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

Pirabakaran Navaratnam • Haitham Cruickshank • Rahim Tafazolli

Published online: 11 June 2008 © Springer Science + Business Media, LLC 2008

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 improving quality of service performance in multi hop wireless networks. Our proposed protocol, Link Adaptive Transport Protocol provides a systemic way of controlling transport layer offered load 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 improve quality of service performance metrics, such as end-to-end delay, jitter, packet loss rate, throughput smoothness and fairness for media streaming applications. In addition, our scheme requires few 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 \cdot medium contention \cdot transport protocols \cdot multimedia streaming \cdot rate control \cdot 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 However, this assumption is no longer valid in MHWN as packet losses may be due to channel bit errors, network congestion or route failures.

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 networks. 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

^{P. Navaratnam (⊠) · H. Cruickshank · R. Tafazolli} Centre for Communication Systems Research, University of Surrey, Guildford GU2 7XH, UK e-mail: p.navaratnam@surrey.ac.uk

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 [4–7] 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, and packet loss rate. In MHWN, real time applications suffer from increased end-to-end delay and packet loss rate 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 transport protocols pump more traffic into the networks than can be supported [8, 9]. Thus, the offered load at the transport layer should be controlled adaptively according to the network condition in MHWN, which is the main focus of this paper.

Traditionally, real time services have used user datagram protocol (UDP) or UDP based protocols, without implementing any congestion control mechanism at the transport layer. Please note that congestion control mechanisms implemented at the application layer are liable to misuse as users can easily modify the functions and attempt to consume larger share of the bandwidth. Therefore, such mechanisms are not considered in this paper. Recently, TCP friendly rate control protocol (TFRC) was proposed as an unreliable, equation based transport layer 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]. Fundamentally, new research focus and solutions are required to provide transport layer support for real time applications in MHWN.

In this paper, we propose a new link adaptive transport protocol (LATP) to improve quality of service (QoS) performance metrics 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 provides the available capacity of the path to the sender. This allows the transport layer rate to be controlled accordingly, so that the sender does not overload the network and improves QoS performance.

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 [4–7, 9] and some solutions have been proposed in [4–7, 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 applicable 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 can be used to control the total offered load to support QoS requirements in WLAN. Motivated by this work, the same authors have proposed a new mechanism in [11] for improving TCP performance in mobile ad hoc networks. The proposed mechanism utilizes the channel busyness ratio measurements to detect the network status around a node and to control the end-to-end rate. However, this mechanism does not consider the possibilities of collisions in MHWN, when estimating the rate feedback. In fact the actual rate feedback depends on the probability of packet collision too. The probability of packet collision is very high due to hidden terminal problem in MHWN; unlike in single hop wireless networks [12]. Moreover the proposed 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 incoming data packet for MAC layer estimations at each intermediate node. This 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, 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 performance for real time applications.

3 Congestion problem in MHWN

Unlike the congestion nature in traditional wired networks, in MHWN, there can be many reasons for network congestion due to its multi hop wireless channel nature. In this section, we first study the congestion problem in contention based MHWNs, using ns2 simulations. We particularly diagnose the main 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 given 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 the offered load increases, packet loss rate (fraction of sent data packets not received at the destination) and end-to-end delay increase dramatically.

Figure 1a 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 Fig. 1b. 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. However the



Figure 2 Channel busyness ratio at nodes along the path

channel busyness ratio of a node alone is not sufficient to estimate the available channel capacity of a link (sourcedestination pair at MAC layer) in MHWNs due to hidden terminal problem. Thus, the available capacity estimation should consider not only the network status around a node but also the impacts due to hidden terminal problem in MHWN. We discuss this in detail below.

In fact in MHWN, traffics from different location have different impacts on a link in the network due to the shared nature of wireless links. Not only the traffics require a link contend for the medium, but also the traffics passing through the links in carrier-sensing range or interference range may impact the performance of that link. Consequently, the capacity of a link becomes time variant due to these impacts. In this paper, we illustrate these impacts on a link in two cases. We discuss the impact due to transmissions on carrier sense links in the first case and in the second case the impact due to transmissions on hidden terminal links (akin to [12]). Please note that in [12], the authors have considered 'carrier-sensing range' of a node as 'interference range'. However in our discussion we consider the ranges, carrier-sensing range and interference range as two separate ranges. Unlike carrier-sensing range, interference range is determined by the channel capture threshold and varies with the link distance [15].

We denote the link with transmission from node *i* to node *j* as l_{ij} and the distance between node *i* and node *j* as d_{ij} . Also we assume all the nodes in the network have the same transmission range, R_X and carrier-sensing range, R_S $(R_S \ge R_X)$. As an example, we consider a network with four nodes *a*, *b*, *c*, and *d* in four different scenarios, as shown in Fig. 3, for illustrating the two cases mentioned earlier. In all scenarios, we assume that the transmission on link l_{cd} starts first.

In the first case, the link, l_{cd} can be considered as carrier sense link of candidate link, l_{ab} if:

- i $d_{ac} \leq R_S$ —This means that node *a* (candidate source) can physically sense the transmission from node *c* to node *d* and will then consider the medium as busy as shown in Fig. 3a; or
- ii $d_{ad} < R_X$ —This means that node *a* will virtually sense the medium as busy as it can receive CTS from node *d* as illustrated in Fig. 3b. The transmissions on carrier sense links can be sensed by the candidate source. Therefore, the source will freeze its contention window and will not transmit new packets, during the period of transmissions on the carrier sense links according to IEEE 802.11 DCF standard. This will avoid potential collisions in the network.

In this case, channel busyness ratio measurement at a node will provide an accurate measure of medium contention level experienced by that node. Indeed, the



Figure 3 Medium contention in MHWN. **a** $d_{ac} \leq R_S$. **b** $d_{ad} \leq R_X$. **c** $d_{ac} \geq R_S$ and $d_{bc} \leq R_S$. **d** $d_{ac} > R_S$ and $d_{bd} \leq R_S$.

channel busyness ratio measurement can be used to predict actual channel availability around a node including the time wasted in unnecessary back-off states.

Unlike in single hop networks, transmissions on a link could still collide with transmissions on hidden terminal links in MHWN and increase the medium contention level in the network. We discuss this type of impact in the second case, where link, l_{cd} can be considered as hidden terminal link of l_{ab} if:

- i $d_{ac} > R_S$ and $d_{bc} \le R_S$ —In this scenario, shown in Fig. 3c, node *a* will not sense the transmission from node *c* and therefore can attempt a new transmission by sending an RTS packet to node *b*. However node *b* can physically sense the transmission from node *c* and will ignore the RTS packet from node *a*. Thus, node *a* will interpret this as a collision and trigger the exponential back-off algorithm of 802.11 MAC; or
- ii $d_{ac} > R_S$ and $d_{bd} \le R_S$ —As before, node *a* is unaware of the transmission on l_{cd} and can send an RTS packet to node *b*. This RTS packet also has the probability to be ignored by node *b* since node *b* can virtually sense the transmission on l_{cd} if $d_{bd} \le R_X$ as it can receive CTS packet from node *d* and sets its network allocation vector as busy for the transmission period on l_{cd} or if $d_{bd} > R_X$, node *b* can not understand the CTS packet from node *b* but may physically sense the acknowledgement packet transmission from node *d* when the RTS packet from node *a* reaches it, since node *d* is within its carrier-sensing range ($d_{bd} \le R_S$). This scenario is shown in Fig. 3d.

In the above scenarios, we considered only the possibility of RTS collisions on l_{ab} since we assumed that the transmission on l_{cd} starts first. On the other hand, consider that the transmission on l_{ab} starts first in Fig. 3c, as an example. The data packet from node a will be interfered at node b by a transmission attempt on the hidden link, l_{cd} if node c lies within the interference range of node b (i.e. at node b, the power level of the packet received from node a is less than the power level of the packet from node c plus the capture threshold), otherwise the data packet transmission on l_{ab} will be successful. Please refer [15] for more detail on this kind of collisions. Thus, in MHWN, RTS and data packets could collide due to the hidden terminal problem and the probability of collisions can not be predicted locally using channel busyness ratio measurement at a node. However, channel throughput measurement based on successful transmissions at a node will account this impact locally. Therefore in our proposed solution, we use both the channel busyness ratio and the throughput measurements in order to predict the medium contention level of a link locally at the source node, without utilizing any additional signaling mechanism.

4 Link adaptive transport protocol

The primary goal of LATP is to provide an efficient transport layer rate for media streaming applications in contention based MHWN. The proposed protocol, LATP exploits cross layer coordination in a systematic way, as shown in Fig. 4. The MAC layer provides permissible throughput information to the cross layer information base.



This information is used by the transport layer to determine the end-to-end transport layer rate for the media streaming application. Here, we assume an appropriate rate adaptation technique is incorporated with the streaming application to support the end-to-end rate determined by the transport layer protocol.

In particular, an LATP sender transmits a stream of data packets to the receiver and controls the offered load (sending rate) at the transport layer based on the feedback information from the receiver. Each intermediate node provides a feedback on permissible throughput (P) to update rate-feedback (R) on every outgoing LATP data packet, based on MAC layer estimations. An options field in the internet protocol (IP) header is used to carry the ratefeedback on all LATP data packets. Finally, the ratefeedback is used by the receiver to estimate the degree of medium contention on the path and to inform the sender to control the sending rate (S) at the transport layer appropriately, using periodic feedback packets. Details of this processing are presented in the following sections.

4.1 Intermediate node

Intermediate nodes in the connection path play a key role in providing accurate and stable feedback information for LATP operation. 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 based single hop wireless networks. However, in MHWN the channel busyness ratio can be used to determine the medium contention level experienced by a node but not by a link, as we discussed in Section 3. 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 20µs (one back-off timeslot period). An estimation B at that node is derived using the well-known exponentially weighted moving average (EWMA) filtering with weight, α 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 μ is the sample average and σ is the sample standard deviation. If B_{sample} falls within the observation window, i.e. within $B \pm 3\sigma_B$, where σ_B is the standard deviation of the samples obtained so far, *B* is updated with a small value for

weight α , otherwise we assume the sample is too noisy and update *B* with a high value for weight α . We empirically selected these small and high values for α as 0.125 and 0.875 respectively. And also, we estimate σ_B as follows.

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

where weight, β is set to 0.875 for a stable window size.

Since, packet collisions can still happen due to the hidden terminal problem in MHWN we include this impact in throughput estimation as mentioned earlier. 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, as shown in Fig. 5. 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 (ACK) is received for that transmission [16]. 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:

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

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

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

Finally, for every outgoing LATP data packet, the node calculates its permissible throughput, P based on the channel busyness ratio and the throughput estimations, as given below, to efficiently utilize the channel, while avoiding severe medium contention. It then updates the rate-feedback, R in the IP header to the value of P, if P is smaller than R.

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



Figure 5 Packet transmission sequence in IEEE 802.11 DCF

where B_{TH} is the threshold value for *B* and is set 95% in order to keep the collision probability low at the operating point [8]. 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 and DATA collisions [12]) in MHWN. Further more, it does not require any transport layer information at the MAC layer for ratefeedback 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 field (IP header field) and utilizes the cross layer coordination in a systematic way, i.e. LATP requires cross layer information flow only in one direction; from lower layer to upper layer as shown in Fig. 4, unlike the scheme proposed in [11]. This makes the system can cooperate with other types of upper layer protocols as well.

4.2 LATP receiver

An LATP receiver sends feedback packets at regular reporting periods in order to assist the sender to determine the transport layer 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, ω 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 LATP 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/rrt ensures a minimum sending rate of one packet per *rtt*, as in TFRC.

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 LATP, 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 δS , where δ 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 δS , 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 $\frac{1}{9}S$, once per *rtt* period, for the negative feedbacks. This rate reduction value is taken by considering the facts that LATP takes continuous rate control action, normally every rtt period, in particular, rate increment is small when medium contention is high and below threshold, and LATP 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.

4.4 LATP packet formats

This section discusses the header formats for the proposed LATP protocol. An LATP header follows the IP header, supplying information specific to the LATP protocol. Moreover, an options field in the IP header is allocated to carry the bottleneck node's permissible throughput information to the LATP receiver, as mentioned before. Figure 6 presents the header formats of LATP data and feedback packets. The LATP data packets are used to send control packets, during connection initiation and termination, and payload packets from the LATP sender to the LATP receiver, while the LATP feedback packets are used by the receiver to send any feedback or response to the sender.

As shown in Fig. 6a, an LATP data packet consists of the following fields and in total has a header size of 12 bytes (excluding payload). Please note that, an appropriate header compression technique can be incorporated at the transport layer to reduce the header size, as all the fields defined in the LATP header are only used by the end stations.



Figure 6 LATP packet formats. a LATP data packet. b LATP feedback packet

- Source port and destination port: 16 bits each—These fields are used to uniquely identify the transport connection at both ends.
- Sequence number: 24 bits—This provides identification to LATP data packets, so that losses can be detected and reported. The sender increases the sequence number by one per packet.
- Packet type (PT): 4 bits—This field specifies the type of the packet. Different packet types are used for connection initiation, data transfer and connection termination purposes.
- Urgent (UG): 1 bit—This bit is set by the sender to request immediate feedback from the receiver.
- Reserved (Res): 3 bits—Reserved for future use.
- Time stamp: 16 bits—This indicates when the data packet is sent and is echoed by the receiver on its feedback packets to compute the round trip time at the sender. The resolution of the time stamp is typically in milliseconds. This field can be made optional, if the sender keeps the time stamp records of the sent packets.
- Round trip time (RTT): 16 bits—This gives the current *rtt* estimate of the sender to the receiver. The resolution is in milliseconds.

An LATP feedback packet, as shown in Fig. 6b, has a fixed size of 20 bytes (This can also be compressed as mentioned before) and consists of some other fields, defined below, specific to the feedback packet.

- Feedback ID: 24 bits—This provides the last sequence number received by the receiver and uniquely identify the feedback packet. This is also useful at the sender to update the *rtt* estimation.
- Loss detected (LD): 1 bit—When a new loss is detected, the receiver sets this bit and sends a feedback packet immediately to the sender, as mentioned in section 4.2.
- Time stamp: 16 bits—In this field, the receiver copies the time stamp of the last packet received when it generates a feedback packet.
- Time elapsed: 16 bits—This gives the time elapsed between the reception of the last data packet and the generation of this feedback packet at the receiver and is used to make *rtt* estimation more accurate at the sender. Please note that the sender estimates the *rtt* based on the time stamp and therefore the time taken by the receiver to send the feedback packet after receiving the last data packet should be informed.
- Rate feedback: 32 bits—The current rate-feedback estimation (R_{avg}) at the receiver is sent in this field to the sender. The resolution is in bytes per second.
- Receive rate: 32 bits—The average receiving rate (*s*) at the receiver during the last reporting period is reported to the sender in this field, in bytes per second.

5 LATP performance evaluation

In this section, we evaluate the performance of LATP over a variety of scenarios using *ns2* simulations. LATP 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 as we study the impacts of medium contention on transport layer performance in this paper. The IEEE 802.11 DCF MAC protocol, with some modifications to support LATP, and ad hoc on demand distance vector (AODV) routing protocol [17] 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: LATP, TFRC and TCP have always data to send to the destination. The results presented were taken over 10 simulation runs, unless otherwise specified.

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 [18] as given below.

$$\frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2}$$

where x_i is the throughput of the *i*th flow and *n* is the number of flows competing for the network resources.

 Table 1
 Simulation parameters

Parameters (units)	Value	
Propagation model	Two ray ground	
Transmission range (m)	250	
Carrier-sensing range (m)	500	
Capture threshold (dB)	12	
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	



Figure 7 n-hop Chain topology

5.1 Chain topology

The proposed protocol was first evaluated over a chain topology as shown in Fig. 7, 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. It is quite common to have chain-like topologies in ad hoc and mesh networks where the deployment is not highly dense in the whole network.

First, LATP, 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 Fig. 8.

Figure 8a shows the delay performance of LATP, TFRC and TCP flows. We can clearly observe from the figure that both TFRC and TCP experience higher delay than LATP flows. In particular, TFRC flows experience at least 100% more delay than LATP 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 connections. Then, for each packet the MAC layer attempts multiple retransmissions and backoffs 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 Fig. 8a, LATP provides much better delay performance than other flows, in all number of hops



Figure 8 LATP, TFRC and TCP performance with increasing number ► of hops connection. **a** End-to-end delay. **b** Jitter. **c** Throughput. **d** Packet loss rate. **e** Smoothness



connections. Since LATP controls the sending rate according to the network condition, 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 Fig. 8b. Although TFRC and LATP both provide considerably good jitter performance for streaming applications over the chain topology, LATP outperforms TFRC.

Packet loss rate (PLR—fraction of packets sent and not received by the destination) is also estimated and presented in Fig. 8d. LATP exhibits much better performance in terms of PLR than TFRC and TCP. Throughput smoothness is also measured at the receiver and presented in Fig. 8e. Both, LATP and TFRC show good throughput smoothness performance. However, LATP's smoothness is better than TFRC's smoothness.

In Fig. 8c, we observe that TFRC obtains higher throughput in small number of hops connections than LATP, and both TFRC and LATP achieve almost same throughput in connections with more than 8 hops, under the given simulation parameters in Table 1. Although LATP looses some throughput in small number of hops connections, we believe that the performance improvement achieved with LATP for delay, jitter, PLR and throughput smoothness will significantly improve the performance of media streaming applications in MHWN.

In order to evaluate the performance with competing flows in the network, we performed the second set of





Figure 11 Performance comparison in grid topology. a End-to-end delay. b Jitter. c Throughput. d Fairness. e Packet loss rate. f Throughput smoothness

simulations over a 7-hop chain topology, with the same simulation configurations as before.

We performed simulations for 400s for LATP, TFRC and TCP protocols separately with 4 competing flows. The first flow commenced at 10s and the subsequent flows were started in 10s intervals. All the flows were stopped at the end of simulation time i.e. at 400s. We took the measurements over a steady period of 150s to 400s and present the average results taken for each flow over 10 simulation runs, in Fig. 9.

With competing flows, the end-to-end delay and PLR for each flow are much reduced with LATP compared to TFRC, as shown in Figs. 9a & d respectively. Although TCP exhibits comparatively better delay performance than TFRC, it shows very poor performance in terms of jitter, throughput and smoothness. Moreover, LATP outperforms TFRC and TCP in terms of jitter and smoothness performance, as shown in Figs. 9b & e. Figure 9c shows that, LATP achieves little less throughput than TFRC, while providing excellent fairness among the competing flows.

5.2 Grid topology

We also evaluated LATP performance in an 8×8 grid topology as shown in Fig. 10, in four different scenarios. These scenarios were selected so that the flows in the

Table 2 Performance comparison in random topology

Flow type	Avg. delay (ms)	Avg. PLR	Avg. smoothness	Fairness index	Aggregate throughput (Kbps)	Avg. jitter (ms)
LATP	232.93	0.154	0.460	0.894	257.3	129.04
TFRC	938.27	0.285	0.328	0.799	303.2	285.92

network experience various conditions. (1) Scenario 1-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-sensing range and out of the transmission range of each other (see Fig. 10). (2) Scenario 2-In this scenario, 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-sensing range. (3) Scenario 3-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. (4) Scenario 4 - Finally, 2×2 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 network. The flows were started in 20s intervals and the measurements were taken over a steady period.

The average results obtained are shown in Fig. 11. In all scenarios, LATP outperforms TFRC in terms of end-to-end delay, jitter, PLR, and smoothness while providing almost same performance as TFRC in terms of aggregate throughput and fairness index. In particular, the performance improvement achieved with LATP in terms of end-to-end delay and PLR is highly significant in all scenarios.

5.3 Random topology

Finally we performed simulations in a random topology. We placed 100 nodes uniformly and randomly in a $1600m \times 1600m$ 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 2. We still observe from the results that the end-toend delay, jitter, PLR, throughput smoothness and fairness performances of LATP 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.

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, LATP to improve the performance of media streaming applications in such networks. Our proposed protocol, LATP exploits cross layer coordination in a systemic way and performs transport layer load control for end-to-end flows efficiently, based on the level of medium contention information received from the intermediate nodes on the end-to-end path.

In fact, LATP provides an offered load 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 PLR, which are key QoS metrics for media streaming applications. Simulation results over chain, grid and random topologies under various network conditions confirmed that LATP achieves considerable performance improvement over TFRC protocol, in terms of end-to-end delay, jitter, PLR, throughput smoothness, and fairness for media streaming applications. We have also noted that although LATP does not improve the throughput performance, it provides almost same throughput as with TFRC protocol in MHWN. Thus, it clearly demonstrates that by using an efficient transport layer load control mechanism based on the level of medium contention information from the network, we can improve QoS performance of multimedia services in MHWN. LATP provides this with less complex estimations and mechanisms. Further, LATP can be incorporated with the existing application layer rate adaptation mechanisms, such as real time transport protocol as a transport protocol for such mechanisms and we aim to investigate this in our future work.

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

References

- Holland G, Vaidya NH (1999) Analysis of TCP performance over mobile ad hoc networks. Proc. ACM Mobile Communications Conf., Seattle, WA, August 1999219–230
- Liu J, Singh S (2001) ATCP: TCP for mobile ad hoc networks. IEEE J Sel Areas Commun 19(7):1300–1315 Jul
- Fu Z, Meng X, Lu S (2003) A transport protocol for supporting multimedia streaming in mobile ad hoc networks. IEEE J Sel Areas Commun 21(10):1615–1626 Dec
- Fu Z, Luo H, Zerfos P, Lu S, Zhang L, Gerla M (2005) The impact of multihop wireless channel on TCP performance. IEEE Transactions on Mobile Computing 4(2):209–221 Mar/Apr
- Nahm K, Helmy A, Jay Kuo C-C (2005) TCP over multihop 802.11 networks: issues and performance enhancement.Proc. ACM MobiHoc'05, Urbana-Champaign, May 2005
- Chen K, Xue Y, Nahrstedt K (2003) On setting TCP's congestion window limit in mobile ad hoc networks.Proc. IEEE ICC 2003, Seattle, May 2003

- Xu K, Gerala M, Qi L, Shu Y (2003) Enhancing TCP fairness in ad hoc wireless networks using neighbourhood RED.Proc. ACM MobiCom'03, San Diego, September 2003
- Zhai H, Chen X, Fang Y (2005) How well can the IEEE 802.11 wireless LAN support quality of service. IEEE Trans Wirel Commun 4(6):3084–3094 Nov
- Li M, Lee CS, Agu E, Claypool M, Kinicki R (2004) Performance enhancement of TFRC in wireless ad hoc networks.Proc. DMS'2004, San Francisco, September 2004
- Handley M, Floyd S, Padhye J, Widmer J (2003) TCP friendly rate control (TFRC): Protocol specification, IETF RFC3448, Jan
- Zhai H, Chen X, Fang Y (2005) Rate-based transport control for mobile ad hoc networks.Proc. WCNC2005, New Orleans, March 2005
- Wang K, Yang F, Zhang Q, Xu Y (2007) Modeling path capacity in multi-hop IEEE 802.11 networks for QoS servces. IEEE Trans Wirel Commun 6(2):738–749 Feb
- Chen K, Nahrstedt K, Vaidya N (2004) The utility of explicit ratebased flow control in mobile ad hoc networks. Proc. WCNC2004, Atlanta, March 2004
- Sundaresan K, Anantharaman V, Hsieh HY, Sivakumar R (2005) ATP: A reliable transport protocol for ad hoc networks. IEEE Trans Mob Comput 4(6):588–603 Nov/Dec
- Ng PC, Liw SC (2007) Throughput analysis of IEEE802.11 multihop ad hoc networks. IEEE Trans Netw 15(2):309–322 Apr
- 16. Shah SH, Chen K, Nahrstedt K (2003) Dynamic bandwidth management for single hop ad hoc wireless networks.Proc. IEEE International Conference on Pervasive Computing and Communications, Fort Worth, March 2003
- 17. Perkins C, Royer E, Das S (2003) Ad hoc on demand distance vector (AODV) routing, IETF RFC 3561
- Jain R, Chiu D-M, Hawe W (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. In Technical Report TR-301, DEC Research Report, Sept

Pirabakaran Navaratnam received the B.Sc.Eng. degree in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka. He is currently working toward the PhD degree in mobile communications at the University of Surrey, UK. His research interests include congestion control in multi hop wireless networks, cross layer design, and multimedia wireless sensor networks.

Haitham Cruickshank is a lecturer at University of Surrey, U.K. He worked there since January 1996 on several European research projects in the ACTS, ESPRIT, TEN-TELECOM and IST programmes. His main research interests are network security, satellite network architectures, VoIP and IP conferencing over satellites. He also teaches in the Data and Internet Networking and satellite communication courses at University of Surrey. He is a member of the Satellite and Space Communications Committee of the IEEE ComSoc, also he is a chartered engineer and corporate member of the IEE in UK.

Rahim Tafazolli is Professor of Mobile/Personal communication and Head of Mobile Communications Research Group at the University of Surrey, UK. He has been active in research for 20 years and has coauthored more than 300 papers in referred international journals and conferences. Professor Tafazolli has been consultants to many mobile companies, lectured and chaired a number of IEE summer schools and IEEE workshops and conferences. He has carrier out many consultancies for mobile companies, Home Office, and European Union all in the field of mobile communications. He is the founder and chairman of IEE International Conference on 3rd Generation Mobile Communications. He is the chairman of e. mobility technology platform expert group tasked with defining the strategic research agenda for EU FP7 programme of research on mobile / wireless communications.