

# A Link Adaptive Transport Protocol for Multimedia Streaming Applications in Multi Hop Wireless Networks

Pirabakaran Navaratnam, Haitham Cruickshank and Rahim Tafazolli  
Centre for Communication Systems Research  
University of Surrey  
Guildford, GU2 7XH, U.K.

{p.navaratnam, h.cruickshank, r.tafazolli}@surrey.ac.uk

## ABSTRACT

Transport layer performance in multi hop wireless networks has been greatly challenged by the intrinsic characteristics of these networks. In Particular, the nature of congestion, which is mainly due to medium contention in multi hop wireless networks, challenges the performance of traditional transport protocols in such networks.

In this paper, we first study the impact of medium contention on transport layer performance and then propose a new transport protocol for supporting quality of service requirements in multi hop wireless networks. Our proposed protocol, Link Adaptive Transport Protocol provides a systemic way of controlling end-to-end rate for multimedia streaming applications, based on the degree of medium contention information received from the network. Simulation results show that the proposed protocol provides an efficient scheme to support quality of service requirements, such as end-to-end delay, jitter, packet loss rate, throughput smoothness and fairness for media streaming applications. In addition, our scheme requires few processing cycles and minimum overhead and does not maintain any per-flow state table at intermediate nodes. This makes it less complex and more cost effective.

## Keywords

multi hop wireless networks; medium contention; transport protocols; multimedia streaming; rate control; TFRC.

## 1. INTRODUCTION

Multi Hop Wireless Networks (MHWN), such as mobile ad hoc networks, wireless mesh networks and wireless sensor networks have gained a lot of attention in recent years, both in the industry as well as the research community as these networks are flexible and resilient. Unfortunately, they face some acute challenges due to their intrinsic characteristics. For instance, the traditional transport layer protocols, like transmission control protocol (TCP), perform very poorly in MHWN as they implicitly assume that any packet loss is due to congestion and invoke congestion control whenever a new packet loss is detected. However, this assumption is no longer valid in MHWN as packet losses may be due to channel bit errors, network congestion or route failures.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIMEDIA 2007, August 27-29, Nafpaktos, Greece  
Copyright © 2007 ICST 978-963-06-2670-5  
DOI 10.4108/ICST.MOBIMEDIA2007.1580

Several schemes have already been proposed to alleviate the problems due to wireless channel bit errors and route failures in MHWN. Most of the proposed schemes make use of various loss differentiation techniques to distinguish packet losses due to channel bit errors or route failures from those due to network congestion. They employ either explicit notifications [1], [2] or implicit end-to-end measurements [3] to traditional transport protocols, such as TCP, and react appropriately in MHWN. However, the nature of the network congestion in MHWN is significantly different from that of traditional wired network. In traditional wired networks packet losses due to network congestion are caused by buffer overflows at intermediate routers, and the traditional congestion control mechanisms, like TCP, are tailored to mitigate this type of congestion in the network. On the other hand, in MHWN buffer overflows at the intermediate nodes are rare, while the packet losses due to network congestion are largely caused by medium contention [4]. In [4], the authors showed that TCP performance degrades due to medium contention and TCP further produces increased medium contention at intermediate nodes. This is because its congestion window size is allowed to grow beyond its optimal value, which is typically very small in MHWN. Some solution have already been proposed in [4], [5], [6], and [7] for TCP. However, solution for real time transport protocols has not been studied well in literature.

Real time applications such as telephony and multimedia streaming have strict requirements in terms of end-to-end delay, jitter, packet loss rate and throughput smoothness. In MHWN, real time applications suffer from increased end-to-end delay caused by increased medium contention in the network. Thus, provisioning of multimedia services in MHWN is a challenging task as the transport protocols react inappropriately to the network condition. The situation is particularly worsened when the network is over loaded [8], [9].

Traditionally, real time services have used user datagram protocol (UDP) or UDP based protocols, without implementing any congestion control mechanism at the transport layer. Recently, TCP friendly rate control protocol (TFRC) was proposed as an unreliable, equation based rate control protocol to provide smooth, low delay and TCP friendly packet transfer for telephony and media streaming applications [10]. However, as TFRC was originally designed for wired networks, it faces challenges, like TCP in MHWN due to mobility, wireless channel bit errors and medium contention. Particularly, TFRC is unaware of the medium contention, overloads the network until a packet loss is detected and reported to the sender, and thus exacerbates the medium contention. Eventually it operates in a sub optimal stable state with increased end-to-end delay in MHWN, even with no mobility and no wireless channel bit errors [9].

In this paper, we propose a new link adaptive transport protocol (LATP) to provide quality of service (QoS) support for media streaming applications in carrier sense multiple access (CSMA) based MHWN. LATP is a transport layer end-to-end rate control scheme based on medium access control (MAC) layer feedback of the bottleneck node's permissible throughput information. The permissible throughput feedback information indicates the degree of medium contention on the path to the sender. This allows the rate to be controlled accordingly, so that the sender does not overload the network and supports QoS requirements.

The rest of this paper is organized as follows. We first outline the motivation for the new protocol and discuss some related work in Section 2. We study the nature of congestion in MHWN and its impact on transport layer performance in Section 3. The details of LATP are described in Section 4. In Section 5, we evaluate the performance of LATP with various simulation scenarios and present the results. Finally we conclude the paper in Section 6.

## 2. MOTIVATION AND RELATED WORK

The impact of medium contention on the transport layer performance in MHWN has been studied in [4], [5], [6], [7], and [9] and some solutions have been proposed in [4], [5], [6], [7], and [11] for improving TCP performance for reliable data transfer. However, very little work has focused the problems due to medium contention on transport protocols for real time applications in MHWN. To the best of our knowledge, only the scheme proposed in [9] addresses the problems due to medium contention on TFRC performance in MHWN and proposes a rate estimation technique for TFRC, based on a theoretical round trip time estimation model. However, the theoretical model is only applicable in perfect linear chain topology networks. Thus, this scheme is not general and does not work in other topologies.

QoS support in IEEE 802.11 WLAN has been extensively studied in [8] through theoretical analysis and simulations. The results suggest that 802.11 WLAN can perform well in supporting QoS as long as it is tuned to operate at the optimal point, lies below saturation level. They also demonstrate that the channel busyness ratio, which is easy to obtain and accurately and timely represent network utilization, can be used to control the total input traffic to support QoS requirements. Motivated by this work, the same authors have proposed a new mechanism, in [11], for improving TCP performance in mobile ad hoc networks. The mechanism uses channel busyness ratio to estimate the total rate-feedback at the intermediate nodes. However, they have not considered the possibilities of collisions in MHWN, which is very high due to hidden terminal problem in such networks; unlike in single hop wireless networks [12]. Moreover the mechanism introduces additional complexity and overhead in order to improve throughput and fairness of TCP flows. In particular, it requires some specific transport layer information, such as end-to-end rate and end-to-end round trip time from every data packet for MAC layer estimations performed at each intermediate node. We strongly believe that the requirement for 'protocol specific' upper layer information from each data packet for lower layer estimations at each intermediate node makes system design limited to that particular upper layer protocol and difficult to upkeep.

Explicit rate-feedback from intermediate nodes has also been used in [13] and [14] to estimate end-to-end rate for reliable data transfer over MHWN. However they have not addressed the problems arising from medium contention and also the QoS requirements for real time applications.

## 3. CONGESTION PROBLEM IN MHWN

In this section, we study the congestion problem in contention based multi hop wireless networks, using *ns2* simulations. We particularly diagnose the reasons for packet losses due to congestion in MHWN and the impact on QoS metrics. For this purpose, the wireless channel is assumed to be perfect with no bit errors and there is no mobility in the network considered. We set the simulation parameters as detailed in section 5 and performed constant bit rate applications using UDP from left end node to right end node, over a 7-hop linear chain network as detailed in section 5.1.

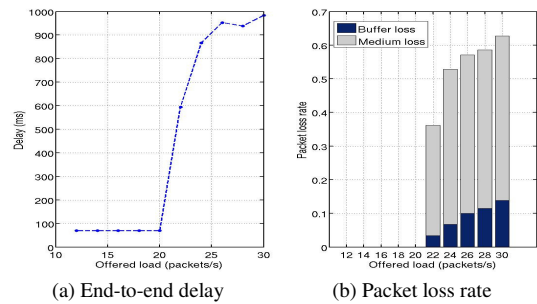


Figure 1. QoS metrics vs. offered load

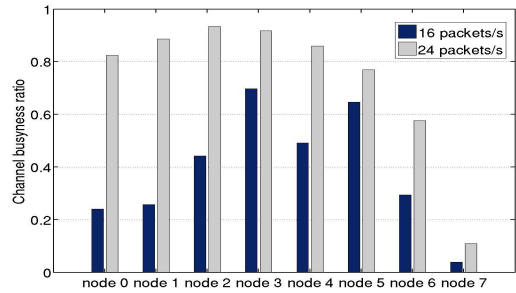


Figure 2. Channel busyness ratio at nodes along the path.

Figure 1(b) shows the packet loss rate, due to medium contention as well as buffer over flows, for various offered load. We see that packet losses are mainly caused by medium contention and increase quickly with offered load. Further, end-to-end delay increases by around 12 times, when the network state changes from no-congestion state to congestion state as shown in Figure 1(a). This dramatic change in delay is mainly due to repeated retransmission attempts and back-offs in IEEE 802.11 distributed coordination function (DCF).

Figure 2 shows average channel busyness ratio (fraction of channel busy period over total period) observed at nodes along the path over the simulation period, for 16 packets/s (no-congestion observed) and 24 packets/s (congestion observed) offered loads. The channel busyness ratio increases with offered load along the path. Thus, the channel busyness ratio can be used as a measure of current network state and to predict the future state based on measurements in MHWN.

## 4. LINK ADAPTIVE TRANSPORT PROTOCOL (LATP)

The primary goal of LATP is provide an efficient transport layer rate control mechanism for media streaming applications in contention based MHWN. An LATP sender transmits a stream of

data packets to the receiver and controls the sending rate based on the feedback information received from the receiver. Details of this mechanism are presented in the following sections.

#### 4.1 Intermediate Node

Each node in the network maintains two estimations, channel busyness ratio ( $B$ ) and throughput ( $T$ ) at the MAC layer. It has been shown in [8] that the channel busyness ratio provides precise and robust information about the network status in IEEE 802.11 networks. Further, our investigation in section 3 confirms the same result in MHWN as well. Thus, we use the channel busyness ratio at a node as a direct measure of the level of medium contention experienced by that node. Every node computes current channel busyness ratio,  $B_{sample}$  between two consecutive transmission attempts by sampling the channel state (idle/busy) every 20 $\mu$ s (one back-off timeslot period). An estimation  $B$  at that node is derived using exponentially weighted moving average (EWMA) filtering with weight,  $\alpha$  as follows.

$$B = \alpha B + (1 - \alpha) B_{sample}$$

In order to improve the stability and the flexibility of the estimation, an observation window is introduced to detect whether the sample,  $B_{sample}$  is reasonable or too noisy to be used. We use the 3-sigma rule to obtain the observation window. The observation window is defined as  $\mu \pm 3\sigma$ , where  $\mu$  is the sample average and  $\sigma$  is the sample standard deviation. If  $B_{sample}$  falls within the observation window, i.e. within  $B \pm 3\sigma_B$ , where  $\sigma_B$  is the standard deviation of the samples obtained so far,  $B$  is updated with a small value for weight  $\alpha$ , otherwise we assume the sample is too noisy and update  $B$  with a high value for weight  $\alpha$ . We empirically selected these small and high values for  $\alpha$  as 0.125 and 0.875 respectively. And also, we estimate  $\sigma_B$  as follows.

$$\sigma_B = \beta \sigma_B + (1 - \beta) |B - B_{sample}|$$

where weight,  $\beta$  is set to 0.875 for a stable window size.

Since, collisions can still happen due to hidden terminal problem in MHWN [12] we include this impact in throughput estimation. For this purpose, the current throughput,  $T_{sample}$  is measured over a time period; including the time wasted due to collisions. The measurement is taken for every successful data transmission at the MAC layer using the RTS-CTS-DATA-ACK sequence in IEEE 802.11 DCF mode. For each transmitted packet,  $T_{sample}$  can be computed as  $X/(t_a - t_d)$ , where  $X$  is the packet size,  $t_d$  is the time when the packet is ready for transmission at the MAC layer and  $t_a$  is the time when the acknowledgement is received for that transmission [15]. We believe that this calculation is feasible for most of the CSMA based wireless networks at the MAC layer as they have similar mechanisms for link level explicit acknowledgment.

Then, the throughput,  $T$  is estimated using EWMA filtering and the 3-sigma rule as we estimated  $B$ :

$$T = \alpha T + (1 - \alpha) T_{sample}$$

$$\sigma_T = \beta \sigma_T + (1 - \beta) |T - T_{sample}|$$

where  $\sigma_T$  is the standard deviation of the current throughput samples. As we described before, either 0.125 or 0.875 is used for  $\alpha$ , based on whether  $T_{sample}$  falls within  $T \pm 3\sigma_T$  or not and 0.875 is used for  $\beta$ .

Finally, for every outgoing LAMP data packet, the node calculates its permissible throughput  $P$  based on channel busyness ratio and throughput estimations, as given below, to efficiently utilize the

channel, while avoiding severe medium contention. It then updates the rate-feedback,  $R$  in the header to the value of  $P$ , if  $P$  is smaller than  $R$ .

$$P = (B_{TH} - B)T / B \quad \text{if } B > 0, \text{ otherwise } P = B_{TH}T$$

where  $B_{TH}$  is the maximum threshold for  $B$ . According to [5],  $B_{TH}$  is set to 95%. Unlike the metric used in [11], metric  $P$  uses an efficient throughput estimation mechanism in order to include the impact due to collisions (both RTS-RTS and DATA-RTS collisions [12]) in MHWN. Further more, it does not require any transport layer information at MAC layer for rate-feedback estimation, and provides end-to-end fairness at the transport layer itself, like TCP does. Also, it uses only one field in a special header and utilizes the cross layer coordination in a systematic way, i.e. LAMP requires cross layer information flow only in one direction; from lower layer to upper layer, unlike the scheme proposed in [11]. This makes the system can co-operate with other types of upper layer protocols as well.

#### 4.2 LAMP Receiver

An LAMP receiver sends feedback packets at regular reporting periods in order to assist the sender to determine the sending rate according to the network conditions. When the receiver receives a data packet,  $R$  in the header gives the minimum permissible throughput estimation of the path. The receiver copies  $R$  and estimates the average rate-feedback ( $R_{avg}$ ) as described below.

$$R_{avg} = \eta R_{avg} + (1 - \eta)R$$

$$\sigma_R = \omega \sigma_R + (1 - \omega) |R_{avg} - R|$$

When the new value  $R$  falls within the observation window,  $R_{avg} \pm 3\sigma_R$ , the receiver updates  $R_{avg}$  using the above EWMA filter with  $\eta=0.5$  in order to obtain a mean value for the reporting period. When the  $R$  value falls outside the observation window, the value  $R$  is not used to update  $R_{avg}$ ; instead it is stored in a suspect window. If the receiver receives at least three  $R$  values in its suspect window within a round trip time ( $rtt$ ), it will assume a major change has taken place in the network condition and will immediately use all these  $R$  values to update  $R_{avg}$ . The weight,  $\omega$  is set to 0.875 in order to maintain stability. Here, the  $rtt$  value is informed by the sender on the data packet header.

In addition, the receiver looks for new packet losses by observing the sequence numbers of the received data packets. The loss of a data packet is detected by the arrival of at least three data packets with a higher sequence number than the lost packet, like TFRC does. In addition, it calculates the average receiving rate ( $s$ ) within the last reporting period and includes  $s$  and  $R_{avg}$  in its feedback packets to the sender. Feedback packets are normally sent to the sender every  $rtt$ . However, a feedback packet will be immediately sent whenever a new packet loss is detected at the receiver without waiting for the end of the current  $rtt$  period.

#### 4.3 LAMP Sender

On connection start-up, the sender sends out the packets using a small initial sending rate until it receives the first feedback packet from the receiver. Once the first feedback packet is received, it follows a slow start mechanism to probe the network capacity. During the slow start, the sending rate,  $S$  will be updated every  $rtt$  as follows.

$$S = \max(2s, X / rtt)$$

where  $s$  is the receiving rate informed by the receiver and  $X$  is the data packet size. The term  $X / rtt$  ensures a minimum sending rate of one packet per  $rtt$ .

The slow start mechanism will terminate when the sender receives a negative value for  $R_{avg}$  in the feedback packet or the sender's "no feedback timer" expires. In LAMP, the "no feedback timer" timeout interval is set to  $4rtt$  in order to maintain smoothness for streaming applications, like in TFRC. After slow start, the sender follows a normal rate control operation based on the  $R_{avg}$  value received, as described below.

When the average rate-feedback,  $R_{avg}$  value from the receiver is positive, and it is greater than a threshold value  $\delta S$ , where  $\delta$  is a small constant (empirically set to 0.05) used to reduce fluctuations in the sending rate, the sender increase the sending rate as follows.

$$S = \max(\min(2s, (S + R_{avg}), (S + NX / rtt)), X / rtt)$$

where  $N$  is a time period in terms of number of  $rtts$  from the last rate change and is used to maintain a smooth rate change while providing convergence and fairness. In particular, it ensures that the sending rate will not be increased by more than one packet per  $rtt$ . If the positive  $R_{avg}$  is smaller than  $\delta S$ , the same sending rate will be maintained.

On the other hand, when the sender receives a negative  $R_{avg}$  value it reduces the sending rate in order to reduce the medium contention level in the network. It reduces the rate by  $\frac{1}{8} S$ , once per  $rtt$  period, for the negative feedbacks. This rate reduction value is taken by considering the facts that LAMP takes continuous rate control action, normally every  $rtt$  period, in particular, rate increment is small when medium contention is high and below threshold, and LAMP protocol is meant for smooth rate change applications. Therefore, it is reasonable to keep the rate reduction small for each negative rate-feedback. This rate reduction value also ensures that the sending rate will be reduced by half if negative rate-feedbacks are received subsequently in  $4rtt$  period. This is equivalent to the rate reduction taken, when "no feedback timer" expires. Note that the above rate control actions can be taken by the sender only when it receives new feedback packets from the receiver. Moreover, it maintains a "no feedback timer", which is reset to expire after  $4rtt$  periods, as mentioned earlier, by the new feedback packets. When the "no feedback timer" expires the sender assumes a severe change has taken place in the network and reduces the sending rate by half, like TFRC. Since the timeout value is four times the normal feedback reporting period, it is reasonable, even for smooth rate applications, to reduce the rate by half when the timer expires. In addition, the sender ensures a minimum sending rate of one packet per  $rtt$  at all time.

## 5. LAMP PERFORMANCE EVALUATION

In this section, we evaluate the performance of LAMP over a variety of scenarios using  $ns2$  simulations. LAMP is compared with TFRC for end-to-end delay, jitter, packet loss rate, throughput smoothness and fairness performance over chain, grid and random topologies. Some results observed with TCP NewReno are also presented for comparison.

We performed the simulations in MHWN with static nodes in order to avoid the problems induced by mobility. In addition, the wireless channel was assumed to be perfect with no bit errors. The IEEE 802.11 DCF MAC protocol, with some modifications to support LAMP, and ad hoc on demand distance vector (AODV) routing protocol [16] were used in the protocol stack of each node for wireless connectivity. The other main simulation parameters were set as in many previous works [5], [9] and are presented in Table 1. It was also assumed that the transport protocols: LAMP, TFRC and TCP have always data to send to the destination. The

results presented were taken over 10 simulation runs, unless otherwise specified.

**Table 1. Simulation Parameters**

Parameters (units)	Value
Propagation model	Two ray ground
Transmission range (m)	250
Carrier-sense range (m)	500
Channel capacity (Mbps)	2
Antenna	Omni-directional
MAC protocol	IEEE 802.11 DCF
Routing protocol	AODV
Interface queue size (packets)	25
Data packet size (bytes)	1000
Simulation time (s)	400
Simulation runs	10

In this evaluation, we define the throughput smoothness as the smallest ratio between the instantaneous throughput values obtained over a particular period. In our simulations, the instantaneous throughput was measured every 1s and the smoothness was calculated every 10s. Further more, we computed the long term fairness of the flows using Jain's fairness index [17].

### 5.1 Chain Topology

The proposed protocol was first evaluated over a chain topology where the nodes are placed in a line with a uniform distance of 200m between each pair. This ensures that only neighboring nodes can directly communicate with each other.

First, LAMP, TFRC and TCP flows were performed separately from node 0 to node n. The simulation time was set to 400s and the flow was started at 10s in each simulation run. For all the data flows, the measurements were taken at the transport layer over a steady period of 50s to 400s. This measurement period was used in order to avoid the transient measurements at the beginning. The average results obtained with an increasing number of hops for the end-to-end connection are presented in Figure 3.

Figure 3(a) shows the delay performance of LAMP, TFRC and TCP flows. We can clearly observe from the figure that both TFRC and TCP experience higher delay than LAMP flows. In particular, TFRC flows experience at least 100% more delay than LAMP flows. The reason behind this is, that TFRC over loads the network since it produces a sending rate that is above the rate supported by the IEEE 802.11 MAC layer in multi hop networks. Then, for each packet the MAC layer attempts multiple retransmissions and back-offs before transmitting or dropping the packet. This increases the end-to-end delay of the TFRC packet as TFRC waits for the sender to be notified of packet losses in order to control the sending rate. Although TFRC eventually resorts some packet losses caused by medium contention, it receives them too late due to MAC layer retransmission and back-off. Thus, TFRC will experience maximum delay if the nodes participating in the connection have sufficient interface queue (buffer) size. On the other hand, as shown in Figure 3(a), LAMP provides much better delay performance than other flows, in all number of hops connections. Since LAMP controls the sending rate based on the degree of medium contention level in the network, it operates at a rate supported by the MAC layer and does not over load the network. Therefore, it shows better delay performance than TFRC and TCP.

Jitter performance is presented in Figure 3(b). Although TFRC and LAMP both provide considerably good jitter performance for real

time applications over the chain topology, LAMP outperforms TFRC.

Packet loss rate (PLR – fraction of packets sent and not received by the destination) is also estimated and presented in Figure 3(d). LAMP exhibits much better performance in terms of PLR than TFRC and TCP. Throughput smoothness is also measured at the receiver and presented in Figure 3(e). Both, LAMP and TFRC show good throughput smoothness performance. However, LAMP's smoothness is better than TFRC's smoothness.

In Figure 3(c), we observe that TFRC obtains higher throughput in small number of hops connections than LAMP, and both TFRC and LAMP achieve almost same throughput in connections with more than 8 hops. Although LAMP loses some throughput in small number of hops connections, we believe that the performance improvement achieved with LAMP for delay, jitter, PLR and throughput smoothness will significantly improve the performance of media streaming applications in MHWN.

Further more, we evaluated LAMP in chain topology with competing flows as well. The simulation results showed that the end-to-end delay and the PLR are much reduced with LAMP flows compared with TFRC flows. In addition, LAMP outperforms TFRC and TCP in terms of jitter and smoothness while providing excellent fairness with the competing flows. Due to lack of space the results are not presented here.

## 5.2 Grid Topology

We also evaluated LAMP in an 8x8 grid topology as shown in Figure 4 under various scenarios.

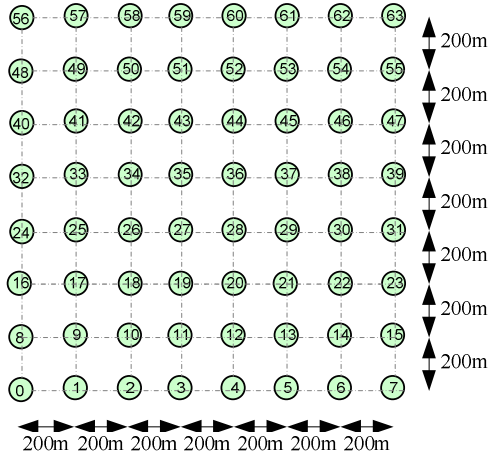
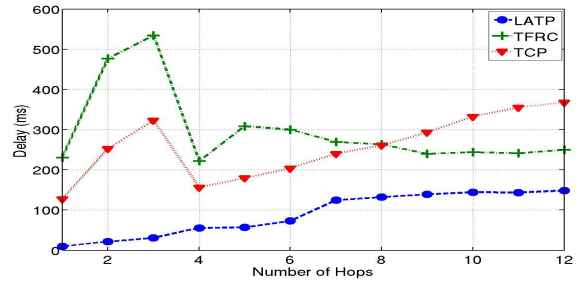
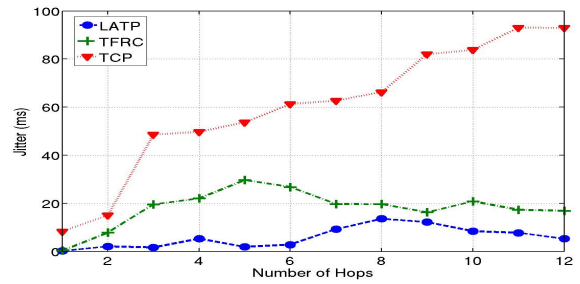


Figure 4. 8x8 Grid topology

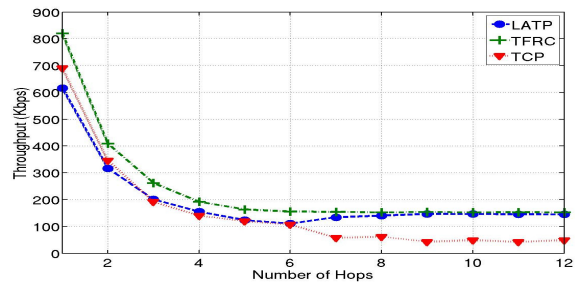
We performed several flows under several conditions and studied the performance. First, two parallel flows were initiated from nodes 16 and 32 to nodes 23 and 39 respectively. These two, source and destination pairs were selected 400m apart, such that they lie in the carrier-sense range and out of the transmission range of each other (see Figure 4). Similarly, four parallel flows were performed from nodes 0, 16, 32, and 48 to nodes 7, 23, 39, and 55 respectively. The neighboring pairs lie 400m apart in carrier-sense range. Also, eight parallel flows were performed from nodes 0, 8, 16, 24, 32, 40, 48, and 56 to nodes 7, 15, 23, 31, 39, 47, 55, and 63 respectively. Here, the neighboring pairs lie 200m apart, i.e. in the transmission range of the neighbors. Finally, 2x2 crossing flows were performed from nodes 2, 5, 16, and 40 to nodes 58, 61, 23, and 47 respectively. Here, two parallel flows (2-58 and 5-61) were set to cross the other two parallel flows (16-23 and 40-47) in the



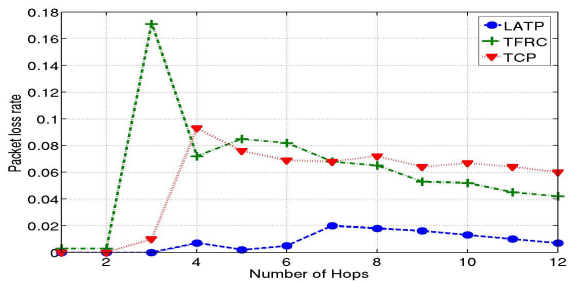
(a) End-to-end delay



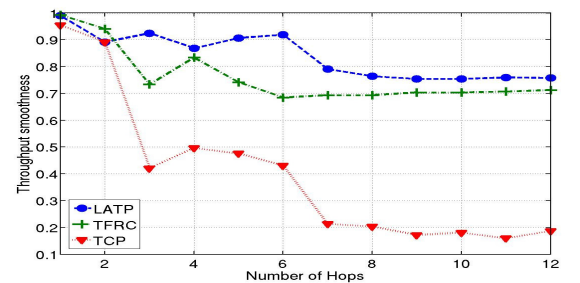
(b) Jitter



(c) Throughput



(d) Packet loss rate



(e) Throughput smoothness

Figure 3. LAMP, TFRC and TCP performance with increasing number of hops connection.

network. The flows were started in 20s intervals and the measurements were taken over a steady period.

The average results obtained per flow are summarized in Table 2. In all cases, LAMP outperforms TFRC in terms of end-to-end delay, jitter, PLR, and smoothness. In particular, the performance improvement achieved with LAMP in terms of end-to-end delay and PLR is highly significant, in all cases. We observe good fairness results with LAMP and TFRC in this topology too.

**Table 2. Performance Comparison in Grid Topology.**

Flow Type		Avg. Delay (ms)	Avg. PLR	Avg. Smoothness	Fairness Index	Avg. Jitter (ms)
2 parallel flows	LAMP	125.45	0.015	0.72	0.995	14.50
	TFRC	337.79	0.091	0.65	0.995	40.43
4 parallel flows	LAMP	130.40	0.017	0.67	0.865	25.88
	TFRC	369.83	0.109	0.58	0.954	60.89
8 parallel flows	LAMP	178.29	0.044	0.61	0.815	63.08
	TFRC	588.50	0.175	0.45	0.799	153.47
2x2 crossing flows	LAMP	135.40	0.026	0.69	0.976	34.72
	TFRC	344.36	0.172	0.49	0.954	89.54

### 5.3 Random Topology

Finally we performed simulations in a random topology. We placed 100 nodes uniformly and randomly in a 1600mx1600m area. Ten flows were sent simultaneously between ten randomly chosen source and destination pairs, with a minimum hop distance (between a source and its destination) of five hops. The average results obtained per flow are summarized in Table 3. We still observe from the results that the end-to-end delay, jitter, PLR, throughput smoothness and fairness performances of LAMP are better than that of TFRC even in such a complex simulation scenario. It should be noted that, in this evaluation we used the fairness index only as a comparative metric and not as an actual fairness measurement. This is because the index we used may not be applicable to this scenario, where some flows may not compete with some other flows at all for the network resources.

**Table 3. Performance Comparison in Random Topology.**

Flow Type	Avg. Delay (ms)	Avg. PLR	Avg. Smoothness	Fairness Index	Avg. Jitter (ms)
LAMP	232.93	0.154	0.460	0.894	129.04
TFRC	938.27	0.285	0.328	0.779	285.92

## 6. CONCLUSIONS

In this paper, we addressed the problems arising from medium contention on transport layer performance in MHWN and proposed a new transport protocol, LAMP to support media streaming applications in such networks. Our proposed protocol, LAMP exploits cross layer coordination in a systemic way and performs transport layer rate control for end-to-end flows efficiently, based on the degree of medium contention information received from the intermediate nodes. LAMP always provides an end-to-end rate supported by the MAC layer in MHWN. This prevents network overloading and thereby helps to deliver the packets to the receiver with small delay and jitter and minimum packet loss rate, which are key QoS metrics for media streaming applications. Simulation results confirmed that LAMP achieves good performance in terms of end-to-end delay, jitter, packet loss rate, throughput smoothness, and fairness for media streaming applications over various simulation scenarios. Thus, it clearly

demonstrates that by using an efficient rate control mechanism based on the degree of medium contention information from the network, we can support QoS requirements for multimedia services in MHWN. LAMP provides this with less complex estimations and mechanisms.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank the colleagues from the IST e-Sense project. The authors also thank Nadeem Akhtar and Lajos Hanzo II for their reviews and comments.

## 8. REFERENCES

- [1] G. Holland and N.H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in Proc. ACM Mobile Communications Conf., Seattle, WA, Aug. 1999, pp. 219-230.
- [2] Jian Liu and Suresh Singh, "ATCP: TCP for mobile ad hoc networks," in *IEEE Journal on Selected Areas in Communications*, vol. 19, no.7, pp. 1300-1315, Jul. 2001.
- [3] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," in *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1615-1626, Dec. 2003.
- [4] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP performance," in *IEEE Transactions on Mobile Computing*, vol. 4, no.2, pp. 209-221, Mar./Apr. 2005.
- [5] K. Nahm, A. Helmy, and C.-C Jay Kuo, "TCP over multihop 802.11 networks: Issues and performance enhancement," in *Proc. ACM MobiHoc'05*, Illinois, May 2005.
- [6] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks," in Proc. IEEE ICC 2003, Seattle, May 2003.
- [7] K. Xu, M. Gerala, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighbourhood RED," in Proc. ACM MobiCom'03, San Diego, Sep. 2003.
- [8] H. Zhai, X. Chen, and Y. Fang, "How well can the IEEE 802.11 wireless LAN support quality of service?," in *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 3084-3094, Nov. 2005.
- [9] M. Li, C. S. Lee, E. Agu, M. Claypool, and R. Kinicki, "Performance enhancement of TFRC in wireless ad hoc networks," in *Proc. DMS'2004*, San Francisco, Sep. 2004.
- [10] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," IETF RFC3448, Jan 2003.
- [11] H. Zhai, X. Chen, and Y. Fang, "Rate-based transport control for mobile ad hoc networks," in *Proc. WCNC2005*, New Orleans, Mar. 2005.
- [12] K. Wang, F. Yang, Q. Zhang, and Y. Xu, "Modeling path capacity in multi-hop IEEE 802.11 networks for QoS services," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 738-749, Feb. 2007.
- [13] K. Chen, K. Nahrstedt and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in *Proc. WCNC2004*, Atlanta, Mar. 2004.
- [14] K. Sundaresan, V. Anantharaman, H.Y. Hsieh, and R. Sivakumar, "ATP: A reliable transport protocol for ad hoc networks," *IEEE Transactions in Mobile Computing*, vol. 4, no. 6, pp. 588-603, Nov./Dec. 2005
- [15] S. H. Shah, K. Chen, and K. Nahrstedt, "Dynamic bandwidth management for single hop ad hoc wireless networks," in *Proc. IEEE International Conference on Pervasive Computing and Communications*, Texas, Mar. 2003.
- [16] C. Perkins, E. Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing," IETF RFC 3561, 2003.
- [17] R. Jain, D-M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," in *Technical Report TR-301, DEC Research Report*, Sept. 1984.