

A Novel transport protocol for wireless mesh networks

KAFI Mohamed Amine, TANDJAOUI Djamel
DTISI
CERIST Research Center
Algiers, Algeria
{kafi, dtandjaoui}@mail.cerist.dz



ABSTRACT: *Throughput is the main concern in Wireless Mesh Networks (WMNs). The standard congestion control mechanism of TCP[7] is not able to handle the special properties of a shared wireless multi-hop channel well. Given that the unreliable wireless links and congestion are likely to be the source of throughput degradation in the network, reliable transport protocol conception dealing with wireless links properties can significantly improve the performance of such networks.*

This paper presents a novel transport protocol called MTCP, which covers these needs. Simulation results show that MTCP outperform TCP and its ad-hoc wireless variants in terms of reliability and congestion control.

Keywords: Wireless Mesh Networks; Transport protocol; TCP; Reliability; Congestion control

Received: 19 January 2011, Revised 22 February 2010, Accepted 1 March 2011

© 2011 DLINE. All rights reserved

1. Introduction

Wireless Mesh Networks (WMNs) [1] are dynamically self-organized, where nodes form automatically an ad-hoc network and maintain its connectivity. WMNs are composed of two types of nodes: the mesh routers and the mesh clients.

Through multi-hop communications, mesh routers can ensure the same coverage as long range communications but with less energy consuming. A mesh router may be equipped with more than one radio and support different access technologies. The WMN architecture can be classified into three basic groups depending on nodes' functionalities: i) The infrastructure WMN (mesh routers form infrastructure for clients), ii) Clients WMN (built by clients) and iii) the hybrid architecture (combination of the two types). WMNs can be deployed in an incremental way. As more nodes are installed, reliability and connectivity increase consequently.

Protocol layers need to be re-examined for this type of networks and adapted to the needs since they don't share same characteristics with wired networks. Currently, there is no reliable transport protocol suggested specifically for WMNs. However, several transport protocols are available for ad-hoc networks ([2], [5], [8]...). Reliable data transport protocols can be classified into two types: TCP variants and new transport protocols.

TCP variants seem to be the most adapted because of interoperability with existing networks. However, TCP utilization shows performance degradation in wireless networks. This is due to the invocation of congestion control mechanisms when losses caused by wireless links happened.

In this paper, MTCP (Mesh TCP) protocol is presented. Our main contribution is to adapt TCP for WMNs environment and to cope with the well known drawback of TCP in wireless environments.

MTCP consists of an intermediate layer between the network layer and the transport layer, which differentiates between the losses caused by the congestion - for which it keeps TCP standard behaviour- and losses caused by the wireless environment- for which the packet retransmission (in the event of losses) or the suspension of the transmissions (in the network partitioning event) will be adequate. Moreover, MTCP takes into account the effects of route changes while using quality of service routing protocol.

In order to evaluate MTCP protocol, several simulation scenarios using the NS2 simulator were performed. The obtained results support our approach and show that MTCP performance is better compared with those of standard TCP[7] and TCP versions adapted to Ad-Hoc networks.

The rest of the paper is organized as follows. Section II describes some background issues and related work. Next, Section III gives detailed description of the proposed protocol MTCP. Section IV is devoted to simulation and performance evaluation and finally Section V draws the conclusions and perspectives.

2. Related Work

In TCP [7], it is assumed that all segment losses are resulted from congestion and will trigger window-based flow control and congestion control. This style will incur that TCP will unwisely reduce transmission rate when there is no congestion but packet losses from bit-error or route failures. This behavior will lead to low throughput especially under multiple wireless hops.

The confusion between route failures and network congestion and as a result the congestion control call is not recommended since they are independent phenomena which must be treated independently. Consequently, a diagram called TCP-F was proposed [2]. This diagram enables the source to know about the route failure, and consequently to avoid unnecessarily congestion control. It abstains from sending any other packet until the route is reconstituted. As soon as the network layer at intermediate nodes (called failure point: FP) detects the route failure due to the mobility of the next node along this route, it explicitly sends a notification packet (RFN: route failure notification) to the source and records this event. If an intermediate node knows another route towards the destination, this route can be used for the next communications and the RFN is cancelled. Otherwise, the intermediate nodes propagate simply the RFN towards the source. At the RFN reception, the source enters a frozen state, in which it stops any sending. The source remains in this state until it is notified about the route restoration by the reception of a route reestablishment notification packet (RRN). As soon as the source receives the RRN, it changes into an active state and resumes the transmission again based on the window stored values. These stages reduce the TCP congestion control effect when the retransmissions start again. The communication resumes again at the same rate before the failure occurred (the source's rate is set to the last transmission rate used before the failure occurred).

Another approach called ATCP [5] suggests putting TCP transmitter in a persistent state, congestion control state, or retransmission state. Thus, when the network is partitioned, TCP transmitter is put in the persistent mode so that it does not transmit unnecessarily and does not retransmit the packets. In addition, when the packets are lost because of errors (in opposition to the congestion), TCP transmitter retransmits these packets without calling the congestion control. Lastly, if the network is really congested, TCP transmitter calls the congestion control. In this implementation, standard TCP itself is not modified to maintain compatibility with the standard TCP/IP. Consequently, a thin layer is inserted called ATCP (ad hoc TCP) between IP and TCP which listens to the network state information provided by ECN messages [4] (explicit congestion notification) and by ICMP messages "unreachable destination"[9] and puts then TCP transmitter into the suitable state. Thus, with the reception of ICMP message, the TCP state at the transmitter is frozen (the transmitter returns in persistent state [8]) until a new route is found ensuring that the transmitter does not call the congestion control.

Another solution to adapt protocols (of wired network) for wireless networks and ad hoc networks in particular is coordination between the various protocol layers. ATP [6] uses sub-layers information and explicit feedback of other nodes to help the transport layer mechanisms. Specifically, ATP employs the network feedback for three different goals: feedback of initial rate for the starting rate evaluation, the feedback of progressive rate for congestion detection and congestion avoidance and route failure notification.

ATP decouples congestion control and reliability contrary to TCP where the congestion control and reliability are narrowly coupled by the dependence on ACK arrival. In ATP, the congestion control is carried out by using the network feedback, whereas the reliability is ensured by the receiver feedback and selective ACKs. TCP interoperability is not a constraint under which ATP is conceived, since it is conceived for the ad hoc environments where the network nodes will have a dedicated protocol layer. However, in mesh networks interoperability is one of the main concerns.

3. MTCP Protocol

Our approach is founded on ATP [5], which is itself an improvement of TCP to deal with the emerging problems of ad hoc networks. As WMNs networks share several characteristics with ad-hoc networks, improving ATP is an interesting approach.

MTCP uses the network layer feedback (intermediate hops) to put TCP transmitter in a persistent state, congestion control state, or retransmission state. Thus, when the network is partitioned, TCP transmitter is put in the persistent mode so that it does not transmit unnecessarily and does not retransmit packets. In addition, when the packets are lost because of errors (in opposition to congestion), transmitter TCP retransmits these packets without calling the congestion control. Lastly, if the network is really congested, TCP transmitter calls the congestion control.

The solution consists of the insertion of a thin layer called MTCP (Mesh TCP) between IP and TCP, which listens to the network state information provided by ECN messages (explicit congestion notification) and by ICMP messages “unreachable destination” and then puts TCP transmitter into the suitable state. Thus, with the reception of an ICMP message, TCP state at the transmitter is frozen (the transmitter returns in persistent state) until a new route is found ensuring that the transmitter does not call the congestion control. Moreover, the transmitter does not send packets since no route exists between the source and the destination.

ATCP protocol adopts as strategy “congestion window initialisation” to one packet after new route reconstitution following a route failure. While the preceding improvements of TCP kept the same congestion window as before failure[2], the two approaches do not provide the best window value. Keeping the same congestion window as the old one (before failure) can cause congestion if the new route does not allow a rather window; or cause under estimation of the window if the new route is better in sense of bandwidth. But as connection is in “congestion avoidance” stage, convergence towards the optimum window value can be reached in a rather slow way. In this case, it is preferable to reinitialise the window size to one packet and increase it exponentially to converge more quickly to the optimum value, rather than keeping the old window and probe for the new window linearly since we are in “congestion avoidance” stage. However, also in this case (starting with one packet, and “slow start” stage), the frequent route change can cause the stagnation in the “slow start” stage and does not reach the ideal efficiency since each time the route change the window will be reinitialized with one packet.

Our approach use a window probing method to reach the optimal window just after the establishment of a new route and sending packets of the first congestion window. This technique has been used by TCP-RC [8]. The Cwnd becomes:

$$Cwnd_{aft} = \gamma * \sqrt{\frac{RTT_{bef} * HOP_{aft}}{RTT_{aft} * HOP_{bef}}} Cwnd_{bef}$$

RTT_{bef} and RTT_{aft} are the RTT values before and after route failure respectively.

HOP_{bef} and HOP_{aft} are the hop-count values from source to destination before and after failure respectively. Cwnd_{bef} and Cwnd_{aft} are congestion values window before and after failure respectively.

ATCP does not take into account the feedback from QOS routing protocols, which can have consequences on route changes. The route change due to quality of service can occur when using a routing protocol with metrics for route selection. If the routing protocol finds a new route more optimal compared to the current route then this route change causes the congestion window re-evaluation as described previously.

Since the quality of service handling is essential and must be addressed by WMN protocols, MTCP takes into account these feedbacks from the route changes due to quality of service. This also yields the congestion window re-evaluation by using the TCP-RC approach.

ATCP protocol does not also use selective ACKs (SACKs), since one of indications when using SACK option [3] is that the value of multiplying transmission time by the bandwidth must be high. As it is not the case for ad hoc networks, in general, the use of this option generates only an additional overload. On the other hand, in WMNs the use of this option will significantly improve the network throughput since the multiplying value is rather high in WMNs (an example is the satellite network which can be part of the WMN and which has this high value).

MTCP acts only at the TCP transmitter (in full-duplex communication, MTCP layer will be present at the two participating nodes).

During TCP connection initialization, MTCP is in the normal state, and then doesn't act. Let us examine MTCP behavior in the four circumstances. See figure 1:

- The channel with loss probability: When connection has high loss probability, segments can not arrive at the receiver or arrive in disorder. This causes the receiver to produce double ACKs in response to segments in disorder. When TCP receives three consecutive double ACKs, it calls the congestion control. It is also possible that because of lost ACKs, the TCP transmitter RTO expires involving the segment retransmission and the congestion control call.

MTCP in its normal state, exactly like ATCP, counts the number of double ACKs received for each segment. With the reception of the third double ACK, it does not dispatch it but puts TCP in the "persistent mode". When the RTO is about to expire, MTCP puts TCP in "the persistent mode". Therefore, by doing this, we make sure that the congestion control is not called. MTCP enters thereafter a "loss state". In the "loss state", MTCP retransmits the segments not acquitted from the TCP sending buffer. MTCP maintains its own timers separated to retransmit these segments if their ACKs will not be received. As a new ACK arrives (an ACK for a segment previously not acquitted), MTCP follows this ACK to TCP layer which also removes TCP from the persistent mode. MTCP returns then to its normal state.

- Congestion: It is supposed that when the network detects congestion, ECN bit is placed in the data packets and ACKs [4]. Since MTCP receives this message while it is in its "normal state", it enters in "congested state" and does not do anything. It is unaware of any double ACKs which arrive and it is unaware also of RTO expiries. After TCP transmits a new segment, MTCP returns to its normal state.

- Network Disconnected: Nodes' mobility in ad-hoc networks causes re-computation of routes or even provisional network partition. This scenario is not critical in WMNs due to wireless infrastructure presence, but it can appear, for example, in the case of routers failure, which ensures connection of a given path, or clients mobility in architecture with no infrastructure.

When this occurs, it is supposed that the network produces an ICMP message "inaccessible destination" in response to packets transmission. When MTCP receives this message, it puts TCP transmitter in "the persistent mode" and enters in "disconnected state". TCP produces probe packets periodically while it is in "the persistent mode". Thereafter, if the receiver is connected to the transmitter again, it answers these packets with a double ACK (or a data packet, if bidirectional communication). This behavior removes TCP from "persistent mode" and again puts MTCP in "the normal state". In order that TCP does not continue to employ the old congestion control window value, MTCP recalculate the new window value with the method explained earlier.

- QOS route change: When using MTCP with QOS routing protocol, it is supposed that a bit is positioned by the routing protocol to indicate to MTCP when route change happened because of QOS metric. This change, cause also window re-initialization, but using MTCP method, explained above.

4. Performance Evaluation

In order to evaluate the performance of the proposed protocol (MTCP), simulation tests were carried out by using NS2 simulator.

The goal of the various scenarios is to examine the performance of MTCP in the presence of BER (bit errors), network partition, congestion, and route changes caused by the quality of service which is held as a metric in certain routing protocols, each parameter alone in a scenario. As a last scenario the various parameters are put together.

As already mentioned in the previous sections, MTCP is an improvement of ATCP. Therefore the comparison was made with standard TCP RENO and ATCP protocol.

We have chosen network architecture with five hops through which various simulations were done; the simulation time was

fixed to 120 seconds for each scenario. See figure2.

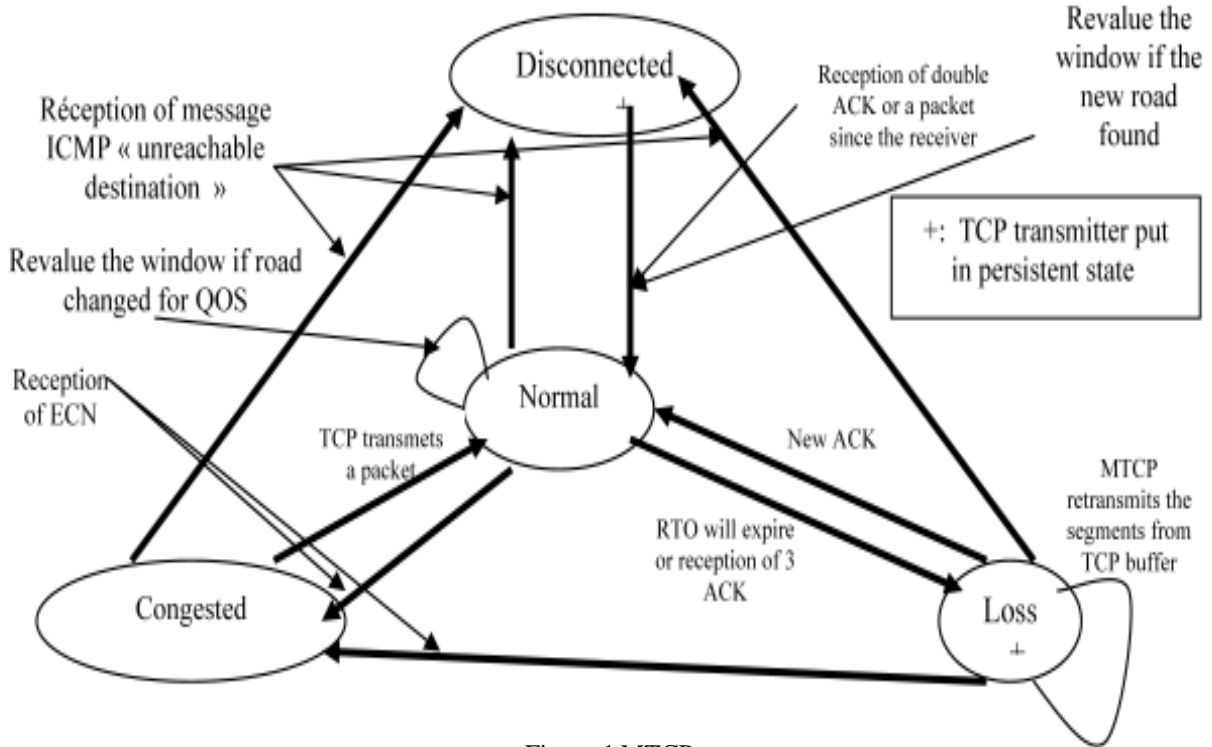


Figure.1 MTCP states

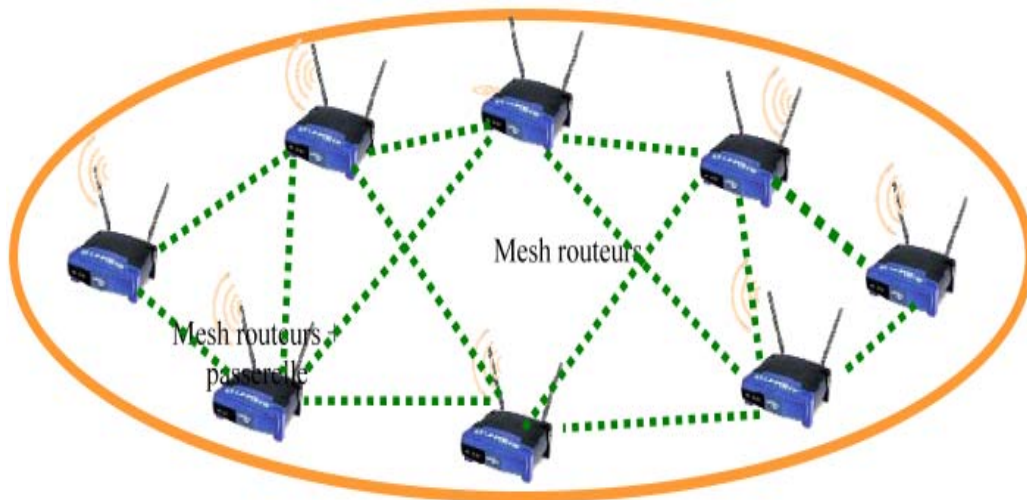


Figure 2. network architecture

4.1 Losses case

The first test did not include the partition or congestion events. The connection was only subjected to the bit error which occurred with a BER at each hop. This is done in simulation by sending packets with a probability. The packets evolution sequence numbers according to time is shown in figure 3.

ATCP and MTCP evolution are better than TCPRENO since the two protocols have a mechanism to make difference between wireless losses and congestion losses.

In TCP, packets or ACKs losses result in timeouts and retransmissions. Moreover, three double ACKs result in a retransmission.

In these cases, TCP reduces its congestion window, resulting in performance degradation. ATCP and MTCP have an identical behaviour in the losses event caused by BER, thus we don't find a TCP timeout because they put TCP in the persisting mode and retransmit the not acquitted packet.

TCP employs the "slow start" to increase its transmission window. ATCP and MTCP, in the other hand, put TCP transmitter in the persisting mode and retransmit the packets whose retransmission timer is about to expire. The TCP congestion window can never increase because the losses due to the BER result in congestion control. In opposite, ATCP's and MTCP's congestion window never decrease. This explains the dramatic difference in TCP, ATCP and MTCP performances.

Finally, the difference between ATCP and MTCP is explained by the fact of selective ACKs use [3], since we cannot completely eliminate congestion to happen in this first test, and thus the advantage of the selective ACKs to manage the retransmissions due to congestion appears.

4.2 Congestion case

In congestion test case, the congestion will be introduced into the network by increasing the data packets size in addition to the restriction of receiver window (small buffer size) to saturate the network.

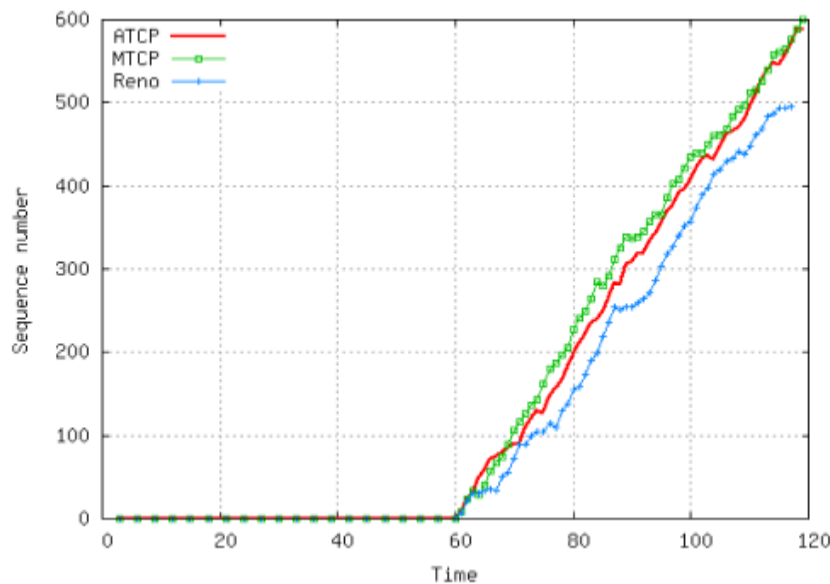


Figure 3. Protocols comparison in case of packets losses

There are many reasons causing the difference in performance between MTCP, ATCP and TCP RENO. For TCP RENO, the congestion control detection will be after a timeout or at the reception of three double ACKs. On the other hand in the ATCP case, the congestion control behaviour is anticipated since the reception of ECN message which indicates congestion somewhere in the intermediate nodes. Thus ATCP behaviour would be better than that of RENO.

In addition, MTCP improves ATCP behaviour with the selective ACKs option which decreases the packets number retransmitted following the congestion. Therefore, MTCP behaviour is better compared to TCP for the same reasons than ATCP. Furthermore, its behaviour is better than ATCP since MTCP uses the selective ACKs option, which enables it to face several losses in the same congestion window [3]. See figure 4.

4.3 Partition case

In this section, we consider the case where the network suffers from periodic partitions. See figure 5.

We simulated these partitions by moving away the nodes so that they are put in different coverage zones.

It is easy to explain the data packets sequence numbers evolution in time for ATCP. By comparing the sequence numbers evolution in absence of partitions with the case where the network is divided, we expect that the evolution delay of the sequence numbers will be at least equal to the partition duration.

ATCP puts TCP in the persistent mode at the time of the reception of ICMP message “inaccessible destination”. In the persistent mode, TCP produces probe packets. Poor TCP performance is caused by the periodic timeouts behaviour when the network is divided.

For MTCP the output is better compared to ATCP and TCP. This behaviour is since ATCP puts its congestion window to one packet after the re-connexion then its increase will be slower than MTCP which reach optimal window congestion value in only one RTT(Round trip time).

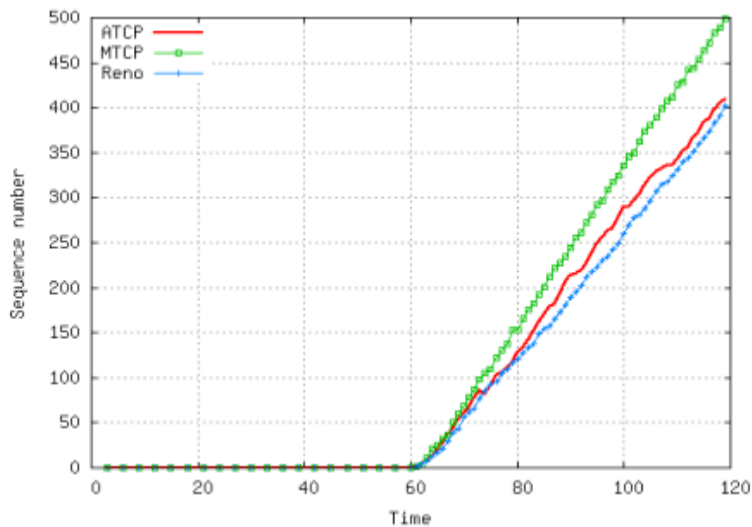


Figure 4. Protocols comparison in congestion case

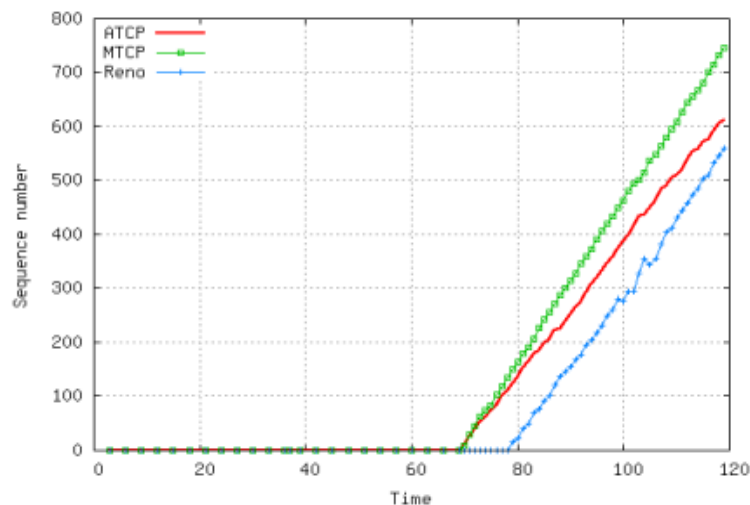


Figure 5. Protocols comparison in case of partitions case

4.4 Packets disorder case

The packet disorder can occur when there are multiple valid routes from the source to the destination, or when the route recomputation occurs. When a router has more than one outgoing interface which leads to the same destination, it can distribute the packets among these various interfaces providing that the packets go to the same destination. These packets can, therefore, arrive at the destination in disorder because they took various routes.

Another reason of packet disorder is the route re-computation. In wireless networks (especially ad hoc networks) the route failures occur frequently. The packets in the old route and those which take the new route can reach their destination in a different order.

TCP needs much more time to transfer the same number of data packets as ATCP and MTCP. The necessary time for ATCP

and MTCP remains almost constant while the transfer time for TCP increases (roughly) linearly as the hop to hop delay increases.

The reason of this difference is that ATCP and as well MTCP since their behaviour in this case is the same- put TCP transmitter in the persisting mode when they receive three consecutive double ACKs. In the other hand, TCP will begin the congestion control algorithm after three double ACKs, having as a result a congestion window appreciably smaller and a slow growth of its size. For this test case we have chosen route re- computation. See figure 6.

4.5 Case of route change for QOS

In this case our protocol shows better results compared to TCP and ATCP, since the route change due to the quality of service routing support is done in a transparent way for TCP and ATCP. See figure 7. This change is done on the routing layer level and thus TCP and ATCP will not change their congestion window. On the other hand, MTCP knows about this change through the network layer feedback. Therefore it will revalue its congestion window by using the TCP-RC method and thereafter will converge quickly to the optimal congestion window value of this new route. The new congestion window is surely larger since the routing protocol with QOS changes route if only the new route will be better than the old one.

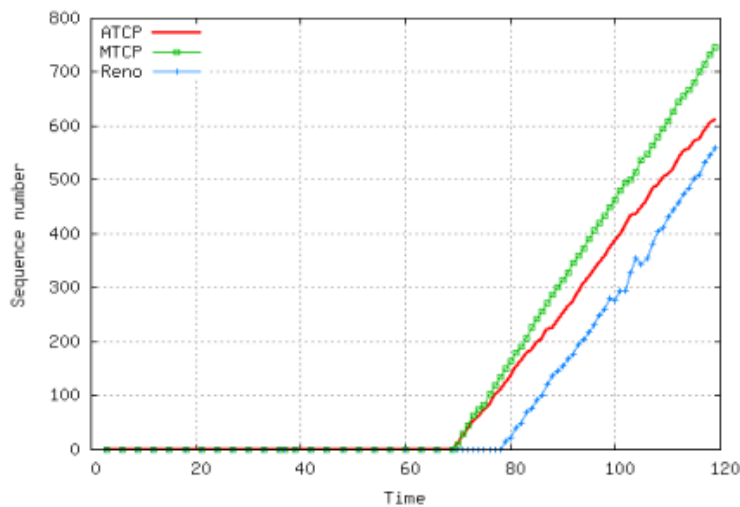


Figure 6. Protocols comparison in packet disorder case

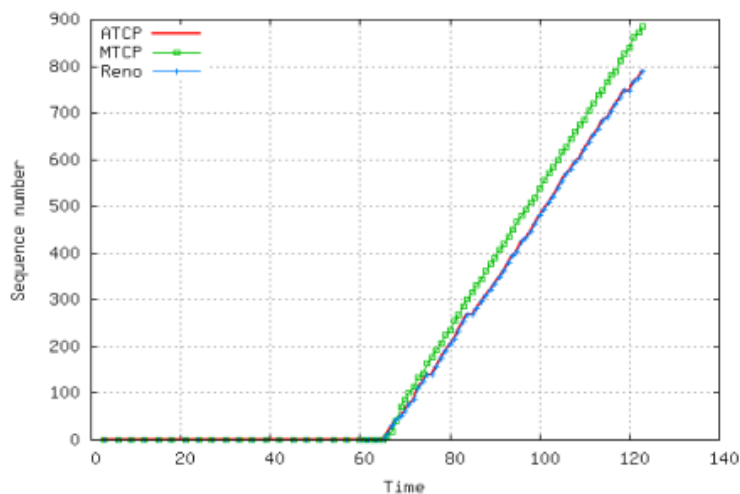


Figure 7. Protocols comparison in route change for QOS case

4.6 All cases together

Finally, we have run MTCP, ATCP and TCP in a network that presents all scenarios tests together, i.e. the multi way routing, the congestion, BER and route change due to QOS routing protocol. Simulations show that MTCP behaviour is better than ATCP and TCP which has demonstrated the worst performance among the three protocols. See figure 8. This result is a

logical behaviour since all previous test results.

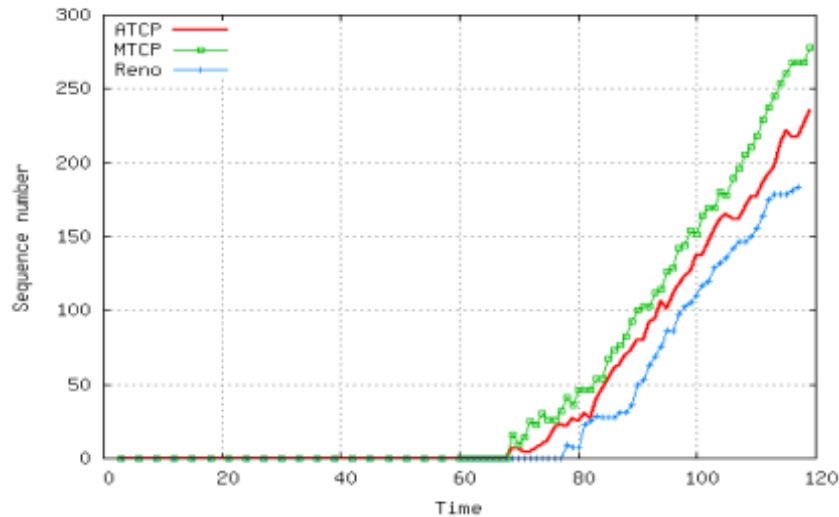


Figure 8. Protocols comparison with all cases together

5. Conclusion

In this paper we present MTCP a new protocol for wireless mesh networks that ensures interoperability with TCP standard, but its strong point is that it covers TCP drawbacks when used in wireless environments. MTCP presents the appropriate behaviour in cases of: road failures, road changes, packet disorder, real congestion and packet losses caused by wireless links.

Many simulation tests were done to argue the improvements of MTCP protocol by taking into account several circumstances in a wireless environment. Simulations had showed good results.

We envisage as future work to implement our protocol in a testbed environment to show the real benefit of the approach.

References

- [1] Akyildiz, I.F., Wang, X., Wang, W (2005). Wireless mesh networks: a survey, *Computer Networks* 47 (2005) 445–487.
- [2] Chandran, K., Raghunathan, S., Prakash, S.R (2001). A feedback- based scheme for improving TCP performance in ad hoc wireless networks, *IEEE Personal Communications* 8 (1) 34–39.
- [3] Mathis, M., Mahdavi, J., Floyd, S., Romanow, A (1996). TCP Selective acknowledgment Options, Sun Microsystems October RFC2018.
- [4] Ramakrishnan, K., Floyd, S. (1999). A Proposal to add Explicit Congestion Notification (ECN) to IP, RFC 2481, Network Working Group, LBNL.
- [5] Singh, S., Liu, J. (2001). ATCP: TCP for Mobile Ad Hoc Networks, *IEEE Journals on Selected Areas in Communications* 19 (7) 1300–1315.
- [6] Sundaresan, K., Anantharaman, V., Hsieh, H.-Y., Sivakumar, R (2003). ATP: a reliable transport protocol for adhoc networks, *In: ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)* p. 64–75.
- [7] Transmission Control Protocol. (1981) September, RFC793.
- [8] Zhou, J.X., Shi, B.X., Zou, L (2003). Improve TCP performance in Ad hoc network by TCP-RC, *The 14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings*.
- [9] Postel, J. (1981). RFC 792: INTERNET CONTROL MESSAGE PROTOCOL. September 1981.