# Design and Analysis of Rate Aware Ad Hoc 802.11 Networks

Sandhya G., K. Gopinath
Computer Science & Automation
Indian Institute of Science, Bangalore

***Abstract*— 802.11 WLANs are characterized by high bit error rate and frequent changes in network topology. The key feature that distinguishes WLANs from wired networks is the multi-rate transmission capability, which helps to accommodate a wide range of channel conditions. This has a significant impact on higher layers such as routing and transport levels.**

**While many WLAN products provide rate control at the hardware level to adapt to the channel conditions, some chipsets like Atheros do not have support for automatic rate control. We first present a design and implementation of an FER-based automatic rate control state machine, which utilizes the statistics available at the device driver to find the optimal rate. The results show that the proposed rate switching mechanism adapts quite fast to the channel conditions.**

**The hop count metric used by current routing protocols has proven itself for single rate networks. But it fails to take into account other important factors in a multi-rate network environment. We propose transmission time as a better path quality metric to guide routing decisions. It incorporates the effects of contention for the channel, the air time to send the data and the asymmetry of links.**

**In this paper, we present a new design for a multi-rate mechanism as well as a new routing metric that is responsive to the rate. We address the issues involved in using transmission time as a metric and presents a comparison of the performance of different metrics for dynamic routing.**

***Index Terms*—802.11, rate-aware networks**

## I. Introduction

SINCE 1997, when IEEE 802.11 standard came into action, there has been a tremendous interest in the use of wireless LAN hardware. The main advantages of wireless networks over fixed networks are mobility, ease and speed of deployment and flexibility. But the speed of wireless networks is constrained by the available bandwidth.

The 802.11 physical layer is capable of operating at different rates. The rationale behind multi-rate support is the need to provide network coverage. Long distance transmissions are not possible at high data rates. Using lower data rates always is not a good idea either, since it leads to decreased throughput. It has been observed that the presence of even a single slow host in the system can degrade the overall throughput of the system to a level below the lowest rate. Keeping the data transmission rate fixed is thus not desirable. This motivates the need for a highly adaptive rate switching mechanism. Multi-rate support has a significant impact on higher layers such as routing and transport levels. In this paper, we present a new design for a multi-rate mechanism as well as a new routing metric that is responsive to the rate.

Many WLAN products provide rate control support as part of their chipset. But certain chipsets (for eg., Atheros) do not allow automatic selection of transmission rate. An alternative is to do it at the device driver level using the statistics available. We discuss the design and implementation of an FER based approach, which could be used at the device driver for finding the optimal rate. We report on the performance of the proposed state machine using simulations and real world experiments. It is seen that the performance is quite close to that of manual configuration.

802.11 networks can operate in two different modes - infra-structure and ad-hoc mode. In an infra-structure mode, all the communication between the nodes go through a central entity called the access point. Hence routing has few implications in an infra-structure based network. Ad-Hoc networks are multi-hop wireless networks consisting of a collection of peer nodes communicating with each other without support from a fixed infrastructure. Packets might take a multi-hop path to reach the destination in such a network due to the lack of access points. Highly dynamic routing protocols are required to adapt the wireless system to the frequent changes in channel conditions and topology. These protocols exchange messages to decide the shortest path between the source and destination based on some metric.

The hop count metric used by current routing protocols proves to be good for single rate networks. But it fails to take into account other important factors in a multi-rate network environment. For example, the rate information

provided by the lower layers can be used to drive the routing decisions. Consider the scenario in Figure 1. The hop count based metric routing protocols use the path A-B-C that consists of two slow links. But choosing the path A-D-B-E-C can boost the system throughput because the transmission time required is less at higher data rates.

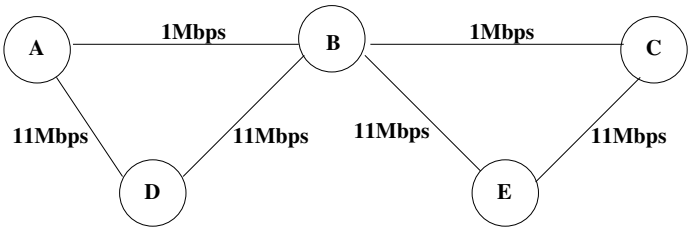


Fig. 1. Need for multi-rate aware routing

The two main performance measures that are substantially affected by the routing algorithm are throughput (Quantity of service) and average packet delay (Quality of service)[5]. Choosing high rate paths guarantee quantity of service. The issue is to provide a bound on the delay. Since high data rate links are shorter, it might be required to traverse more number of hops to reach the destination. The extra backoffs required and the congestion at intermediate nodes can add to the end-to-end delay experienced by the packet. The metric used for routing should strike a balance between throughput and delay. Another critical issue to be taken into account for routing in a wireless network is the asymmetry of links. The link characteristics in the forward direction may not be the same as that in the reverse direction. 802.11 MAC expects an ACK for each packet and hence the reverse path quality also has an impact on the successful transmission of a packet. It is desirable to come up with a metric that takes into consideration the transmission characteristics of link to aid the computation of optimal routes.

We propose transmission time as a better path quality metric. Transmission time is the amount of time between when a packet is sent and the time when the corresponding ACK is received. It incorporates the effects of the following factors.

- link rate
- contention for the channel
- asymmetry of links

We present a performance comparison of the various metrics that could be used for dynamic routing using real world experiments run on two testbeds consisting of a 3-node and 5-node setup respectively.

It is evident that the link rate characteristics of the system can be used to drive the different layers of the TCP/IP stack. Our work focuses on making the L2 and L3 layers of the protocol stack multi-rate aware. As a prelude, we first develop a simple model of the 802.11 MAC and its performance in the presence of slow transmitters. This is to establish a framework for analysis that will be later used for the multi-rate environment. Next, a similar analysis is done for the design of the automatic rate switching mechanism and its performance is studied using simulations and experiments. The rate switching mechanism keeps track of the link status to its neighbors and this information is exported to the IP layer to guide the flows through optimal routes. We use two test-beds to study the effectiveness of using link quality as the metric for routing in wireless networks.

The rest of the report is organized as follows. Section 2 develops a simple model of performance analysis of the 802.11 MAC. Section 3 gives the design for the proposed rate switching mechanism and discusses the result. The design and implementation details of the transmission time metric and the performance comparison of the different metrics are presented in Section 4. Section 5 provides a summary of the related work in this area. Section 6 concludes the paper.

## II. Performance Analysis of 802.11 MAC

To esatablish a framework for analysing the multi-rate system, we first discuss a simple model of 802.11 DCF to study the performance of the system for varying number of nodes.

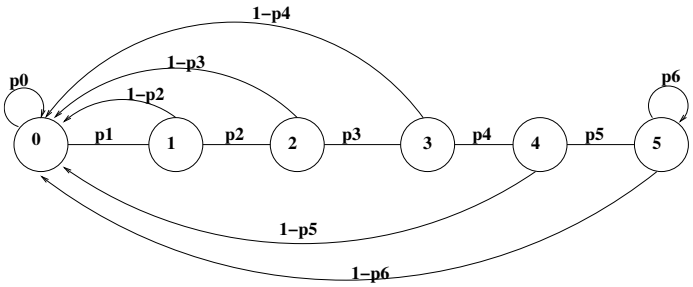


Fig. 2. Model for Performance Analysis

### A. Performance Analysis of DCF

RTS/CTS handshake has been introduced to avoid the problems due to hidden and exposed nodes[3]. In case of a collision, the cost incurred is low for RTS packets due to its relatively smaller size. But RTS/CTS is an additional overhead for smaller packets. Hence an RTS Threshold is defined which determines when to use the RTS/CTS handshake for transmission. The first scheme of sending

packets without RTS/CTS is called Basic Access Scheme (BAS) and the latter scheme is called RTS/CTS Scheme (RCS). A model of the 802.11 MAC has been developed and a study has been done to analyze the performance under BAS and RCS.

*1) Assumptions:* A simple model that uses Distributed Coordination Function for providing access to the channel is assumed. The system operates in saturated conditions, that is, the station queue is never empty. It is assumed that hidden and exposed terminals are not present and the channel conditions are ideal. The number of nodes present is varied between 2 to 100 and the performance of the system is evaluated. All nodes transmit at the same rate.

Initially, basic access mechanism is assumed where a packet is sent after a period equal to the DIFS period followed by a random backoff chosen from the contention window. In case of a collision, the station doubles the contention window and chooses another backoff. This is continued till the packet is sent successfully or the packet is dropped after 7 retries (as specified in the standard)[1]. The same is repeated for RTS/CTS scheme.

*2) State Transition Diagram:* Assume an average packet size of $S$ (including PHY, MAC) normalized in terms of 11Mbit time units. For $0 < i < 6$, $P_i$ refers to the probability that a node suffers a collision and has to increase its congestion window from $CW_i$ to $CW_{i+1}$ (Figure 2). $P_0$ is the probability that a node succeeds in transmitting without moving from $CW_0$ to $CW_1$. $P_6$ is the probability that a node suffers a collision and uses $CW_5$ again. The relationship amongst the various $P_i$ is as follows:

$P_0 + P_1 = 1$

$P_0 + P_1(1 - P_2) + P_2P_1(1 - P_3) + P_3P_2P_1(1 - P_4) + P_4P_3P_2P_1(1 - P_5) + P_5P_4P_3P_2P_1(1 - P_6)(1 + P_6 + P_6^2) + P_5P_4P_3P_2P_1P_6^3 = 1$

Assume that SIFS, DIFS, slots, AckTimeOut, etc. are normalized in terms of 11Mbit time units. The saturation throughput is given by $P_{succ} * packetsize/effS$ where

$P_{succ}$ = Probability of successful transmission of packet

$P_{coll}$ = Probability of collision

*effS* = Effective system time for sending packet = average latency seen by system + $P_{succ}$ * time taken for successful transmission of packet + $P_{coll}$ * collision overhead

Average system delay = the fraction of the idle channel time seen by the system including DIFS periods and slots

Hence, $effS_{BAS}$ = average system delay + $P_{succ}$($S$ + SIFS + $ACK$) + $P_{coll}$($S$ + AckTimeOut)

$effS_{RTS}$ = average system delay + $P_{succ}$($RTS$ + $CTS$ + $S$ + $ACK$ + 3*SIFS) + $P_{coll}$($RTS$ + AckTimeOut)

TABLE I

PARAMETERS USED FOR THE PERFORMANCE ANALYSIS

| Parameter | Value |
|---|---|
| PHY | DSSS |
| Slot time | 20 $\mu s$ |
| Cwmin | 32 |
| Cwmax | 1024 |
| Packet Payload | 512B, 1024B, 1500B |
| MAC header overhead | 34B |
| PHY header overhead | 192b |
| LLC header overhead | 8B |
| IP + TCP hdr overhead | 40B |
| RTS | 160b + PHY hdr |
| CTS/ACK | 112b + PHY hdr |
| AckTimeOut | 334 $\mu s$ |
| CTSTimeOut | 334 $\mu s$ |
| Channel Bit Rate | 11Mbit/s |
| SIFS | 10 $\mu s$ |
| DIFS | 50 $\mu s$ |
| No of retries | 7 |

*3) Analysis:* Table I summarizes the parameters used for the study. Figures 3 and 4 plot the saturation throughput against the number of nodes. As can be seen from the plot, the RTS/CTS scheme (Figure 4) shows less variation compared to the basic access mechanism (Figure 3). The throughput is slightly lesser than the basic access scheme under lightly loaded situations because of the extra handshaking involved in the RTS/CTS scheme. But as offered load increases, the system throughput of RTS/CTS scheme outweighs that of BAS. A careful study of the graph reveals that the RTS/CTS throughput remains unaffected by the number of nodes, whereas throughput under BAS suffers a gradual degradation. This is due to the fact that the RTS packet involved in the contention process is smaller than the data packet. The results correlate with a similar study done by Bianchi[6] and Miller et. al.[13].

### B. Performance Anomaly in 802.11 DCF

The bit rate of hosts has a major impact on the performance of the system. In 802.11 DCF, all hosts have equal long term channel access probability. So slow hosts can capture the channel for long periods. If there is at least one host with low bit rate, the overall performance of the system drops below the level of the lower rate.

In order to find the impact of slow nodes on the performance of the system, the above studies were repeated assuming one slow node with a bit rate of 1 Mbps. All

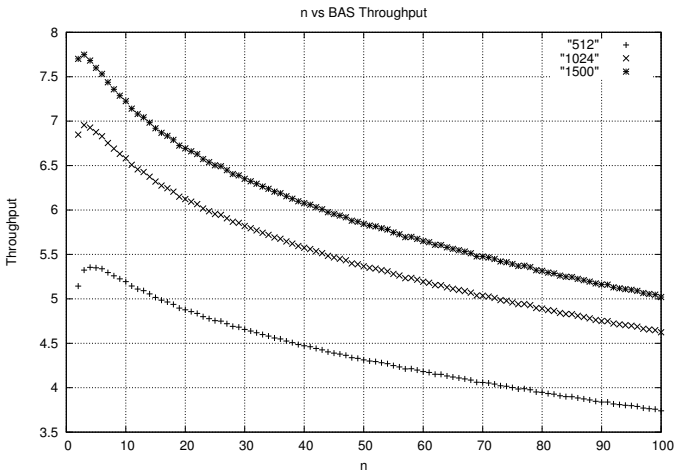


Fig. 3. Number of nodes vs Throughput (BAS)

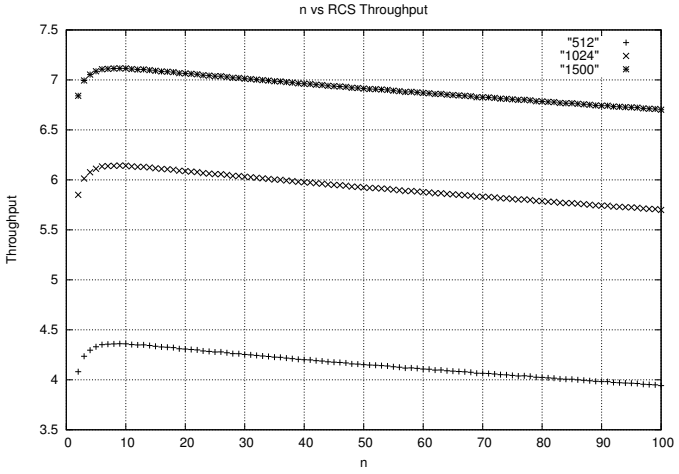


Fig. 4. Number of nodes vs Throughput (RCS)

other hosts transmit at the rate of 11Mbps. The overall throughput can be calculated as described earlier. The effective system time is given by

$effS_{BAS}$ = average system delay + $P_{succslow}$($S_{slow}$ + SIFS + $ACK_{slow}$) + $P_{succfast}$($S_{fast}$ + SIFS + $ACK$) + $P_{collslow}$($S_{slow}$ + AckTimeOut) + $P_{collfast}$($S$ + AckTimeOut)

$effS_{RTS}$ = average system delay + $P_{succslow}$($RTS_{slow}$ + $CTS_{slow}$ + $S_{slow}$ + $ACK_{slow}$+ 3*SIFS ) + $P_{succfast}$($RTS_{fast}$ + CTS + $S_{fast}$ + $ACK$ + 3*SIFS) + $P_{collslow}$($RTS_{slow}$ + AckTimeOut) + $P_{collfast}$($RTS$ + AckTimeOut)

where

$P_{succslow}$ = Probability of a successful transmission by a slow host

$P_{succfast}$ = Probability of a successful transmission by a fast host

$P_{collslow}$ = Probability of an unsuccessful transmission by a slow host

$P_{collfast}$ = Probability of an unsuccessful transmission by a fast host

$ACK_{slow}$ = Time taken by a slow node to send an ACK

$ACK_{fast}$ = Time taken by a fast node to send an ACK

$RTS_{slow}$ = Time taken by a slow node to send RTS

$RTS_{fast}$ = Time taken by a fast node to send RTS

$CTS_{slow}$ = Time taken by a slow node to send CTS

$CTS_{fast}$ = Time taken by a fast node to send CTS

$S_{slow}$ = Time taken by a slow node to send packet S

$S_{fast}$ = Time taken by a fast node to send packet S

$ACK = 1/(n-1)ACK_{slow} + (n-2)/(n-1)ACK_{fast}$

$CTS = 1/(n-1)CTS_{slow} + (n-2)/(n-1)CTS_{fast}$

$RTS = 1/(n-1)RTS_{slow} + (n-2)/(n-1)RTS_{fast}$

$S = 1/(n-1)S_{slow} + (n-2)/(n-1)S_{fast}$

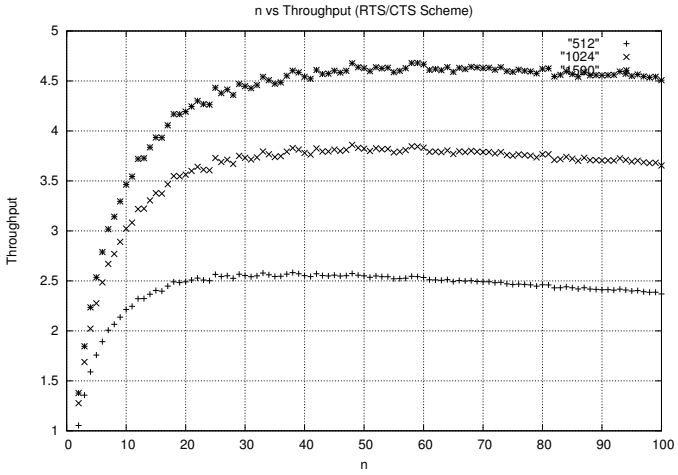


Fig. 5. Performance Anomaly in 802.11

It is observed that the throughput obtained is smaller compared to the case when all hosts transmit at the same rate (Figure 5). This is due to the capture of the channel by the slow host. But the impact of the low bit rate on the throughput is reduced as the number of hosts is increased because the slow host uses a diminishing portion of the channel. The plot indicates that throughput increases till *n* reaches 32. The variation of the curve is less beyond that point, since the share of the channel given to the slow host is decreased. [11] has done a similar study using the basic access scheme.

## III. Rate Switching Mechanism

### A. Design

The IEEE 802.11 standard[1] does not specify the algorithm for switching the data rate. It is up to the vendors

to design and implement such algorithms. Different approaches that can be used for rate switching include FER, SNR and throughput based methods. The SNR based method is not practically used, as the timely and reliable delivery of the SNR information, which is available at the receiver, cannot be guaranteed. It is also observed that the SNR information reported by most of the cards is not accurate. A better alternative is to go for FER based methods using the information available at the transmitter. This allows the design to be applicable even in systems where the interface does not provide SNR information.

*1) Statistics available at the device driver:* The driver reports the following events.

- Successful transmission of the packet. In this case, the number of ACK failures (retries) is reported.
- Frame Error.

Rate switching is done based on the information provided by the driver. An unsuccessful transmission implies that even after 7 MAC-level retries (as specified in the 802.11 standard[1]), the packet is not delivered at the receiver end and hence the rate is to be brought down to a lower value. Another situation when rate is to be lowered is when transmissions are possible at higher rates but with more retries. To study the effectiveness of down scaling the rate when packets are transmitted with more retries, the following analysis has been done.

The transmission time for a packet requiring n number of retries (excluding contention for channel) is calculated as:

$effS_{BAS}$ = ($S$+SIFS+$ACK$) + $n$*($S$+AckTimeOut)

$effS_{RTS}$ = ($RTS$+$CTS$+$S$+$ACK$+3*SIFS) + $n$*($RTS$+AckTimeOut)

where

$effS_{BAS}$ = effective system time for sending a packet under Basic Access Scheme

$effS_{RTS}$ = effective system time for sending a packet under RTS/CTS Scheme

$S$ = Time taken to send a data packet

$n$ = number of retries, it is varied from 0 to 6.

Throughput = $pktsize/effS$ where $pktsize$ is the packet size in bits.

Figures 6, 7, 8, 9, show the throughput for Basic Access Scheme (BAS) and RTS/CTS Scheme (RCS). As can be observed from the graphs, even if multiple retries are required at 5.5Mbps or 2Mbps, throughput cannot be enhanced much by switching to lower rates. The results indicate that if the number of retries required at 11Mbps is greater than a threshold, say RETRY_THRES, it is beneficial to down scale the rate to 5.5Mbps. If we examine Figures 6 and 7, we can find that the RETRY_THRES value is 0. For Figure 8, the retry value at which the switching needs to be done is 1 and Figure 9 shows the value to be 2. The optimal value of RETRY_THRES is taken to be 2, considering the different scenarios.

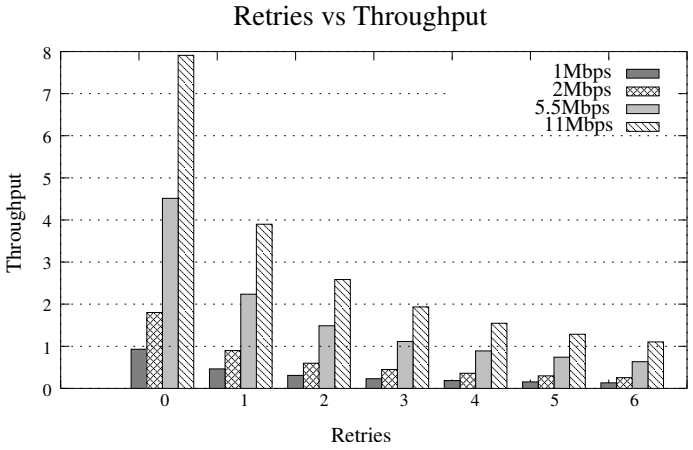


Fig. 6. Retries vs Throughput (BAS,Pktsize=512)

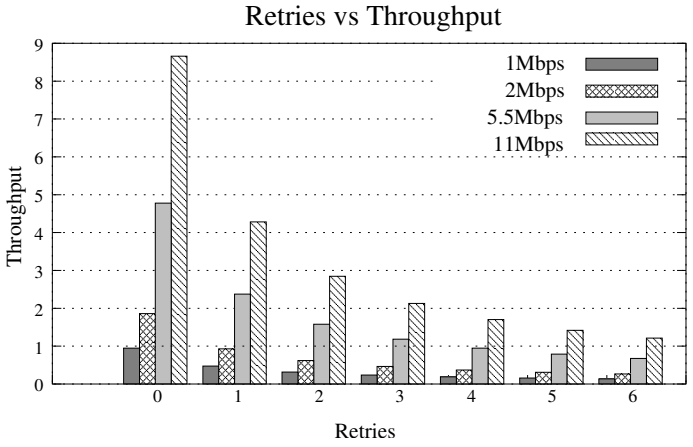


Fig. 7. Retries vs Throughput (BAS,Pktsize=1024)

*2) State Machine:* The state machine shown in Figure 10 uses frame error for predicting the optimal rate for 802.11b networks. The various states maintained are the following.

- STATE_110
  When transmission is at the highest possible rate (11Mbps), the station is in STATE_110 state. If the number of retries required for a successful transmission is greater than RETRY_THRES, the rate should be lowered. Let *succ_with_retrygtthres* represent the number of consecutive successful transmissions requiring more than RETRY_THRES retries. If *succ_with_retrygtthres* is greater than a particular

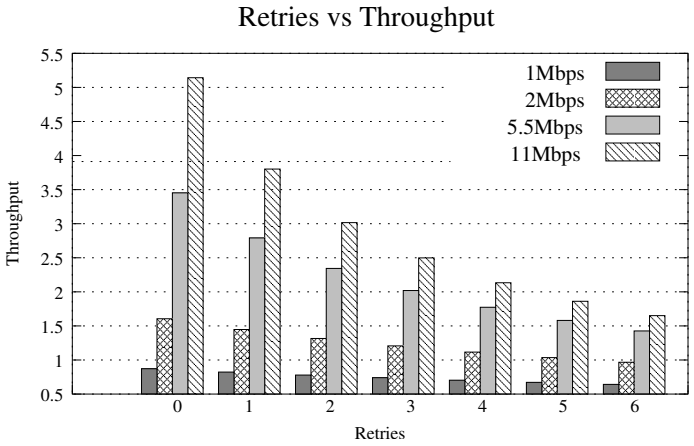


Fig. 8. Retries vs Throughput (RCS,Pktsize=512)

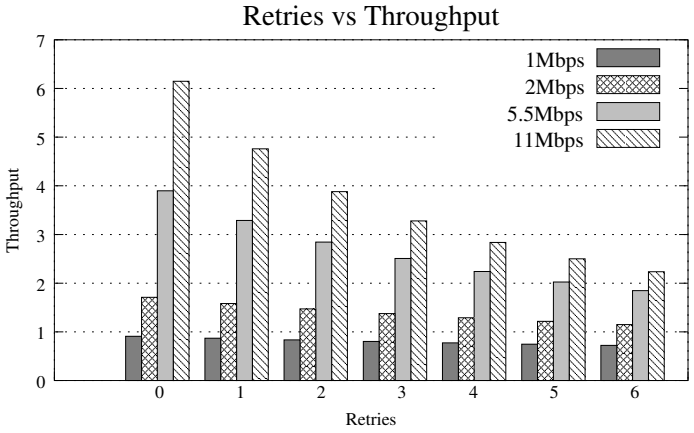


Fig. 9. Retries vs Throughput (RCS,Pktsize=1024)

threshold, say SUCC_RETRYGTTHRES_THRES, we move on to the FALLBACK_55 state. A packet loss indicates that it has undergone 7 retries and it is a safe measure to decrease the rate. Let *error* be the count of the number of consecutive packet losses. If *error* is greater than ERR_THRESHOLD, the rate is reduced to 5.5Mbps and system goes to STATE_55.

- FALLBACK_55
  This is to check if transmission is possible at 5.5Mbps with no retries. A successful transmission in a single attempt takes the system to STATE_55. Otherwise, there is no advantage in down scaling the rate to 5.5Mbps. As in STATE_110, we monitor the number of successful transmissions requiring retry (*succ_with_retry*). When *succ_with_retry* is greater than the threshold, SUCC_RETRY_THRES, the system goes back to STATE_110. A packet loss causes the system to move to STATE_55 as in STATE_110.
- STATE_55

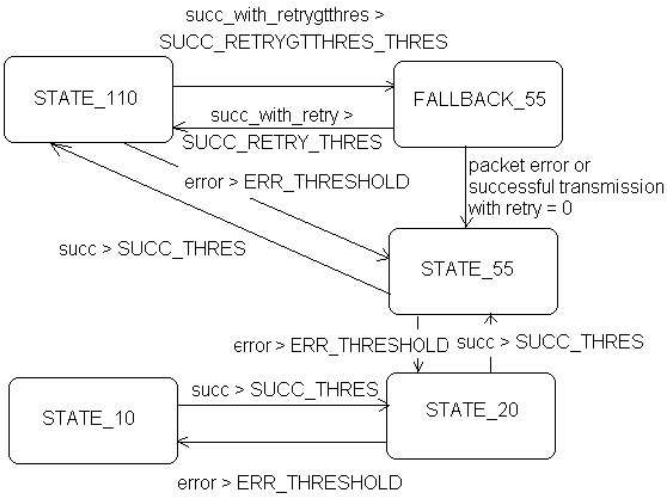


Fig. 10. Rate Switching state diagram

Transmission rate is 5.5Mbps at this state. If the number of consecutive packet losses is greater than ERR_THRESHOLD, it decrements the rate to 2Mbps and goes to STATE_2. A count of the number of successful transmissions(*succ*) is maintained and when it reaches SUCC_THRES, rate is incremented to 11Mbps.

- STATE_20
  This state stands for transmission at 2Mbps. The same transitions as in STATE_55 are used for incrementing and decrementing the rate.
- STATE_10
  Transmission at 1Mbps.

The state machine is maintained for each neighbor. Initially, the system is in STATE_110. As the nodes start communicating with each other, the system adapts to the optimal rate. In addition, an inactivity timer is maintained as part of the state machine, which resets the state to STATE_110 if there is no communication between the nodes for a reasonably long period.

### *B. Simulation*

To study the effectiveness of the proposed state machine, simulations have been run using CMU Monarch group's wireless extension to ns2. But ns2 does not provide multi-rate functionality. Multi-rate support has been added to ns2 and the state machine has been implemented.

*1) Modifications to ns2:* We implemented the Modulation classes in ns2 so as to add multi-rate capability. The probability of bit error, Pe for each of the modulation schemes is calculated using the analysis done by [9] for an 802.11 system in an indoor environment. The analysis computes Pe assuming multi-path Rayleigh channel. The

probability of bit error for the various modulation schemes can be observed from Figure 11.

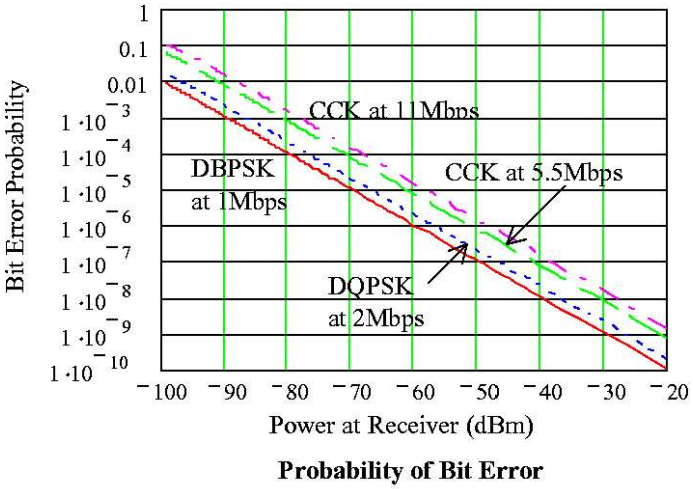


Fig. 11. Receiver Power vs BER

Once Pe of the system is determined, the packet error rate (PER) can be determined as follows:

$$PER = 1 - (1 - Pe)^n$$

where n is packet size in bits. If PER is within acceptable limits (Pa), the packet is passed on to the higher layers.

The automatic rate control state machine has been added to ns2. A link table is maintained at each node which keeps track of the current state of the state machine and the transmission rate to all its neighbors.

*2) Simulation:* We have simulated the automatic rate control mechanism using the modified ns2 simulator. A two node set up is used and the distance between the two nodes are varied so that communication is possible at 11Mbps, 5.5Mbps, 2Mbps and 1Mbps. The experiments has been run with manual configuration (explicitly setting the rate) and with automatic rate control. Table II lists the various parameters used for the simulation.

Figure 12 shows the TCP throughput for the various transmission rates. It can be seen that the throughput for the proposed rate control mechanism is quite close to the manual configuration. The decrease in throughput is due to packet losses resulting from the attempts to increment the transmission rate. When transmissions are possible at 11Mbps but with multiple retries, automatic rate control outperforms manual configuration, the reason being that while manual configuration sends data always at 11Mbps, rate control mechanism switches to 5.5Mbps whenever required. At 11Mbps, throughput remains the same for both the schemes.

TABLE II
SIMULATION PARAMETERS FOR AUTOMATIC RATE CONTROL

| Parameter | Value |
|---|---|
| Number of Hops | 2 |
| Routing Protocol | DSDV |
| Propagation Model | Shadowing |
| Traffic (TCP) | FTP |
| Traffic (UDP) | CBR |
| Packet Size | 1024 |
| SUCC_THRES | 50 |
| ERR_THRESHOLD | 1 |
| SUCC_RETRYGTTHRES_THRES | 2 |
| SUCC_RETRY_THRES | 2 |

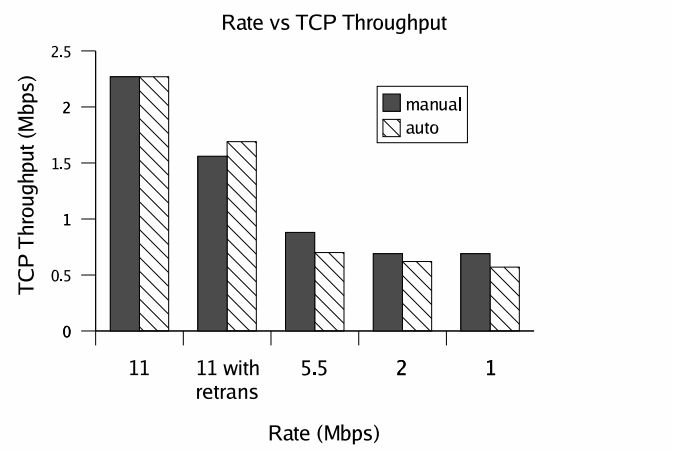


Fig. 12. Rate vs TCP Throughput(Simulation)

The UDP throughput (Figure 13) shows similar behavior.

### C. Implementation

To understand how the system adapts to the optimal rate under real-world situations, the rate control mechanism has been implemented in the device driver. A link table is maintained at each node, which keeps track of the current state of the state machine and the transmission rate to each of its neighbors. The same set of experiments as that of simulation has been done to analyze the performance. The experimental setup consists of one laptop equipped with a D-Link 650+ card and a desktop machine with a SparkLAN card. The distance between the machines has been varied so that transmission is done at 1Mbps, 2Mbps, 5.5Mbps and 11Mbps. Iperf traffic has been used to generate traffic and to collect the statistics.

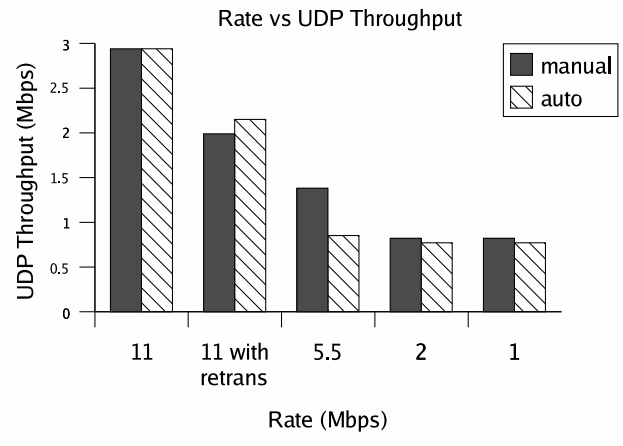


Fig. 13. Rate vs UDP Throughput(Simulation)

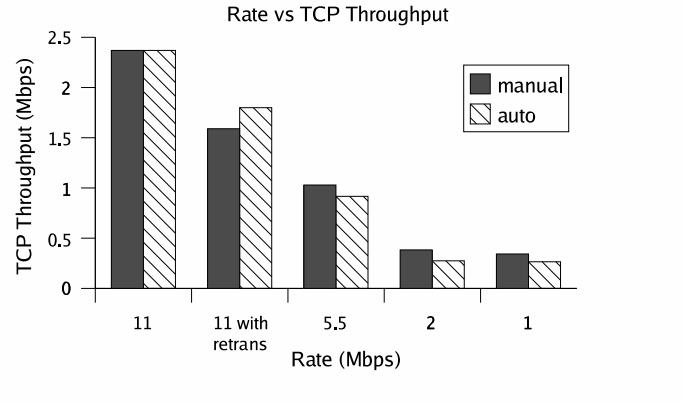


Fig. 14. Rate vs TCP Throughput

Figure 14 plot the TCP throughput against the transmission rate. The results correlate with the simulation results.

## IV. Metrics for Ad-Hoc Routing

In an ad-hoc wireless network, each node acts as a router, forwarding packets for other nodes. Static routing may not be adequate for such networks because of the frequent changes in topology and channel conditions. The most widely used dynamic routing protocols include DSDV, DSR and AODV.

Current dynamic routing protocols like DSDV use minimum hop count as the metric for computing optimal path. We added transmission time and rate metrics into the Grid Ad-hoc networking project's DSDV implementation. Grid is a system for routing in wireless ad hoc mobile networks. It is implemented as part of the Click modular router and is written in C++. The Grid code is a set of Click elements that can be put together in various ways to run DSDV, DSR or geographic forwarding. It can be run at user level or kernel level in Linux. In addition to hop count metric, Click supports ETX[7].

The following sections explain the basic operation of DSDV and provide the implementation details of the metrics.

TABLE III
Typical DSDV Routing Table

| Destination | Next hop | Seq# | Metric |
|---|---|---|---|
| 192.168.16.2 | 192.168.16.2 | 23 | 1 |
| 192.168.16.5 | 192.168.16.2 | 21 | 2 |
| . | | | |
| . | | | |

### A. Operation of DSDV

DSDV is a distance vector protocol, which uses periodic broadcasts to set up the routes. Each node maintains a routing table whose format is given by Table III. Route advertisement packets containing the complete routing table is broadcasted periodically by each node (full dump). A sequence number is associated with each routing table entry, which reflects the age of that entry's routing information. Each node maintains a sequence number of its own, which is incremented and included in its own entry in every full dump that it originates. The sequence numbers for the other entries in the full dump are copied from its routing table.

When a node receives a routing advertisement, it does the following actions. Suppose the update is from node Y for destination D with metric $m$ and sequence number $s$. If there is no entry for destination D, an entry is made in the routing table with Y as next hop and $m+1$ as the metric. The increment in the metric is to account for the hop count value. If there exists an entry for D in the routing table and its sequence number is lesser than $n$, the routing update from Y replaces the current entry. If the sequence numbers are the same, the new update is accepted provided $m$ is better than the current metric. Otherwise, the update is ignored.

Each routing table entry has an associated weighted settling time (WST). Settling time is the time between when a route with a given sequence number was first received and when the best route with the same sequence number was received. When a route entry is replaced, only the changed information is propagated to the other nodes (triggered update). Triggered updates are not sent until at least 2*WST has passed since first hearing the current

sequence number. This is to prevent nodes from sending a route entry, which might be replaced by a better route later.

### B. Modifications to DSDV

To use the rate and Xtime metric, the following changes have been made to the original DSDV protocol. Suppose node X receives an advertisement from Y for destination D, which has a better metric $m$, an entry is made for destination D with next hop as Y. But in stead of incrementing $m$ by one as done for hop count metric, the cost of reaching Y is added to $m$. The cost of reaching Y is obtained from the link table maintained by each node. The link cost is calculated from the transmission rate to the next hop in the case of rate metric and from the transmission time in case of Xtime metric.

TABLE IV
LINK COST FOR RATE METRIC

| Link Rate | Weight |
|---|---|
| 11 Mbps | 1 |
| 5.5 Mbps | 2 |
| 2 Mbps | 5 |
| 1 Mbps | 10 |

### C. Computation of Link cost

*1) Rate metric:* Cost assigned to each link is proportional to its link rate in the case of rate metric. It is computed as

$$metric = 1/r * 10$$

where *r* represents the current link rate. The multiplication is done so as to round off the link cost to an integer value. Table IV lists the weights assigned for different link rates.

*2) Xtime metric:* The average transmission time taken by each packet is computed by measuring the time taken to get the ACK, after the packet is sent. This can be used as a measure of the link quality, since it takes into consideration the contention for the channel, the air time taken to send the packet and receive the ACK. Hence both the forward and reverse channel characteristics are taken into account for routing.

The transmission time for each packet $P_{xtime}$ is measured and the current transmission time per bit is given by

$$current\_xtime = P_{xtime}/packetsize$$

Since the transmission time fluctuates, smoothening of the link cost needs to be done. This is done by computing new link cost as

$$link\_cost_{new} = w*link\_cost_{old}+(1-w)*current\_xtime$$

More weightage is given to the past history by choosing a higher value for the smoothing factor $w$. Experiments have been done to determine the right value of $w$ and a value of 0.85 seems to be good. When there is no communication between the nodes, link cost is initialized to a value proportional to the current link rate, *r*.

$$link\_cost_{initial} = 1/r * 10$$

As the nodes start communicating, the actual transmission time is learnt.

TABLE V
TYPICAL LINK TABLE

| Neighbor | Rate | Link Cost | Timer |
|---|---|---|---|
| 00:0D:88:C0:AA:75 | 55 | 2 | T1 |
| 00:0D:88:F3:F0:D4 | 1 | 10 | T2 |
| . | | | |
| . | | | |

### D. Implementation in Click

The linux kernel module of the Click modular router has been used for the implementation. At kernel level, the Click module runs a separate kernel thread. Click code sits between the kernel's network stack and the device drivers. Click presents a pseudo-device to the kernel for sending and receiving packets to the kernel. Running Click in kernel provides more flexibility and eases the task of exporting the link cost information provided by the device drivers to the router.

The link table (Table V) maintained at each node has an entry for the link cost. The link cost is computed from the current transmission rate or the transmission time and is stored in this field. An inactivity timer is associated with each link, which resets its cost to its initial value, if there is no data transfer through the link for a reasonably long period. The timeout for the experiments has been set to 15 seconds. The metric used for DSDV routing is implemented as a different class in the Click modular router. Click has support for hop count and ETX metric. We added the Ratemetric class so as to support rate and Xtime metrics. The class is the same for both the metrics; only information about the link cost, which is used to

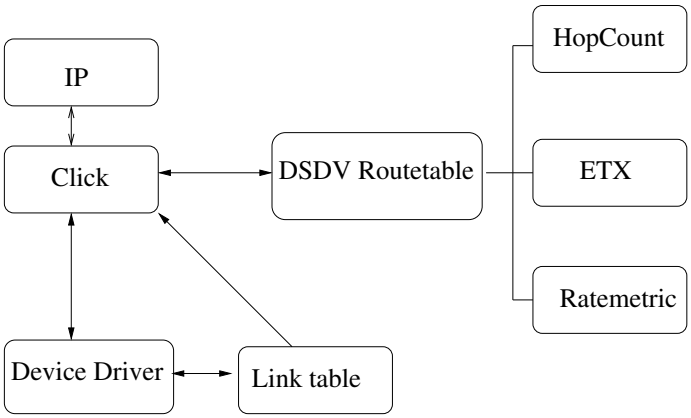


Fig. 15. Design

compute the metric is different. The interactions between the different layers is shown in Figure 15.

The Grid implementation of DSDV has support for two options namely, USE_OLD_SEQUENCE_NUMBER and USE_GOOD_NEW_ROUTE. The first option essentially prevents the use of the current update until it is ready for advertisement. The second option is a modification of USE_OLD_SEQUENCE_NUMBER. It suggests the use of the new update, even if it is not ready for advertisement, as long as it has a better metric. These options were enabled for the experiments so as to prevent the use of a route with bad metric. Usually, new sequence numbers along one-hop path is heard first. There might exist a better multi-hop path, but if the above mentioned options are not used, packets are routed along the bad metric path. With those options enabled, it can be ensured that DSDV uses the previous best route until WST has expired and the best route for the new sequence number has been heard. In case the new update has a better route than previous one, new route is used without waiting for the expiry of WST.

### *E. Experiments*

Experiments have been run for different scenarios to gain an insight into the performance characteristics of different routing metrics. The four metrics compared include minimum hop count (used by standard DSDV), ETX, rate and Xtime metric. We have used five machines running Click modular router. Two of the machines have been equipped with SparkLAN cards and three with D-Link 520+ cards. The TCP throughput and round trip delay for the packets have been measured for the following scenarios.

*1) Scenario 1:* A three node setup as shown in Figure 16 has been used for running the experiments.

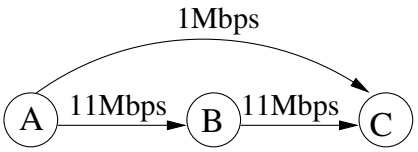


Fig. 16. Scenario 1

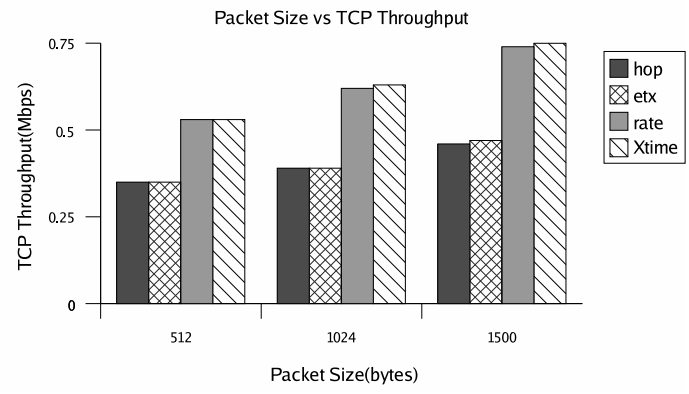


Fig. 17. TCP throughput(Scenario 1(one conn))

The TCP throughput and RTT has been measured with one connection (A to C) and three connections (A to B, B to C, A to C). The results can be observed from Figures 17-19.

It can be observed that when the number of connections is one, the performance of rate and Xtime metrics is almost the same and is much higher than that of hop count and ETX metric (Figure 17). This is because, hop count metric does not take into consideration, the link rate of the path through which packet is routed. ETX uses the

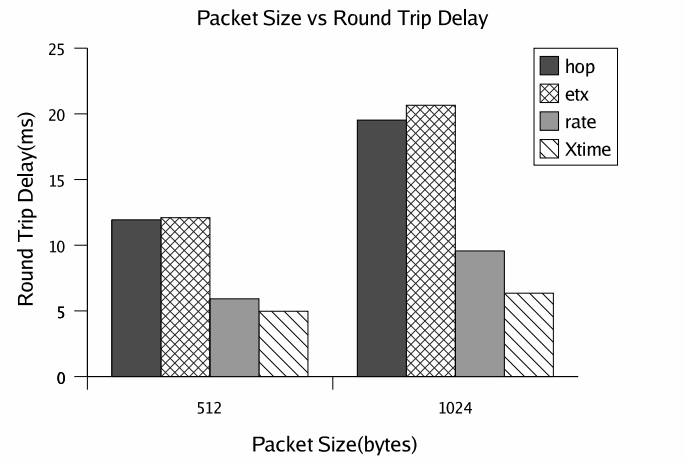


Fig. 18. Round Trip Time (Scenario 1)

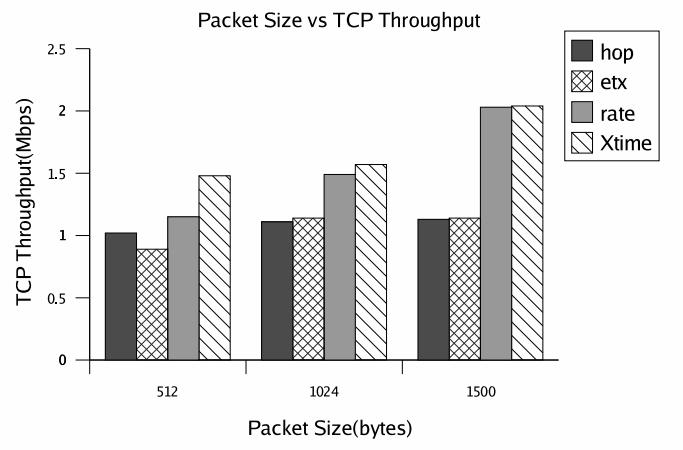


Fig. 19. TCP throughput(Scenario 1(3 conns))

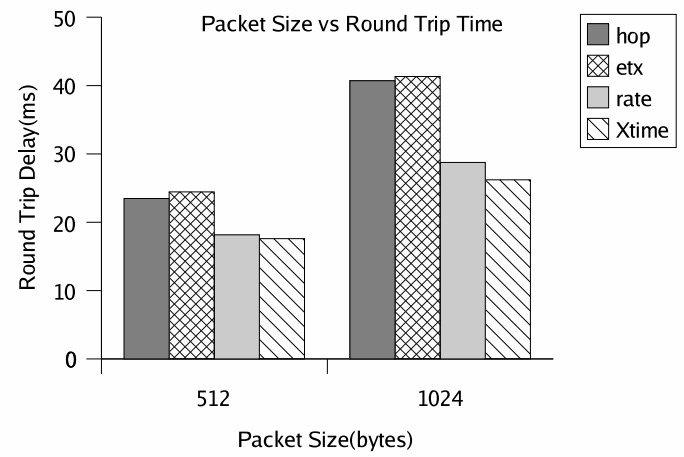


Fig. 21. Round Trip Time (Scenario 2)

delivery ratio to compute the path. The reason for the decreased throughput for ETX metric can be the overhead due to link level probes and the use of single sized packets(134 bytes) for finding optimal path. Also, ETX metric is not designed for networks with links that run at a variety of bit rates.

The round trip time for the packets were measured using ping test for different packet sizes. It is seen that packets routed using rate and Xtime metric has less delay compared to the other metrics (Figure 18).

Experiments have been repeated with the same setup for three connections, A to B, B to C and A to C. The idea has been to load the high link rate path and Xtime metric is found to perform a bit better than rate metric. As in the previous case, hop count and ETX throughput is less (Figure 19).

*2) Scenario 2:* A 5-node wireless testbed as shown in Figure 1 has been used for the next round of experiments.

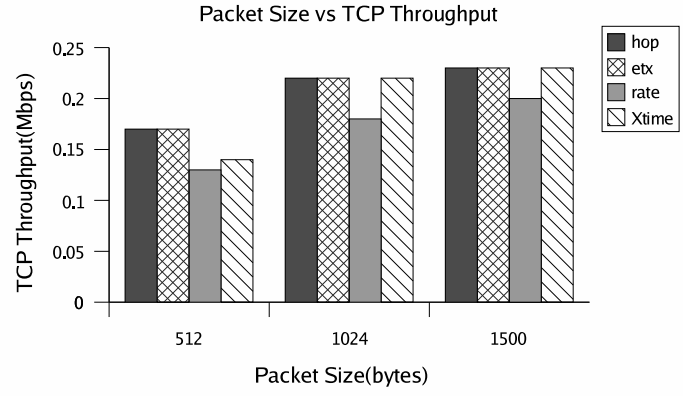


Fig. 20. TCP throughput(Scenario 2(one conn))

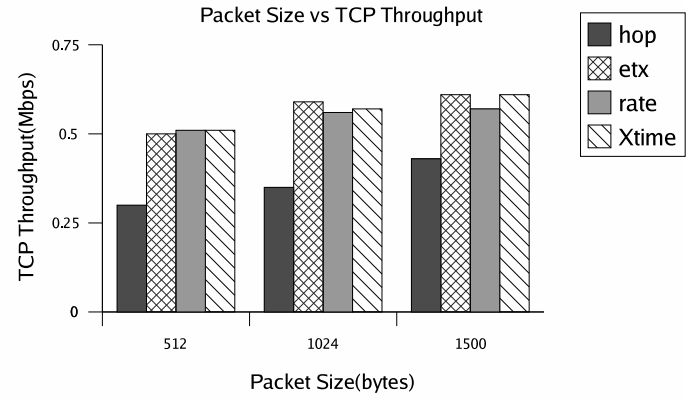


Fig. 22. TCP throughput(Scenario 2(5 conns))

With one connection (A to C), it can be observed that rate metric throughput is less, because of the extra backoffs required for traversing more number of hops (Figure 20). In this case, the optimal path from A to C would be to follow the slow links from A to B and B to C. ETX metric and hop count metric performs well in such a case. Xtime metric performance is similar to that of ETX and hop count metric except when the packet size is 512 bytes. This deviation can be attributed to the route oscillations. The round trip delay for scenario 2 shows similar behavior as that of the three node setup (Figure 21).

We repeated the experiments with 5 connections (A to D, D to B, B to E, E to C and A to C). The four one-hop connections along the high throughput path (A-D, D-B, B-E, E-C) helps to build up the throughput in the case of rate metric. Throughput is less for hop count metric, since all traffic for the connection A-C goes only through the slow links and the slow hosts capture the channel for long period (Figure 22).

In addition to the above experiments, they were repeated with multiple flows from A to C; the behaviour/throughput was found to be about the same with respect to all the metrics considered. Due to lack of space, we omit discussion of these results here.

## V. Related Work

### A. Rate switching mechanism

[10] classifies the current approaches for rate switching into three main categories - throughput based, FER based and SNR based methods. SNR provides reliable information about the link quality without any delay or need for collecting the statistics, since it is firmly related to BER. A critical issue in using this approach is the exact relationship between link quality and SNR. It depends to a much extent on the radio model assumed. Also, not all interfaces export the SNR information to the upper layers. SNR information is available at the receiver and the reliable and timely delivery of this information to the sender is of prime concern. All these factors make it an unsuitable candidate to trigger rate switch. ARF algorithm[16] used in Agere Systems uses the number of ACK misses as a parameter for rate control. [8] proposes the use of a combination of FER based method and Received Signal Strength (RSS). But RSS cannot be used for rate control at a node since the propagation characteristics of the forward link may not be equivalent to the reverse link.

### B. Routing metrics

Dynamic routing protocols proposed for use in wireless networks include DSR[12], DSDV[14], AODV and so on. These protocols use hop count as the metric for computing shortest path. Currently, research community has turned their attention towards utilizing the link characteristics as routing metric. The major works in the field of multi-rate aware routing protocols are MAS and MTM. [15] proposes the use of a thin layer, MAS (Multi-rate Aware Sub layer) in between IP and the link layer. Routing overhead is more in this case as both the IP and MAS layer flood the system with periodic broadcast messages. Another related work is MTM (Medium Time Metric) [4], which is independent of the routing protocol used. The metric takes into account the transmission rate as well as an estimate of the back off delay (310 us). But this estimate can be deceiving when the path is congested or under bad channel conditions. ETX[7] uses the delivery ratio to find optimal path. But the probe messages add more traffic into the system. The use of a single packet size for probing can lead to inaccurate metrics for other packet sizes.

## VI. Conclusions and Future Work

This report introduces the design and implementation of an FER-based automatic rate control mechanism, which uses the statistics available at the device driver. The results drawn from simulation studies and real world experiments show that its performance is comparable to the manual configuration. We propose transmission time as a metric to guide routing decisions. It accounts for the contention for the channel, air time to send the packet and the asymmetry of the links. Measurements done on two wireless testbeds show that transmission time metric performs consistently well across the different scenarios considered.

The proposed automatic rate control mechanism uses a specific threshold for incrementing the rate. It is seen that for gaining higher throughput, the value of the threshold need to be large. In our experiments, the value has been taken as 50. As a future work, adaptive modification of the threshold value depending on the channel conditions can be done. The proposed metric was implemented and tested on DSDV, which is a distance vector protocol. Distance vector protocols take more time to converge than link state protocols. It will be interesting to see how the metric performs when used with link state protocols like OSPF.

## References


[1] IEEE std 802.11b-1999. http://grouper.ieee.org/groups/802/11/

[2] Rice Monarch Project, Wireless and mobility extensions to ns-2, http://www.monarch.cs.rice.edu/cmu-ns.html.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. Wireless Sensor Networks: A Survey. *Comput. Networks*, 38(4): 393–422, 2002.

[4] Baruch Awerbuch, David Holmer, Herbert Rubens. High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks. WONS 2004: 253-270

[5] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, 1987.

[6] G. Bianchi. Performance Analysis of IEEE 802.11 Distributed Coordination Function. *JSAC Wireless series*, 18(3):535–547, 2000.

[7] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, Robert Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *Proc. of 9th annual international conference on Mobile computing and networking*, pages 134–146. ACM Press, 2003.

[8] del Prado Pavon J. and Sunghyun Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *IEEE International Conference on Communications*, pages 1108–1113, May 2003.

[9] Michael Fainberg. A Performance Analysis of the IEEE 802.11b Local Area Network in the presence of Bluetooth Personal Area Network. Master's thesis, Polytechnic University, June 2001.

[10] Ivaylo Haratcherev, Koen Langendoen, Inald Lagendijk, Henk Sips. Application Directed Automatic Rate Control, Telematica Instituut, Dec'02.

[11] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, USA, March-April 2003.

[12] David B Johnson, David A Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, eds., *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[13] Byung-Jae Kwak, Nah-Oak Song, and L. E. Miller. Analysis of the Stability and Performance of Exponential Backoff. In *IEEE Wireless Communications and Networking Conference*, 2003.

[14] Charles Perkins, Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94 Conf. on Communications Architectures, Protocols & Applications*, pages 234–244, 1994.

[15] Yongho Seok, Jaewoo Park, and Yanghee Choi. Multi-rate Aware Routing Protocol for Mobile Ad Hoc Networks. IEEE VTC 2003-Spring, Jeju, Korea, April, 2003

[16] A J van der Vegt. Auto Rate Fallback Algorithm for the IEEE 802.11a Standard. Referred in [10].

[17] Kaixin Xu, Mario Gerla, and Sang Bae. Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks. *Ad Hoc Networks Journal*, 1(1):107–123, 2003.