Explicit Rate-based Congestion Control for Multimedia Streaming over Mobile Ad hoc Networks

Nazia Zaman, Kazi Chandrima Rahman, Syed Faisal Hasan

Abstract— **Design of congestion control mechanism for multi-media streaming over the Mobile Ad-hoc Networks (MANET) is challenging. Streaming applications require a smooth transmission rate which the Internet is unable to provide whenever there is congestion in the network. The standard TCP congestion control mechanism is not able to handle the special properties of a shared wireless multi hop channel well. In particular, the frequent changes of the network topology and the shared nature of the wireless channel pose significant challenges. In this paper, we propose a router assisted approach, where routers provide explicit feedback which allows quick increase of throughput.**

Index Term— **Congestion, MANET, multimedia streaming, rate-based.**

I. INTRODUCTION

THE proliferation of wireless networks is driving a revolutionary change in information society. Due to the availability of wireless interfaces on mobile devices like laptops, PDAs and cell phones etc., wireless networks are getting very popular day by day. The wireless channel now supports a higher data rate which has made real time multimedia applications like radio broadcasting, video conferencing, and real-time environment monitoring, etc. possible. Usage of these applications through mobile ad hoc networks is gaining immense popularity now a day.

A mobile ad hoc network (MANET) is a wireless network, consisting of many mobile nodes connected by wireless links. Each node functions not only as an end-system, but also as a router and rely on each other to keep the network connected. In MANET, nodes are free to move randomly and organize themselves arbitrarily. This random behavior of ad hoc networks causes the topology of wireless network to be changed rapidly and unpredictably. Moreover, node's mobility puts an extra burden on TCP's congestion control mechanism. As a result, traditional congestion control mechanism, applied by the Transport Control Protocol TCP [1], is unable to catch up the network dynamics of ad hoc networks.

Kazi Chandrima Rahman is with the Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh. E-mail: kchandrima@gmail.com.

Syed Faisal Hasan is with the Department of Computer Science and Engineering, University of Dhaka, Bangladesh. E-mail: hasansf@gmail.com.

In this paper, we have proposed a novel Explicit Rate-based Congestion Control mechanism (XRCC), for supporting applications like multimedia streaming over MANET. The following subsections give the brief idea about the problem, our proposed solution to solve the problem, and an outline of this paper respectively.

A. Problem statement

In this section, we have tried to point out the problems of using TCP congestion control mechanism for supporting multimedia applications in mobile ad hoc networks:

• Multimedia applications usually have a higher bandwidth requirement as compare to the usual Internet applications like file transferring [2].

• Supporting multimedia streaming over MANET presents a number of practical challenges.

• Transmission Control Protocol (TCP) [1], which is the most widely used transport protocol, is not suitable for applications like streaming in mobile ad hoc wireless network. This is because of the fact that TCP interprets a missing packet as an indication of network congestion which is not always true for mobile ad hoc networks. Packet loss may occur due to MANET's unique characteristics, like node mobility, channel bit errors, medium contention and route failures. Because of these special characteristics, packet loss rates on wireless links are much higher than the corresponding wired links. The TCP protocol reacts to these wireless losses in the same fashion as it would react to packet losses due to congestion, because it has been designed to react to losses in only that way.

• Moreover, upon any congestion event, TCP reacts conservatively and halves its transmission rate. Such a drastic change in transmission rate could deteriorate the performance of these streaming applications. So uniformly applying congestion control for each loss will lead to unacceptable performance degradation.

Applications like multimedia streaming is often transported using User Datagram Protocol (UDP) [3], [4]. But UDP has the problem that it does not incorporate any congestion control mechanism. If UDP is used for multimedia applications, these unresponsive flows will compete unfairly with other

TCP flows. As a result, the congested network may cause significant degradation of the network performance. To prevent such congestion collapse, multimedia applications need a protocol with appropriate congestion control mechanism. However, TCP controls congestion in an end to end approach. It responds conservatively to packet losses as it does not get accurate information about network's congestion state. Again, TCP does not allow quick increase of throughput. At most one packet can be increased in a RTT which is not suitable for streaming applications. Sometimes streaming applications need to increase more to maintain a smoother rate. In addition to these, TCP's retransmission scheme may be unnecessary for loss tolerant multimedia streaming applications.

B. Proposed solution

In this paper, a router assisted approach, which allows quick increase of throughput, is proposed. Since routers are the central places where congestion takes place, they are in a better position to detect and respond to such condition. Hence, an explicit rate-based congestion control where senders' flow is controlled by the explicit information in the feedback packets from the routers can outperform the TCP and TCP like protocols' conservative behavior for multimedia streaming over MANET.

End to end approach of congestion control [5] lacks the knowledge of how much the network is congested. With this approach, the end hosts assume routers as a black box and thus cannot measure the actual congestion states of the network. Moreover, one important feature of multimedia streaming applications is to maintain a smoother data rate to keep the play-out time stable. In a mobile ad hoc network (MANET), the problem become more critical due to MANET's properties, such as--node mobility, channel bit errors, medium contention and route failures. So, it is necessary to know the actual congestion state and responds accordingly. As the routers are responsible for dropping packets when there is congestion in the network, they should respond such situations.

A router assisted approach, Explicit Rate-based Congestion Control (XRCC), which improves the real-time streaming performance over MANET, is proposed in this paper. With this approach, routers will provide feedback through inserting the rate information into the passing packets. After receiving the packets with explicit rate information, the destination node should propagate this information to the sender through an acknowledgment packet.

XRCC contributes in the following two fields:

- *Detecting losses due to congestion*. With the feedback information, sent by the intermediate nodes, our proposed solution is able to distinguish congestion losses from wireless losses. As a result, the performance degradation of multimedia streaming, caused by the drastic change in sending rate, is reduced.
- *Adjusting the sending Rate using feedback information*. In this paper, the sender nodes adjust their data sending by the rate feedback from the intermediate nodes, using XRCC. An intermediate node uses its queue length to calculate the explicit rate feedback and stamps it in the passing packet's header. After receiving this rate feedback, the sender adjusts its data sending rate accordingly.

The remainder of the paper is structured as follows:

Section I introduces the related works and background in the area of mobile ad hoc networks, congestion control and multimedia streaming. Section II describes the limitations of current approaches and the motivation behind the new proposal. Section III illustrates the proposed solution to improve the real-time streaming performance over mobile ad hoc networks. Section IV describes the simulation results of the proposed mechanism. Section V concludes this paper with possible future research directions.

II. BACKGROUND AND RELATED WORKS

Many of the Internet design concepts have been changed by the emergence of new multimedia applications. Moreover, as wireless networks are getting available on mobile devices, multimedia applications are becoming a popular medium for communication. But, multimedia applications over MANET, faces some problems due to MANET's unique properties. Thus, a pertinent congestion control mechanism is needed for these applications to operate in mobile ad hoc networks. As existing approach of congestion control has deficiency of the knowledge of actual situation, a router-assisted congestion control approach is proposed in this paper.

Wireless networks can generally be classified as wireless fixed networks, and wireless, or mobile ad-hoc networks. MANETs (mobile ad-hoc networks) are based on the idea of establishing a network without taking any support from a centralized structure. By nature, these types of networks are suitable for situations where either no fixed infrastructure exists, or to deploy one is not possible.

Communication in mobile ad-hoc networks is normally achieved through other mobile devices in the network. Each node of an ad hoc network is the destination of some information packets, while at the same time it also functions as an intermediate station for other packets on the way to their final destination. This multi-hop support in ad hoc networks makes communication between nodes outside direct radio range of one another possible. It's also the main difference between mobile ad-hoc networks and wireless LANs.

With the emergence of broadband wireless networks, local networks (both structured and ad-hoc WLANs) and wide area mobile networks, multimedia applications are also emerging. As wireless interfaces are becoming available on mobile devices like laptops, PDAs and cell phone etc., mobile ad hoc networks are getting very popular. Moreover, the wireless channel now supports a much higher data which has made possible real time multimedia streaming like radio broadcasting, video conferences, real-time environment monitoring, etc.

Multimedia traffic in the current Internet can be transported over either TCP or UDP. UDP does not perform any congestion control. Multimedia traffic is sent over UDP at a constant rate equal to the drain rate at the receiver. TCP does perform congestion control, but this control creates large fluctuations in the fill rate in the receiver buffer. This is far from optimal for the multimedia traffic, since a typical video traffic flow is highly sensitive to sudden and large rate changes. Multimedia traffic requires congestion control, but it requires a new approach to it, which would be more suitable for the end-to-end performance of multimedia applications.

In recent years, the need of multimedia streaming over the Internet drives the design of TCP-friendly congestion control mechanism [10], [11]. Among the classes of TCP-friendly congestion control mechanisms, the TCP equation-based approach has been one of the most well studied algorithm [12]. It relies on a 'TCP throughput equation, which captures the TCP throughput over a network path with certain loss rate and round-trip time (RTT). Past studies have shown that the TCP equation is able to achieve a reasonable fairness with competing TCP flows under a wide range of traffic conditions in wired networks, [12], [13]. Although TFRC serves multimedia streaming reasonably well in wire line networks, it degrades the performance of mobile ad hoc networks.

With the increasing availability of wireless medium on mobile devices multimedia applications are becoming a popular communication medium. Since MANET is a special type of network, congestion control mechanism for this field needs to be adapted to the specific properties of MANETs. In early research, multimedia streaming is carried by UDP flows [4]. But UDP is an unreliable protocol as it does not have any congestion control mechanism and this characteristic of UDP makes it suitable for transmission of multimedia applications with a smoother data rate. Since it does not deploy any congestion control mechanism, the unresponsive multimedia flows carried by UDP will compete unfairly with other responsive TCP flows. As a result, network performance may significantly degrade by the congested network. A number of studies [14],[15], [16], [17], [18] has shown than streaming audio and video are better served by a congestion mechanism which reacts slowly on packet losses, achieving smooth throughput changes.

Recent research in this field has focused on the development of TCP friendly congestion control since TCP's retransmission scheme may be too expensive or somewhat unnecessary for real-time multimedia streaming applications. But a research [19] shows that although TCP friendly mechanism for congestion control maintains a smoother throughput than TCP in mobile ad hoc networks, it obtains fewer throughputs than the competing TCP flows. Moreover, in MANET, congestion control often depends on the characteristics and nature of the applications being transported. However, we can broadly categorized the taxonomy of congestion control into two types, depending on how the congestion state of the network is measured, implicit congestion control, and explicit congestion control.

1) *Implicit Congestion Control:* Implicit congestion control is based on end-to-end measurement i.e. the end-systems measure the network congestion state.

TCP's AIMD (Additive Increase Multiplicative Decrease) controls flow implicitly. It assumes congestion by packet loss occurrences, and responds to the absence or presence of packet loss by additively increasing or multiplicative decreasing its transmission window size. The implicit AIMD algorithm performs poorly in MANET because of several reasons. First, it presumes packet loss as an indication of network overload and hence shrinks its transmission window size.

However, in MANET, packet loss can be occurred due to its special properties, such as re-routing, route failure. Second, AIMD's additive increase policy restricts its ability to acquire spare bandwidth to one packet per round trip time. In case of frequent re-routing the algorithm may never be able to catch up with the network dynamics. Third, as AIMD algorithm senses network overload by packet loss, bottleneck router queues may kept full even in the steady state. This can cause long queuing delay and a number of packets may be dropped due to the bandwidth of the wireless link fluctuations (wireless medium contention, inference, mobility).

In recent years, some variants of the AIMD [10], [11], and [20] have been proposed for the Internet. These algorithms differ in the increase and decrease equations to adjust the transmission window size. But, as they still rely on the bandwidth probing and congestion avoidance strategies, they exhibits almost the same problems as the original AIMD algorithm when applying over MANET.

TCP friendly congestion control [21], [18], which is also known as TCP equation-based approach, measures a flow's packet loss event rate and RTT during a steady state of the network. These measurements are used to obtain the flow's TCP-equivalent rate by the TCP equation. This approach of using statistical measurement helps equation-based method to react slowly to the network dynamics and to achieve a smooth rate control, which is beneficial to multimedia applications in the Internet [18], [12]. However, in MANET, it is difficult to obtain reliable statistics for the packet loss events at the end nodes.

2) *Explicit Congestion Control:* This type of congestion control relies on intermediate gateways, i.e. routers, to measure the network congestion state.

Explicit Congestion Notification (ECN) [22] is such a scheme in which each router marks a bit in passing packets IP header if there is any possibility of network congestion. This early detection of congestion is done by monitoring routers queue size. ECN indicates whether there is congestion, but it provides no idea about how much the congestion is. This binary information causes the end-systems to behave like the AIMD algorithm and as a result ECN suffers the similar problems as with the AIMD algorithm over MANET.

There is another scheme with implicit congestion control ATM forum's rate-based congestion control scheme for the Available Bit Rate (ABR) service [23], [24], [25]. ABR congestion control tries to fairly split the bandwidth left over from higher priority traffic to fully exploit the available throughput of the links. Intermediate routers convey the precise explicit rate information to the receivers and this is done by using a special cell called Resource Management (RM) cell. But there are some problems to use ATMs ABR congestion control in MANET as it assumes symmetric circuit and does not consider route failure and re-routing which are common scenarios in MANET.

A variation of the XCP [16] transport protocol for wired networks with high bandwidth-delay product is the WXCP [26]. WXCP uses explicit feedback from within the network and multiple congestion metrics. These are evaluated at the intermediate nodes, in order to avoid the necessity of probing for the highest available bandwidth. WXCP is a window-based approach and it cannot provide a smooth data rate, which is important for multimedia applications.

III. A NEW APPROACH: EXPLICIT RATE-BASED CONGESTION CONTROL (XRCC)

In mobile ad hoc networks, TCP is unable to maintain a smooth data sending rate, a requirement for supporting streaming applications, due to the unique properties of MANET. TCP uses a conservative end-to-end mechanism for controlling the congestion which deteriorates the network throughput. To solve this performance issue an explicit congestion control mechanism, which uses feedback from intermediate nodes on a network connection, is proposed in this paper.

A. Motivation

In the past, users had to download an entire multimedia file to their local hard disk drive before playing the multimedia content on the Internet. But, the situation has been changed during the last number of years. Streaming has matured and gained high acceptance among users of Internet-enabled devices. Again, with the integration of wireless network with various types of mobile devices, multimedia streaming is becoming a popular medium for exchanging information. As mobile ad hoc networks have a higher loss rate as compared to wired networks, streaming of multimedia content over these network needs some special cares to deal with these losses.

The traditional TCP congestion control mechanism works very well on the Internet. But, mobile ad hoc networks exhibit some unique properties which greatly affect the design of appropriate protocol in general and of a congestion control mechanism in particular. This vastly differing nature of a mobile ad-hoc network imposes a great problem on standard TCP congestion control mechanism.

Node mobility and a shared, wireless multi-hop channel are the most important among the special features of MANET. These properties cause unsteady packet delivery delays and packet losses. These delays and losses may be misinterpreted as congestion losses which are not desirable. Although this depends on the network type, packet losses which are not caused by network congestion can be much more frequent in wireless networks. This wireless loss can lead to wrong reactions of TCP congestion control. Moreover, observing packet losses is much harder in MANET, because transmission times and thus, also round trip times vary much more. These properties of MANET cause the performance degradation of multimedia streaming as TCP maintains an aggressive congestion control mechanism. Therefore, an appropriate congestion control is an absolute need for stable network and acceptable performance.

Motivation of our work is to design a router-assisted congestion control mechanism for multimedia streaming over mobile ad hoc networks which can outperform the traditional TCP. The router can provide feedback through which the sender node can distinguishes between congestion based packet loss and non-congestion based losses. In addition, to determine the type of loss, the sender also controls its sending rate using this feedback.

B. XRCC Design

We have discussed the design of the proposed solution, XRCC (explicit Rate-based Congestion Control), to the congestion control problem for multimedia streaming over MANET in the following subsections.

1) *Distinguishing Congestion Losses:* Our solution tries to determine the losses caused by congestion. As we have discussed earlier, in mobile ad hoc network, congestion is not the only reason behind packet loss. MANET's special properties, like node mobility, route failure, medium contention, also play a vital role behind these losses. But TCP, as an end-to-end protocol, could not distinguish the losses due to congestion, and as a result, number of false detection occurs. False detection comes up in two forms. We can consider congestion as an example to illustrate this. Network congestion may go undetected, or conversely congestion can be detected when the network is not congested at all. Using end-to-end measurements, the probability of undetected congestion is not high. When the network is congested, measurement metrics such as RTT or inter-arrival time indeed increase. However, with this single metric measurement, the probability of false congestion detection in a non-congested ad hoc network is quite high due to noisy end-to-end observations. As TCP does not have any knowledge about the type of loss, which has occurred at the intermediate nodes, it uniformly applies congestion control for all of these losses. And false detection makes the matter worse. This conservative behavior causes severe performance degradation, which is not desirable at all. In case of multimedia streaming over MANET, TCP also failed

to maintains a smooth data rate due to its saw tooth congestion control mechanism. Thus detecting the type of packet loss is of great importance in mobile ad hoc networks and is one of main contributions of this paper.

2) *Explicit Rate-based Congestion Control:* XRCC tries to control the congestion for applications like multimedia streaming. Our congestion control scheme depends on the feedback, form the intermediate nodes. The intermediate nodes send the rate information in the packet's IP header's options field, and this is done using the current queue length. Each of the intermediate nodes on the path from sender to receiver stamps the rate feedback in passing packet's IP header. After receiving this feedback information from the intermediate nodes, the receiver sends this rate information to the sender through acknowledgement packet. The sender then makes the adjustment by using this feedback information and also using its current sending rate.

The contribution of XRCC can be listed with the following two phases--

- Detecting Losses due to Congestion
- Adjusting the Sending Rate.

C. XRCC for Multimedia Streaming over MANET

Our congestion control mechanism XRCC depends on the feedback from the intermediate nodes which includes both information about the network congestion and the rate information. In this section, we describe the steps used by our congestion control mechanism. The intermediate nodes provide congestion feedback to the sender via the receiver. We describe the procedure of detecting the congestion losses in the next subsection. The following subsections describe the role of the sender node, intermediate nodes, and receiver node respectively.

1) Determining Type of Packet Loss: As TCP performs congestion control uniformly for each of the losses that take place in the network without having any knowledge about the type of loss; this deteriorates the network throughput severely. To solve this problem, we use the priority field of the IP header. Each of the intermediate node sets the value of the priority field, *prio* of each passing packet's IP header to 1 if the percentage of queue length of that node, Q_{len} , reaches a predefined threshold value L_{th}. The value of L_{th} is set to 0.9 for our proposed solution. If the queue length percentage of that node, during the traversal of the packet at that node, is below this threshold value the priority field value is set to 0. After receiving this modified packet, the receiver copies this value along with other information to a new packet and sends this packet to the sender. Upon receiving this feedback information, the sender copies the value contained in the packet's priority field to a variable called P_{prv} . Whenever a retransmission time out is triggered by a loss event, the sender node first tries to identify the reason behind the loss before slowing down. This task is performed by checking the value of P_{prv}. As the value of packet's priority field, as well as P_{prv}, is set

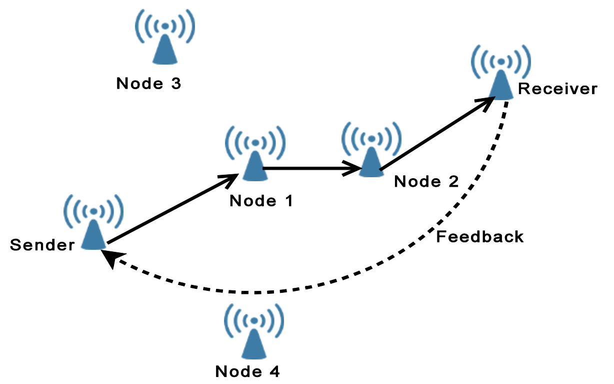


Fig. 1. A simple network scenario illustrating the idea.

to 1 if the queue of that intermediate node is above 90% full, this indicates with a high probability that the network is congested. The sender then performs a slowing down operation if P_{prv} is set to 1. Otherwise, the sender continues with its normal operation as it assumes that the loss occurred due to some other reason other than congestion. Thus, with this approach the sender can distinguish among the losses, i.e., can make difference among the congestion based losses and wireless losses. We can illustrate the steps with the network scenario of figure: Fig 1.

The path from sender to receiver is passes through the intermediate nodes Node 1, and Node 2. When a packet passes through these nodes, each node compares its queue length, Q_{len} , to the predefined threshold L_{th} and set the priority field accordingly. After arrival of a packet at the receiver side, it feedbacks this information to the sender through an acknowledgement packet. When the sender receives an acknowledgement, it checks the value of priority field, set by the intermediate nodes, and uses this information to adjust the data sending rate. The pseudo code of our proposed solution for detecting congestion losses is presented in Algorithm 1, 2, and 3.

Algorithm 1: SENDER (*Packet*)
procedure RECV(*Packet*)
P. prv← *Packet.prio*
procedure TIMEOUT(*Packet*)
if *Packet.prio* = *1*
then SLOWDOWN()

Algorithm 2: INTERMEDIATENODE(*Packet*)
procedure RECV(*Packet)*
if $Q_{len} >= 0.9 \cdot Q_{lim}$
then *Packet.prio* ← *1*

Algorithm 3: RECEIVER (Packet)
procedure RECV(*Packet*)
Ack.prio ← *Packet.prio*
SEND(*Ack*)

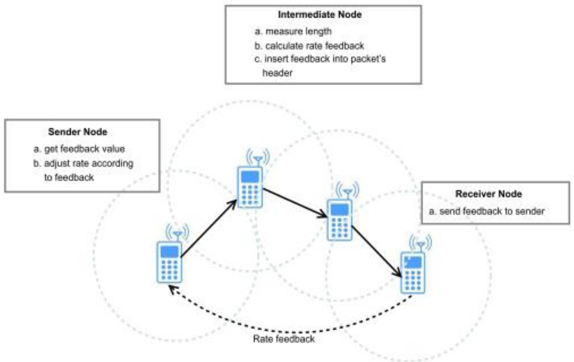


Fig. 2. Illustration of XRCC congestion control mechanism

As in XRCC, the sender uses the feedback from the intermediate nodes to determine the losses due to congestion in the network, and uses this information to control the congestion; this mechanism serves a better throughput than the traditional TCP congestion control. This mechanism also helps to keep the packet loss rate lower as compare to the traditional one.

2) *Adjusting the Sending Rate*: XRCC improves the smoothness of the sending rate, which is a requirement for streaming applications, by using explicit rate information from the intermediate nodes. In our proposal, the rate information, which is a function of the queue length of the node being traversed, is inserted into the passing packet's IP header. The receiver then propagates this explicit rate information to the sender, and the sender, based on this feedback, adjusts its sending rate. Based on the value of the rate feedback, the sender either choose to maintain the current rate, or can increase/decrease the rate. Fig. 2 illustrates the idea of XRCC mechanism.

Unlike TCP, our mechanism depends on the explicit feedback information from the intermediate nodes on a connection path. The receiver, after receiving these information from the intermediate nodes, feedback it to the sender. And the sender then takes appropriate steps. We implement our idea as modified TCP, which takes feedback from router. Implementation of a complete transport protocol, with complete reliability and fairness issue, for supporting streaming applications in MANET is considered as our future enhancement. In the following subsections, a description of the actions taken by sender nodes, receiver nodes, and the intermediate nodes, is depicted.

Intermediate Node's Behavior

The intermediate nodes on a connection path can play the vital role in determining the congestion state of the network as they are in the place where congestion actually takes place. In our proposal, congestion control for supporting multimedia streaming over MANET, the intermediate nodes on the path from the sender to the receiver calculate the rate information and propagate it as a feedback. The rate value is normalized using a factor α. The value of α is equal to 0.2. The intermediate node stamps the rate feedback R_{fb} based on the current queue length Q_{len} and the already stamped value in the options field of passing packet's IP header. The rate value is normalized using a factor α. The value of α is equal to 0.2.

To calculate the rate feedback, the following two equations have been used:

$$R_{cur} = \frac{1}{Q_{len}} \qquad (1)$$

$$R_{fb} = \alpha \times R_{prv} + (1 - \alpha) \times R_{cur} \qquad (2)$$

Here, Q_{len} is the current queue length and we take the inverse of Q_{len} as the current rate, R_{cur} , to calculate the rate feedback as in equation (1). The smoothed rate feedback, R_{fb} , is then calculated according to the equation (2). To get a smoothed feedback we use factor α and the already stamped value of feedback field (usually by the previous node). this rate information is inserted in the options field of the passing packet's IP header. Moreover, other than the single level priority, as used in algorithm 3, for detecting congestion losses, now we are using two level priorities. The queue length is checked and the *prio* field of packet's IP header is set to 2 if the queue is more than 90% full. If it is not, the queue is further checked to find whether more than 85% of the queue limit is full or not. In this case, the value of priority field is set to 1; otherwise it is set to 0. This feedback information is then passed through the IP header of the packet. Algorithm 5 presents the pseudo code of the actions taken by an intermediate node.

Algorithm 4: SENDER(*Packet*)
procedure RECV(*Packet*)
rate _feedback ← *Packet.fb*
procedure OPENCWND()
if *cwnd* < *rate_feedback*
then *cwnd* ← *cwnd* + *rate_ feedback_* · *0.2*
else ***if*** *cwnd* > *rate_feedback*
then *cwnd* ← *rate_ feedback*
else *maintain current rate*

Algorithm 5: INTERMEDIATNODE (*Packet*)

procedure RECV(*Packet*)
Q_{len}← *get_ current_ queue_length*
R_{prv}← *Packet.fb*
if $Q_{len} > 0.9 \cdot Q_{lim}$
then *Packet.prio* ← *2*
else if $Q_{len} > 0.85 \cdot Q_{lim}$
then *Packet.prio* ← *1*
$R_{cur} = 1/Q_{len}$
$R_{fb} = \alpha \cdot R_{prv} + (1 - \alpha) \cdot R_{cur}$
if $R_{fb} > R_{prv}$
then *Packet.fb* = R_{fb}
SEND(*Packet*)

***Receiver's* Behavior**

In our proposed solution, the task of receiver node is kept as simple as possible. Upon receiving a packet, the end node checks it for feedback information. Then along with other necessary fields, value of the feedback carrying field is also copied to an acknowledgement packet. The receiver then sends this acknowledgement packet to the sender. The task performed by receiver node is almost as same as algorithm 3.

Sender's Behavior

After arrival of an acknowledgement packet, the sender set its parameter *cwnd* to the value of the feedback field R_{fb}, of this packet. Before setting the value of *cwnd*, R_{fb} is compared with the current value of *cwnd*. If current value of *cwnd* is greater than the rate feedback then R_{fb} is set as the new value of *cwnd*. Otherwise, an increment factor is added to the current value of *cwnd*, based on the value of R_{fb}. Since the value of *cwnd* is adjusted based on the feedback information from the intermediate nodes, this mechanism provides a better performance than adjusting *cwnd* value with the static value of *increase_ num*. Pseudo code of sender's action is presented in Algorithm 4.

IV. SIMULATION RESULTS

This section evaluates the performance of our proposed solution, explicit rate-based congestion control for multimedia streaming in mobile ad hoc networks, through extensive ns-2 [57] simulations. We compare these results with traditional TCP congestion control mechanism.

A. Simulation Environment

We use the network simulator ns2 for our simulation purposes. To generate the random topologies for the simulations, the setdest tool in ns2 is used. We use the random way point mobility model for generating the topology of our simulation. All the simulations are performed for a 1000m x 1000m grid consisting of 100 nodes, distributed randomly over the two-dimensional grid. The source-destination pairs are randomly chosen from the set of 100 nodes in the network. We consider speeds of 1 m/s, 10 m/s and 20 m/s in our simulations. We also study the effect of load on the network by investigating scenarios with 1, 5, 15 and 25 connections respectively.

The Ad hoc On Demand Distance Vector (AODV) [41] routing protocol is used for all of our simulations and FTP is the application that we use over TCP for all the flows in the network. The packets generated are of size 512 bytes in all the simulations. The performance of XRCC is evaluated and compared against default TCP for network scenarios outlined above. We also compare the results of our solution with CBR application over UDP protocol. To measure the performance of our new congestion control mechanism, we employ matrices like instantaneous throughput, aggregate throughput, and number of dropped packets. By instantaneous throughput we refer to the size of the packet received by a node at each time interval, both for default TCP and for XRCC. The aggregate throughput is measured in kbps and reflects the number of packets successfully received at the destination. All the simulations are run for 100 seconds and each data point on the graph is averaged over 5 simulation runs. Fig. 3 depicts a snapshot of our simulation topology for 5 flows.

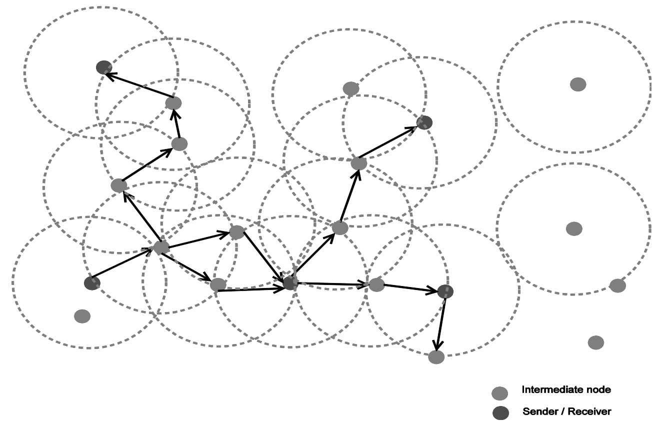


Fig. 3. A snapshot of simulation topology for 5 connections

B. Simulation Results

This section describes the simulation results based on four metrics, instantaneous throughput, aggregate throughput, fairness index, and nature of dropped packets.

1) *Instantaneous Throughput:* The instantaneous throughput results for standard TCP, UDP, and XRCC for 25 connection scenario for a speed of 20 m/s in Fig. 4 and Fig.5. Fig. 5 also includes a comparative result of these three mechanisms. TCP unnecessarily halves its congestion window and performs a slow-start whenever it experiences a time out. The slow start is triggered even on a wireless loss since TCP does not distinguish between congestion losses and mobility losses. This conservative behavior severely affects TCP's performance.

The following key observation can be made from the simulation result of our explicit rate control mechanism XRCC:

• XRCC uses rate feedback, stamped in the received packet's IP header to calculate the sending rate of the next packet, and thus it does not decrease its rate upon wireless losses.

• XRCC tries to maintain a steady sending rate. It is clear from Fig. 5 (b) that the sharp transition of sending rate is much less as compared to TCP

• It can also be observed that XRCC is able to achieve more instantaneous throughput as compared to TCP and UDP.

2) *Aggregate Throughput*: The aggregate throughput achieved by our explicit congestion control mechanism XRCC for single connection scenario is shown in Fig. 6(a). As we can observe from Figure 6(a), XRCC gains a better throughput than the traditional TCP congestion control mechanism. Since XRCC uses rate feedback from intermediate nodes, it preforms

better in mobile ad hoc networks.

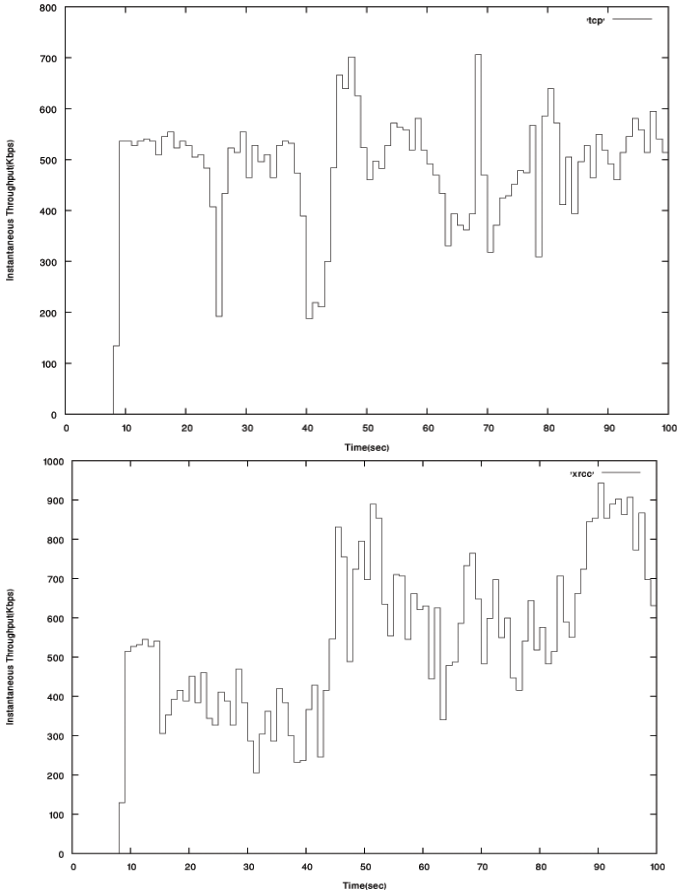


Fig. 4. Instantaneous Throughput of (a) TCP, (b) XRCC for 25 connections

better in mobile ad hoc networks.

Fig. 6 and Fig.7 also show the aggregate throughput for

TABLE I
FAIRNESS INDEX COMPARISON OF XRCC AND TCP

Speed	XRCC	TCP
1 m/s	0.34	0.311
10 m/s	0.43	0.413
20 m/s	0.57	0.54

multiple connections, 5, 15, and 25 flows, respectively. Although XRCC performs better than TCP congestion control mechanism, its performance is affected by increasing network load. With increasing mobility speed of nodes, the performance of aggregate throughput is decreasing. As the load on the network increases, despite of performance degradation of some flows, other flows in the network that can potentially utilize the underlying bandwidth. The performance

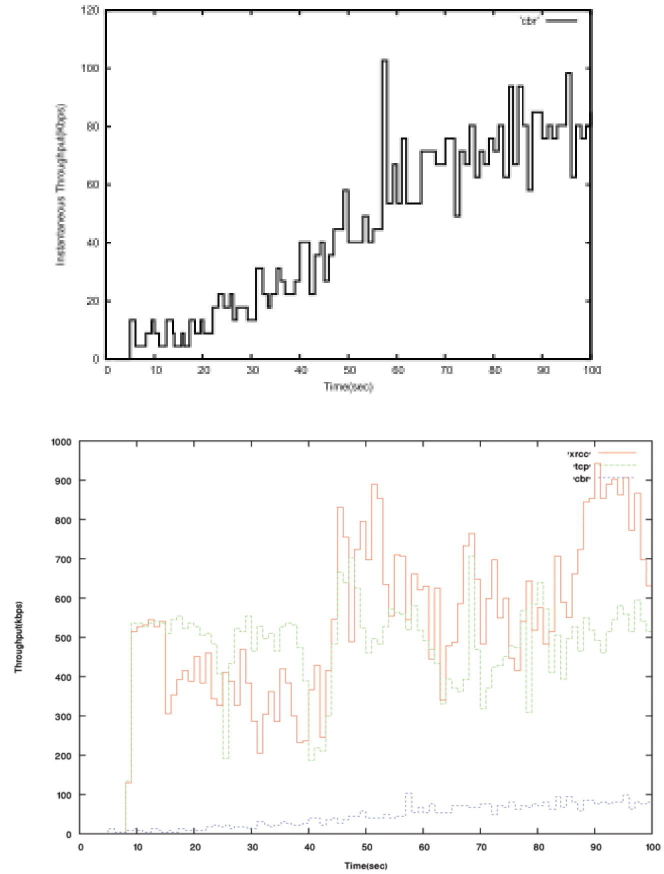


Fig. 5 Instantaneous Throughput of (a) UDP and (b) Comparison of XRCC, TCP and UDP for 25 connections

of XRCC can further be improved by designing a complete transport protocol for mobile ad hoc networks, which can support streaming applications. This is one of our directions to future improvements.

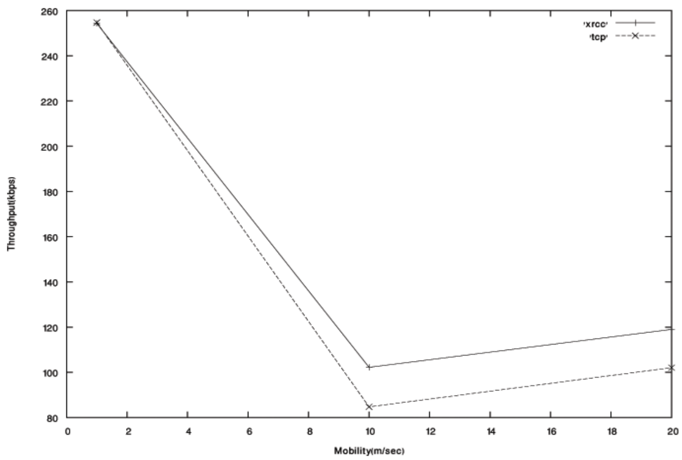
3) *Fairness Index*: In order to address the degree of fairness provided by XRCC in comparison to standard TCP congestion control mechanism, we have used Jain's fairness index. Given a set of flow throughputs (x_1 , x_2 , ..., x_n), the following function assigns a fairness index to the flows [32]:

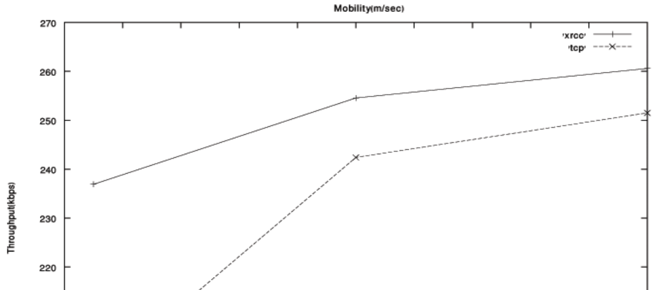
$$f(x_1, x_2, \ldots, x_n) = \text{[illegible]} \quad (3)$$

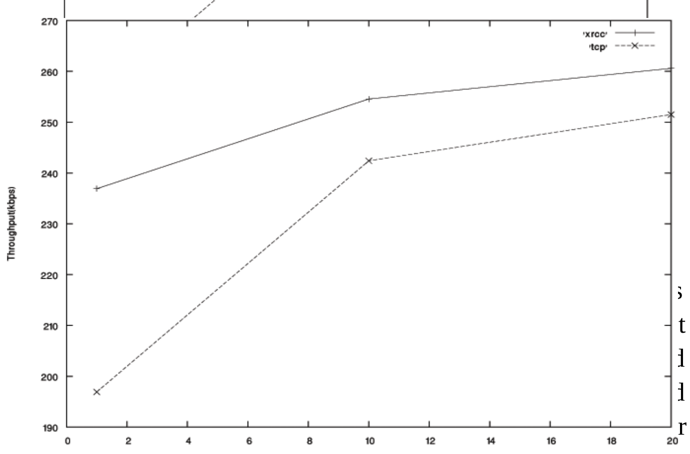
Table I represents the comparison of fairness index between TCP and XRCC congestion control mechanism for 5 connections at different speeds. As we can see, XRCC exhibits a better fairness as compared to TCP. The reason behind that is when an intermediate node is servicing several flows it sends back feedback about network load to all the sources of the flows being currently served.

4) *No. of Dropped Packets*: In this subsection, nature of packet loss causes by network congestion, both for XRCC and TCP is shown. We can observe from these graphs --Fig. 8, 9,

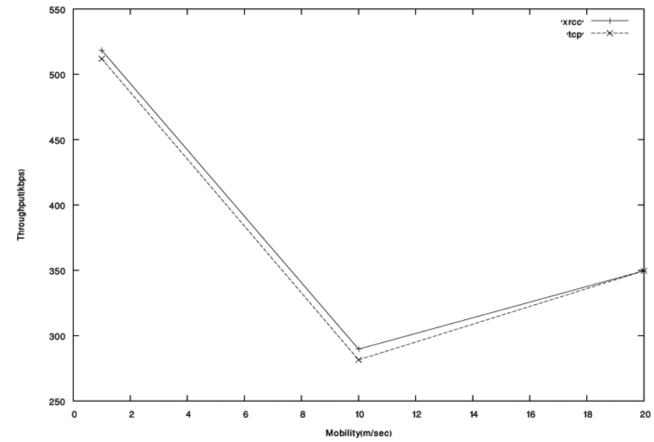
that TCP experiences more congestion losses as compared to XRCC with some exceptions. These exceptions are due to the fact that, the nature of mobile ad hoc networks is largely dependent on the number of connections and the speed of mobile nodes. TCP's higher packet loss nature results in performance degradation. As we have already discussed, TCP uniformly applies congestion control mechanism for all losses it experiences. Thus with the increase of packet loss occurrences, network performance and throughput of MANET







XRCC and TCP, we have noted the total number of losses (including wireless losses) and the number of congestion losses (i.e. losses occurred at Interface Queue).



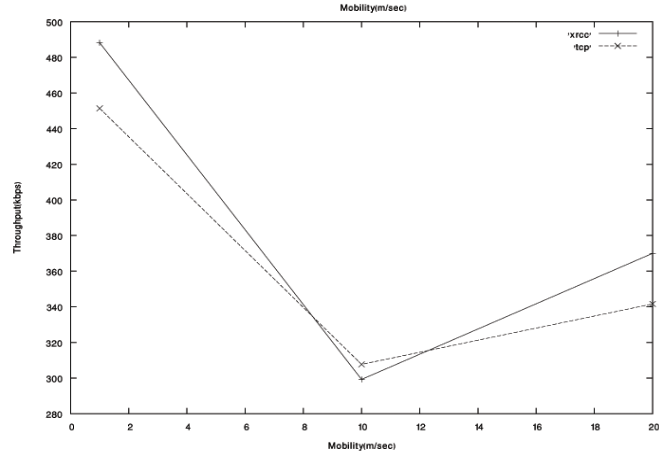


Fig. 7. Throughput vs. Mobility (a) 15 Flows, (b) 25 Flows

TABLE II
CONGESTION LOSS RATIO FOR TCP

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
1 Flow	72	523	13.77 %
5 Flows	286	1571	18.21 %
15 Flows	239	1972	12.12 %
25 Flows 215	215	1443	14.90 %

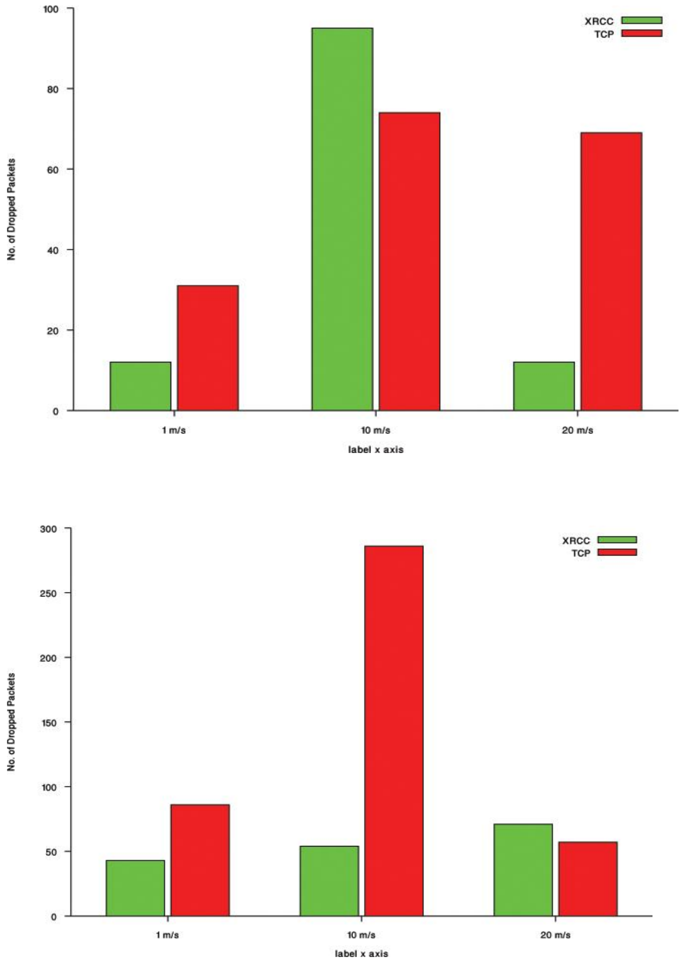


Fig. 8. Congestion loss vs. Mobility (a) 1 Flow, (b) 5 Flows

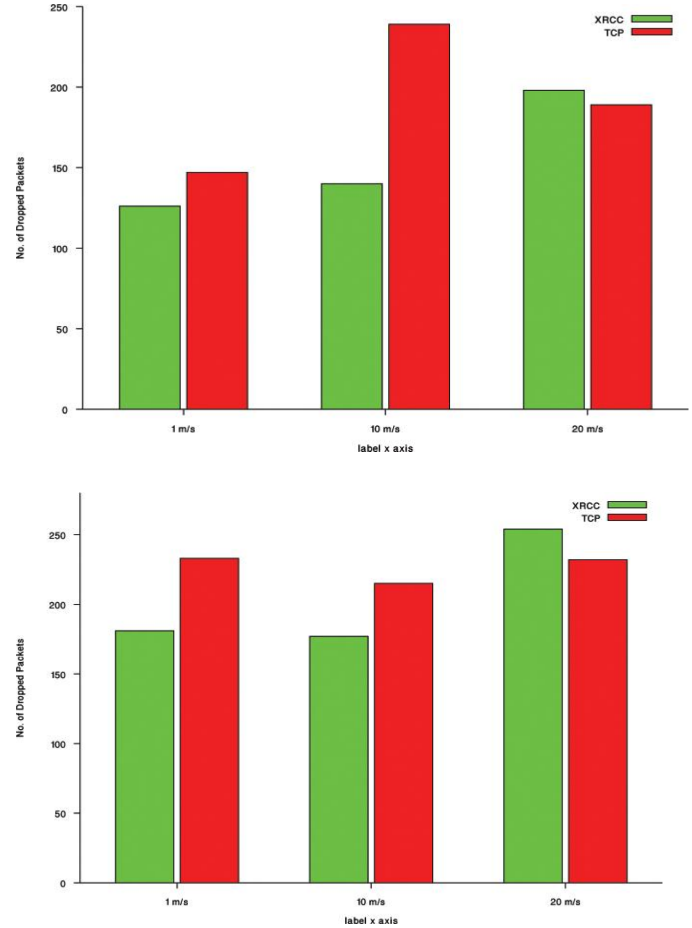


Fig. 9. Congestion loss vs. Mobility (a) 15 Flows, (b) 25 Flows

TABLE III
CONGESTION LOSS RATIO FOR XRCC

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
1 Flow	95	535	17.76 %
5 Flows	54	505	10.69 %
15 Flows	140	1665	8.41 %
25 Flows 215	177	2373	7.46 %

Table II and III reflects the result of our observations for 1, 5, 15, and 25 connections, with TCP and XRCC respectively. The maximum mobility speed of nodes is considered to be 10 m/s.

Tables 2, 3 reflect the fact that, XRCC experiences a lower packet loss rate at the router queue as compared to TCP. Usage of rate feedback from the intermediate nodes on a network path, helps XRCC to avoid a number of drops caused by congestion. We can further improve this feature of XRCC considering the starting phase of a connection, and this is one of our future directions.

V. CONCLUSION

The fundamental problem of congestion control mechanism, designed for multimedia applications in mobile ad hoc networks is caused by MANET's dynamic and random behavior. These network behaviors, like channel error, congestion, route failure, need to be detected and reacted with a reliable mechanism. Our solution tries to solve these issues in this paper.

Simulation results show that an XRCC mechanism outperform TCP congestion control mechanism and thus is well suited for applications like multimedia streaming in MANET. But still we have some limitations which are as follows:

Although XRCC minimizes packet drops caused by network congestion as compared to TCP congestion control mechanism, it still suffers from packet drops. This causes rate fluctuation and needs to be solved.

• XRCC does not take any wireless losses into consideration. And this also affects XRCC's throughput.

The limitations of our proposal lead to some directions for future improvement.

• The rate feedback can be made more accurate by considering the available network bandwidth.

• By identifying and performing appropriate actions for router failure and channel error induced packet losses, performance of XRCC congestion control mechanism can further be improved.

The behavior of TCP over ad-hoc networks is studied extensively in this paper. It can be inferred from the results that a majority of the components of TCP are not suitable for the unique characteristics of ad hoc networks and this motivate a new congestion control mechanism called XRCC, which is better suited for ad hoc networks, especially for applications like multimedia streaming. The XRCC congestion control mechanism addresses the problems that TCP faces when deployed over ad-hoc networks, and thus shows considerable performance improvement over TCP.

REFERENCES

[1] J. Postel. RFC 793: Transmission control protocol, 1981.

[2] J. Postel. RFC 793: Transmission control protocol, 1981.Real-Time-Multimedia. URL:http://encyclopedia.jrank.org/articles/Real-Time-Multimedia.html, 4 October, 2009.

[3] J. Postel. RFC 766: User Datagram Protocol. 1980.

[4] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. RTP: A transport protocol for real-time applications. 1996.

[5] D.P. Reed J.H. Saltzer and D.D. Clack. End-to-end Arguments in System Design. ACM Transactions in Computer Systems 2, pages 277–288, 4 November, 1984.

[6] Constant Bit Rate. URL: ttp://mp3.about.com/od/glossary/g/CBR-def.htm, 4 October, 2009.

[7] Variable Bit Rate. URL: http://mp3.about.com/od/glossary/g/VBR-def.htm, 4 October, 2009.

[8] Average Bit Rate. URL: http://mp3.about.com/od/glossary/g/ABR-def.htm, 4 October, 2009.

[9] M.J.K. Van Jacobson. Congestion avoidance and control. In Proceedings of the Sigcomm'88.

[10] D. Bansal and H. Balakrishnan. Binomial congestion control algorithms.In IEEE INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, volume 2, 2001.

[11] Y.R. Yang and S.S. Lam. General AIMD congestion control. In Proc. IEEE ICNP, pages 187–198, 2000.

[12] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In Proceedings of the 2001 SIGCOMM conference, volume 31, pages 263–274. ACM New York, NY, USA, 2001.

[13] Y.R. Yang, M.S. Kim, and S.S. Lam. Transient behaviors of TCP-friendly congestion control protocols. Computer Networks, 41(2):193–210, 2003.

[14] E. Kohler, M. Handley, S. Floyd, and J. Padhye. Datagram congestion control protocol (DCCP). Work in progress.

[15] N. Feamster, D. Bansal, and H. Balakrishnan. On the interactions between layered quality adaptation and congestion control for streaming video. In 11th International Packet Video Workshop, 2001.

[16] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In Proceedings of the 2002 SIGCOMM conference, volume 32, pages 89–102. ACM New York, NY, USA, 2002.

[17] Q. Zhang, W. Zhu, and Y.Q. Zhang. A Cross-layer QoS Supporting Framework for Multimedia Delivery over Wireless Internet. In International Packet video Workshop 2002.

[18] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. ACM SIGCOMM Computer Communication Review, 30(4):43–56, 2000.

[19] K. Chen and K. Nahrstedt. Limitations of equation-based congestion control in mobile ad hoc networks. In Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on, pages 756–761, 2004.

[20] S. Jin, L. Guo, I. Matta, and A. Bestavros. TCP-friendly SIMD congestion control and its convergence behavior. In Proceedings of ICNP, volume 2001, 2001.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. ACM SIGCOMM Computer Communication Review, 28(4):303–314, 1998.

[22] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP, 2001.

[23] S. Sathaye. ATM forum traffic management specification version 4.1. In ATM Forum, pages 95–0013, 1995.

[24] F. Bonomi and KW Fendick. The rate-based flow control framework for the available bit rate ATM service. IEEE Network, 9(2):25–39, 1995.

[25] A. Charny, KK Ramakrishnan, and A. Lauck. Time scale analysis scalability issues for explicit rate allocation in ATM networks. IEEE/ACM Transactions on Networking (TON), 4(4):569–581, 1996.

[26] Y. Su and T. Gross. WXCP: Explicit Congestion Control for Wireless Multi-hop Networks. In Quality of service: IWQoS 2005: 13th international workshop, IWQoS 2005, Passau, Germany, June 21-23, 2005: proceedings, volume 13, page 313. Springer Verlag, 2005.

[27] C. Lochert, B. Scheuermann, and M. Mauve. A survey on congestion control for mobile ad hoc networks. Wireless Communications and Mobile Computing, 7(5):655–676, 2007.

[28] D. Clark. The design philosophy of the DARPA Internet protocols. ACM SIGCOMM Computer Communication Review, 18(4):106–114, 1988.

[29] M. Allman, V. Paxson, W. Stevens, et al. TCP congestion control, 1999.

[30] Van Jacobson. Modified TCP Congestion Control Avoidance Algorithm. End-2-end-interest mailing list, April 30, 1990.

[31] TCP (Transport Control Protocol) congestion control mechanism. URL: http://monet.postech.ac.kr/new2008/research.html.

[32] L.L. Peterson and B.S. Davie. Computer networks: a systems approach. Morgan Kaufmann, 2007.

[33] W. Stevens. RFC2001: TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC Editor United States, 1997.

[34] IEEE. IEEE 802.11 WG, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Standard. Aug. 1999.

[35] S. Xu and T. Saadawi. Does the IEEE 802. 11 MAC protocol work well in multihop wireless ad hoc networks? IEEE communications Magazine, 39(6):130–137, 2001.

[36] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang. TCP over wireless multi-hop protocols: Simulation and experiments. In 1999 IEEE International Conference on Communications, 1999. ICC'99, volume 2, 1999.

[37] J. Li, C. Blake, D.S.J. De Couto, H.I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In Proceedings of the 7th annual international conference on Mobile computing and networking, pages 61–69. ACM New York, NY, USA, 2001.

[38] Ahuja, S. Agarwal, JP Singh, and R. Shorey. Performance of TCP over different routing protocols in mobile ad-hoc networks. In 2000 IEEE 51st Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo, volume 3, 2000.

[39] T.D. Dyer and R.V. Boppana. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pages 56–66. ACM New York, NY, USA, 2001.

[40] D. Kim, C.K. Toh, and Y. Choi. TCP-BuS: Improving TCP performance in wireless ad hoc networks. Journal of Communications and Networks, 3(2):175–186, 2001.

[41] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 2:90–100, 1999

[42] B.A. Forouzan and S.C. Fegan. Data communications and networking. McGraw-Hill Science Engineering, 2003.[43]

[43] M. Weiser. The computer of Twenty-First century. Scientific American, 1991.

[44] Z. Fu, X. Meng, and S. Lu. How bad TCP can perform in mobile ad hoc networks. In Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on, pages 298–303, 2002.

[45] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In IEEE Infocom, volume 3, pages 1405–1413. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1997.

[46] S. Floyd, M. Handley, and J. Padhye. A comparison of equation-based and AIMD congestion control. Manuscript available at http://www.aciri. Org/floyd/papers. Html, 2000.

[47] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. Wireless Networks, 8(2):275–288, 2002.

[48] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE, pages 153–179, 1996.

[49] J. Postel. RFC 792: Internet Control Message Protocol.

[50] J.P. Monks, P. Sinha, and V. Bharghavan. Limitations of TCP-ELFN for ad hoc networks. MOMUC'00.

[51] J. Liu and S. Singh. ATCP: TCP for mobile ad hoc networks. IEEE Journal on selected areas in communications, 19(7):1300–1315, 2001.

[52] Z. Fu, B. Greenstein, X. Meng, and S. Lu. Design and implementation of a TCP-friendly transport protocol for ad hoc wireless networks. In 10th IEEE International Conference on Network Protocols, 2002. Proceedings, pages 216–225, 2002.

[53] Z. Fu, X. Meng, and S. Lu. A transport protocol for supporting multimedia streaming in mobile ad hoc networks. IEEE Journal on Selected Areas in Communications, 21(10):1615–1626, 2003.

[54] S. Floyd, M. Handley, J. Padhye and J. Widmer. Equation-based congestion control for unicast applications. ACM SIGCOMM Computer Communication Review, 30(4):43–56, 2000.

[55] W. Sun, T. Wen, and Q. Guo. A Performance Comparison of Equation-Based and GAIMD Congestion Control in Mobile Ad Hoc Networks. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering-Volume 03, pages 147–152. IEEE Computer Society, 2008.

[56] B. Ramesh and D. Manjula. An Adaptive Congestion Control Mechanism for Streaming Multimedia in Mobile Ad-hoc Networks. IJCSNS, 7(6):290, 2007.

[57] Network. Simulator. 2 (NS2). URL: http://www.isi.edu/nsnam, 2005.

Nazia Zaman completed her B.Sc and M.Sc. in Computer Science and Engineering in 2007 and 2009 respectively from University of Dhaka, Bangladesh. Currently she is working as a Senior Software Engineer (Team Leader) in Divine IT Limited, Bangladesh. Her research interest includes wireless sensor networks, mobile ad-hoc networks, cloud computing, network security, cyber-physical systems, etc.

Kazi Chandrima Rahman Completed her B.Sc. (Hons) in Computer Science in 2000 and M.Sc. in Computer Science in 2002 from The University of Dhaka. She worked as Software Developer in Grameen Solutions from May 2002 to January 2003. She was a Lecturer at BRAC University in Department of CSE, from September 2003 to September 2005. She was also a Lecturer at The University of Asia Pacific from October 2005 to October 2009. She has been serving as an Assistant Professor at the Department of Computer Science and Engineering (CSE), The University of Asia Pacific from October 2009. She has research paper on Research and Educational Network for Bangladesh and Sensor network. Her research interests are on sensor network, network security, distributed, ad hoc and wireless network etc. She is a member of Executive Committee of Dhaka University Alumni Association.

Syed Faisal Hasan is currently working as an Assistant Professor at the department of computer science and engineering, Dhaka University, Dhaka, Bangladesh. His research interest is on TCP congestion control, rate control for streaming media, distributed social networking protocols and grid computing. Currently he is representing Bangladesh at the Asia Pacific Advanced Network (APAN). He is also a member of Asia New Zealand foundation's young leaders' network. He is collaborating with Altruists International towards building a trust based distributed social network.

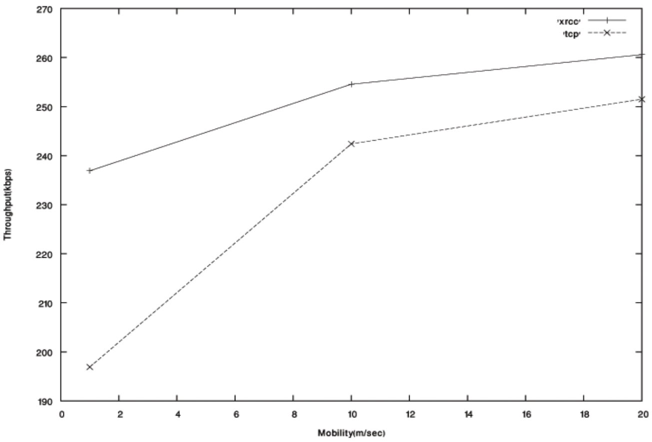


Fig .6. (b)

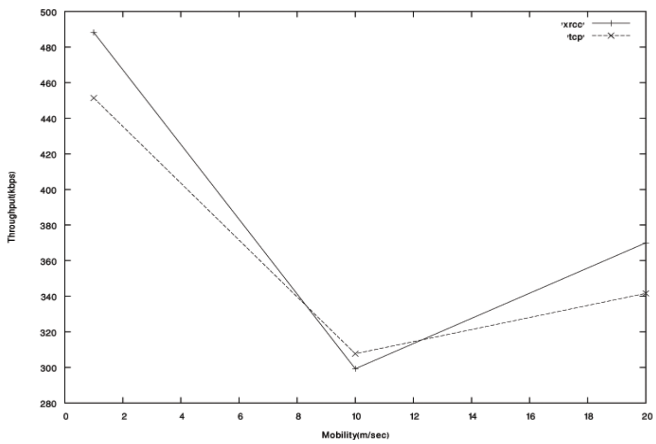


Fig. 7. (b)

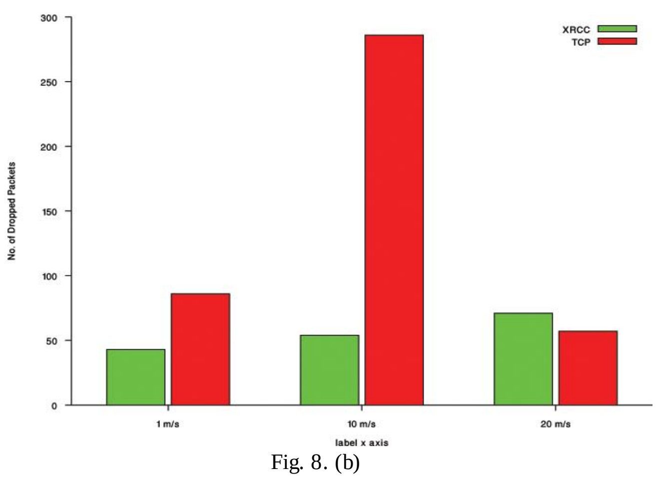


Fig. 8. (b)

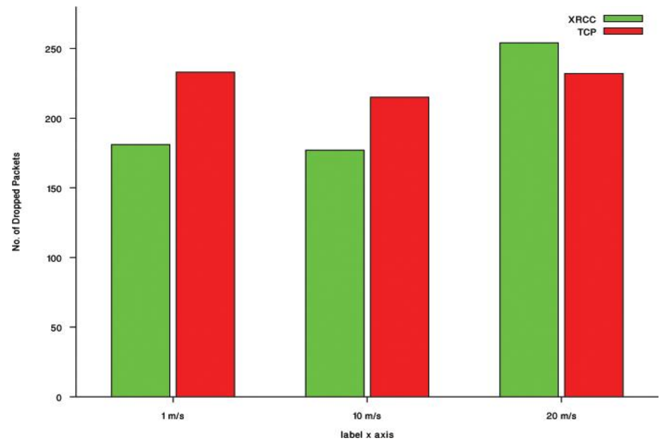


Fig. 9. (b)

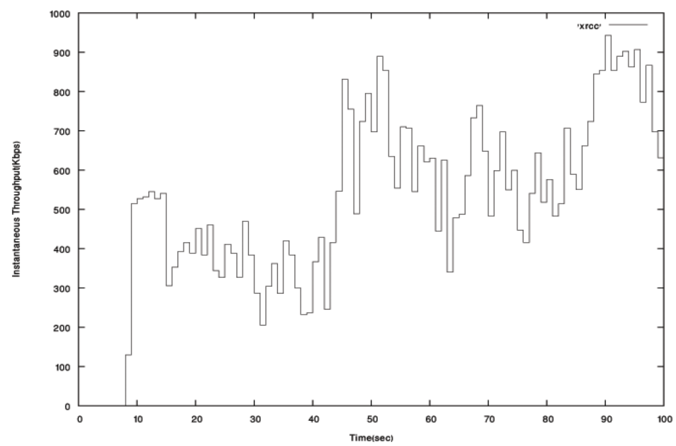


Fig. 4. (b)

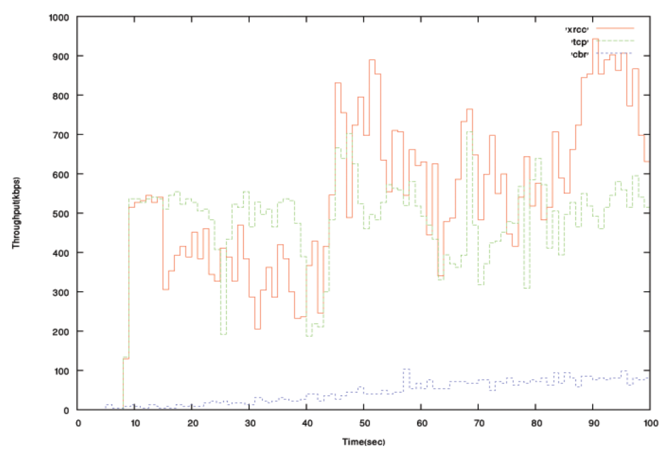


Fig. 5. (b)