Congestion Avoidance Routing Protocol for Ad-Hoc Networks

YI LU
Microsoft
One Microsoft Way, Redmond, WA, USA
yi.lu@microsoft.com
and
BHARAT BHARGAVA
Department of Computer Science, Purdue University
305 N. University Street, West Lafayette, IN, USA
bb@cs.purdue.edu

The objective of this research is to bring the consideration of congestion into the design of ad hoc routing protocols. The main thrust is to avoid congestion by minimizing contentions for channel access. The intermediate delay (IMD) is proposed to replace the hop count as a routing metric. IMD estimates the delay introduced by the intermediate nodes along the route using the sum of delays from each node. It characterizes the impacts of channel contention, traffic load, and the length of a route. The self-adjusting congestion avoidance (SAGA) routing protocol is designed to use IMD as the routing metric. SAGA reduces congestion at intermediate nodes. It selects routes that bypass the hot spot where contention is intense.

The packet transmission procedure of the distributed coordination function (DCF) in the IEEE 802.11 standard is analyzed and used as a study case for evaluation and experimentation. An estimate of the transmission delay is derived based on local information available at a node. The estimation takes the impact of active traffic in the neighborhood into account without exchanging messages with neighbors.

The performance of SAGA is evaluated and compared with that of ad hoc on-demand distance vector (AODV), dynamic source routing (DSR), and destination-sequenced distance-vector (DSDV) protocols using simulation. TCP and two types of UDP traffic are considered: constant bit rate traffic and traffic exhibiting long range dependency. SAGA can sustain heavier traffic load and offers higher peak throughput than AODV and DSR. The overhead of SAGA can be as low as 10% as that of AODV and 12% as that of DSR. The average end-to-end delay of SAGA is the lowest among the protocols. It is shown that considerations of congestion and intermediate delay instead of the hop count can enhance routing performance significantly.

Keywords: Ad hoc networks, congestion avoidance, routing protocols

1. INTRODUCTION

A mobile ad hoc network is a collection of mobile nodes that are deployed as a multi-hop wireless network without the aid of any preexisting infrastructure. The network connectivity and functionality are maintained through cooperations among nodes. Ad hoc networks use wireless links, which have significantly lower capacity than their hardwired counterparts (e.g., 54Mbps for 802.11g versus 9.952Gbps for OC192). The real throughput of a wireless link is affected by multiple access, fading, noise, and interference conditions. It is usually lower than the maximum transmission rate. The aggregated traffic demand easily reaches or exceeds the link capacity. Congestion is typically the norm rather than exception in ad hoc networks [Corson and Macker 1999].

Many research efforts that address the congestion avoidance/control problem are based on the principle of conservation of packets [Jacobson 1988]. Examples include TCP and its varieties [Brakmo et al. 1994; Mo et al. 1999; Padhye et al. 2000]. The conventional TCP-type mechanisms use packet loss to infer congestion and provide per-connection congestion control. In ad hoc networks, wireless transmission loss (high bit-error rate) and route reconstruction (network partition) are significant causes for packet loss. They degrade the effectiveness of congestion inference schemes [Biaz and Vaidya 1998; Liu and Singh 2001]. Mechanisms have been proposed to improve TCP's performance over wireless and ad hoc networks [Parsa and Garcia-Luna-Aceves 2000; Chandran et al. 2001; Zhang and Tsaoussidis 2001; Wang and Zhang 2002; Cen et al. 2003; Kunniyur and Srikant 2003]. The essence of TCP congestion control algorithms is to reduce the sending rate of traffic upon the occurrence of a congestion.

Two characteristics of ad hoc networks are the existence of multiple routes and the node-based routing. Routing protocols can make use of them to reduce network congestion with little sacrifice in the sending rate of traffic. Routing with load balancing has been investigated in [Toh 1997; Hassanein and Zhou 2001; Lee and Gerla 2001]. The idea is to provide extra information, such as a secondary metric based on the current load on each node, to help in distributing traffic load. It prevents a single node from being overwhelmed. In an ad hoc network, the wireless channel is shared by multiple nodes. They contend for the channel not only for sending but also for receiving packets because of the hidden terminal problem [Bertossi and Bonuccelli 1995]. The experimental study in [Lu et al. 2003] shows that contention for the channel is the primary reason for network congestion. The impact of the channel contention should be taken into account in the congestion reduction. For example, if the contention is already intense among a node's neighbors, it should not be chosen to forward packets even if there is no load on the node itself.

The main thrust of our work is to reduce network congestion by minimizing channel contentions. The objective is to avoid the *hot* spots where multiple nodes are in contention with each other. The global coupling effect of wireless channel access in ad hoc networks poses a challenge in determining the contentions locally. In addition, traffic load on a node must be taken into account, as the store-and-forward process may also cause congestion when the capacity of a node is exceeded. The shorter routes should be given higher priority because they are less likely to be involved in contentions with other nodes.

Our approach for reducing contention is as follows: (1) A single server queueing system is used to model nodes. The impact of channel contention is quantified by using the service time (the time to successfully transmit a packet over the channel). The routing cost at each node is computed as the estimated delay. It reflects the effects of channel contention, current load, and expected load in the immediate future. (2) When a node has active traffic, statistical methods are used to evaluate the mean of the delay. In absence of active traffic, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of the delay. (3) The intermediate delay (IMD) routing metric is proposed to measure the communication delay introduced by the nodes connecting the source and destination. (4) The self-adjusting congestion avoidance (SAGA) routing protocol is designed to avoid network congestion. Lazy route query operation that is presented in section 4 is used by SAGA to accelerate the establishment of needed routes. Experimental studies are conducted to evaluate the performance of SAGA and compare it with AODV [Perkins et al. 2003], DSR [Johnson and Maltz 1996], and DSDV [Perkins and Bhagwat 1994] protocols.

This research is conducted in the framework of CSMA/CA (carrier sense multiple access with collision avoidance) paradigm, which is adopted by the widely used IEEE 802.11 standard. For unicast packets, CSMA/CA requires the sender and receiver to exchange the request-to-send/clear-to-send (RTS/CTS) frames prior to the transmission of the actual data frame. Broadcast packets are sent out without RTS/CTS. In this paper, packets refer to unicast packets unless otherwise stated.

The rest of this paper is organized as follows. Section 2 illustrates the idea of ad hoc routing based on the intermediate delay. Two methods are proposed in section 3 to estimate delay locally. Section 4 presents the detail of SAGA protocol. In section 5, the performance of the proposed protocol is evaluated and compared with AODV, DSR, and DSDV. The related work is discussed in section 6. Section 7 gives analysis and guidelines resulting from this research.

2. AD HOC ROUTING BASED ON INTERMEDIATE DELAY

The intermediate delay (IMD) routing metric estimates the communication delay introduced by the intermediate nodes using the sum of delays from each node. It characterizes the impacts of channel contention, the length of the route, and traffic load at individual nodes. The following imaginary examples show the use of the IMD metric in ad hoc routing. For the purpose of demonstration, in these examples, we simply assume the delay at a node is X if there is no contending neighbor; it is nX if there are n contending neighbors. The methods to estimate the delay in realistic scenarios are presented in section 3.

The examples use a ten-node ad hoc network as shown in figures 1 and 2. A line between two nodes denotes that they are neighbors and within each other's transmission range. Neighbors will contend for the channel in order to send or receive a packet. A line with an arrow head represents a connection session, i.e. a series of packets being sent from the source to the destination. A connection between nodes F and G is to be established. Figure 1 illustrates the route selection

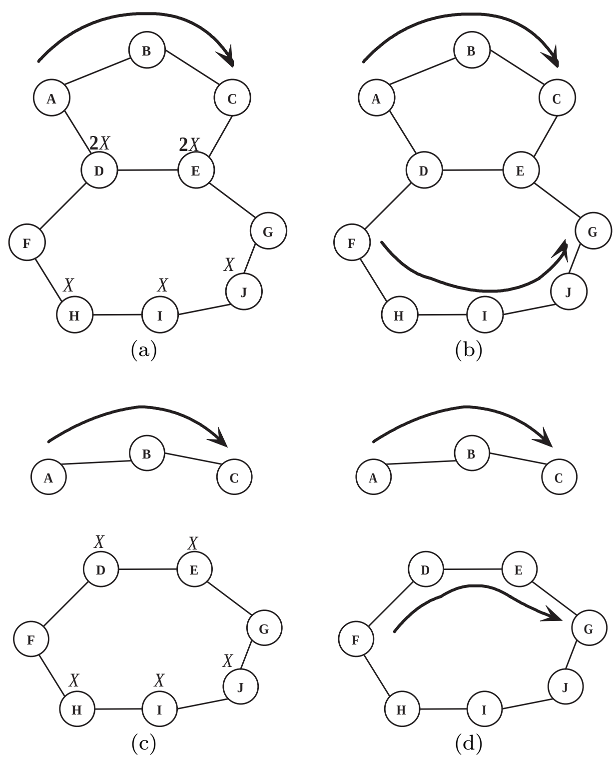


Figure. 1: Select a route with presence of other connections and then adapt to changes in network topology

process and the adaption to changes in network topology. As shown in figure 1a, there is a connection session between A and C when F wants to establish a connection with G. D is aware of the contention with A and computes the delay to be $2X$. Similarly, E's delay is $2X$. The delay computed by nodes H, I, and J is X. The IMD of the route F→D→E→G is $4X$, while that of the route F→H→I→J→G is $3X$. The later route is chosen even though it is one hop longer (figure 1b). This route is better in terms of channel reuse and congestion avoidance. It introduces a lower end-to-end delay.

Suppose nodes A and C have moved and are no longer contending with D and E. F will observe that the route F→D→E→G has become better since its IMD is $2X$. The connection is re-established as shown in figure 1d.

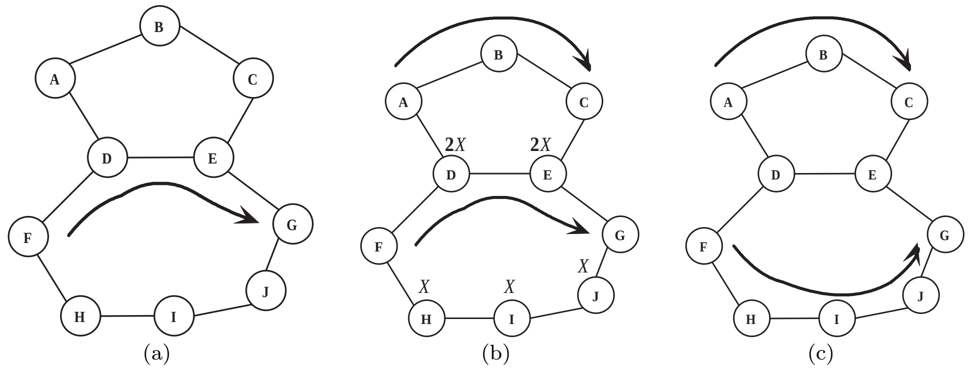


Figure. 2: Adapt to changes in traffic

Figure 2 illustrates the adaption to changes in traffic. At the beginning, there is no traffic in the network. Every node introduces a communication delay of X. The shortest route in terms of hop count is chosen to establish the connection (figure 2a), since it has the smallest IMD. After the establishment of the connection, a new connection session from A to C is established. This connection will follow its best route A→B→C. The new connection causes channel contention between A and D as well as C and E. The new delays at D and E are $2X$. The IMD of the route F→D→E→G changes to $4X$. The route F→H→I→J→G has become a better choice, as shown in figure 2b. Node F re-establishes the connection via the new route. Figure 2c shows the result after adapting to the new connection session.

These examples demonstrate the essential idea of congestion avoidance by using IMD. For the design of a practical routing protocol, we must consider the following: (1) At the time a node computes the delay, it may not know the number of neighbors who are contending with it. (2) Due to the locality of contention, access to a wireless channel creates global coupling effects in the entire network [Luo et al. 2001]. Even if the number of contending nodes is known, the share of capacity cannot be predetermined. (3) The successive links of a route may interfere with each other.

3. DELAY ESTIMATION

Estimating the delay for sending a packet is critical in SAGA protocol. It is impractical to compute the accurate delay due to the dynamics and complexity of the network. Furthermore, an accurate value is not required because the delay is transient. The proposed methods calculate an approximation of the delay.

3.1 The model

A node can be modeled as a single server queueing system [Gelenbe and Pujolle 1998]. The following assumptions are made for delay estimation.

(1) The incoming traffic is localized with respect to time, i.e., in a short period of time, it obeys approximately the same distribution.
(2) The channel access is localized with respect to both time and location. If a node finds that the channel is busy, so do its neighbors.
(3) A node has a queue of sufficient size.
(4) The incoming traffic rate and outgoing traffic rate are independent. This assumption is based on (a) the complexity of channel contentions washes out the dependency and (b) the

incoming traffic includes packets coming from other mobile nodes as well as from upper layer applications.

(5) The incoming traffic and outgoing traffic are Poisson processes.

The assumptions reduce the complexity of the computation and allow us to develop a model to compute the delay. The computation yields a good estimate of the real delay. As shown in section 3.4, the relative error between the computed delay and the measured delay is less than 10%. The simulation results presented in section 5 show that SAGA protocol significantly improves the performance of routing by using the proposed delay estimation methods.

The following notations are used to describe the parameters of the queueing system.

Δt: The interval between two consecutive delay estimations.

λ: The arriving rate of packets. It is estimated using $\frac{N_A}{\Delta t}$. N_A is the number of packets arrived within the time interval Δt.

μ: The service rate, i.e., the number of packets transmitted over the wireless channel per second. The capacity of the channel and the contention with neighbors determine this parameter.

T_Q: The wait in the queue before a packet is transmitted.

T_S: The average service time for transmitting a packet ($T_S = \frac{1}{\mu}$).

T_D: The total delay at a mobile node ($T_D = T_Q + T_S$).

L: The current length of the queue.

If $\lambda \geq \mu$, the maximum allowed value of the delay is assigned to T_D since the wait in queue T_Q may be arbitrarily large [Gelenbe and Pujolle 1998]. Otherwise, T_Q can be evaluated using equation 1 by applying the Little's law [Gelenbe and Pujolle 1998].

$$T_Q = \frac{\lambda}{\mu(\mu - \lambda)} + T_S L \tag{1}$$

The delay T_D is calculated as follows.

$$T_D = T_Q + T_S = \frac{T_S(L+1) - \frac{N_A}{\Delta t}(T_S)^2 L}{1 - \frac{N_A}{\Delta t} T_S} \tag{2}$$

L and N_A can be easily computed. Two cases are considered in the estimation of the service time T_S: a node with recent traffic (i.e., it has sent out unicast packets over the wireless channel since the last delay estimation) and a node without recent traffic.

3.2 Node with recent traffic

The mean value of the service time can be calculated using the statistical method. Let N_S be the number of packets and T_B be the time that the node spent on transmitting packets. T_B is less than or equal to Δt because the node may not be transmitting packets all the time.

$$T_S = \frac{T_B}{N_S} \tag{3}$$

The delay T_D is computed using equation 2 once the value of T_S is obtained.

3.3 Node without recent traffic

No recent traffic on a node does not imply that a packet can be sent with the smallest delay. This is because the neighbors may be using the channel. The expectation of the service time can be determined by using probability methods to study the procedure of packet transmission. The IEEE 802.11 distributed coordination function (DCF) is analyzed as an example. The methods are applicable to other MAC protocols.

Figure 3 illustrates the procedure of transmitting a unicast packet using RTS/CTS. We briefly review it for the purpose of evaluating the expectation of the transmission time. The detailed description of the process is available in the IEEE 802.11 standard.

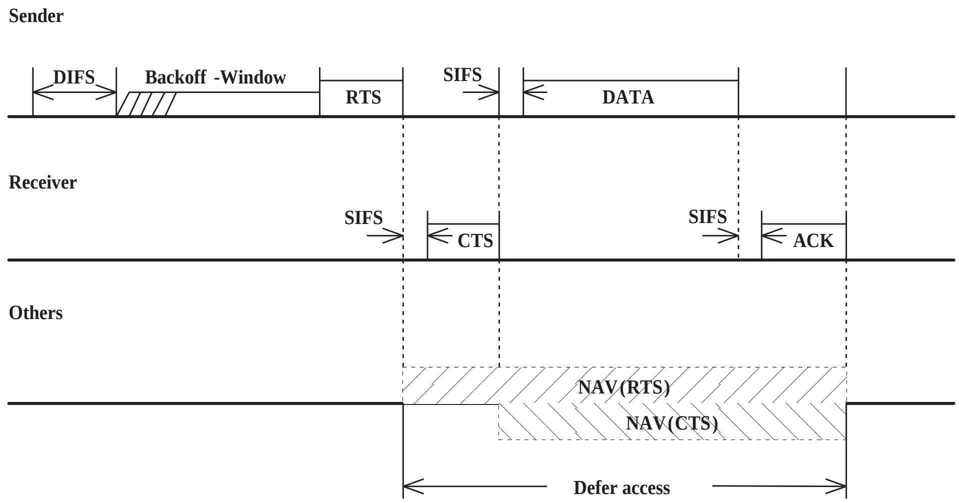


Figure. 3: Transmission of a unicast packet using RTS/CTS in the IEEE 802.11 standard

When a packet is ready to transmit, the sender picks up a backoff time $b \times T_{slot}$ after observing an idle channel for the time period T_{DIFS}. b is a random number uniformly distributed over $[0, CW]$. T_{slot} and T_{DIFS} are values specified by the physical layer. The sender starts to transmit the RTS frame when the backoff time reaches zero. The receiver transmits a CTS frame after time T_{SIFS} upon receiving the RTS frame, if the media is idle. The neighbors of the sender and receiver set the network allocation vector (NAV) correspondingly to indicate that the media is reserved. The sender waits for time T_{SIFS} after receiving the CTS frame and then transmits the data. The receiver waits for time T_{SIFS} after receiving the data and replies with an acknowledgement (ACK) frame. The expectation of the transmission time for a successful attempt is

$$E[T_{succ}] = T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3T_{SIFS} + E[T_{backoff}] \quad (4)$$

where T_{DATA}, T_{RTS}, T_{CTS} and T_{ACK} are, respectively, time for transmitting a data frame, a RTS frame, a CTS frame, and an ACK frame. T_{DIFS} and T_{SIFS} are the DCF interframe and short interframe time, and $T_{backoff}$ is the time spent on the backoff procedure.

The attempt fails if a CTS frame has not been received at the end of $T_{timeout}$ period following the transmission of the RTS frame. The sender then restarts this procedure. The expected time spent on a failed attempt is

$$E[T_{fail}] = T_{RTS} + T_{timeout} + E[T_{backoff}] \quad (5)$$

Now we compute the expected time spent on the backoff procedure $E[T_{backoff}]$. According to the assumption 2 in section 3.1, the probability that a channel is busy in a unit time (i.e., the smallest time unit in the MAC specification) will not change during the transmission period. It is denoted as p. Observing an idle channel for time t is then a Bernoulli trial [Mendenhall and Beaver 1991]. It stops if the channel has been idle for a continuous time t. Let $\mathcal{T}_{idle}(t)$ be the time needed for this trial. The expectation of $\mathcal{T}_{idle}(t)$ is computed recurrently as follows for a given p.

$$E[\mathcal{T}_{idle}(t)] = E[\mathcal{T}_{idle}(t-1)] + (1-p) * 1 + p * (1 + E[\mathcal{T}_{idle}(t)]) \quad (6)$$

Equation 6 aggregates two cases, assuming that the channel has been idle for a continuous time $t-1$ in the trial: (1) The channel is idle in the next unit of time and the trial stops. The probability for this case is $1-p$. (2) Otherwise, the trial still needs time $\mathcal{T}_{idle}(t)$ to stop. Solving this recurrence formula with the initial condition $E[\mathcal{T}_{idle}(1)] = \frac{1}{1-p}$ yields

$$E[\mathcal{T}_{idle}(t)] = \frac{1 - (\frac{1}{1-p})^{t+1}}{1 - \frac{1}{1-p}} = \frac{(\frac{1}{1-p})^t - 1}{p} + 1$$

Let $\mathcal{T}_{backoff}(b)$ be the time needed for a backoff procedure reaching 0 from $b \times T_{slot}$.

$$\mathcal{T}_{backoff}(b) = \mathcal{T}_{idle}(T_{DIFS}) + \sum_{i=1}^{b} t_0, \quad t_0 = T_{slot} \text{ or } T_{slot} + \mathcal{T}_{idle}(T_{DIFS} + T_{slot})$$

t_0 is the time needed to decrease the backoff time by T_{slot}. There are two cases.

(1) During the backoff procedure, if no channel activity is detected for the duration of a particular backoff slot, the backoff time is decreased by T_{slot}. The probability is $(1-p)^{T_{slot}}$. In this case, $t_0 = T_{slot}$.

(2) Otherwise, the procedure is suspended without decreasing the backoff time. It resumes after observing an idle channel for time T_{DIFS}. To decrease the backoff time, the channel must be idle for a continuous time $T_{DIFS} + T_{slot}$. $t_0 = T_{slot} + \mathcal{T}_{idle}(T_{DIFS} + T_{slot})$.

The expectation of t_0 is

$$E[t_0] = (1-p)^{T_{slot}} T_{slot} + (1 - (1-p)^{T_{slot}})(T_{slot} + E[\mathcal{T}_{idle}(T_{DIFS} + T_{slot})])$$

For a given random number b, the expectation of the time spent on the backoff procedure is

$$E[\mathcal{T}_{backoff}(b)] = E[\mathcal{T}_{idle}(T_{DIFS})] + \sum_{i=1}^{b} E[t_0] = E[\mathcal{T}_{idle}(T_{DIFS})] + bE[t_0]$$

The expected time for the backoff procedure in the transmission attempt is

$$E[T_{backoff}] = E[E[\mathcal{T}_{backoff}(b)]] = E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW}{2} E[t_0] \tag{7}$$

The parameters CW (contention window), $aCWmin$, and $aCWmax$ are defined in the IEEE 802.11 standard. CW takes an initial value of $aCWmin$ for the first attempt. Every time an attempt fails, CW takes the next value in the series, until it reaches $aCWmax$. A successful attempt resets CW to $aCWmin$. The CW values are powers of 2 minus 1, sequentially ascending from $aCWmin$ to $aCWmax$. They are specific to the physical layer. For example, direct sequence spread spectrum (DSSS) physical layer management information base (MIB) sets $aCWmin$ to 31 and $aCWmax$ to 1023. In this paper, we assume DSSS is used as the physical layer. Let CW^n be the CW of the n-th attempt to transmit, then

$$CW^n = \begin{cases} 2^{n+4} - 1, & 1 \leq \text{n} \leq 6; \\ 2^{10} - 1, & \text{n} > 6. \end{cases}$$

Let T^n_{succ} and T^n_{fail} be the time spent on a successful transmission and a failed transmission for the n-th attempt, respectively. From equations 4, 5, and 7, we have

$$E[T^n_{succ}] = T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK} + 3T_{SIFS} + E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW^n}{2} E[t_0]$$

$$E[T^n_{fail}] = T_{RTS} + T_{timeout} + E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW^n}{2} E[t_0]$$

The receiver gets the RTS frame if there is no collision during the transmission of the frame. It will transmit a CTS frame after time T_{SIFS} if the NAV indicates that the channel is idle. Otherwise, the receiver will not respond to the RTS frame (we assume this is the major reason for a failed attempt). The channel must be idle in this duration of $T_{RTS} + T_{SIFS}$ for a successful RTS/CTS exchange. Since channel access has locality characteristic, the probability of a successful transmission P_s is approximately $(1-p)^{T_{RTS} + T_{SIFS}}$.

The expected transmission time makes sense only for successfully delivered data packets. We assume that there is no limit on retry and the sender will keep trying until the packet is delivered. The expected transmission time is

$$E[T_{trans}] = P_s E[T^1_{succ}] + \sum_{i=1}^{\infty}((1-P_s)^i P_s(E[T^{i+1}_{succ}] + \sum_{j=1}^{i} E[T^j_{fail}])) \tag{8}$$

CW^n is fixed when $n \geq 6$, so are $E[T^n_{succ}]$ and $E[T^n_{fail}]$. Solving equation 8 yields

$$\begin{aligned} E[T_{trans}] = \quad & P_s E[T^1_{succ}] + \sum_{i=1}^{5}((1-P_s)^i(E[T^{i+1}_{succ}] - E[T^i_{succ}] + E[T^i_{fail}])) \\ & + \sum_{i=6}^{\infty}((1-P_s)^i E[T^6_{fail}]) \\ E[T_{trans}] = \quad & P_s E[T^1_{succ}] + \sum_{i=1}^{5}((1-P_s)^i(E[T^{i+1}_{succ}] - E[T^i_{succ}] + E[T^i_{fail}])) \\ & + (1-P_s)^6 \frac{1}{P_s} E[T^6_{fail}] \end{aligned} \tag{9}$$

Once the physical layer parameters are determined, the time for transmitting a packet T_S is estimated by $E[T_{trans}]$, which can be computed by using equation 9 with the given probability that the channel is busy. The total delay is calculated by applying the value of T_S to equation 2.

3.4 Accuracy of delay estimation

Experiments are needed to evaluate the accuracy of the proposed delay estimation methods. Two nodes A and B are put in an ad hoc network that has active traffic. Node A randomly sends dummy packets to node B, which contains a time stamp, the estimated delay, and garbage data. The size of the dummy packet is the same as that of the real data packet. The measured delay is computed using the time the packet is sent and the time stamp in the packet. Four dummy packet are sent per 10 seconds, so that the evaluation has little impact on the real traffic.

The results of six experiments are shown in Figure 4. It can be observed that both methods produce reasonable good estimates of the delay. For the statistics based method, the relative error between the estimated value and the measured value is less than 1.5%. The relative error for the probability based method is less than 8.5%.

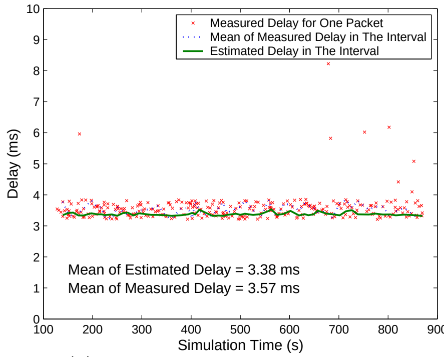
Both methods compute the estimated delay at a node in a constant time. The IMD metric of a route is obtained by aggregating the delays from every intermediate node along the route. In the proposed delay estimations, the impact of active traffic in the neighborhood is reflected by the service time or the probability of a busy channel. The estimation of delay can be done without exchanging information with neighbors.

4. SELF-ADJUSTING CONGESTION AVOIDANCE ROUTING PROTOCOL

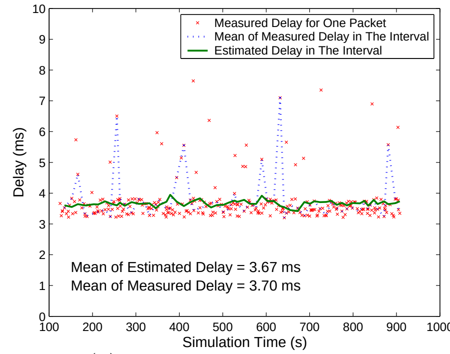
4.1 Introduction

The self-adjusting congestion avoidance (SAGA) routing protocol is designed based on the ideas presented in sections 2 and 3. SAGA is a distance vector routing protocol. One of the major differences between SAGA and other distance vector based routing protocols is that SAGA uses IMD instead of hop count as the distance. It gives SAGA the capability of balancing traffic load and dealing with congestion.

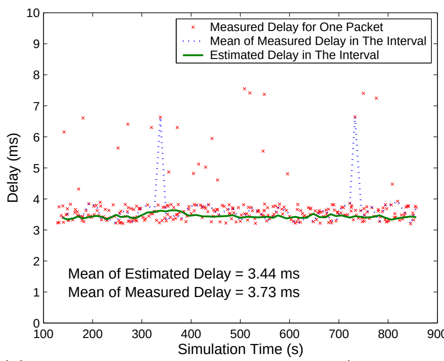
To send packets, every node maintains a routing table that contains entries to all known nodes in the network. *seq* is a field of a routing entry that stores the sequence number representing the "freshness" of a route as in DSDV and AODV. It is maintained by the destination. Routes



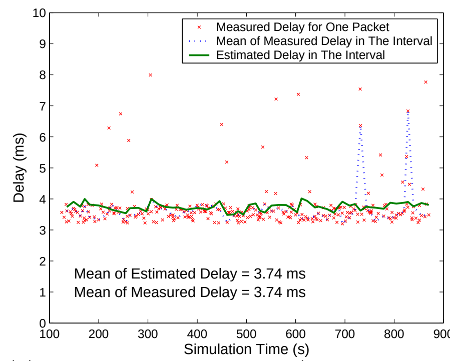
(a) Sender without recent traffic



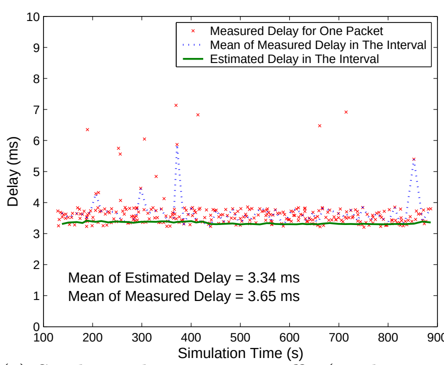
(b) Sender with recent traffic



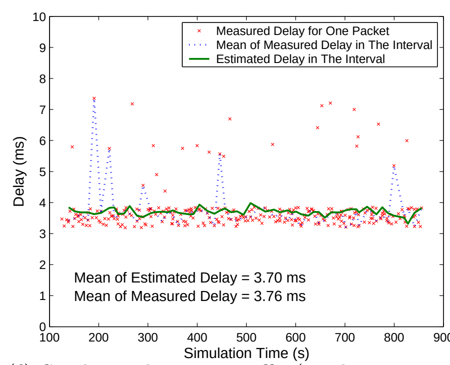
(c) Sender without recent traffic (receiver is sending packets to other nodes)



(d) Sender with recent traffic (receiver is sending packets to other nodes)



(e) Sender without recent traffic (sender is receiving packets from other nodes)



(f) Sender with recent traffic (sender is receiving packets from other nodes)

Figure. 4: Comparison of estimated delay and measured delay

with more recent sequence numbers are always preferred for routing decisions. For routes with identical sequence number, the one with the smallest IMD is chosen.

SAGA is a proactive protocol like DSDV, which requires every node to periodically advertise the routing table to its neighbors. Significant new information such as a new route or a broken route may also trigger an advertisement. The estimated delay at this node is included in advertisement packets. A broken or unavailable route is assigned a delay of ∞, which is a value greater than the maximum allowed value of the delay. A route with ∞ delay is considered as invalid and is usually not included in advertisements.

4.2 Operations

SAGA protocol uses the following operations to estimate the delay, maintain the routing table, and discover needed routes. The route advertisement, broken link handling, and packet forwarding procedures are similar as those in DSDV. They are not discussed in this paper.

Delay estimation. Every node estimates the delay using the methods presented in section 3. The number of arrived packets N_A, the number of sent packets N_B, the time during which the node is transmitting packets T_B, the current length of the queue L, and the probability of a busy channel p are needed for estimation. The duration Δt determines how frequently the delay is estimated. It is set to the time interval between two full advertisements. N_A, N_B, and T_B are counted using a MAC callback function. This function is invoked when a packet arrives at MAC, when a packet is ready to transmit, and after a packet is transmitted. The probability p is determined by using randomly sampling. $p = \frac{N_{busy}}{N_{sample}}$, where N_{sample} is the number of samples and N_{busy} is the number of samples that indicate a busy channel. Each node maintains a sampling timer, which will be randomly triggered about 200 times per second. When the timer is triggered, SAGA checks state of the channel. This timer is set when a new estimate process begins. It is cleared if active traffic is detected.

Route maintenance. Each routing entry in SAGA has two fields associated with the distance: (a) *imd* stores the intermediate delay and is used for route advertisement; and (b) *min_advertised_imd* stores the minimum value of *imd* in all advertisements since the last update of *seq*. Assume that a node i receives an advertisement of a route to a node x from a neighbor j. The route has a sequence number seq_j^x and an intermediate delay imd_j^x. Node i updates its routing table if and only if one of the following four conditions is true.

(1) Node i does not have a valid route to the destination x.

(2) Node j is the next hop of the current route.

(3) The new route contains a fresher (valid) sequence number ($seq_j^x > seq_i^x$) and $imd_j^x < \infty$.

(4) $seq_j^x = seq_i^x$ and $min\_advertised\_imd_i^x > imd_j^x$.

These constraints guarantee that SAGA will not introduce loops in routes. In the third condition, as proved in [Perkins and Bhagwat 1994], a loop cannot be created if nodes use fresher sequence numbers to pick routes. The loop-free property holds in the fourth condition due to the theorem proved in [Jaffe and Moss 1982], which states that distance vector algorithms always maintain loop-free routes in presence of static or decreasing link weights.

Some distance vector routing protocols use a single field of distance for a routing entry. This field is used for both route advertisement and routing decision. Because *imd* reflects the extent of congestion along a route, its value may change even if the route is static. As *imd* is not static or decreasing, the loop-free property may not hold if it is used to make routing decisions as in the fourth condition. The use of *min_advertised_imd* assures loop-free routes.

Adaptive routing metrics such as *imd* sometimes suffer from oscillation. After choosing a route and beginning to send packets, other routes become attractive. The tendency of the routing decision to switch excessively from one choice to alternates makes the routes unstable. This increases routing overhead and decreases performance. The oscillation problem was stated in [De Couto et al. 2003]. Using *min_advertised_imd* in route decision prevents a node from switching back and forth among alternative routes and helps in reducing the oscillation of the *imd* value. The associated cost is a possible delay in adopting a better route whose intermediate delay is lower than *imd* but higher than *min_advertised_imd*.

The reception of an advertisement with an older sequence number will trigger a partial advertisement to help the neighbor to obtain the up-to-date routes.

Lazy route query. SAGA does not provide a dedicated route query operation as in the on-demand protocols. When a node wants to send packets to a destination but does not have a valid route, it uses a technique called *lazy route query*. Usually, a route with ∞ delay in an advertisement packet is used to report a broken link. In this case, *seq* is an odd number. A route with ∞ delay and an even number of *seq* is treated as a query instead of an advertisement. It indicates that this node needs a route to the destination. The route's *seq* must be greater than the one in the query. Neighbors who have a valid route will include it in the next advertisement packet as a response to the query. Lazy route query works well with the proactive approach, because (1) each node periodically advertises its routing table, it is likely that one of the neighbors has already had a valid route; and (2) multiple routes may be queried in one advertisement packet.

These operations enable SAGA protocol to handle the dynamic and unpredictable changes in the network topology and traffic load, and to deliver packets through routes with less congestion. They are the basis of a complete implementation for experimental studies. Please refer to [Lu and Bhargava 2003] for the details of SAGA protocol.

5. EXPERIMENTAL EVALUATION

The objective of the experiments is to study the performance of routing protocols under congestion. The use of intermediate delay in SAGA is contrasted against the use of hop count in AODV, DSR, and DSDV protocols, which have received wide attention in the literature [Broch et al. 1998; Johansson et al. 1999; Perkins et al. 2001]. The study is conducted via simulation using the network simulator ns2 [ns2]. All optimizations for AODV and DSR are enabled for the comparisons. The implementation of SAGA is based on the operations presented in section 4.

The wireless interface simulates the 914 MHz Lucent WaveLAN DSSS radio interface [wav 1996]. The IEEE 802.11 DCF with CSMA/CA is used as the MAC protocol. The random waypoint model [Broch et al. 1998] is used to generate movements for mobile nodes. These settings are commonly used in studies reported in the literature. Five independent scenarios are generated for each experiment. The average values are used for analysis.

5.1 Performance metrics

The following metrics are used to evaluate the routing protocols. They are based on a list of quantitative metrics suggested by the RFC 2501 [Corson and Macker 1999].

- *Delivery Ratio*: The ratio of the data delivered to the destinations (i.e., throughput) to the data sent out by the sources. The throughput is also studied in the experiments.
- *Protocol Overhead*: The ratio of the routing load to the data delivered to the destination. The routing load is measured as the number of bytes of protocol messages transmitted hop-wise. The transmission on each hop is counted once.
- *Average End-to-end Delay*: The average time it takes for a packet to reach the destination. It includes all possible delays in the source and intermediate nodes. The delay can be caused by routing discovery, queueing at the interface queue, transmission over the wireless channel, etc. Only successfully delivered packets are counted.

5.2 Simulation and input parameters

UDP and TCP connections are used in the simulation. Each connection is specified as a randomly chosen source-destination pair. Every connection starts at a time uniformly distributed over 0 to 100 seconds, so that the proactive protocols have sufficient time to bootstrap. The size of packets is 512 bytes. Two types of UDP traffic are considered in the study.

- *Constant Bit Rate (CBR) traffic:* It is generated at a deterministic rate [ns2]. This type of traffic is widely used in the study of ad hoc network routing protocols and provides a good basis for evaluating SAGA protocol.
- *Pareto On/Off (POO) traffic:* It is generated according to a pareto on/off distribution [ns2]. Packets are sent at a fixed rate during on periods, and no packets are sent during off periods. Both on and off periods are drawn from a pareto distribution. POO traffic exhibits long range

Table I: Simulation parameters

Simulation time	1000 seconds	Wireless transmission range	250m
Simulation area	1000m × 1000m	Channel capacity	2 Mb/s
Pause time	10 seconds	Number of mobile nodes	50
POO on/off time	500/1000 ms	POO shape	1.5

dependency. It closely matches with the empirically measured network traffic [Willinger et al. 1997].

For UDP, the offered traffic load is taken as the input parameter in order to highlight the impact of congestion on the routing performance. For TCP, the number of connections is used as the input parameter, because TCP will automatically adjust the traffic load according to the network condition. Seven experiments have been conducted by varying the maximum speed of the movement of nodes and the number of connections. The maximum speeds of 4m/s and 20m/s are considered as low and high mobility respectively. The values of parameters used in the simulation are given in table I.

5.3 Measurements and observations

5.3.1 *Experiments 1 and 2.* These two experiments study the routing performance when the number of connections is 10. In the simulation, 20% of the nodes are generating CBR traffic. The results of the low and high mobility experiments are shown in figures 5 and 6 respectively.

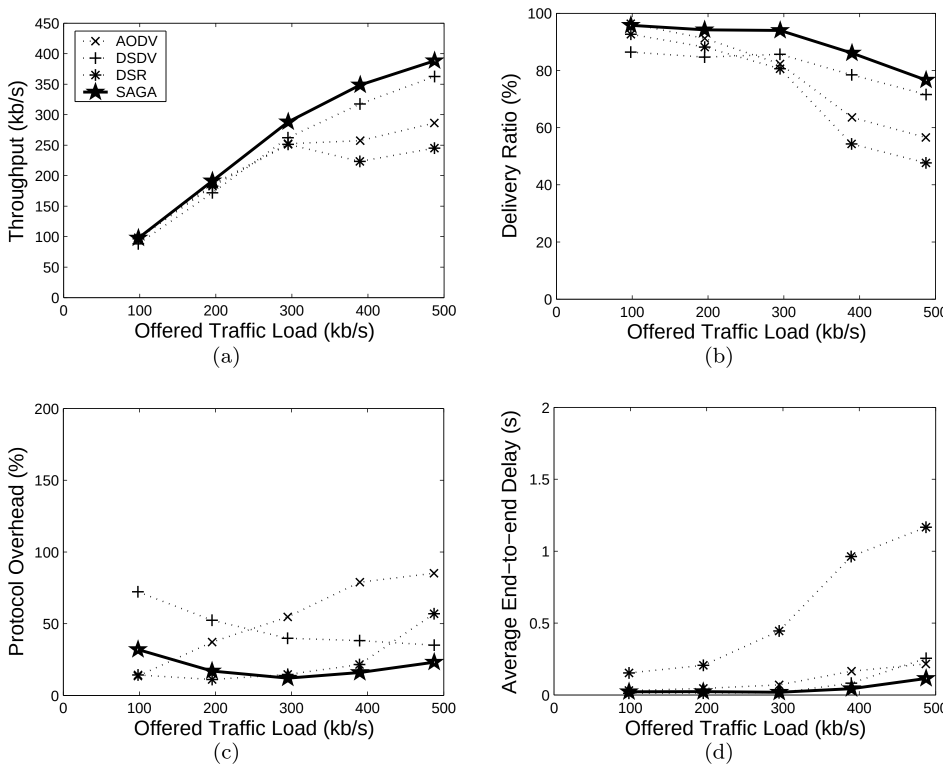


Figure. 5: 10 CBR connections, low mobility

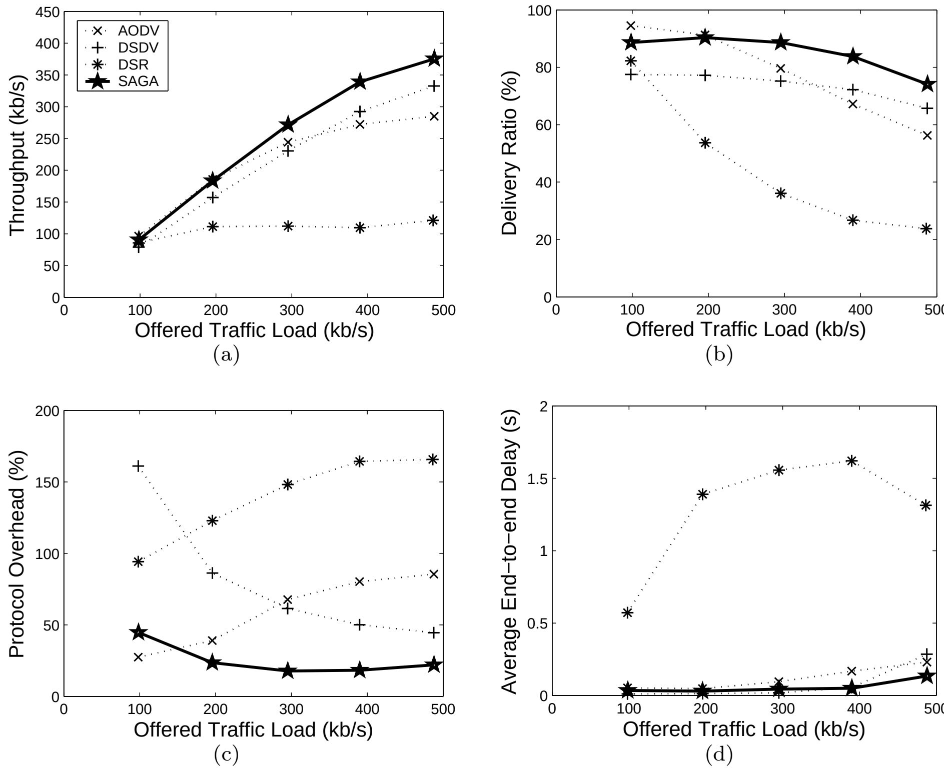


Figure. 6: 10 CBR connections, high mobility

As shown in figures 5b and 6b, the delivery ratios of AODV and DSR drop quickly with the increase of the offered traffic load. The delivery ratios of SAGA and DSDV stay stable when the offered traffic load is less than 300 kb/s. SAGA delivers 10% more packets than DSDV.

As shown in figures 5c and 6c, the overheads of AODV and DSR increase with the offered traffic load. When mobility changes from low to high, the overhead of DSDV is almost doubled, and that of SAGA slightly increases by 5%.

The average end-to-end delay of DSR increases significantly with the offered traffic load or mobility. The delays of other protocols increase gradually with the offered traffic. They are not affected much by mobility (figures 5d and 6d).

Comparing figure 6 with figure 5, we can conclude that mobility greatly affects the performance of DSR. For SAGA and DSDV, the increase of mobility has a greater impact when the offered traffic load is lower. Mobility does not have much effect on the performance of AODV.

Because SAGA performs better when mobility is low, the results of the low mobility scenarios are omitted in the following experiments due to space limit. Please refer to [Lu and Bhargava 2003] for detailed results.

5.3.2 *Experiments 3 and 4.* These two experiments study the performance of routing protocols when 60% of the mobile nodes are generating traffic. The aggregated traffic load is the same as in experiments 1 and 2.

The results of the high mobility experiment are shown in figure 7.

Comparing figure figure 7 with figure 6, we can conclude that routing performance decreases with the number of connections, which has a greater impact on AODV and DSR than on SAGA and DSDV.

The throughput of DSR is saturated at 100 kb/s, almost the same as in the experiment with

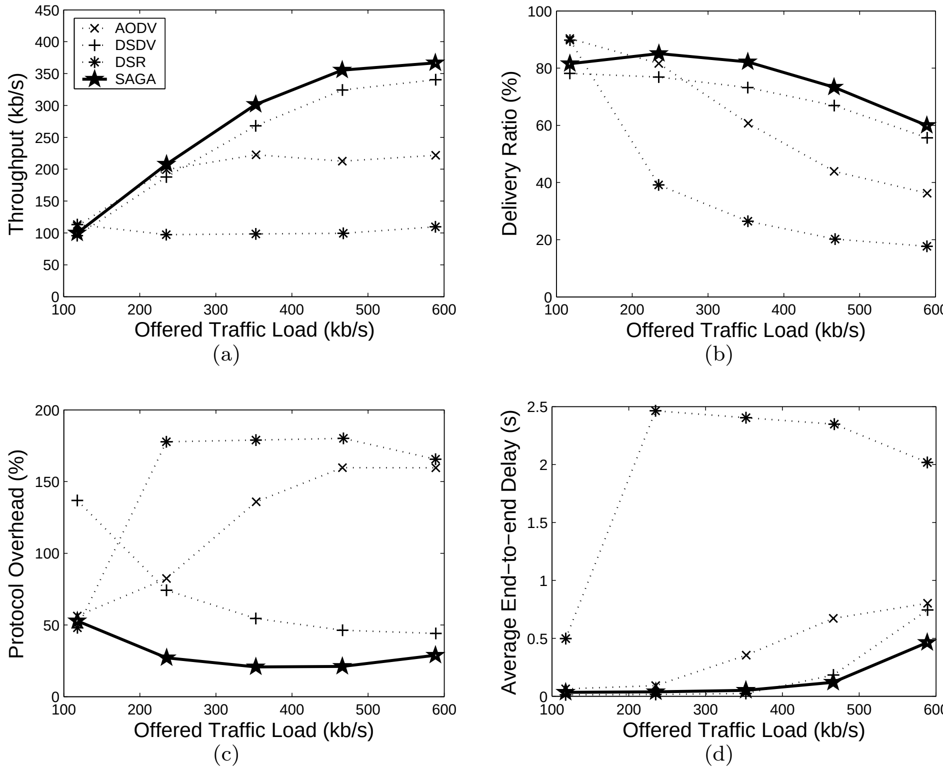


Figure. 7: 30 CBR connections, high mobility

10 connections. The throughput of AODV is saturated at about 200 kb/s, while the saturation is not obvious in the corresponding 10-connection experiment (figures 7a and 6a).

5.3.3 *Experiments 5 and 6.* In addition to CBR traffic, experiments have been conducted to study the performance of routing protocols using POO traffic. The long range dependency of the aggregated POO traffic closely matches with the actual network traffic. The study provides a better understanding on the performance when the routing protocols are implemented for ad hoc networks in practice. The simulation parameters in these experiments are the same as in the 10-connection experiments, except that every source of a connection generates POO traffic instead of CBR traffic. The results of the high mobility experiment is shown in figure 8.

The performance of DSDV and DSR is almost the same as in the 10-connection experiments. SAGA performs even better when the offered traffic load is in the range of 100 to 300 kb/s. In terms of delivery ratio, SAGA outperforms all evaluated protocols in all cases except for DSR in the 67 kb/s traffic and high mobility (figure 8b).

AODV delivers less than 40% of packets in the low mobility experiment and about 20% in the high mobility experiment. Figure 8c shows that the overhead of AODV is less than 10%, which is much lower compared with the results of the CBR traffic experiments. It indicates that AODV does not exchange much routing information when traffic bursts. Many packets are dropped due to congestion.

5.3.4 *Experiments for TCP traffic.* Two experiments have been conducted to evaluate the performance of SAGA with TCP traffic in low and high mobility scenarios. The results are shown in Figure 9. All evaluated routing protocols except for DSR have almost the same end-to-end delay regardless the number of connections as shown in Figure 9c and 9d. The proactive

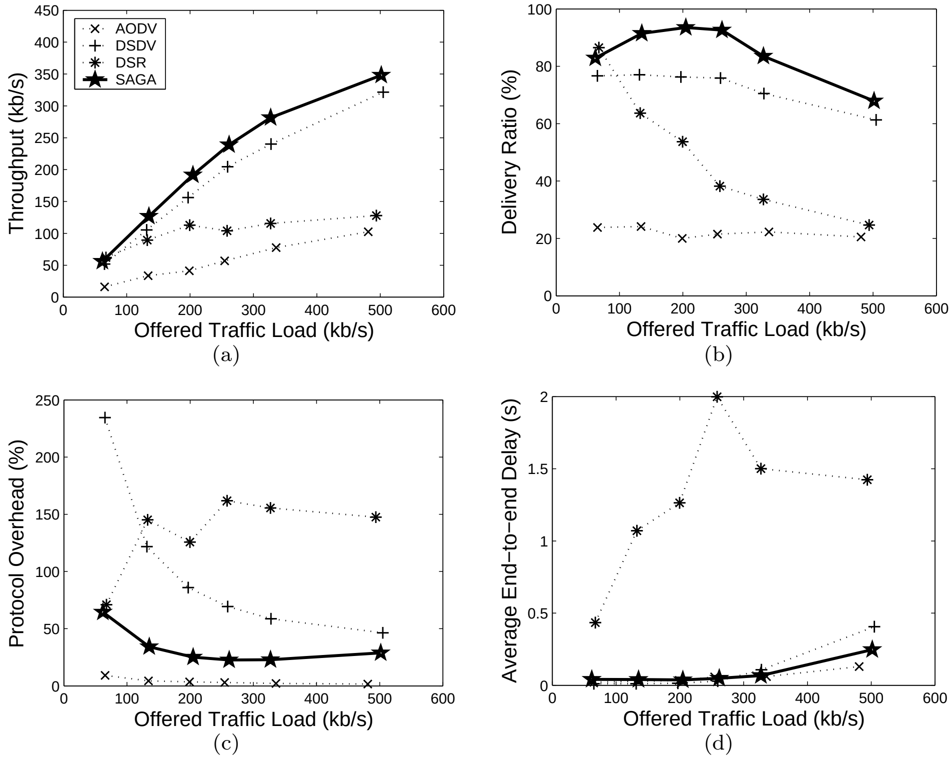


Figure. 8: POO traffic with long range dependency, high mobility

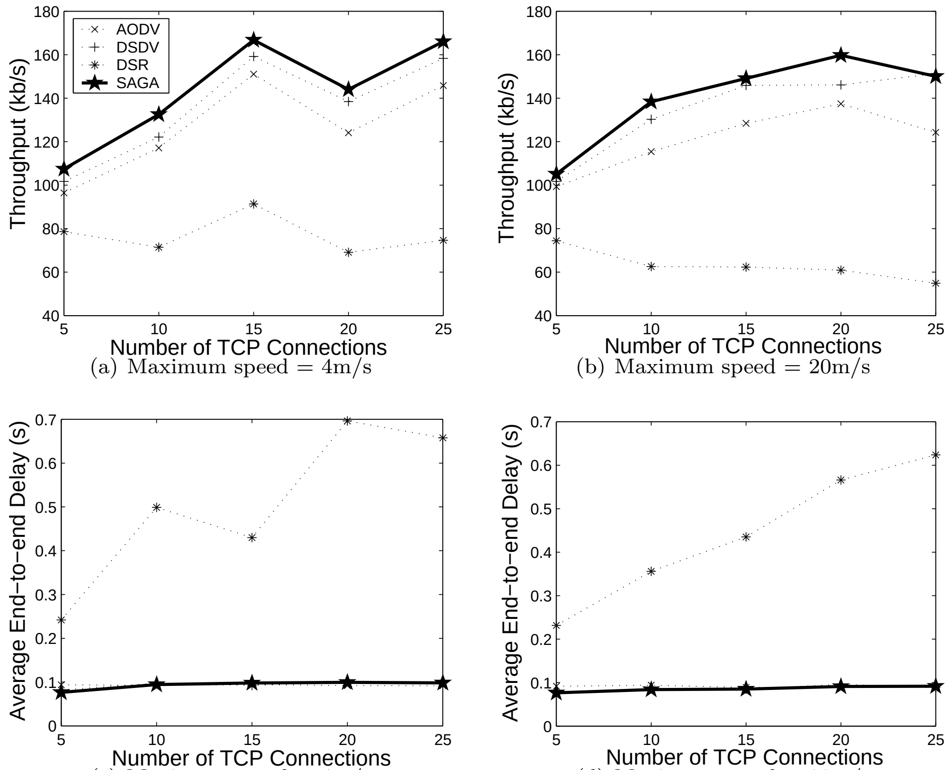
protocols achieve higher throughput than the on-demand ones. This is consistent with the results obtained from the study of packet loss. SAGA still performs better than DSDV, but not much. This is because TCP also tries to control congestion, thus diminishes the advantage of SAGA in terms of congestion avoidance.

5.4 Analysis and discussion

We classify the traffic load offered by CBR connections into *low*, *moderate*, and *high* based on whether it is less than 200 kb/s, between 200 and 400 kb/s, or greater than 400 kb/s. For the traffic load offered by POO connections, the two classifying values are 132 kb/s and 330 kb/s.

5.4.1 *SAGA versus on-demand protocols*

- *Throughput:* Because of its capability of balancing traffic load and avoiding congestion, SAGA is able to sustain heavier traffic load and offers higher peak throughput than AODV and DSR. When mobility is low, AODV and DSR saturate at around 250 kb/s (figure 5a). In high mobility scenarios, DSR saturates at 120 kb/s. The peak throughput of AODV is 220 to 280 kb/s. It decreases as the number of connections increases. POO traffic does not have much impact on the peak throughput of SAGA and DSR. It causes the peak throughput of AODV to drop to 100 kb/s in high mobility scenarios (figure 8a). SAGA can consistently offer a peak throughput of 370 to 400 kb/s in all cases, which is 1.5 to 3.5 times of the peak throughput achieved by the on-demand protocols.
- *Delivery ratio:* SAGA does not achieve high delivery ratio in high mobility and low traffic load (figures 6b, 7b, and 8b). More than 95% of the dropped packets are caused by broken routes, because the routes obtained from advertisements are stale by the time they are used. In the SAGA implementation, a link is considered broken if two consecutive packets to the



(a) Maximum speed = 4m/s (b) Maximum speed = 20m/s

(c) Maximum speed = 4m/s (d) Maximum speed = 20m/s

Figure. 9: TCP traffic

same neighbor are dropped. This increases the accuracy of broken link detection at the cost of more dropped packets. When mobility is high and traffic load is low, one packet might be enough to infer a broken link.

With the growth of the traffic load from low to moderate, the delivery ratio of SAGA increases because of the accuracy of broken link detection.

SAGA performs as well as the on-demand protocols in low traffic load and low mobility. It outperforms them when the offered traffic load is moderate or high.

➢ *Protocol overhead:* The protocol overhead of SAGA is in the range of 150 to 500 bytes per kilo-byte delivered data. Because SAGA uses one-hop broadcast, its overhead is not affected much by traffic load, mobility, and the number of connections.

The overhead of AODV increases rapidly with the offered traffic load. The POO traffic experiments are exceptions where AODV fails to deliver most of the packets. AODV uses network-wide broadcast to re-discover a route when a packet is dropped and the route is considered as broken. With the increase of the offered traffic load, a larger number of packets are dropped due to congestion. This causes AODV to initiate additional route re-discoveries. AODV introduces less overhead than SAGA only in low traffic load.

The overhead of DSR is affected by mobility. It is almost tripled when the maximum moving speed of nodes changes from 4 m/s to 20 m/s. DSR uses route cache and snooping that are not effective in highly dynamic networks. Only when mobility and the offered traffic load are low, DSR can outperform SAGA in terms of protocol overhead. Otherwise, it introduces up to 8 times overhead as SAGA does (figure 7c).

➢ *End-to-end delay:* SAGA offers lower average end-to-end delay than AODV and DSR, because it uses the intermediate delay instead of the hop count as the routing metric. The advantage of using the new metric is significant when the offered traffic load is high. In those cases, the delay of SAGA is 50% less than that of AODV and 80% less than that of DSR (figures 5d, 6d, and 7d).

5.4.2 *SAGA versus DSDV.* SAGA outperforms DSDV in the measured metrics at the conducted experiments. It delivers 10% more packets than DSDV with less than half of the protocol overhead. The average end-to-end delay of SAGA is almost the same as that of DSDV when the offered traffic load is less than 300 kb/s. It is around 50% to 70% of the delay of DSDV with 500 kb/s traffic, depending on mobility and the number of connections. DSDV fails to provide high delivery ratio in low traffic load. It delivers about 85% of the packets while the other protocols can deliver 95% (figure 5b). In addition, it introduces 1 to 2 times more overhead than other protocols in high mobility and low traffic load (figures 6c and 7c).

Additional experiments have been done with various maximum speeds ranging from 4 m/s to 24 m/s and numbers of connections ranging from 10 to 50. They lead to the similar conclusions.

6. RELATED WORK

Associativity-based routing (ABR) [Toh 1997] is one of the first protocols that consider load as a part of the routing metric. The load is based on the number of routes in which a node is involved. Load balancing routing protocols [Hassanein and Zhou 2001; Lee and Gerla 2001] use a similar idea as ABR but different methods to compute load. Various traffic loads on different routes have not been considered.

Multipath routing protocols [Tsirigos and Haas 2001; Marina and Das 2001] can be adjusted for load balancing by allowing sources to deliver packets through different paths. The source-based load balancing may still cause congestion. Even though every single source evenly distributes load over multiple paths, nodes that are involved in several paths can be overloaded.

A. Boukerche and S.K. Das present a new approach to control congestion in wireless ad hoc networks in [Boukerche and Das 2003]. They propose a randomized version of the DSDV routing protocol called R-DSDV. R-DSDV propagates the routing messages according to a routing probability distribution rather than on a periodic basis. It controls congestion in the store-and-forward procedure. If the current queue size is over the congestion level, a newly arrived packet is dropped or queued according to a probability. The data packets have higher priorities than the advertisement packets.

The experimental evidence from two empirical wireless test-beds presented by D.S.J. Couto, D. Aguayo, B.A. Chambers, and R. Morris in [De Couto et al. 2003] shows that the minimum-hop-count routing often chooses routes that have significantly less capacity than the best paths in a multi-hop wireless network. A new metric, the expected transmission count (ETX), is designed for routing protocols to find high-throughput paths [De Couto et al. 2003]. The expected number of transmission is determined by the forward and reverse delivery ratios of a link, which are measured using dedicated link probe packets. The ETX metric incorporates the effect of link loss ratio and the interference among the successive links of a path. It does not account for mobility and does not route around congested links. It is complementary to the IMD metric proposed in this paper.

C. Cordeiro, S.R. Das, and D.P. Agrawal propose contention-based path selection (COPAS) for TCP over multi-hop wireless networks [Cordeiro et al. 2002]. COPAS monitors the MAC layer contention and accordingly changes the forward and reverse paths for a TCP connection. It enhances the performance of TCP by minimizing the likelihood of the capture problem [Xu and Saadawi 2002]. The number of backoffs is used to measure contention. Because the number of backoffs is closely related to the number of packets that are sent during the measured time, research is needed for a more precise indication of channel contention. Intermediate nodes continuously piggyback their contention information on packets that pass through them. If the number of backoffs exceeds a predefined threshold, the route is reconstructed. In a network with heavy traffic or lossy links that result in a large number of backoffs, unnecessary route reconstructions can be caused.

MR^{2}RP is a delay-oriented multi-rate/multi-range routing protocol for IEEE 802.11 ad hoc networks [Sheu et al. 2003]. It is designed to maximize the channel utilization and minimize the network transfer delay. The medium access control (MAC) protocol is analyzed to predict the transfer delay of a routing path. The authors assume: (a) the packet arrival process is a Poisson process, (b) all nodes have the same packet arrival rate, (c) each node knows the buffer information of every other node, (d) every node knows the connectivity matrix of the network so that the Dijkstra algorithm can be employed to find the shortest path. SAGA is based on weaker assumptions as discussed in section 3.1. It will be preferable if the delay is estimated locally without exchanging information among neighbors.

Quality-of-Service (QoS) routing protocols for ad hoc networks select routes with sufficient resources to satisfy certain requirements such as delay or bandwidth [Chen and Nahrstedt 1999; Lin 2001; Zhu and Corson 2002]. They work on a per-connection basis. The QoS routing requires the underlying MAC protocol to support and guarantee resource reservation as well as provide information and constraints about delay and bandwidth, etc. If QoS support is not available, SAGA's delay estimation methods can be extended for contention-based media access protocols to provide this information to the upper layer protocols and applications.

7. CONCLUSION

Congestion is a serious problem that degrades the performance of ad hoc networks. Compared to the traditional solutions at the transport layer, the SAGA routing protocol attacks the problem at the network layer. It integrates the channel spatial reuse with the multi-hop nature of ad hoc routing to reduce congestion. SAGA is a distance vector routing protocol that uses intermediate delay (IMD) instead of hop count as the distance. The use of IMD enables routing protocols to select routes that bypass mobile nodes in contention. The lazy route query operation in SAGA protocol uses a special route advertisement for route discovery. Multiple queries can be included in one advertisement packet to accelerate the establishment of needed routes. SAGA provides an approach to reduce the oscillation of the value of IMD and makes the routes stable.

The use of IMD in routing decisions can enhance the performance of existing routing protocols. It is especially of benefit to networks where topology changes are less frequent than traffic changes. The lazy route query can be applied to other proactive routing protocols that do not have a dedicated route discovery operation. SAGA can reduce congestion at intermediate nodes and be used as a supplement to the end-to-end congestion control/avoidance mechanisms. The delay estimation methods can be extended for contention-based media access protocols to provide quality of service (QoS) data to upper layer protocols and applications. The intermediate delay can be used to improve the accuracy of round-trip-time (RTT) estimation for TCP connections.

Methods are proposed to estimate the delay at a node using only local information. When a node has recent traffic, statistical methods are used to evaluate the mean of the delay. Otherwise, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of the delay. We analyze the packet transmission procedure of the distributed coordination function in the IEEE 802.11 standard as a case of the practical study. These methods are applicable to other contention-based media access protocols.

A series of experiments have been conducted to study the performance of routing protocols under congestion. The maximum moving speed of nodes and the number of connections are varied. SAGA performs better than DSDV in all our measurements. A summary of comparison of SAGA with AODV and DSR for throughput, overhead, and end-to-end delay is as follows.

- ➢ SAGA is able to deliver around 90% of the data packets with an offered traffic load up to 300 kb/s. It can offer a peak throughput of 370 to 400 kb/s in all experiments. This is 1.5 to 3.5 times as compared to the throughput of AODV and DSR.
- ➢ Overhead is measured as the ratio of the routing load to the data successfully delivered to the destination. The overhead of SAGA remains in a range of 15% to 50%. In similar cases,

the overhead of AODV and DSR varies widely and increases fast as the offered traffic load goes high. The overhead of SAGA is as low as 10% of that of AODV and 12% of that of DSR in high traffic load.

➢ For low traffic load, the average end-to-end delay of SAGA is the same as that of AODV and DSR. When traffic reaches 500 kb/s, the delay of SAGA is 50% less than that of AODV and 80% less than that of DSR.

Evaluating SAGA protocol in an emulation instead of simulation environment is preferable for its success in real world. In the future, we plan to use the mobile ad hoc emulator MobiEmu [Zhang and Li 2002] to conduct experimental studies. The impact of the accuracy of delay estimation on the performance of SAGA protocol will be investigated. The results of the research on the lifetime of routes in mobile ad hoc networks [Tseng et al. 2003] will be adopted to improve the accuracy of delay estimation. Research will be conducted to integrate SAGA's congestion reduction mechanism with the TCP congestion control algorithms. The idea of randomization [Boukerche and Das 2003] may be adjusted for SAGA protocol to decrease routing overhead and provide better congestion reduction. Multi-rate and multi-range link adaption is being studied to improve the performance of ad hoc networks [Wu et al. 2004]. The impact of the link distance on the end-to-end delay as well as the integration of the adaptive date transmission rate and congestion control will be investigated.

REFERENCES

The network simulator - ns-2. *http://www.isi.edu/nsnam/ns/*.

1996. WaveLAN/PCMCIA card user's guide, Lucent Technologies.

Bertossi, A. A. and Bonuccelli, M. A. 1995. Code assignment for hidden terminal interference avoidance in multihop packet radio networks. *IEEE/ACM Transactions on Networking (TON) 3,* 4, 441–449.

Biaz, S. and Vaidya, N. H. 1998. Distinguishing congestion losses from wireless transmission losses: A negative result. In *Proceedings of IEEE ICCCN*. New Orleans, 722–731.

Boukerche, A. and Das, S. K. 2003. Congestion control performance of R-DSDV protocol in multihop wireless ad hoc networks. *Wireless Networks 9,* 3, 261–270.

Brakmo, L. S., O'Malley, S. W., and Peterson, L. L. 1994. TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of SIGCOMM*. 24–35.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y., and Jetcheva, J. 1998. A performance comparison of multihop wireless ad hoc network routing protocols. In *Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 85–97.

Cen, S., Cosman, P. C., and Voelker, G. M. 2003. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Transactions on Networking (TON) 11,* 5, 703–717.

Chandran, K., Raghunathan, S., Venkatesan, S., and Prakash, R. 2001. A feedback-based scheme for improving TCP performance in ad hoc wireless networks. *IEEE Personal Communications 8,* 1, 34–39.

Chen, S. and Nahrstedt, K. 1999. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications 17,* 8, 1488–1505.

Cordeiro, C., Das, S. R., and Agrawal, D. P. 2002. COPAS: Dynamic contention-balancing to enhance the performance of TCP over multi-hop wireless networks. In *Proceedings of the 10th International Conference on Computer Communication and Networks (IC3N)*. Miami, 382–387.

Corson, S. and Macker, J. 1999. Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations. *RFC 2501, http://www.ietf.org/rfc/rfc2501.txt*.

De Couto, D. S. J., Aguayo, D., Bicket, J., and Morris, R. 2003. A high-throughput path metric for multihop wireless routing. In *Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom)*. 134–146.

De Couto, D. S. J., Aguayo, D., Chambers, B. A., and Morris, R. 2003. Performance of multihop wireless networks: Shortest path is not enough. *ACM SIGCOMM Computer Communication Review 33,* 1, 83–88.

Gelenbe, E. and Pujolle, G. 1998. *Introduction to Queueing Networks, Second Edition*. John Wiley & Sons.

Hassanein, H. and Zhou, A. 2001. Routing with load balancing in wireless ad hoc networks. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 89–96.

Jacobson, V. 1988. Congestion avoidance and control. In *Proceedings of Symposium on Communications Architectures and Protocols*. 314–329.

JAFFE, J. AND MOSS, F. 1982. A responsive distributed routing algorithm for computer networks. *IEEE Transactions on Communications 30,* 7, 1758–1762.

JOHANSSON, P., LARSSON, T., HEDMAN, N., MIELCZAREK, B., AND DEGERMARK, M. 1999. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. Seattle, 195–206.

JOHNSON, D. B. AND MALTZ, D. A. 1996. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, Chapter 5, 153–181.

KUNNIYUR, S. AND SRIKANT, R. 2003. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. *IEEE/ACM Transactions on Networking (TON) 11,* 5, 689–702.

LEE, S.-J. AND GERLA, M. 2001. Dynamic load-aware routing in ad hoc networks. In *Proceedings of ICC*. Vol. 10. 3206–3210.

LIN, C. 2001. On-demand QoS routing in multihop mobile networks. In *Proceedings of INFOCOM*. Vol. 3. 1735–1744.

LIU, J. AND SINGH, S. 2001. ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications 19,* 7, 1300–1315.

LU, Y. AND BHARGAVA, B. 2003. Self-adjusting congestion avoidance routing protocol for ad hoc networks. Technical Report CSD 03-018, Department of Computer Sciences, Purdue University.

LU, Y., ZHONG, Y., AND BHARGAVA, B. 2003. Packet loss in mobile ad hoc networks. Technical Report CSD 03-009, Department of Computer Sciences, Purdue University.

LUO, H., MEDVEDEV, P., AND LU, S. 2001. A self-coordinating approach to distributed fair queueing in ad hoc wireless networks. In *Proceedings of INFOCOM*. Vol. 3. 1370–1379.

MARINA, M. K. AND DAS, S. R. 2001. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of the International Conference on Network Procotols (ICNP)*. Riverside, 14–23.

MENDENHALL, W. AND BEAVER, R. J. 1991. *Introduction to Probability and Statistics*, 8th ed. PWS-Kent Pub. Co., Boston.

MO, J., LA, R., ANANTHARAM, V., AND WALRAND, J. 1999. Analysis and comparison of TCP Reno and Vegas. In *Proceedings of INFOCOM*. 1556–1563.

PADHYE, J., FIROIU, V., TOWSLEY, D. F., AND KUROSE, J. F. 2000. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking (TON) 8,* 2, 133–145.

PARSA, C. AND GARCIA-LUNA-ACEVES, J. J. 2000. Improving TCP performance over wireless networks at the link layer. *Mobile Networks and Applications 5,* 1, 57–71.

PERKINS, C. E., BELDING-ROYER, E. M., AND DAS, S. R. 2003. Ad hoc on-demand distance vector (AODV) routing. *RFC 3561, http://www.ietf.org/rfc/rfc3561.txt*.

PERKINS, C. E. AND BHAGWAT, P. 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*. 234–244.

PERKINS, C. E., ROYER, E. M., DAS, S. R., AND MARINE, M. K. 2001. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications 8,* 1, 16–28.

SHEU, S.-T., TSAI, Y., AND CHEN, J. 2003. MR^{2}RP: The multi-rate and multi-range routing protocol for IEEE 802.11 ad hoc wireless networks. *Wireless Networks 9,* 2, 165–177.

TOH, C.-K. 1997. Associativity-based routing for ad-hoc mobile networks. *Wireless Personal Communications Journal 4,* 2, 103–139.

TSENG, Y.-C., LI, Y.-F., AND CHANG, Y.-C. 2003. On route lifetime in multihop mobile ad hoc networks. *IEEE Transactions on Mobile Computing 2,* 4, 366–376.

TSIRIGOS, A. AND HAAS, Z. 2001. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine 39,* 11, 132–138.

WANG, F. AND ZHANG, Y. 2002. Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*. 217–225.

WILLINGER, W., TAQQU, M. S., SHERMAN, R., AND WILSON, D. V. 1997. Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking (TON) 5,* 1, 71–86.

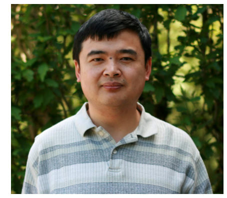
WU, X., DING, G., AND BHARGAVA, B. 2004. Impact of link distance on end-to-end throughput in multi-rate ad hoc networks. Tech. Rep. TR-04-026), Department of Computer Sciences, Purdue University.

XU, S. AND SAADAWI, T. 2002. Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks. *Journal of Computer Networks 38,* 4, 531–548.

ZHANG, C. AND TSAOUSSIDIS, V. 2001. TCP-Real: Improving real-time capabilities of TCP over heterogeneous networks. In *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. 189–198.

ZHANG, Y. AND LI, W. 2002. An integrated environment for testing mobile ad-hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*. 104–111.

ZHU, C. AND CORSON, M. 2002. QoS routing for mobile ad hoc networks. In *Proceedings of INFOCOM*. Vol. 2. 958–967.



Y. Lu is a senior software development engineer at Office Communications Group at Microsoft. He received his PhD degree in Computer Science from Purdue University. His areas of interests include wireless networking, networking protocols, and scalable video coding. He has several peer-reviewed journal and conference papers and a couple of patents to his credit.

Bharat Bhargava is a professor of the Department of Computer Science with a courtesy appointment in the School of Electrical & Computer Engineering at Purdue University. Professor Bhargava is conducting research in security and privacy issues in distributed systems and Cloud environments. Professor Bhargava is a Fellow of the Institute of Electrical and Electronics Engineers and of the Institute of Electronics and Telecommunication Engineers. In 1999, he received the IEEE Technical Achievement Award for a major impact of his decade long contributions to foundations of adaptability in communication and distributed systems. He has been awarded the charter Gold Core Member distinction by the IEEE Computer Society for his distinguished service. He serves on thirteen editorial boards of international journals. He served on the IEEE Computer Society on Technical Achievement award and Fellow committees. Professor Bhargava is the founder of the IEEE Symposium on Reliable and Distributed Systems, IEEE conference on Digital Library, and the ACM Conference on Information and Knowledge Management.