

Explicit Rate Control for MANET

Nazia Zaman

*Department of Computer Science and Engineering, University of Dhaka
Dhaka-1000, Bangladesh
E-mail: nazia.zaman11@gmail.com*

Morshed Chowdhury

*School of IT, Deakin University
Melbourne, Australia
E-mail: muc@deakin.edu.au*

Received 12 April 2012

Accepted 22 October 2012

Streaming applications over Mobile Ad-hoc Networks (MANET) require a smooth transmission rate. The Internet is unable to provide this service during traffic congestion in the network. Designing congestion control for these applications is challenging, because 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 to the network topology and the shared nature of the wireless channel pose major challenges. In this paper, we propose a novel approach, which allows a quick increase of throughput by using explicit feedback from routers.

Keywords: ad-hoc networks, multimedia streaming, congestion, wireless.

1. Introduction

The massive growth of wireless networks is driving a revolutionary change in information society. Due to the availability of wireless interfaces on mobile devices such as laptops, PDAs, iPad and smart i-phone etc., wireless networks are becoming very popular. The wireless channel now supports a higher data rate which has made real time multimedia applications like radio broadcasting, video conferences, and real-time environment monitoring, etc. possible. Usage of these applications, through Mobile Ad hoc Networks, is increasing in popularity.

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 nodes rely on each other to keep the network connected. The random behavior of ad hoc networks causes the topology of a wireless network to be changed rapidly and unpredictably and puts an extra load on the TCP's (Transport Control Protocol) [1] congestion control

mechanism making it unable to cope with the network dynamics of ad hoc networks. The problem becomes worse for multimedia applications in MANET as they usually have a higher bandwidth requirement compared to the usual Internet applications like file transferring [2].

Moreover, upon any congestion event, the TCP generally reduces the transmission rate to nearly half of its original transmission rate. This change in the transmission rate could worsen the performance of these streaming applications. In addition, the TCP does not allow for a rapid 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 at a faster rate. These applications are often transported using User Datagram Protocol (UDP) [3], [4]. But UDP has the problem that it does not incorporate a congestion control mechanism. If UDP is used for multimedia applications, these unresponsive flows will compete unfairly with other TCP flows.

In this paper, we have proposed a novel Explicit Rate Control Mechanism (ERCM), for supporting applications such as multimedia streaming over MANET. The following subsections give a brief idea about the problem, and our proposed solution to address the problem.

The remainder of the paper is structured as follows: Section 2 introduces the related works and background in the area of mobile ad hoc networks, congestion control and multimedia streaming. Section 3 illustrates the proposed solution to improve the real-time streaming performance over mobile ad hoc networks. Section 4 describes the simulation results of the proposed mechanism. Section 5 concludes this paper with possible future research directions.

2. Related Works

Multimedia traffic in the current Internet can be transported over either TCP or UDP [4]. A number of studies [9], [10], [11], [12], [13] have shown that streaming audio and video is better served by a congestion mechanism which reacts slowly on packet losses, achieving smooth throughput changes. 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. Since MANET is a special type of network, a congestion control mechanism for this field needs to be adapted to the specific properties of MANETs.

We can broadly categorize 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.

2.1. Implicit congestion control

Implicit congestion control is based on end-to-end measurement, that is, the end-systems measure the network congestion state.

TCP's AIMD (Additive Increase Multiplicative Decrease) controls flow implicitly. It presumes packet loss as an indication of network overload and hence shrinks its transmission window size. However, in MANET, packet loss can occur due to its special properties, such as re-routing, and route failure. Again, AIMD's additive increase policy restricts its ability to

acquire spare bandwidth to one packet per round trip time. In the case of frequent re-routing the algorithm may never be able to catch up with the network dynamics. Also, 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 delays and a number of packets may be dropped due to the bandwidth of the wireless link fluctuations (wireless medium contention, inference, mobility) [15].

In recent years, some variants of the AIMD [5], [6], and [14] 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 exhibit almost the same problems as the original AIMD algorithm when applied over MANET.

TCP friendly congestion control [15], [13], [5], [6], 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 the 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 [13], [7], and [8]. However, in MANET, it is difficult to obtain reliable statistics for the packet loss events at the end nodes.

2.2. Explicit congestion control

This type of congestion control relies on intermediate gateways that are routers, to measure the network congestion state. Explicit Congestion Notification (ECN) [15] 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 the router's queue size. ECN indicates whether there is congestion, but it provides no information 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 similar problems to the AIMD algorithm over MANET.

There is another scheme with implicit congestion control that is the ATM forum's rate-based congestion

control scheme for the Available Bit Rate (ABR) service [17]. ABR congestion control tries to split fairly 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. But there are some problems using ATMs ABR congestion control in MANET as it assumes symmetric circuit and does not consider route failure and rerouting which are common scenarios in MANET.

A variation of the Congestion control for high bandwidth-delay product networks (XCP) [10] for wired networks with high bandwidth-delay product is the Explicit Congestion Control for Wireless Multi-hop Networks (WXCP) [16]. WXCP uses explicit feedback from within the network and multiple congestion metrics.

3. A New Approach: Explicit Rate Control for MANET (ERCM)

To solve this performance issue of MANET, a router assisted approach, which allows quick increase of throughput, is proposed in this paper. Since routers are the central places where congestion takes place, they are in a better position to detect and respond to such conditions. The proposed approach (ERCM) improves the real-time streaming performance over MANET. With this approach, routers will provide feedback by 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. 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. ERCM contributes in the following two fields:

- i. Detecting losses due to congestion.
- ii. Adjusting the sending rate using feedback information.

3.1. ERCM over MANET

ERCM depends on the feedback from the intermediate nodes which includes both the 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.

3.1.1. Determining the type of packet loss

To determine the type of loss we use the priority field of the IP header. Each of the intermediate nodes 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 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, that is; can detect the difference between the congestion based losses and wireless losses.

The path from sender to receiver 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 the arrival of a packet at the receiver side, it feeds this information to the sender through an acknowledgement packet. When the sender receives an acknowledgement, it checks the value of the 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 I.1, I.2, and I.3.

Algorithm I.1: SENDER (Packet)

```

procedure RECV (Packet)
Pprv ← Packet.prio_
procedure TIMEOUT (Packet)
if Packet.prio_ = 1
then SLOWDOWN()
    
```

Algorithm I.2: INTERMEDIATENODE (Packet)

```

procedure RECV (Packet)
if Qlen >= 0.9 × Qlim
then Packet.prio_ ← 1
    
```

Algorithm I.3: RECEIVER (Packet)

```

procedure RECV(Packet)
Ack × prio_ ← Packet × prio_
SEND (Ack)
    
```

As in ERCM, 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 provides a better throughput than the traditional TCP congestion control. This mechanism also helps to keep the packet loss rate lower compared to the traditional one.

3.1.2. Determining the type of packet loss

ERCM 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 chooses to maintain the current rate, or can increase/decrease the rate. Figure 1 illustrates the idea of the ERCM mechanism.

Unlike TCP, our mechanism depends on the explicit feedback information from the intermediate nodes on a connection path. The receiver, after receiving this information from the intermediate nodes, feeds it back to the sender. The sender then takes appropriate steps.

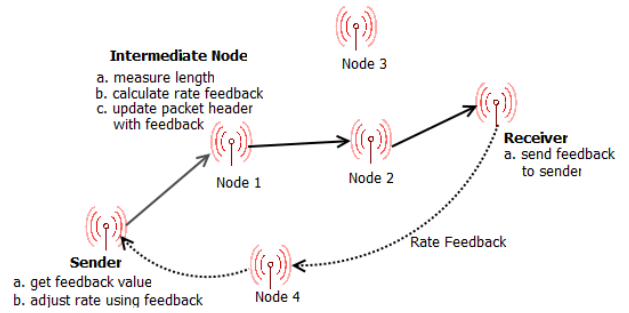


Fig. 1. Illustration of ERCM congestion control mechanism Lists of items

We implement our idea as a modified TCP, which takes feedback from the router. Implementation of a complete transport protocol, with complete reliability and fairness issues addressed, for supporting streaming applications in MANET is considered to be our future enhancement. In the following subsections, a description of the actions taken by sender nodes, receiver nodes, and the intermediate nodes, is presented.

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 feedback. 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:

$$R_{cur} = \frac{1}{Q_{len}} \tag{1}$$

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

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. 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. In addition, other than the single level priority, as used in algorithm I.3 for detecting congestion losses, we are using two level priorities. The queue length is checked and the prio_ field of the 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 the 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 I.4 presents the pseudocode of the actions taken by an intermediate node.

Algorithm I.4: SENDER (Packet)

```

procedure RECV (Packet)
rate_feedback_ ← Packet x fb
procedure OPENCWND ( )
if cwnd_ < rate_feedback_
then cwnd_ ← cwnd_ + rate_feedback_ x 0.2
else if cwnd_ > rate_feedback_ && not eln_
then cwnd_ ← rate_feedback_
else maintain_current_rate

```

Algorithm I.5: INTERMEDIATENODE (Packet)

```

procedure RECV (Packet)
Qlen ← get current queue length
Rprv ← Packet × fb
if Qlen > 0.9 × Qlim
then Packet.prio ← 2
else if Qlen > 0.85 × Qlim
then Packet.prio ← 1
Rcur = 1/Qlen
Rfb = α × Rprv + (1 - α) × Rcur
if Rfb > Rprv
SEND Packet)

```

Receiver's behavior

In our proposed solution, the task of the 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, the value of the feedback carrying field is also copied to an acknowledgement packet. The receiver then sends this acknowledgement packet to the sender. The tasks performed by the receiver node are almost as same as algorithm I.3.

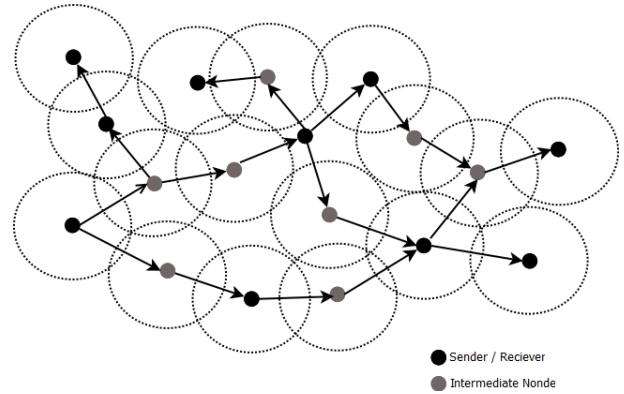


Fig. 2. Simulation topology for 10 connections

Sender's behavior

After the arrival of an acknowledgement packet the sender sets 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 the current value of $cwnd_$ is greater than the rate feedback and also the sender has observed that there is link failure notification through $eln_$ 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 feed-back information from the intermediate nodes, this mechanism provides a better performance than adjusting the $cwnd_$ value with the static value of $increase_num_$. The pseudocode of the sender's action is presented in Algorithm I.5.

4. 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 [19] simulations. We compare these results with traditional TCP congestion control mechanisms.

4.1. Simulation setting

We use the network simulator ns-2 for our simulation purposes. To generate the random topologies for the simulations, the setdest tool in ns-2 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, 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s, 30 m/s, 35 m/s, 40 m/s in our simulations. We also

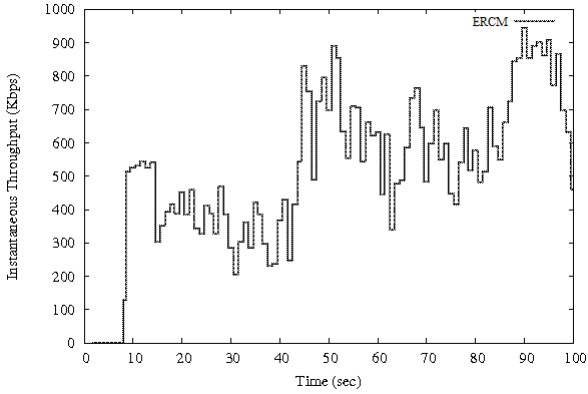


Fig. 3. Instantaneous throughput of ERCM

study the effect of load on the network by investigating scenarios with 1, 5, 10, 15, 20, 25, 30, 60, 65, 70, 75, 80, 85, 90, 95 and 100 connections respectively.

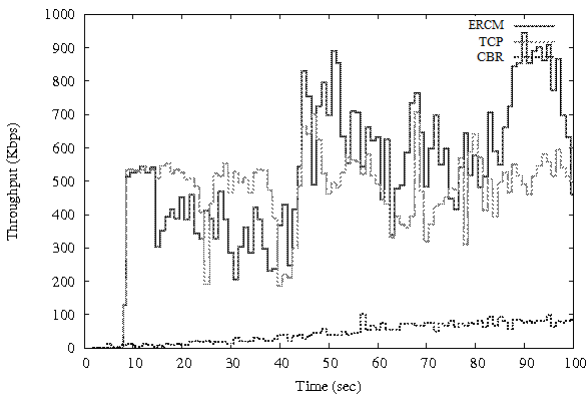


Fig. 4. Instantaneous throughput comparisons of ERCM, TCP and UDP

The Ad hoc On Demand Distance Vector (AODV) [18] 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 ERCM 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 metrics such as 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 ERCM. 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. Figure 2 is a snapshot

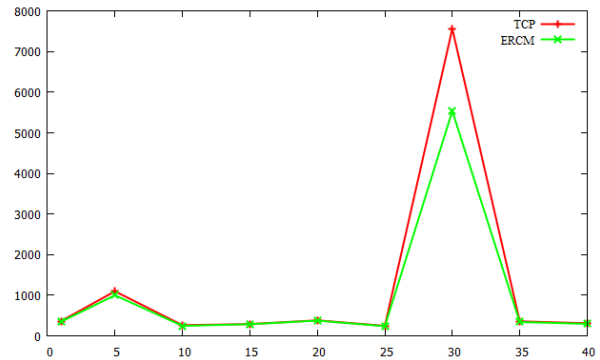


Fig. 5. Congestion loss vs. Mobility for 60 flows

of our simulation topology for 10 flows.

4.2. Simulation behavior

This section describes the simulation results based on

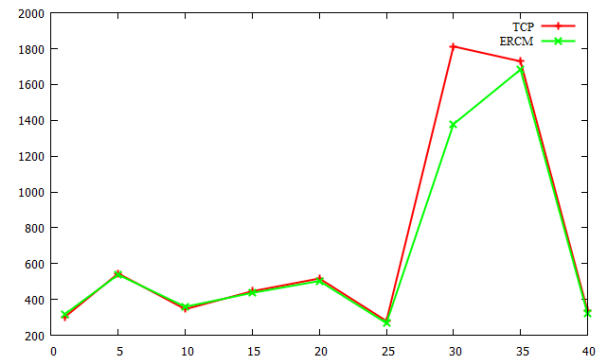


Fig. 6. Congestion loss vs. Mobility for 65 flows

four metrics- instantaneous throughput, aggregate throughput, fairness index, and nature of dropped packets.

4.2.1 Instantaneous throughput

The instantaneous throughput results for standard TCP, UDP, and ERCM for 30 connection scenario for a speed of 20 m/s are set out in Figures 3 and 4. Figure 4 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

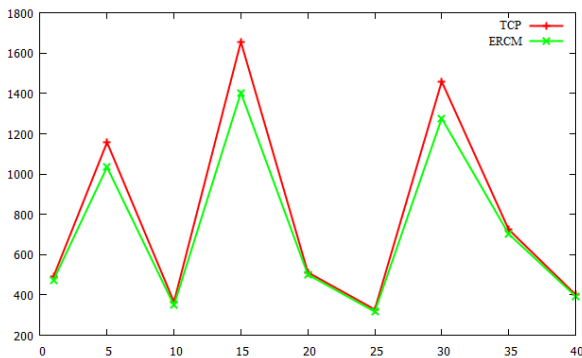


Fig. 7. Congestion loss vs. Mobility for 70 flows

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 observations can be made from the simulation result of our explicit rate control mechanism ERCM:

- ERCM 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.

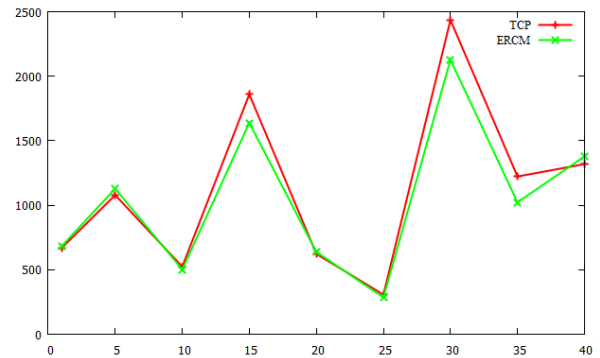


Fig. 8. Congestion loss vs. Mobility for 75 flows

- ERCM tries to maintain a steady sending rate. It is clear from Figure 4 that the sharp transition of the sending rate is much less compared to TCP
- It can also be observed that ERCM is able to achieve more instantaneous throughput compared to TCP and UDP.

4.2.2 Aggregate throughput

The aggregate throughput achieved by our explicit rate control mechanism ERCM for 60, 65, 70, 75, 80, 85, 90, 95, 100 flows have been shown in the Table 2 and Table 3 respectively. As we can observe from Table 2, 3,

Table 1. Aggregate throughput for TCP.

Speed	60	65	70	75	80	85	90	95	100
1 m/s	413.08	458.53	376.00	346.55	346.55	346.55	346.55	346.55	346.55
5 m/s	391.04	355.37	276.43	407.77	407.77	407.77	407.77	407.77	407.77
10 m/s	261.63	287.71	270.74	291.96	291.96	291.96	291.96	291.96	291.96
15 m/s	341.40	392.47	418.12	478.87	478.87	478.87	478.87	478.87	478.87
20 m/s	352.71	408.93	339.79	104.22	352.68	352.68	352.68	352.68	352.68
25 m/s	247.57	224.48	242.40	248.67	248.67	248.67	248.67	248.67	248.67
30 m/s	671.72	564.84	566.48	540.70	540.70	540.70	540.70	540.70	540.70
35 m/s	311.39	295.02	333.21	420.33	420.33	420.33	420.33	420.33	420.33
40 m/s	364.84	404.16	374.60	319.45	319.45	319.45	319.45	319.45	319.45

Table 2. Aggregate throughput for ERCM.

Speed	60	65	70	75	80	85	90	95	100
1 m/s	421.24	452.23	379.33	347.23	347.23	347.23	347.23	347.23	347.23
5 m/s	400.21	351.32	278.98	409.36	409.36	409.36	409.36	409.36	409.36
10 m/s	264.66	289.10	273.91	297.03	297.03	297.03	297.03	297.03	297.03
15 m/s	339.98	397.04	421.01	483.24	483.24	483.24	483.24	483.24	483.24
20 m/s	355.29	412.22	337.98	112.76	312.21	312.21	312.21	312.21	312.21
25 m/s	249.23	252.29	251.24	249.23	250.20	251.24	249.23	250.20	253.50
30 m/s	673.25	569.35	571.00	547.02	547.02	547.02	547.02	547.02	547.02
35 m/s	314.72	298.34	337.28	425.32	424.23	424.23	424.23	424.23	424.23
40 m/s	406.98	372.13	317.21	318.45	317.21	317.21	317.21	317.21	317.21

ERCM gains a better throughput than the traditional TCP congestion control mechanism. Since ERCM uses rate feedback from intermediate nodes, it performs better in mobile ad hoc networks.

Table 3. Fairness index comparison of ERCM and TCP.

Speed	ERCM	TCP
1 m/s	0.34	0.311
5 m/s	0.37	0.334
10 m/s	0.43	0.413
15 m/s	0.54	0.431
20 m/s	0.60	0.541
25 m/s	0.62	0.582
30 m/s	0.64	0.612
35 m/s	0.71	0.641
40 m/s	0.82	0.730

Although ERCM performs better than TCP congestion control mechanisms, its performance is affected by increasing network load. With increasing mobility speed of nodes, the performance of aggregate through-put is decreasing. As the load on the network increases, despite the performance degradation of some flows, other flows in the network can potentially utilize the underlying bandwidth. The performance of ERCM can be further improved by designing a complete transport proto-col for mobile ad hoc networks, which can support streaming applications. This is one of our directions for future improvements.

Table 4. Congestion loss ratio for TCP (Speed 30 m/s)

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
60 Flow	7559	24682	30.62 %
65 Flows	1812	4338	41.77 %
70 Flows	1459	3701	39.42%
75 Flows	2436	4469	54.50 %
80 Flows	2436	4469	54.50 %
85 Flows	2436	4469	54.50 %
90 Flows	2436	4469	54.50 %
95 Flows	2436	4469	54.50 %
100 Flows	2436	4469	54.50 %

4.2.3 Fairness index

To address the degree of fairness provided by ERCM in comparison to standard TCP congestion control mechanisms, we have used Jain’s fairness index. Given

a set of flow throughputs (x1, x2, ..., xn), the following function assigns a fairness index to the flows [16].

Table 3 represents the comparison of the fairness index between TCP and ERCM congestion control mechanisms for 10 connections at different speeds. As we can see, ERCM exhibits improved fairness compared to TCP. The reason for this is that when an intermediate node is servicing several flows it provides feedback about network load to all the sources of the flows currently being served.

4.2.4 No. of dropped packets

In this subsection, the nature of packet loss caused by network congestion, both for ERCM and TCP is shown. We can observe from these graphs [Figure 5, 6, 7 and 8] that TCP experiences more congestion losses compared to ERCM 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 a congestion control mechanism for all losses it experiences. Thus with the increase of packet loss occurrences, network performance and throughput of MANET are degraded. It can also be observed that the number of losses is increased not only with increasing speed of node mobility, but also with increasing number of flows.

Table 5. Congestion loss ratio for ERCM (Speed 30 m/s)

No.of Flows	Congestion Loss	Total Loss	% of Congestion Loss
60 Flow	5534	22342	24.77 %
65 Flows	1378	4278	32.21 %
70 Flows	1276	3678	34.70%
75 Flows	2178	4387	49.65 %
80 Flows	2178	4387	49.65 %
85 Flows	2178	4387	49.65 %
90 Flows	2178	4387	49.65 %
95 Flows	2178	4387	49.65 %
100 Flows	2178	4387	49.65 %

We have also observed the percentage of congestion loss experienced by mobile ad hoc networks, for all of our simulation scenarios. Both for ERCM and TCP, we have noted the total number of losses (including

wireless losses) and the number of congestion losses (that is losses occurred at Interface Queue). Table 4 and V reflects the result of our observations for 60, 65, 70, 75, 80, 85, 90, 95, and 100 connections, with TCP and ERCM respectively. The maximum mobility speed of nodes is considered to be 10 m/s.

Tables 4, 5 reflect the fact that, ERCM 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 ERCM to avoid a number of drops caused by congestion. We can further improve this feature of ERCM by considering the starting phase of a connection; and this is one of our future directions.

5. Conclusion

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

Simulation results show that the ERCM mechanism outperforms the TCP congestion control mechanism and thus is well suited for applications like multimedia streaming in MANET. But still we have some limitations which lead to some directions for future improvement. Like, the rate feedback can be made more accurate by considering the available network bandwidth.

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 motivates a new congestion control mechanism called ERCM, which is better suited for ad-hoc networks, especially for applications like multimedia streaming.

6. References

1. J. Postel. *RFC 793: Transmission control protocol*, 1981.
2. 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. Y.R. Yang and S.S. Lam. General AIMD congestion control. *In Proc. IEEE ICNP*, pages 187–198, 2000.
6. 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.
7. 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.
8. E. Kohler, M. Handley, S. Floyd, and J. Padhye. *Datagram congestion control protocol (DCCP)*. Work in progress.
9. 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.
10. 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.
11. 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*.
12. 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.
13. 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.
14. 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.
15. K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP, 2001.
16. 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.
17. L.L. Peterson and B.S. Davie. *Computer networks: a systems approach*. Morgan Kaufmann, 2007.
18. C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *In proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 2:90–100, 1999.
19. Network. Simulator. 2 (NS2). URL: <http://www.isi.edu/nsnam>, 2005