UWL REPOSITORY

repository.uwl.ac.uk

Modelling and Analysis of TCP Performance in Wireless Multihop Networks

Xiao, Hannan, Zhang, Ying ORCID logoORCID: https://orcid.org/0000-0002-6669-1671, Malcolm, James, Christianson, Bruce and Kee Chaing, Chua (2010) Modelling and Analysis of TCP Performance in Wireless Multihop Networks. Wireless Sensor Networks, 2 (7). pp. 493-503. ISSN 1945-3078

This is the Published Version of the final output.

UWL repository link: https://repository.uwl.ac.uk/id/eprint/5286/

Alternative formats: If you require this document in an alternative format, please contact: open.research@uwl.ac.uk

Take down policy: If you believe that this document breaches copyright, please contact us at open.research@uwl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

◆◆ Scientific
◆◆ Research

Modelling and Analysis of TCP Performance in Wireless Multihop Networks

Hannan Xiao[1], Ying Zhang[2], James Malcolm[1], Bruce Christianson[1], Kee Chaing Chua[3]
[1]*School of Computer Science, University of Hertfordshire, Hatfield, UK*
[2]*Datalink Electronics Group, Loughborough, UK*
[3]*Department of Electrical and Computer Engineering, National University of Singapore,*
Lower Kent Ridge Road, Singapore
E-mail: *h.xiao@herts.ac.uk*

Abstract

Researchers have used extensive simulation and experimental studies to understand TCP performance in wireless multihop networks. In contrast, the objective of this paper is to theoretically analyze TCP performance in this environment. By examining the case of running one TCP session over a string topology, a system model for analyzing TCP performance in multihop wireless networks is proposed, which considers packet buffering, contention of nodes for access to the wireless channel, and spatial reuse of the wireless channel. Markov chain modelling is applied to analyze this system model. Analytical results show that when the number of hops that the TCP session crosses is fixed, the TCP throughput is independent of the TCP congestion window size. When the number of hops increases from one, the TCP throughput decreases first, and then stabilizes when the number of hops becomes large. The analysis is validated by comparing the numerical and simulation results.

Keywords: Wireless Multihop Networks, TCP Modelling, TCP

1. Introduction

An adhoc network has no fixed infrastructure: data transmission depends on the temporary location of nodes and the transit distribution of traffic in the fly. Originally designed for the wired Internet, TCP exhibits anomalous performance features in adhoc wireless networks, such as mobility induced retransmission [1], capture effect and unfairness [2] over the IEEE 802.11 MAC layer protocol, and RED-like packet dropping behaviour from link layer contention [3]. TCP performance in multihop adhoc wireless networks has been an active research topic. Extensive simulations have been conducted to obtain a better understanding of TCP behaviour in adhoc wireless networks and to find ways of enhancing TCP performance in that environment. There is not much effort, however, contributing to theoretical analysis of TCP performance in adhoc wireless networks.

This research work was motivated by the interesting observations of TCP in adhoc wireless networks (in simulations) reported in [1,2,4]: when a TCP connection runs over a static *string* topology (**Figure 1**), the TCP throughput measured decreases rapidly when the number

of hops increases from one, and then stabilizes when the number of hops becomes large. Gerla gave an expression in [2] and predicted that the throughput would drop faster than exponentially with hop length, without giving further theoretical analysis.
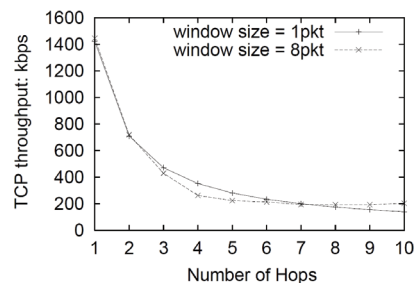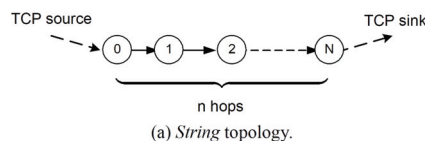
(a) *String* topology.

(b) TCP throughput over the *string* topology.
(pkt=packet)

Figure 1. TCP performance over a *String* topology.

In this paper, a system model for analyzing TCP performance in adhoc wireless networks is proposed after examining the case of running one TCP session over a string topology. The system model considers the features of wireless multihop network such as contention of nodes for access to the wireless channel, packet buffering intermediate nodes, and spatial reuse of the wireless channel. A Markov chain modelling is applied to analyze this system model starting from a single hop case to multiple hop cases. Analytical results show that when the number of hops that the TCP session crosses is fixed, the TCP throughput is independent of the TCP congestion window size. When the number of hops increases from one, the TCP throughput decreases first, and then stabilizes when the number of hops becomes large. The analysis is validated by comparing the numerical and simulation results.

The rest of the paper is organized as follows. Section 2 discusses related work in two areas; TCP performances in multihop wireless networks and system models for TCP analysis in different networks. Section 3 proposes a system model for analyzing TCP performance in multihop wireless networks, following which section 4 applies a Markov chain modelling of the system model. Section 5 presents the theo retical analysis of the TCP throughput over the string topology from specific cases to the general case and section 6 evaluates the analysis by comparing the simulation and numerical results. Finally section 7 concludes the paper.

2. Related Work

2.1. TCP Performance in Wireless Multihop Networks

TCP was initially designed for wired networks to provide reliable data transmission by retransmitting lost packets due to network congestion that is detected at the sender by time-out or duplicated acknowledgments. Packet loss triggers TCP congestion control which reduces the congestion window size, effectively the sending rate, as a reaction to network congestion. TCP, however, has performed poorly in wireless multihop networks where packet loss may happen from various reasons other than network congestion, such as error-prone wireless link, link layer contention and routing failure or breakage. Many approaches have been proposed to improve TCP performance in the environment of wireless multihop networks from different aspects. Based on the nature of the approaches, they may be summarized into three groups as in [5]:

- TCP with feedback [1,6-11] where feedbacks from networks are sent back to the sender to distinguish packet losses due to congestion from link error, link contention or routing breakage or failure.

- TCP without feedback [12,13] where TCP congestion control strategy is modified according to the scenarios in multihop wireless networks.
- TCP with lower layer enhancement [14-18]. This approach is more popular recently as more work have been done to understand the cross-layer interaction, especially lower layer impact on TCP performance in wireless multihop networks.

The above work shares one similarity *i.e.*, they all observe TCP performance in multihop wireless networks by simulations and then propose improvements. Very few works, however, have been devoted on theoretical analysis of TCP performance in multihop wireless networks. We proposed a system model for TCP analysis over one-hop *string* topology as early as in 2001 [19] and then extended the modelling to multihop cases in [20]. So far our analysis only applies when there is no packet loss in the multihop wireless TCP packet transmissions. Multiple lossy links are modelled in [21] which also consider different proportions between the interference range and transmission range in the spatial reuse property of the wireless channel.

2.2. System Models for TCP Analysis

There are various approaches to modelling TCP in the literature. Some papers have tried to capture the essential TCP dynamics through closed-form expressions. Lakshman and Madhow [22] and Kumar [23] use Markovian analysis to develop a closed-form expression for the throughput of TCP connections by observing the cyclical evolution of the TCP transmission window. The latter work introduces some extensions for several versions of TCP, incorporating such features as coarse timers, fast retransmit and fast recovery. Mathis *et al*. [24] focus on the stochastic behavior of the congestion avoidance mechanism, deriving an expression for the throughput that is then applied to study the behavior of several flavors of TCP sources and queueing techniques. Padhye *et al*. [25] have derived a steady-state model that approximates the throughput of bulk TCP flows as a function of loss rate and round trip time, comparing their estimates with real-life traces of TCP traffic. In [26], instead of trying to establish a closed-form expression for key metrics such as throughput, queueing delay and packet loss, a novel methodology, reciprocal model tuning, combines a Markovian model of a single TCP source in isolation, and the analysis of superposition and interaction of several TCP sources through standard queueing analysis techniques. Notably, some researchers [27,28] have chosen a different approach whereby the observation of "actual" TCP traces is the foundation of empirical models. These efforts imply collecting hours' (days') worth of data, and finding suitable statistical distributions for the observed data.

There are two system models in the literature most similar to our model. The first one is used by Chaskar, Lakshman and Madhow in [29] to analyze TCP performance over a wireless channel with link level error control. There, ACK packets are assumed to arrive in the source node after a constant delay and are never lost. Thus the path of ACKs is omitted (**Figure 2a**). The second one is proposed by Lakshman, Madhow and Suter in the same research group to analyze TCP performance with random loss and bidirectional congestion [30]. There, for the first time the path followed by ACKs is explicitly modeled (**Figure 2b**). These models, however, cannot be used in our case of multihop wireless networks where the TCP packets and ACKs are contented to use the same wireless channel. Instead of starting with analyzing the TCP protocol with assumptions as in [22-26], our approach is to study the process of how TCP works between a source and destination pair by examining the simulation trace file. The resulting model considers the channel access probability of both the source and the destination (**Figure 3**).

3. System Model

Consider the simple adhoc wireless network scenario shown in **Figure 1** nodes form a *string* with length N. Each node has the same transmission radius, the same carrier sense radius, and the same interference radius. One TCP connection is run from node 0 to node N crossing all the intermediate nodes in the *string*.

We model the communication process of the TCP connection as in **Figure 3** after analyzing the trace files from simulations carefully. The source and destination nodes both have a First In First Out (FIFO) forward buffer of size The source has infinite data to send, so that TCP packets are always of the maximum packet size. For each packet that is received by the destination, a cumula-
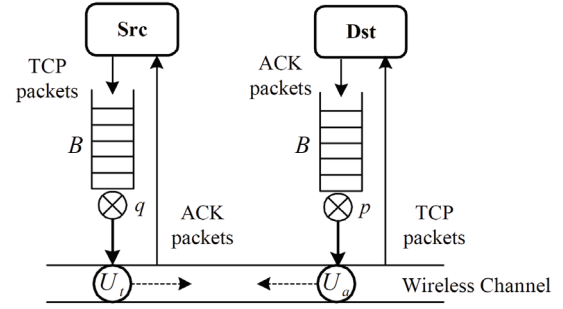

a. Model for TCP over wireless with link error control.

b. Model for TCP with random loss and bidirectional congestion.

Figure 2. Reference system models for TCP analysis.

Figure 3. System model of TCP in wireless ad-hoc networks.

tive acknowledgment (ACK) is generated which can be modeled as containing the next expected segment number. TCP packet service rate is U_t, the typical number of TCP packets the system serves per unit time when it is constantly busy. ACK packet service rate is U_a, the typical number of ACK packets the system serves per unit time when it is constantly busy. The "unit time" includes TCP/ACK packet transmission delay, propagation delays, processing delays at the nodes from layer to layer and the average MAC layer contention delays. The TCP/ACK packet service rates are affected by the spatial reuse of wireless channel in the *string* topology. As all nodes compete to use the channel, the chance for any node to send packet is determined by the underlying MAC protocol and traffic distribution. Generally, we denote the *average* probability of the source to send a TCP packet as q, and the *average* probability of the destination to send an ACK packet as p. The reason to use the average probability of the node accessing the channel is because the performance metric of interest is the average TCP throughput.

The system model captures the unique communication features of TCP in adhoc wireless networks including link layer channel contention and channel spatial reuse. It is similar to but different from system models used in analyzing TCP performance in other kind of networks [29,30].

4. Markov Chain Modelling

We use Markov chain modelling to analyze the proposed system model. Firstly, a discrete Markov chain is formed for the case when the source and the destination are single-hop away. The Markov chain is then extended to the general case when the source and the destination are multiple hops away. The assumptions made are as follows:

● There is no random loss of packets due to channel error. The channel appears error-free to the upper layer because of the error coding schemes and link layer retransmission protocols.
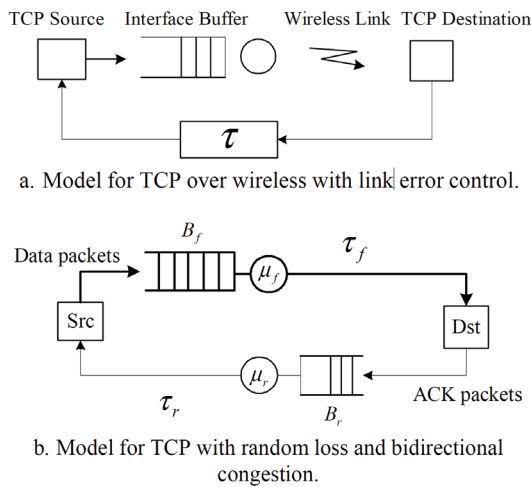
- The channel also appears collision-free to the upper layer because the MAC layer protocol is collision avoided, such as MACAW [31].
- The maximum TCP congestion window size is less than the buffer size at each node.
- There is no packet dropping due to buffer overflow, and no packet loss due to link layer contention. This assumption is supported by simulation results demonstrating that packet loss due to buffer overflow is rare and packet loss due to link contention is very low for single TCP over a *string* topology.
- The distribution of inter-service time of TCP packets and ACK packets are exponentially distributed which include packet retransmissions. This assumption is based on the fact that exponential backoff during contention is commonly used in MAC protocols, e.g., IEEE 802.11.

4.1. Single-Hop Case

There are only TCP packets at the source and only ACK packets at the destination. Either the source or the destination sending packets will change the number of the packets queued in their buffers. Ignoring the slow start phase which is a small part of the data transmission, the sum of the TCP packets at the source and the ACK packets at the destination equals to the TCP congestion window size in packets. As it is assumed that no packet loss exists due to channel error, buffer overflow, or link contention, the TCP connection thus increases its congestion window size to the maximum value and stabilizes there.

Discrete-time Markov chain is applied to analyze TCP performance here. Let us focus attention at times 0, δ, 2δ …, $k\delta$,… as in [32] (pages 162-173), where δ is a small positive number. Let F denote as the number of TCP packets in the source node buffer at time $k\delta$. F thus is a Markov chain on the state-space $\{m : W \geq m \geq 0\}$, where W the maximum TCP congestion window size. The transition diagram of the Markov chain is shown in **Figure 4**.

When there are W TCP packets in the source node, *i.e.*, the first state in **Figure 4**, the destination node does not have ACK packets to send. Therefore, the source node catches the wireless channel for sure. The resulting

transition probability from the state of $F = W$ to $F = W - 1$ is δU_t, where U_t is the service rate of TCP packets and the inter-service time is assumed to be exponentially distributed. Similarly, the transition probability from the state of $F = 0$ TO $F = 1$ is δU_a Where U_a is the service rate of ACK packets and the inter-service time is assumed to be exponentially distributed.

When there are $F (W - 1 \geq F \geq 1)$ TCP packets in the source, there are $W - F$ ACK packets in the destination node. The source node and destination node compete with each other to use the channel. The source node has average probability of q to access the channel (**Figure 3**), the transition probability from the state of $F = W - i$ to $F = W - i - 1$ is thus $q\delta U_t$, where $1 \leq i \leq W - 1$. Similarly, the transition probability from the state of $F = W - i$ to $F = W - i + 1$ is $p\delta U_a$, where p is the average probability of the destination to access the channel.

4.2. Multiple-hop Case

When the source and destination are multiple hops away, an accurate Markov model at times $k\delta$ would be on the state space $\{(F_0, A_0), (F_1, A_1), ..., (F_N, A_N)\}$, where (F_i, A_i) are the numbers of TCP and ACK packets at node i. This multi-dimensional Markov chain modelling considers the number of TCP packets and ACK packets in the source, the destination and the intermediate nodes along the *string* topology. Unfortunately, such a modelling is difficult to tackle. For example, even when the *string* topology is 2-hops long, and the TCP maximum congestion window size is 2 packets, the Markov chain already has 11 states as shown in **Figure 5**, and the number of states increases quickly with the increase of the TCP maximum congestion window size and the length of the *string*. This happens due to the sharing of wireless channel between nodes. When two nodes within interference range both have packets to send, there are two possible next states. The more states the Markov chain has, the more difficult it is to solve. A heuristic alternative is applied as follows.

The more states the Markov chain has, the more paths are available in the transition diagram for a TCP packet

$$Q = \begin{pmatrix} 1 - \delta U_t & \delta U_t & 0 & 0 & \cdots & 0 \\ p\delta U_a & 1 - p\delta U_a - q\delta U_t & q\delta U_t & 0 & \cdots & 0 \\ 0 & p\delta U_a & 1 - p\delta U_a - q\delta U_t & q\delta U_t & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & p\delta U_a & 1 - p\delta U_a - q\delta U_t & q\delta U_t \\ 0 & \cdots & 0 & 0 & \delta U_a & 1 - \delta U_a \end{pmatrix} \tag{1}$$
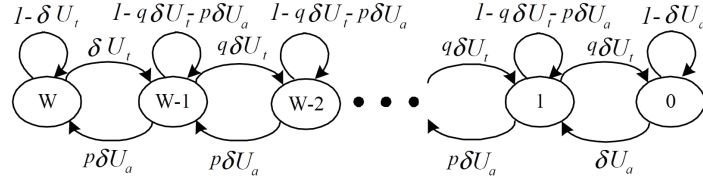

Figure 4. Transition diagram when the number of hop is 1 and TCP maximum congestion window size is *W*.
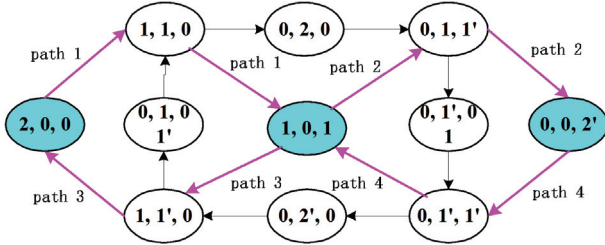

Figure 5. Transition diagram when the number of hop is 2 and the maximum TCP congestion window size is 2 packets. The number of TCP packets at nodes are shown as 1 or 2; the number of ACK packets are shown as 1' or 2'. (1, 1', 0) denotes that there is 1 TCP packet at the source node, 1 ACK packet at the middle node, and none packet at the destination node.

to be transmitted from the source to the destination, or for an ACK packet from the destination to the source. Whatever states the packet goes through in between, it finally passes all the nodes of the *string*. Let us denote the path for a TCP packet to be transmitted from the source to the destination as a *forward* path; and the path for an ACK packet to be transmitted from the destination to the source as a *backward* path. A transmission round of TCP is composed of one *forward* path and one *backward* path.

The average utilization of the *string* topology should be the average of all the transmission rounds. Intuitively, ***for each forward path, a corresponding complementary backward path exists which makes this round of transmission to be exactly the average value***. With this understanding, the analysis can be sought out based on a specific transmission round with a pair of complementary *forward* and *backward* paths only. All the TCP packets go through the specific *forward* path, and all the ACK packets go through the specific *backward* path in such analysis.

A simple transmission round is chosen as follows: for the *forward* path, when the source node gets chance to start sending a TCP packet, the packet passes all the intermediate nodes continuously until it reaches the destination; the *forward* path can be path 1 or path 2 in **Figure 5**. Likewise, for the *backward* path, when the destination node gets chance to start sending an ACK packet, the ACK packet passes all the intermediate nodes continuously until it reaches the source; the *backward* path can be path 3 or path 4 in **Figure 5**. These paths are

complementary.

In the above specific transmission round, once one packet in the source node is sent out, the number of packets in the buffer of source node decreases by 1. After the packet goes continuously to the destination, the destination absorbs the TCP packet and generates an ACK packet. The reverse process is the same. Consequently, the changing of number of packets in the source and the destination is the same as in the single-hop scenario. The Markov modelling of this complete transmission round is thus exactly the same as in **Figure 4** when we only consider the marked paths and states in **Figure 5**.

In the following subsections, we are going to analyze the discrete Markov chain in **Figure 4** for both the single-hop and multiple-hop cases.

5. Analysis

5.1. TCP Throughput

The transition matrix Q of the Markov chain in **Figure 4** is listed in (1). Let $\pi = (\pi w, \pi w-1, \dots \pi 0)$ denote the stationary probability distribution of the Markov chain, we have $\pi = \pi \times Q$ at steady state. From this it is derived that

$$\pi w-1 = \frac{1}{q}\left(\frac{q}{p}\rho\right)\pi w$$

$$\pi w-2 = \frac{1}{q}\left(\frac{q}{p}\rho\right)^2 \pi w$$

$$\dots \qquad \dots$$

$$\pi w-i = \frac{1}{q}\left(\frac{q}{p}\rho\right)^i \pi w, \quad 1 \le i \le W-1 \tag{2}$$

$$\dots \qquad \dots$$

$$\pi 0 = \frac{p}{q}\left(\frac{q}{p}\rho\right)^w \pi w$$

where ρ is the ratio of U_t to U_a, *i.e.*, $\rho = \dfrac{U_t}{U_a}$.

From $\sum_{i=0}^{W}\pi_i = 1$, it is further derived that

$$\pi w = \frac{1}{1 + \sum_{i=1}^{W-1}\frac{1}{q}(\frac{q}{p}\rho)^i + \frac{p}{q}(\frac{q}{p}\rho)^W} \tag{3}$$

From the transition diagram, we denote the TCP throughput of state i by λ_i and the average throughput of TCP packets by λ. Thus $\lambda = \sum_{i=0}^{W} \pi_i \lambda_i$, where $\lambda w = U_t, \lambda_i - qU_t$ when $W - 1 \geq i \geq 1$, and $\lambda_0 = 0$. Therefore

$$\lambda = U_t \pi w \frac{1 - (\frac{q}{p}\rho)^w}{1 - \frac{q}{p}\rho} \qquad (4)$$

The source and destination nodes are located at the two ends of the *string* topology with $N - 1$ intervening nodes. The positions of the source and destination are the same, and they also have the same number of packets to send since we assume that one ACK is generated for each TCP packet. Thus, the source and destination nodes have the same *average* probability to access the channel successfully, *i.e.*, $p = q$. (3) and (4) can then be simplified as

$$\pi w = \frac{1 - \rho}{1 - \rho^{W+1} + (\frac{1}{q} - 1)\rho(1 - \rho^{W-1})} \qquad (5)$$

$$\lambda = U_t \frac{1 - \rho^W}{(1 - \rho^{W+1}) + (\frac{1}{q} - 1)\rho(1 - \rho^{W-1})} \qquad (6)$$

ρ is the ratio of TCP packet service rate to the ACK packet service rate. Since TCP packets and ACK packets are transmitted at the same channel, and they travel the same number of hops between the source and the destination, ρ is always approximately equal to the ratio of the TCP packet service rate to the ACK packet service rate in the case when there is only one hop between the source and the destination. This is further approximately equal to the ratio of the ACK packet size to the TCP packet size. The TCP packet (say, 1460 bytes) is normally much larger than the ACK packet (say 40 bytes), ρ is thus much smaller than 1 (say, around 40/1460 = 0.027), *i.e.*, $\rho \ll 1$. The maximum TCP congestion window size W *pkts* is an integer and when it is big enough, it is expected that

$$1 - \rho^W \approx 1 - \rho^{W-1} \approx 1 - \rho^{W+1} \qquad (7)$$

Thus from (6) and (7), the *average* TCP throughput is derived as

$$\lambda = U_t \frac{1}{1 + (\frac{1}{q} - 1)\rho} \qquad (8)$$

In the above analysis, it is shown clearly that the ratio of ACK packet size to TCP packet size being much less than 1 is required to carry out the approximation.

5.2. TCP Service Rate

We have defined U_t as the service rate of TCP packets seen q as the average probability for the source node to access the wireless channel. Their values, however, change with the length of the *string*, *i.e.*, the number of hops that the TCP session crosses from the source to the destination (**Figure 1**). This is because of the effect of global channel spatial reuse and local channel contention in adhoc wireless networks. By definition, service rate is the "typical number of customers the system serves per unit time when it is constantly busy" [32] (page 152). When the number of hops is N, let the average number of TCP packets being transmitted (served) in the system be $\overline{I_N}$ and the service time to transmit a packet be T_N, the definition of service rate gives:

$$U_{tN} = \frac{\overline{I_N}}{T_N} \qquad (9)$$

Let qN be the average probability that the source node accesses the wireless channel when the number of hops is N. As N changes, $\overline{I_N}$, T_N, and qN as N changes.

5.2.1. N is 1, 2, 3 or 4

When N is 1, $\overline{I_1} = 1$ *pkt*, $q_1 = \frac{1}{2}$. Let $T_1 = T$, where T is the service time to transmit a packet over one hop when the source and destination nodes are one hop away.

When N is 2, each TCP packet travels two wireless links to reach the destination. Therefore, approximately $T_2 = 2T$. During the transmission from node 0 to node 1, and then from node 1 to node 2, only 1 TCP packet can be transmitted in the system without collision. Therefore $\overline{I_2} = 1$ *pkt*. In the *string* topology, the source node only forwards TCP packet, the destination node only forwards ACK packets but the middle node forwards both TCP and ACK packets. Since one ACK packet is assumed for each TCP packet, the middle node sends out packets twice as much as the source and destination node. As a result, the average probability of the middle node to access the channel successfully is twice as much as that of the source and destination node. In addition, the summation of the average probability of all nodes to access the channel is 1. Derivation from these relationships gives $q_2 = \frac{1}{4}$. When N is 3, by similar analysis we have $\overline{I_3} = 1$ *pkt*, $T_3 = 3T$, $q_3 = \frac{1}{6}$.

When N is 4, it looks that node 0 and node 3 could send packets concurrently without collision. However,

although node 3 is outside of node 1's transmission range, it is within the carrier sensing range and interference range of node 1. Node 3 is thus a potential hidden terminal of the transmission pair node 0 to node 3. Consequently, only 1 TCP packet can be transmitted in the system without collision. This analysis is consistent with that in [3]. Further derivation gives $\overline{I_4} = 1\ pkt$, $T_4 = 4T$, $q_4 = \dfrac{1}{8}$.

We summarize the value for $\overline{I_N}$, $\overline{I_N}$ as shown in **Table 1**.

5.2.2. N is 5, 6, 7, or 8

When N is 5, each TCP packet crosses five wireless hops to reach the destination. Among the five hops from node 0 to node 5, the 1st hop (*i.e.*, from node 0 to node 1) and the 5th hop (*i.e.*, from node 4 to node 5) can be utilized at the same time as shown in **Figure 6**. But when the 2nd, 3rd or 4th hop is used, only that hop can be used without collision.

Assuming that the system is utilized fully, the number of packets in the system is:

$$I_5 = \begin{cases} 2\,pkts & \text{when the 1st and 5th hops} \\ & \text{are used} \\ 1\,pkt & \text{when the 2nd, 3rd or 4th hops} \\ & \text{are used} \end{cases}$$

Since each hop has equal opportunity to be utilized, the average number of packets in the system is: $\overline{I_3} = 2 \times \dfrac{2}{5} + 1 \times \dfrac{3}{5}\,pkts$.

Table 1. N is 1, 2, 3 or 4

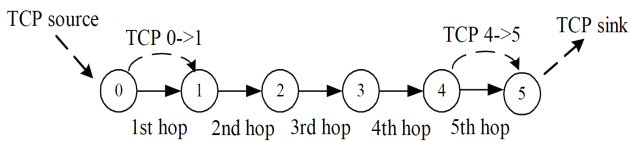| N	T_N	$\overline{I_N}$	q_N
1	T	$1\,pkt$	$\dfrac{1}{2}$
2	2T	$1\,pkt$	$\dfrac{1}{4}$
3	3T	$1\,pkt$	$\dfrac{1}{6}$
4	4T	$1\,pkt$	$\dfrac{1}{8}$

Figure 6. Two TCP packets are transmitted together when $N = 5$.

Although there are six nodes, however, node 0 competes to use the channel locally with nodes 1, 2, 3 and 4 only. The probability for node 0 to access the channel can therefore be taken as the same as when there are 4 hops. Thus $q_5 = \dfrac{1}{8}, T_5 = 5T$.

When N is 6, each TCP packet crosses six wireless links to reach the destination. **Figure 7** shows the scenarios when two of the six wireless links are utilized at the same time. The number of packets in the system is:

$$q_6 = \frac{1}{8}, T_5 = 5T$$

$$I_6 = \begin{cases} 2\,pkts & \text{when the (1st or 2nd) and} \\ & \text{(5th or 6th) hops are used} \\ 1\,pkt & \text{when the 3rd or 4th hop is used} \end{cases}$$

The average number of packets in the system is: $\overline{I_6} = 2 \times \dfrac{4}{6} + 1 \times \dfrac{2}{6}\,pkts$. It is also easy to get $q_6 = \dfrac{1}{8}$, $T_6 = 6T$.

Similarly, when N is 7, we get

$$I_7 = \begin{cases} 2\,pkts & \text{when the (1st, 2nd or 3rd) and} \\ & \text{(5th, 6th or 7th) hops are used} \\ 1\,pkt & \text{when the 4th hop is used} \end{cases}$$

$$\overline{I_7} = 2 \times \frac{6}{7} + 1 \times \frac{1}{7}\,pkts, q_7 = \frac{1}{8}, T_7 = 7T$$

When N is 8, $I_8 = \overline{I_8} = 2\,pkts$, when the (1st, 2nd, 3rd or 4th) and (5th, 6th, 7th or 8th) hops are occupied, and

$$q_8 = \frac{1}{8}, T_8 = 8T.$$

5.2.3. Generalization of TCP Service Rate

Let $N = 4M + j$, where $M \geq 1, j \subseteq \{1, 2, 3, 4\}$, the generalization of the above analysis is as follows.

When N is $4M$, we have $I_{4M} = \overline{I_{4M}} = M\ pkts$.
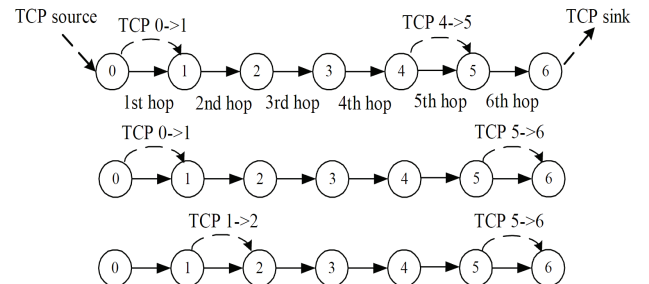
When N is $4M + 1$,

Figure 7. Two TCP packets are transmitted together when $N = 6$.

$$I_{4M+1} = \begin{cases} M = 1\,pkts & \text{when the } 1^{st} \text{ and } 5^{th} \text{ and } \ldots \\ & (4M+1)^{th} \text{ hops are occupied} \\ M\,pkt & \text{when the } (2^{nd},\, 3^{rd} \text{ or } 4^{th}) \text{ and} \ldots \\ & (6^{th},\, 7^{th} \text{ or } 8^{th}) \text{ and } \ldots \\ & ((4M\text{-}2)^{th},\, (4M-1)^{th} \text{ or} \\ & (4M)^{th} \text{ hops are occupied} \end{cases}$$

$$\overline{I_{4M+1}} = (M+1) \times \frac{M+1}{4M+1} + M \times \frac{(4M+1)-(M+1)}{4M+1}\,pkts$$

Similar derivation gives:
When N is $4M+2$,

$$I_{4M+2} = \begin{cases} M+1\,pkts & \text{when the } (1^{st} \text{ or } 2^{nd}) \text{ and} \\ & (5^{th} \text{ or } 6^{th}) \text{and } \ldots \text{ and} \\ & (4M-3)^{th} \text{ or } (4M-2)^{th} \\ & \text{hops are occupied} \\ M\,pkt & \text{when the } (3^{rd} \text{ or } 4^{th}) \text{ and} \\ & (7^{th} \text{ or } 8^{th}) \text{ and } \ldots \text{and} \\ & ((4M\text{-}1)^{th} \text{ or } 4M^{th})) \\ & \text{hops are occupied} \end{cases}$$

$$\overline{I_{4M+2}} = (M+1) \times \frac{2(M+1)}{4M+2} + M \times \frac{(4M+2)-2(M+1)}{4M+2}\,pkts\,.$$

When N is $4M+3$,

$$I_{4M+3} = \begin{cases} M+1\,pkts & \text{when the } (1^{st},\, 2^{nd} \text{ or } 3^{rd}) \text{ and} \\ & (5^{th},\, 6^{th} \text{ or } 7^{th})\text{and } \ldots \text{ and} \\ & (4M-3)^{th} \text{ or } (4M-2)^{th}\text{or} \\ & (4M-1)^{th}\text{hops are occupied} \\ M\,pkt & \text{when the } 4^{th}\,and\, 8^{th}) \text{ and } \ldots \\ & \text{and } 4M^{th} \text{ hops are occupied} \end{cases}$$

$$\overline{I_{4M+3}} = (M+1) \times \frac{3(M+1)}{4M+3} + M \times \frac{(4M+3)-3(M+1)}{4M+3}\,pkts\,.$$

Finally, it is summarized that:

$$\overline{I_{4M+j}} = (M+1) \times \frac{j(M+1)}{4M+j} + M \times \frac{(4M+j)-j(M+1)}{4M+j}\,pkts \tag{10}$$

Where $M \geq 1,\, j \subseteq \{1,2,3,4\}$.

Combining (10) and the analysis when N is 1, 2, 3 or 4 in **Table 1**, we have the average number of packets in the system as:

$$\overline{I_{4M+j}} = \frac{j(M+1)^2 + (4-j)M^2}{4M+j} \tag{11}$$

Where $M \geq 0$, $j \subseteq \{1,2,3,4\}$.

The service time to transmit a packet is generalized as $T_{4M+j} = (4M+j) \times T$. From (9), the TCP service rate is derived as:

$$U_{t(4M+j)} = \frac{j(M+1)^2 + (4-j)M^2}{(4M+j)^2} \times \frac{1}{T} \tag{12}$$

where $M \geq 0,\, j \subseteq \{1,2,3,4\}$.

The average probability of the source to access the channel is:

$$q_{4M+j} = \begin{cases} \dfrac{1}{2j} & \text{when } M=0 \text{ and } j \subseteq \{1,2,3,4\} \\ \dfrac{1}{8} & \text{when } M \geq 1 \text{ and } j \subseteq \{1,2,3,4\} \end{cases} \tag{13}$$

5.3. Generalization of TCP Throughput

Substituting (12) and (13) into (8), the TCP throughput becomes:

$$\lambda_{4M+j} = \begin{cases} \dfrac{1}{j(1+(2j-1)\rho)} \times \dfrac{1}{T} \\ \quad when\ M=0\ and\ j \subseteq \{1,2,3,4\} \\ \dfrac{j(M+1)^2 + (4-j)M^2}{(4M+j)^2} \times \dfrac{1}{1+7\rho} \times \dfrac{1}{T} \\ \quad when\ M \geq 1\ and\ j \subseteq \{1,2,3,4\} \end{cases} \tag{14}$$

When M goes to infinity, $N = 4M + j$ goes to infinity, we have

$$\lim_{M \to \infty} \lambda_{4M+j} = \frac{1}{4} \times \frac{1}{T} \times \frac{1}{1+7\rho} \tag{15}$$

Equation (14) shows that the TCP throughput is independent of the maximum TCP congestion window size W; instead it is decided by (a) $4M + j$, the value of the number of hops of the *string* topology, (b) ρ, the ratio of service rate of TCP packet to ACK packet, and (c) T, the time needed for a TCP packet to be transmitted over a single hop. Furthermore, (15) illustrates that as the number of hops increases and goes to infinity, TCP throughput converges to a constant value.

6. Evaluations

The analytical results are verified by the comparison of

       WSN

simulation results in *ns*2 and numerical results in *matlab*.

In simulations, $N+1$ nodes forma *string* (**Figure 1**) with adjacent nodes 200 *m* apart. All nodes communicate with identical, half-duplex wireless radios which have a bandwidth of 2 *Mbps* and a nominal transmission radius of 250 *m*. Nodes have carrier sense radius of 550 *m* and interference radius of 550 *m*. They are configured with the Dynamic Source Routing (DSR) protocol. One TCP Reno session is introduced from node 0 (TCP source) to node N (TCP destination) to transfer FTP bulk data, crossing N hops. TCP and ACK packets size are of 1460 bytes and 40 bytes respectively. Simulations were run with various numbers of hops and various maximum TCP congestion window sizes. One simulation with the same number of hops and the same maximum TCP congestion window size was run for five times, each for 300 *secs*, and the overall throughput was measured from 50 *secs* to 250 *secs*. During the time the throughput is quite stable, but the average of five simulations was used as the simulation result.

To get the numerical results in *matlab*, two parameters are needed:

- ρ : the ratio of service rate of TCP packet to ACK packet. Here $\rho = \dfrac{40 bytes}{1460 betes} = 0.027$, *i.e.*, the ratio of ACK packet size to TCP packet size.

- $\dfrac{1}{T}$: the average transmission rate of one packet over a single hop link, where T is defined as the average transmission time of one packet over one hop link. However, $\dfrac{1}{T}$ cannot be easily assigned a value considering the bandwidth used on the channel contention, MAC control packets exchange, etc. Instead, we decide the value of $\dfrac{1}{T}$ with the help of simulation results so that the numarical results could best match the simulation results

Figure 8 shows the comparison of simulation and numerical results of the TCP throughput as the number of hops changes from 1 to 11. Simulation results are presented when the maximum TCP congestion window size is 8 packets. Numerical results are presented with two values of $\dfrac{1}{T}$, so there are three curves in **Figure 8**: the simulation results, the numerical results when $\dfrac{1}{T} = 1300$ *kbps*, and the numerical results when $\dfrac{1}{T} = 900$ *kbps* .

It is observed that the numerical results with $\dfrac{1}{T} = 1300$ *kbps* match the simulation results well at $N=1, 2, 3$ or 4 hops , and the numerical results with $\dfrac{1}{T} =$

900 *kbps* match the simulation results well at $N \geq 5$. The reason is as follows. When $N = 1, 2, 3$ or 4 , no link is used simultaneously due to the hidden terminal problem. When $N \geq 5$, there are links used simultaneously because of channel spatial resue. However, although channel spatial reuse helps improve the overall system utilization, it has higher local contention overhead. This results in a lower payload transmission rate. Therefore, $\dfrac{1}{T}$ is chosen as 900 *kbps* when $N \geq 5$ and 1300 *kbps* when $N = 1, 2, 3$ or 4 respectively. In **Figure 8**, TCP throughput is shown to decrease rapidly when the number of hops increases from 1, and stabilizes when the number of hops becomes large. This is in line with the observation we described in the introduction and verifies the analysis in (13) and (14).

Next, we look at the trend of TCP throughputwith varying maximum window size. Simulation results of TCP throughput is shown in **Figure 9** with varying number of hops (from 1 to 11 hops) and TCP maximum window size (from 1 to 12 packets). When the number of hops is fixed, TCP throughput is kept constant independent of the maximum window size, *i.e.*, the throughput curve is almost a straight line parallel to the x-axis. This is consistent with (14) where the TCP throughput (λ)is not a function of TCP maximum window size (W).This result, however, looks inconsistent with the claim in [3] that "given a specific network topology and flow patterns, there exists a TCP window size W^*, at which TCP achieves best throughput via improved spatial channel
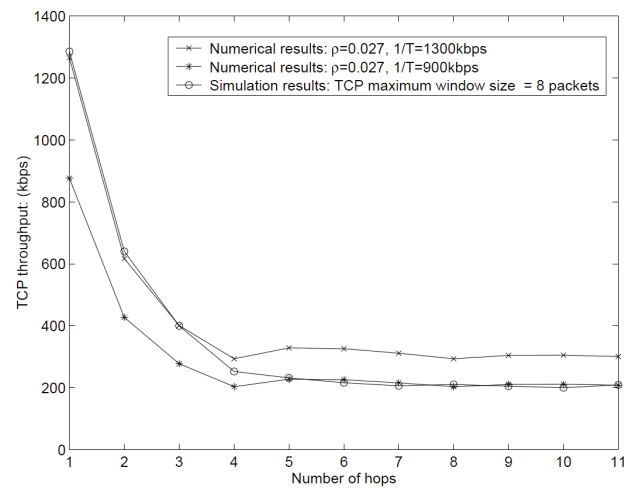

Figure 8. Comparison of simulation and numerical results: TCP throughput with varying number of hops. Simulations run for 300 seconds.
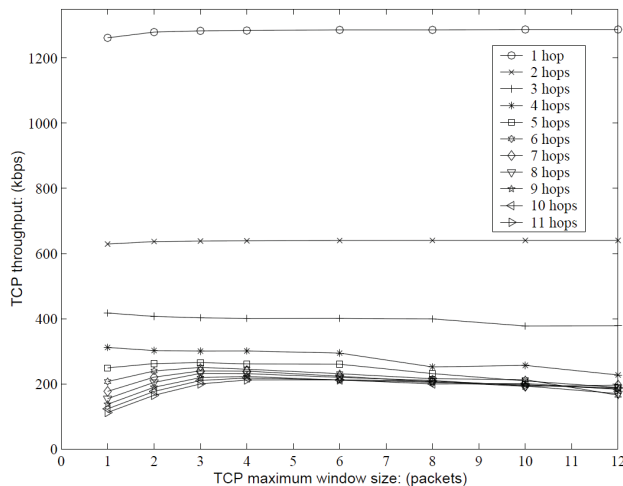
Figure 9. Simulation result: TCP throughput with varying number of hops and maximum window size, simulations run for 300 seconds.

reuse". However, a close look at **Figure 2 (a)** and **(b)** in [3] reveals that TCP throughput at the "optimal" TCP window size is not much higher than those at non-optimal TCP window sizes; furthermore, TCP throughput is a constant at most values of TCP window size. This observation is consistent with our analysis here.

We notice a phenomenon in the left lower part of the figure. When the number of hops is greater than 5 and the maximum window size is 1 or 2 packets, TCP throughput is less than the constant value when the maximum window size is greater or equal to 3 packets. This is due to the fact that we assume the system is always busy, *i.e.*, all the available links are fully used at every moment. However when the window size is 1 packet, the whole system goes into a stop-and-wait process and the source will only issue a TCP packet after it receives an ACK from the destination. Therefore the system is not fully occupied. The achieved throughput is thus less than the fully occupied value. The same reason explains the case when the maximum window size is 2 packets but the *string* topology can support a maximum of 3 packets when the number of hops is 9, 10, or 11.

The above analytical result is interesting as it shows that the setting of TCP congestion window size does not matter that much in wireless multihop networks, which is different fromthe scenario in wired networks where TCP performance is largely affected by its congestion window size. However, our analysis considers single a TCP connection over static chain without interfering background traffic. Further evaluations are necessary.

7. Conclusions

This study takes a step to theoretically analyze TCP performance in adhoc wireless networks. The analysis is

based on a *string* topology containing *N* hops. A system model for analyzing TCP performance in adhoc wireless networks is proposed, which considers packet buffering, contention of nodes for access to wireless channel and spatial reuse of wireless channel. Markov chain modelling is applied to analyze the system model. Analytical results are presented to show that when the number of hops that the TCP session crosses is fixed, the TCP throughput is independent of the TCP congestion window size. When the number of hops increases from one, the TCP throughput decreases first, and then stabilizes when the number of hops becomes large. The analysis is validated by comparing the numerical results with the simulation results.

8. References

[1] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Proceedings of IEEE/ACM MobiCom*'99, Seattle, August 1999, pp. 219-230.

[2] M. Gerla, R. Bagrodia, L. Zhang, K. Tang and L. Wang, "TCP over wireless multi-hop protocols: simulation and experiments," *Proceedings of IEEE ICC*'99, Vancouver, Vol. 2, June 1999, pp. 1089-1094.

[3] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang and M. Gerla, "The impact of multihop wireless channel on TCP performance," *IEEE Transactions on Mobile Computing*, Vol. 4, No. 2, March 2005, pp. 209-221.

[4] H. Xiao, K. G. Seah, A. Lo and K. C. Chua, "On service differentiation in multihop wireless networks," *ITC Specialist Seminar on Mobile Systems and Mobility*, Lillehammer, March 2000, pp. 1-12.

[5] X. Chen, H. Zhai, J. Wang and Y. Fang, "TCP performance overmobile ad hoc networks," *Journal of Electrical & Computer Engineering*, Vol. 29, No. 1/2, 2004, pp. 129-134.

[6] K. Chandran, S. Raghunathan, S. Venkatesan and R. Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks," *IEEE Personal Communications*, Vol. 8, No. 1, 2001, pp. 34-39.

[7] Z. Fu, X. Meng and S. Lu, "A transport protocol for sup porting multimedia streaming in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, 2003, pp. 1615-1626.

[8] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang and M. Gerla, "The impact of multihop wireless channel on TCP performance," *IEEE INFOCOM*'03, San Francisco, Vol. 3, March 2003, pp. 1744-1753.

[9] X. Yu, "Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness," *Proceedings of IEEE/ACM MobiCom*'04, Philadelphia, September 2004, pp. 231-244.

[10] R. Oliveira and T. Braun, "A smart TCP acknowledgment approach for multihop wireless networks," *IEEE Transactions on Mobile Computing*, Vol. 6, No. 2, 2007, pp.

192-205.

[11] X. Li, P. Y. Kong and K. C. Chua, "DTPA: A reliable datagram transport protocol over ad hoc networks," *IEEE Transactions on Mobile Computing*, Vol. 7, No. 10, 2008, pp. 1285-1294.

[12] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad hoc networks with out-of-order dectetion and response," *Proceedings of IEEE/ACM MobiHoc'02*, Lausanne, June 2002, pp. 217-225.

[13] K. Chen, Y. Xue and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks," *Proceedings of IEEE ICC'03*, Anchorage, Vol. 2, May 2003, pp. 1080-1084.

[14] K. Xu, M. Gerla, L. Qi and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED," *Proceedings of IEEE/ACM MobiCom'03*, San Diego, September 2003, pp. 16-28.

[15] L. Yang, W. K. G. Seah and Q. Yin, "Improving fairness among TCP flows crossing wireless ad hoc and wired networks," *Proceedings of IEEE/ACM MobiHoc'03*, New York, June 2003, pp. 57-63.

[16] V. Anantharaman, S. J. Park, K. Sundaresan and R. Sivakumar, "TCP performance over mobile ad hoc networks: a quantitative study," *Wireless Communications and Mobile Computing*, Vol. 4, No. 2, 2004, pp. 203-222.

[17] K. Nahm, A. Helmy and C.-C. Jay Kuo, "TCP over multihop 802.11 networks: issues and performance enhancement," *Proceedings of MobiHoc'05*, New York, 2005, pp. 277-287.

[18] F. Klemm, Z. Ye, S. V. Krishnamurthy and S. K. Tripathi, "Improving TCP performance in ad hoc networks using signal strength based link management," *Ad Hoc Networks*, Vol. 3, No. 2, 2005, pp. 175-191.

[19] H. Xiao, K. C. Chua, K. G. Seah and A. Lo, "A quantitative analysis of TCP performance over wireless multihop networks," *Proceedings of IEEE VTC2001-fall*, Atlantic, Vol. 2, May 2001, pp. 610-614.

[20] H. Xiao, K. C. Chua, J. A.Malcolm and Y. Zhang, "Theoretical analysis of TCP throughput in adhoc wireless networks," *IEEE Global Telecommunications Conference*, Atlanta, Vol. 5, 2005, pp. 2714-2719.

[21] X. Li, P. Y. Kong and K. C. Chua, "TCP performance in IEEE 802.11-based ad hoc networks with multiple wireless lossy links," *IEEE Transactions on Mobile Computing*, Vol. 6, No. 12, 2007, pp. 1329-1342.

[22] T. V. Lakeshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, June 1997, pp. 336-350.

[23] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, August 1998, pp. 485-498.

[24] M. Mathis, J. Semske, J. Mahdavi and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, Vol. 27, No. 3, July 1997, pp. 67-82.

[25] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACMComputer Communication Review*, Vol. 28, No. 4, September 1998, pp. 303-314.

[26] C. Casetti and M. Meo, "A new approach to model the stationary behavior of TCP connections," *Proceedings of IEEE INFOCOM* 2000, Tel-Aviv, Vol. 1, March 2000, pp. 367-375.

[27] V. Paxson and S. Floyd, "Empirically-derived analytic models of wide-area TCP connections," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, August 1994, pp. 316-336.

[28] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *ACM Computer Communication Review*, Vol. 24, No. 4, October 1994, pp. 257-268.

[29] H. M. Chaskar, T. V. Lakshman and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 5, October 1999, pp. 605-615.

[30] T. V. Lakshman, U. Madhow and B. Suter, "TCP/IP performance with random loss and bidirectional congestion," *IEEE/ACM Transactions on Networking*, Vol. 8, No. 5, October 2000, pp. 541-555.

[31] V. Bharghavan, A. J. Demers, S. Shenker and L. Zhang, "MACAW: A media access protocol for wireless LAN's," *ACM SIGCOMM*, Vol. 24, No. 4, 1994, pp. 212-225.

[32] D. Bertsekas and R. Gallager, "Data Networks," 2nd Edition, Prentice Hall, London, 1992.