTT. This confirms the previous analysis that in light-loaded cases, the lack of consideration of self-traffic may not cause performance losing. But in 256kbps and 384kbps cases, the performance of PPTT is much better than ETX. The average goodput using PPTT metric is up to 28% higher than ETX. And the average end-to-end delay of each connection using PPTT metric is up to 52% lower than ETX.

The gain is a result of the fact that PPTT takes self-traffic interference into account as well as neighboring traffic interference. Two simultaneous connections are possibly interfered with each other. Since ETX doest not consider self-traffic interference, it may not select the high-quality path. The ability of PPTT to select good paths is illustrated in Fig.19(a) and 19(b). The figures show the relationship between path length and throughput for ETX and PPTT. We can see that ETX might select paths in a sub-optimal manner. For example, considering the 3-hop path in both two plots, ETX sometimes selects a low throughput path. However, PPTT always selects higher-throughput paths.



(a) ETX (b) PPTT

Fig. 19. Relationship between path lengths and throughput of individual connections in single-radio scenario. Each dot represents a candidate path.



(a) Goodput (b) Delay

Fig. 20. Performance comparison of PPTT and WCETT in multi-radio testbed

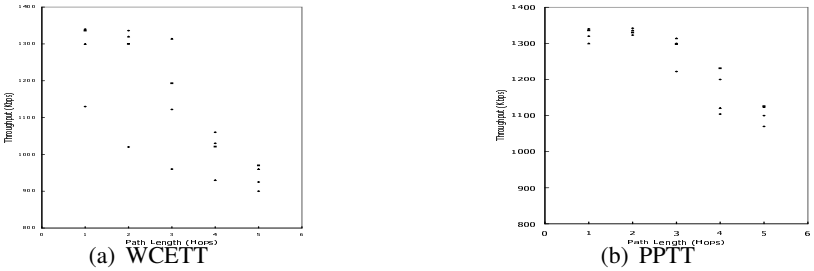
D. Multi-radio scenario

Our proposed PPTT scheme has considered channel diversity of multi-radio network, so it can also serve as multi-radio routing metric.

We first test PPTT scheme in our testbed. Each node equips two wireless network card, one operates on 802.11a mode and the other operates on 802.11g mode. We randomly carried out UDP connections in the testbed and we vary traffic rate of each connections to emulate different loaded cases, both for WCETT and PPTT. The comparison of goodput with different traffic load is shown in Fig.20(a) and 20(b). The average goodput using PPTT is up to 30% higher than WCETT. And the average end-to-end delay using PPTT is up to 57% lower than WCETT.

The ability of PPTT to select high performance path is illustrated in Fig.21(a) and Fig.21(b). WCETT sometimes selects some low throughput path, while PPTT always selects better paths. For example, let us think about an individual flow from node 23 to node 26. In Fig.22, it is shown that there are two candidate paths, Path-1 (23a, 17a, 8ag, 16g, 24ga, 28a, 26a)[3] and Path-2 (23g, 8g, 16g, 24ga, 28a, 26a). WCETT of path-2 is less than that of path-1, thus path-2 is selected by WCETT. However, there is a hidden terminal in path-2 while no hidden terminal in path-1. The self-traffic impact of path-2 is more serious than that of path-1. Therefore, the goodput of path-1 is 1.2Mbps while that of path-2 is 900kbps. Since the self-traffic is explicitly considered in PPTT scheme, PPTT will select path-1 rather than path-2.

[3]23a means the packet is incoming and outgoing both from 802.11a radio of node 23. 8ag means the packet is incoming from 802.11a radio of node 8 and outgoing from 802.11g radio of node 8. The others are likewise.



(a) WCETT (b) PPTT

Fig. 21. Relationship between path lengths and throughput of individual connections in multi-radio scenario. Each dot represents a candidate path.

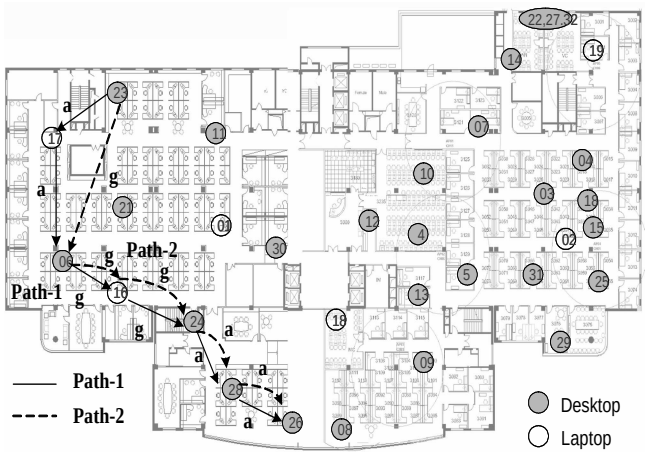
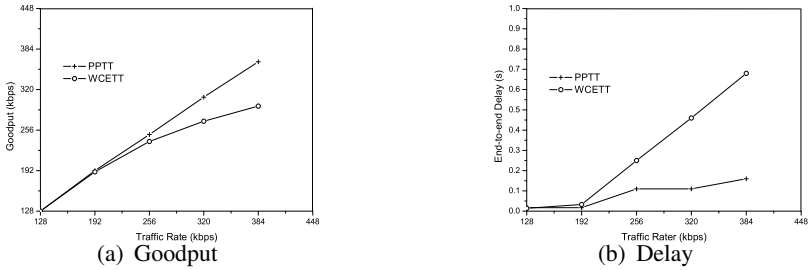


Fig. 22. One flow from node 23 to node 26. Two candidate multi-radio path.

We then compare the performance of PPTT and WCETT by simulations. We generate a random topoloy with 30 static two-radio nodes in 1000m $\times$ 1000m rectangular field. We establish 20 CBR connections randomly over the network. We adjust the traffic rate of each CBR connection to compare the performance under different traffic-load cases. The simulation results are shown in Fig.23(a) and 23(b). The results show that PPTT achieves better performance in terms of goodput and delay than WCETT.

Both the experiment and simulation results show that in light-loaded cases, PPTT can achieve the similar performance as WCETT. In light-loaded case, self-traffic interference is trivial, the path performance is determined by channel diversity. Since PPTT has taken channel diversity into account, it can



(a) Goodput (b) Delay

Fig. 23. Performance comparison of PPTT and WCETT in multi-radio scenario

achieve comparable performance with WCETT. In heavy-loaded cases, PPTT can achieve lower delay and higher goodput than WCETT. This is because that PPTT considers both channel diversity and self-traffic interference, while WCETT considers channel diversity only.

VII. Conclusions

Self-traffic effect should be taken into account in routing metric in order to get a more accurate estimation of transmission time along the path, especially for the RTC flow which has critical delay and bandwidth requirements. This paper proposes a new prediction-based routing metric named PPTT (Path Predicted Transmission Time) to estimate the transmission time for the new coming RTC flow before it is injected into the wireless multi-hop network, and tend to select route with minimal PPTT. We analyze packet service time for rate-controlled RTC traffic, to deduce the expected transmission time on the corresponding link. Based on this model, we estimate the link predicted transmission time (LPTT) according to the interfering traffic from the neighbors including both carrier sensing nodes and hidden terminal nodes. The calculation of LPTT reflects the effect from neighboring traffic, while the PPTT reflects the effect from the self-traffic, thus PPTT offers a unified traffic-aware routing metric for wireless mesh network. Experimental results from both real test bed and network simulator (NS2) show that our PPTT algorithm outperforms other routing schemes without considering self-traffic effect for RTC traffic. The average goodput improvement is about 28%, while the delay improvement is about 52%.

PPTT can also serve as multi-radio routing metric since it calculates the interference effect for different channels accordingly instead of putting them together. In our experiment, PPTT outperform WCETT for RTC traffic in multi-radio environment. For the next step, we are going to evaluate the performance of our proposed scheme in a larger wireless mesh network.

VIII. Acknowledgments

We would like to thank Richard Draves *et al*. for providing MCL code. This is our base for implementing our proposed routing metric. We would also like to thank Yang Yang and Yunxin Liu for the help of testbed experiment.

Appendix
PPTT of MMPP(2) Traffic Model

For simplicity, we use Poisson traffic model in derivation of PPTT previously. In fact, PPTT scheme is independent of traffic model. We can use other more accurate traffic models in PPTT scheme. In this appendix we deduce PPTT of Markov-Modulated Poisson Process, MMPP(2), traffic model. MMPP(2) is a better traffic model for real-time traffic such as voice and video[31] than Poisson traffic model.

Before processing further we introduce some notations of MMPP(2) traffic model. Consider link (i,j), we denote MMPP(2) traffic model with the following infinitesimal generator $\mathcal{Q}_{ij}$ of the underlying Markov chain and with the arrival rate matrix Λ_{ij}:

$$\mathcal{Q}_{ij} = \begin{bmatrix} -\sigma_{ij}^1 & \sigma_{ij}^1 \\ \sigma_{ij}^2 & -\sigma_{ij}^2 \end{bmatrix}, \Lambda_{ij} = \begin{bmatrix} \lambda_{ij}^1 & 0 \\ 0 & \lambda_{ij}^2 \end{bmatrix}. \tag{21}$$

The $\mathcal{Q}_{ij}$ and Λ_{ij} constructs determine the steady-state probability vector π_{ij}, given by

$$\pi_{ij} = \left(\frac{\sigma_{ij}^2}{\sigma_{ij}^1 + \sigma_{ij}^2}, \frac{\sigma_{ij}^1}{\sigma_{ij}^1 + \sigma_{ij}^2} \right), \tag{22}$$

and the mean arrival rate λ_{ij} is

$$\lambda_{ij} = \frac{\lambda_{ij}^1 \sigma_{ij}^2}{\sigma_{ij}^1 + \sigma_{ij}^2} + \frac{\lambda_{ij}^2 \sigma_{ij}^1}{\sigma_{ij}^1 + \sigma_{ij}^2}. \tag{23}$$

To deduce LPTT according to state transmit diagram of basic access method shown in Fig.7(a), we first calculate P_{DIFS}^i, P_{slot}^i and P_{DATA}^i.

P_{DIFS}^i is the probability that no link of CL_{ij} transmitting packets in the time interval of $DATA+DIFS$. For a certain link (k,l), the probability of no packet transmitting along link (k,l) in $DATA+DIFS$ is

$$P_{\mathrm{DIFS}}^{kl} = \frac{\sigma_{kl}^1}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-(DATA_{kl} + DIFS) \cdot \lambda_{kl}^1] + \frac{\sigma_{kl}^2}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-(DATA_{kl} + DIFS) \cdot \lambda_{kl}^2] \tag{24}$$

Thus

$$P_{\mathrm{DIFS}}^i = \prod_{(k,l) \in CL_{ij}} P_{\mathrm{DIFS}}^{kl} \tag{25}$$

Similarly, the probability of no packet transmitting in $slot$ is

$$P_{\mathrm{slot}}^{kl} = \frac{\sigma_{kl}^1}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-slot \cdot \lambda_{kl}^1] + \frac{\sigma_{kl}^2}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-slot \cdot \lambda_{kl}^2] \tag{26}$$

Thus

$$P_{\mathrm{slot}}^i = \prod_{(k,l) \in CL_{ij}} P_{\mathrm{slot}}^{kl} \tag{27}$$

For DATA packet successful transmission probability, the probability of no packet transmitting along link (k,l) in the time interval of two DATA packets transmission is

$$P_{\mathrm{DATA}}^{kl} = \frac{\sigma_{kl}^1}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-(\frac{S_{\mathrm{D}}}{B_{kl}} + \frac{S_{\mathrm{D}}}{B_{ij}}) \cdot \lambda_{kl}^1] + \frac{\sigma_{kl}^2}{\sigma_{kl}^1 + \sigma_{kl}^2} \exp[-(-(\frac{S_{\mathrm{D}}}{B_{kl}} + \frac{S_{\mathrm{D}}}{B_{ij}})) \cdot \lambda_{kl}^2] \tag{28}$$

Thus

$$P_{\mathrm{DATA}}^i = \prod_{(k,l) \in CL_{ij}} P_{\mathrm{DATA}}^{kl} \tag{29}$$

Substituing (25), (27) and (29) into (13), we can get the packet service time, T_{MAC}, of MMPP(2) model.

To calculate the queueing delay, we treat a link as MMPP(2)/G/1 queueing system. For link (i, j), the mean waiting time of MMPP(2)/G/1 system is given by [32]

$$T_{queue} = \frac{1}{2(1-\rho)} \left[2\rho + \lambda_{ij} m_2 - 2m_1((1-\rho)g + m_1 \pi_{ij} \Lambda_{ij})(\mathcal{Q}_{ij} + e\pi_{ij})^{-1}\lambda \right], \tag{30}$$

where e is a 2×1 vector of 1's, $\lambda = (\lambda_{ij}^1, \lambda_{ij}^2)$, ρ is the offered load, m_1 and m_2 are the first two moments of Laplace transform of T_{MAC}, and g is a vector that can be found by an algorithm provided in [32].

Thus, summing the queueing delay and packet service time, we can get LPTT of link (i, j) for the MMPP(2)/G/1 model. And then we can get PPTT of each path.

However, it is a bit more complicated to implementing PPTT based on MMPP(2) model than Poisson model, since we need to transmit four parameters, λ_1, λ_2, σ_1 and σ_2 for MMPP(2) model instead of only one parameter, λ for Poisson model. Considering we can get reasonably accurate estimates of path quality by using Poisson model, we use PPTT based on Poisson model in our real system.

REFERENCES

[1] T.-W. Chen, J.T. Tsai, and M. Gerla. QoS routing performance in multihop, multimedia, wireless networks. *Proceedings of IEEE ICUPC '97*, 1997.

[2] Richard Draves, Jitendra Padhye and Brian Zill, Routing in multi-radio, multi-hop wireless mesh networks, *Proceedings of the 10th annual international conference on Mobile computing and networking* 2004:114 - 128, Philadelphia, USA.

[3] Microsoft Mesh Connectivity Layer (MCL) Software, http://research.microsoft.com/mesh

[4] The Network Simulator - ns-2, 2003. http://www.isi.edu/nsnam/ns.

[5] Paramvir Bahl, Atul Adya, Jitendra Padhye and Alec Walman. Reconsidering wireless systems with multiple radios. *ACM SIGCOMM Computer Communication Review* 2004, 34(5):39-46.

[6] Chenxi Zhu and M. Scott Corson. QoS routing for mobile ad hoc networks. *Proceedings of IEEE Infocom*, June 2001.

[7] Chunhung Richard Lin. On-demand QoS routing in multihop mobile networks. *Proceedings of IEEE Infocom*, April 2001.

[8] W. Liu and Y. Fang. Courtesy Piggybacking: Supporting Differentiated Services in Multihop MobileAd Hoc Networks. *Proceddings of IEEE Infocom*, 2004.

[9] A. Veres G. Ahn, A.T. Campbell and L. Sun. SWAN: Service Differentiation in Stateless WirelessAd Hoc Networks. *Proceedings of IEEE Infocom*, Jun 2002.

[10] Q. Xue and A. Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing* 2003, 63:154-165

[11] C. Perkins and E. Belding-Royer. Quality of Service for Ad hoc On-Demand Distance Vector Routing (work in progress), Oct 2003. draft-perkins-manet-aodvqos-02.txt

[12] K. Al Agha G. Pujolle H. Badis, A. Munaretto. QoS for Ad Hoc Networking Based on Multiple Metrics: Bandwidth and Delay. *Proceedings of The Fifth IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003)*, 2003

[13] R. Sivakumar P. Sinha and V. Bharghavan. CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm. *Proceedings of IEEE Infocom*, Mar 1999

[14] X. Zhang S.B. Lee, A. Gahng-Seop and A.T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks. *Journal of Parallel and Distributed Computing(Special issue on Wireless and Mobile Computing and Communications)* 2000, 60(4):374-406.

[15] Derya H. Cansever, Arnold M. Michelson, and Allen H. Levesque. Quality of service support in mobile ad-hoc IP networks. *Proceedings of IEEE MILCOM 1999*:30-34.

[16] Hongxia Sun and H.D. Hughes. Adaptive QoS Routing Based on Prediction of Local Performance in Ad Hoc Networks. *Proceddings of IEEE WCNC 2003*.

[17] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF RFC 3561.

[18] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The Dynamic Source Routing protocol for mobile ad hoc networks (DSR). Internet draft (work in progress), IETF, April 2003. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt.

[19] V. Park, S. Corson: Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification, 2001. http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-04.txt.

[20] Charles E. Perkins , Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *Proceedings of the conference on Communications architectures, protocols and applications* 1994:234-244, London, UK.

[21] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. *Proceedings of ACM SIGCOMM*, Portland, OR, August 2004.

[22] Douglas S. J. De Couto, Daniel Aguayo, John Bicket and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Proceedings of the 9th annual international conference on Mobile computing and networking*, Pages: 134 - 146, San Diego, CA, USA, 2003.

[23] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. *Proceedings of BroadNets*, 2004.

[24] S. Keshav. A Control-theoretic approach to flow control. *Proceedings of SIGCOMM*, Sep 1991.

[25] Yih-Chun Hu and David B. Johnson. Design and demonstration of live audio and video over multihop wireless ad hoc networks. *Proceedings of the MILCOM* 2002.

[26] Tom Goff , Nael B. Abu-Ghazaleh , Dhananjay S. Phatak , Ridvan Kahvecioglu. Preemptive routing in Ad Hoc networks. *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001:43-52, Rome, Italy.

[27] I. S. Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In IEEE 802.11 Standard. IEEE,ISBN 1-55937-935-9, 1997.

[28] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordinated function. *IEEE JSAC* 2000, 18(3):535-547.

[29] N. Gupta and P. R. Kumar. A performance analysis of the IEEE 802.11 wireless LAN medium access control. *Communications in Information and Systems* 2004, 3(4):279-304.

[30] Josh Broch, David A.Maltz and David B.Johnson. Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks. *Proceedings of Parallel Architectures, Algorithms, and Networks* 1999. 370 - 375.

[31] H. Heffes and D.M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Selected Areas Comm.* 1986, 4(6):856 - 868.

[32] W.Fischer and K.Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook, Performance Evaluation. 1992, 18:149 - 171.