TCP Performance in Mobile Ad Hoc Networks Connected to the Internet

Jonas Karlsson[1] and Andreas J. Kassler[2]

Computer Science Department
Universitetsgatan – 65188 Karlstad - Sweden
[1]jonas.karlsson@kau.se, [2]kassler@ieee.org

Abstract—**A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes (MN) that communicate using wireless links without support from any pre-existing infrastructure network. Packets are delivered from a source to a destination using packet forwarding capabilities of intermediate nodes. Therefore, MNs act as both end systems and routers. Mobile Ad Hoc networking has been considered as one of the most important and essential technologies that support future Pervasive Computing Scenarios. Recently, the usage of MANETs in the scope of 4G scenarios has attracted much research efforts and MANETs are seen as one way to extend coverage of hotspots in order to provide Internet connectivity to mobile users. However, TCP performance is crucial for user satisfaction but TCP is well known to suffer from low performance in wireless environments. In this paper, we evaluate several alternative TCP protocols on their suitability for Internet connected MANETs. We conclude that TCP- Vegas is a viable option as its performance is not significantly worse than those TCP variants highly specialized for MANETs but it is compatible with standard Internet protocols.**

Index Terms—**Ad Hoc Networking, TCP, Performance**

I. INTRODUCTION

Over the last years, the Internet has changed the way people communicate and its success is mainly due to its simplicity and the usage of the TCP/IP protocol suite which is the main communication mechanism. The common de-facto standard for reliable transport layer in the Internet is TCP and UDP is typically used for unreliable end-to-end delivery of e.g. multimedia data. TCP includes congestion control mechanisms and was designed to react in a robust way to changing network conditions that can occur in a wired network. Over the last decade, wireless networks based on e.g. 802.11 standards have been successfully deployed giving the users freedom to move while communicating. As a consequence, researchers have proposed several mechanisms to improve TCP performance over such one-hop wireless networks.

Recently, MANETs have attracted much attention due to their ease of use and fast deployment potential and consequently Ad Hoc networking has been considered as one of the most important and essential technologies that support future Pervasive Computing Scenarios [1] A MANET is a collection of mobile nodes (MN) that communicate using wireless links without support from any pre-existing infrastructure network. Packets are delivered from a source to a destination using packet forwarding capabilities of intermediate nodes. Therefore, MNs act as both end systems and routers. The usage of MANETs in the scope of 4G scenarios has attracted much research efforts and several projects like e.g. IST-DAIDALOS [2] consider the usage of MANETs to provide Internet connectivity to mobile users. In these scenarios, MANETs are supposed to extend the range of hotspots by providing multihop connectivity from MNs towards the Internet through one or more gateway nodes (GW) utilizing packet forwarding capabilities of intermediate nodes via multihop paths (Figure 1). In such an autonomous system, MNs are free to move randomly while still being able to communicate multi-hop in the MANET part with any Internet host which causes several problems not only for state-of-the art routing protocols but also for TCP.

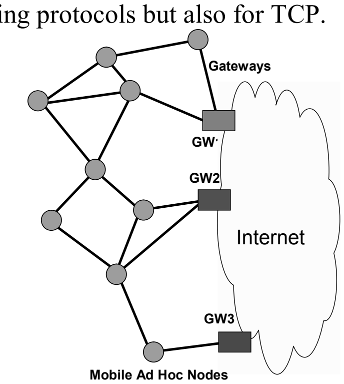

Figure 1. Architecture of Internet connected MANET

TCP uses a connection oriented approach where packets flow from a sender to a destination node. Each TCP flow is identified by certain parameters, with the IP-address being one of them. A change of the IP-address will therefore cause a drop of the connection, unless extra mechanisms such as Mobile IP are used. UDP does not need to re-establish a new connection at each address change or resend lost packets, as UDP does not guarantee delivery and therefore does not use ACK or other mechanisms to recover from lost packets.

. Classical TCP performance degrades significantly in *isolated* Ad Hoc networks mainly due to its inability to distinguish between packet loss caused by congestion and by other factors intrinsic to multihop networks. In MANETs, a significant portion of packet losses are caused by link failures either due to high bit error rate, mobility of nodes resulting in route errors or network partitions. Another source of problems is the complex cross-layer interaction between the MAC, routing and transport layer. When TCP probes for bandwidth aggressively during the slow start phase, there is a high probability for MAC layer contention induced packet loss, which will cause the routing protocol to trigger route error messages regardless if the route is valid thus increasing the problem even more. When TCP starts to react, the routing protocol might use already a different route, which causes unnecessary route changes, even in static scenarios, and large oscillation of the congestion window size [3]. Several solutions for TCP as well as new transport layer protocols have been proposed specifically for standalone MANETs that try to avoid the problems of TCP and a good overview can be found at e.g. [4]. However, only a few studies have been made on TCP performance of Internet connected MANETs.

In this paper, we concentrate on evaluating transport layer performance for Internet connected MANETs. For compatibility issues in Internet connected MANETs, MANET nodes could either run a specialized and optimized transport protocol and the gateway translates between the MANET protocol and standard TCP used in Internet hosts. This introduces overhead due to protocol conversion within the gateways. If the gateway is changed during an ongoing session due to mobility and multihop handover, a state transfer is necessary. This also leads to dropping multiple packets in flight cached at the old gateway or a transfer of those packets towards the new gateway. When TCP detects lost packets or time-out because the state transfer took too long time it assumes congestion. This will force TCP to enter slow start which results in serious performance problems. The transport layer needs therefore to be compatible and interoperable with already deployed transport protocols for integration into 4G networks.

Our contribution in this paper is the performance comparison of different transport layer solutions in Internet connected MANETs and the identification of problems to solve for their efficient operation. In contrast to other work (such as e.g. [5]) we evaluate and compare *several* TCP variants using reactive routing protocol integrated with gateway discovery (such as AODV-UU [5]), where we compare the transparency of transport layer solutions to routing protocols, the overhead and their performance using ns-2 simulations. We conclude that using standardized transport layer such as TCP Vegas [6] with properly tuned congestion control is more beneficial to use in an Internet connected MANET than specialized MANET transport layer solution such as TCP-AP [7], as we will show in the simulation chapter.

The remainder of the paper is structured as follows. The next section describes the issues involved in Internet connected MANETs. We also present background on Internet connectivity and TCP performance in that environment. Section 3 describes our simulation scenarios and presents results on TCP performance in internet connected MANETs. Finally, Section 4 concludes the paper.

II. Internet Connected MANETs and TCP Performance

A. Internet Connectivity for MANETs

4G systems are likely to consist of a combination of heterogeneous wireless technologies and naturally comprise MANETs as one component. Thus, the interconnection of MANETs to fixed infrastructure based IP networks will be very important in order to provide the ubiquitous user Internet access anywhere at any time. In such operator environment, functionalities required are e.g. to support multimedia services with the necessary Quality of Service (QoS), security, Authentication, Authorization, Accounting, Auditing and Charging (A4C/Security) services. Therefore, the Ad Hoc networks need to have the means to deliver a large diversity of services to potentially multihop connected users, with the aforementioned functionalities, for the satisfactory integration of the Ad Hoc cloud with such an operator business environment. In addition to that, an efficient integration with Internet needs to consider gateway discovery and auto-configuration, efficient routing mechanisms and load-balancing among multiple gateways, mobility support and an efficient transport layer implementation.

The discovery of gateways and the automatic configuration of network parameters such as IP addresses of mobile Ad Hoc nodes are key aspects affecting the overall performance of internet connected Ad Hoc networks. Several solutions address these problems and a detailed description and performance evaluation of existing schemes can be found in [8]. Figure 2 demonstrates a typical packet forwarding example in Internet connected MANETs. Note, that individual mobile nodes not only serve as relay stations but also can send their own data. All traffic destined towards a host located in the Internet needs to pass through at least one gateway, which can be mobile as well. A gateway is a node that has two interfaces, one running in infrastructure mode (wired or wireless) and one running in Ad Hoc mode.

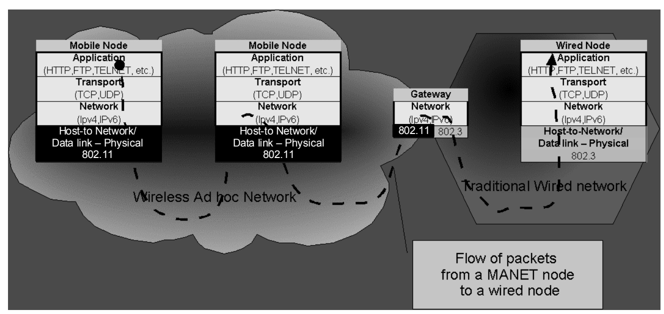

Figure 2. Communication in Internet connected MANETs

One of the main challenges in efficiently integrating Ad Hoc routing protocols in 4G networks is related to the efficient support of multiple gateways as the best route between two nodes within the same Ad Hoc network may go through the Internet crossing two Ad Hoc gateways. Routes towards nodes located in the Internet are generally implemented as default route entries within Ad Hoc nodes, or by setting up tunnels between Ad Hoc nodes and their selected gateways. Tunneling has higher overhead due to the additional header but allows for simplified handovers between gateways and load-balancing by simply switching the tunnel endpoint. Using default routes might lead to cascading route discovery phases thus degrading performance.

In this work, we use the AODV-UU[1] reactive routing protocol to establish routes between MANET nodes, as it also

[1] See http://core.it.uu.se/core/index.php/AODV-UU

enables communication between MANET nodes and the Internet. In AODV-UU, when a node has data packets to send and a route to the destination is not available, it first floods a route request (RREQ) message. If the destination is within the MANET, it unicasts back a route reply (RREP) addressed to the source. Alternatively, if the destination is outside the MANET, a gateway can send a special proxy RREPI on behalf of wired nodes thus integrating gateway discovery. Reverse routes are created during the flooding of the RREQ while forward routes are created towards the destination during the propagation of RREP(I). Once the routes have been established, the source node can send data packets. Packets towards the Internet are encapsulated by the source in a routing header pointing towards the gateway, whereas packets from the Internet are routed at the gateway using standard AODV procedures. The half-tunneling approach eases integration with global mobility management such as MobileIP.

B. TCP Problems in Internet Connected MANETs

The nature of wireless networks leads to route instabilities in the MANET due to mobility, falsely detected routing errors due to MAC layer contention and high collision rates. Present wired transport protocols (UDP/TCP) are inefficient in the given environment. While UDP cannot provide reliability guarantees, TCPs congestion control algorithm performs poorly in Ad Hoc networks due to e.g. its inability to distinguish between packet loss caused by congestion and by Ad Hoc specific MAC problems and routing layer issues.

Researchers proposed several transport layer solutions for standalone MANETs. For compatibility issues in Internet connected MANET, MANET nodes could either run a specialized and optimized transport protocol and the gateway translates between the MANET protocol and standard TCP used in Internet hosts. This introduces overhead due to protocol conversion within the gateways. If the gateway is changed during an ongoing session due to mobility and multihop handover, a state transfer is necessary. This also leads to dropping multiple packets in flight or cached at the old gateway or a transfer of those packets towards the new gateway. Then, TCP is likely to enter slow start resulting in serious performance problems. The second approach is to use standard TCP versions in MANET nodes (such as [6]) and the MANET network and MAC layer could be made more TCP-friendly. Finally, hybrid approaches have been developed recently such as TCP-AP (Adaptive Pacing) [7] for pure MANETs. TCP-AP implements adaptive pacing solely at the sender side while retaining TCP end-to-end semantics and thus is TCP interoperable with any valid TCP implementation but it has not been evaluated for internet connected MANETs.

Another source of problem for TCP is network asymmetry as TCP depends on received acknowledgements to trigger the transmission of new data. Therefore, the choice of routing to/from the gateway greatly influences TCP performance. In [5] it has been suggested to use tunnelling towards the gateway instead of deploying default routes to optimize TCP performance, but the influence of the congestion control on the performance has not been evaluated. The unfairness of the 802.11 MAC layer also poses problems if multiple flows traverse the gateway in different directions. It is assumed that in an internet connected MANET a significant portion of the traffic is exchanged between MANET nodes and internet located nodes which will pass through the gateway therefore increasing contention and the hidden node problem around the gateway covered area. Another problem is that MANETs have typically totally different characteristics as internet connections in delay bandwidth product, which determines the number of packets in flight for a TCP connection. Due to this, and the problem of extensive hidden node problems around the gateway leading to increased packet loss, when multiple flows traverse the gateway, TCP cannot converge to an optimal congestion window size [9]. While some packet scheduling mechanisms have been proposed to alleviate the problem, its usability has only been evaluated in a static scenario [10]. As a consequence, TCP's congestion control mechanism has to be reconsidered and made more adaptive for internet connected ad-hoc networks by studying in more details the complex interactions between MAC layer, gateway discovery, routing, and mobility. Another problem in MANETs in general is that sending too frequent ACKs are not beneficial as ACKs and TCP packets flow along the same route but in opposite directions. As a result, ACKs compete with TCP packets for medium access leading to increased contention. Also, when there are no data packets to picky-back ACKs upon, the overhead for transmitting ACKs is quite large in 802.11 based MANETs. Therefore, Authors have proposed to reduce the frequency of ACK packets applying ACK thinning mechanisms [14]. This is especially beneficial with higher bandwidth links, as the overhead for IEEE 802.11 transmissions increases for sending small packets with the data rate at the link layer. However, as this requires changing the receiver side, it does not allow for incremental deployment.

C. TCP Protocols evaluated

In this section we give a short overview on different TCP variants that have been used in our simulations.

1) TCP NewReno

The most widely deployed variant of TCP in the Internet is probably NewReno [11] and it has proven to be effective and adaptive to different traffic situations in the current wired Internet. The installed user base in the current Internet and the profound knowledge about this TCP variant together with the fact that there are no compatibility issues if NewReno is used within the MANET makes it the standard candidate to compare with. The idea behind congestion control in NewReno is to probe the network for available bandwidth by increasing the number of packets in flight until the network becomes congested and packet drop occurs. This method may lead to situations where TCP oversestimates the available bandwidth resulting in buffer overflows in gateways and routers in the wired network. In MANETs, this leads also to bad interactions with routing protocols. Generally, MANET routing protocols do not distinguish between different types of losses that occur such as losses due to MAC contention, channel errors, mobility etc. When TCP increases its window too fast, MAC layer contention increases which leads to higher back-off intervals and higher queue utilization eventually provoking packet drops. When packet loss occurs, routing protocols assume that the route is not valid, and typically invoke repair mechanisms involving flooding, increasing the contention even more and worsening the situation.. This loop continues until TCP timeout, or as long as the MAC contention is persistent [4]. Clearly, this behavior is not desired in MANETs.

2) TCP Vegas

TCP Vegas [6] is an interoperable variant of TCP to increase throughput, reduce packet loss and retransmissions, while not jeopardizing fairness. It has a different approach to probe for available bandwidth as it proactively adapts the congestion window *CWnd* to *avoid* packet loss. The idea is to measure and control the amount of extra data that a connection has in transit, which would not have been sent if the bandwidth used by the connection exactly matched the available bandwidth of the link. If too much extra data is sent, congestion will arise but if too little extra data is sent, reaction to transient increase in available bandwidth will be delayed.

Vegas computes the expected flow rate by *Expected=CWnd/BaseRTT* using current window size and minimum round trip time, an estimation of the propagation delay of the path. The current flow rate can be calculated as *Actual=CWnd/RTT* using the actual measured round trip time. The extra amount that could be sent is calculated as *Diff=Expected-Actual*. Based on *Diff*, the source updates *CWnd=CWnd+Delta,* where *Delta=+1* if *Diff* $< \alpha$; *Delta= -1* if *Diff* $> \beta$ and *Delta = 0* otherwise. Typically the two thresholds α and β are set between 1 and 3 in a wired network e.g. $\alpha = 1$ and $\beta = 3$, in practise $\alpha = \beta = 2$.

Modifications to the slow start behaviour and retransmission schemes are also introduced to avoid packet loss and unnecessary packet retransmissions. Analytical models, and simulation with wired networks have shown that TCP Vegas measures congestion by end-to-end queuing delay [12] and if the queue sizes at intermediate nodes are large, TCP Vegas achieves better throughput and packet loss ratio than TCP Reno. In the case of inadequate queue sizes, TCP Vegas cannot utilize its improved congestion detection mechanism and reverts to the same behaviour as TCP Reno [13]. Furthermore, [13] shows that the higher the available bandwidth is the more efficient TCP Vegas performs. Further studies in multihop wireless networks [14] verify that packet loss in TCP is mainly due to MAC layer contention, and not due to inadequate queue buffer sizes at intermediate nodes.

A problem could arise in multihop scenarios when mobility is introduced as TCP Vegas performance depends heavily on correct estimation of *BaseRTT.* If re-routing due to mobility occurs, the propagation characteristics of the new path and thus *BaseRTT* are likely to change resulting in substantial reduction of throughput. If the new path is shorter, *BaseRTT* will adjust automatically, but if the new path is longer, Vegas interprets this as congestion resulting in degraded throughput. However, simulations [14] in isolated static MANETs show that TCP Vegas with $\alpha = \beta = 2$ outperforms TCP NewReno and reduced the number of packet retransmissions by 99%. The desirable aspects of TCP Vegas are: It is solely a modification on the sender side and therefore can interoperate with any valid TCP implementation without modifications. It allows for incremental deployment and do not require explicit cross-layer notification or information.

3) TCP AP

TCP with Adaptive Pacing [7] is a hybrid approach that implements adaptive pacing while retaining TCP end-to-end semantics making it thus interoperable with any valid TCP implementation. The main characteristics of TCP-AP are novel congestion detection and control mechanisms. The key idea is to reduce the intra-flow interference in multihop environment and thus substantially reduce MAC layer contention for packets within the same flow at different nodes within interference range within a path. To achieve that, congestion is proactively detected by inspecting fluctuations of RTT samples. Congestion control is achieved by pacing the transmission at the sender, based on the above measure of contention and the delay until the sender can send the next packet, without interfering with the ongoing transmission of previous packets from the same flow that are already some hops away but still within interference range. This delay is calculated by measuring RTT and by knowing available bandwidth and number of hops between the sender and the receiver. Due to the hidden terminal effect, in a chain topology, a TCP sender at node i can only transmit a packet successfully as soon as node (i+3) has finished its transmission. The authors of [7] refer to the time elapsed between transmitting a TCP packet by node i and receiving the packet at node (i+4) as the 4-hop propagation delay (FHD). Two parameters that control the functionality are N and α. N is the number of recent RTT samples to take into consideration and α is the averaging weight parameter in the exponentially weighted moving average (EWMA) algorithm used to estimate the 4-hop propagation delay. As shown in [7], TCP AP outperforms TCP NewReno in static wireless multihop networks. However, it is not clear how beneficial such pacing mechanism is in an internet connected MANET where some hops traverses the Internet backbone.

III. Performance Evaluations

In order to study the performance of TCP in internet connected MANETs, we have setup 96 simulation scenarios in ns-2 [15] with different number of TCP flows and mobility scenarios each using 900 s of simulation time. We use AODV-UU as routing protocol in the half-tunneling mode. TCP packet size was 1460 Bytes and queue size of MANET nodes was set to 50 packets. In our scenarios, one or more MANET nodes are uploading file(s) to a wired host and for each mobility scenario there are 4 different traffic scenarios with different number of competing flows. Transmission range of MANET nodes was set to 250m, interference range to 550m and physical layer bandwidth to 2 mbps. We use NewReno as baseline and compare against TCP Vegas (using $\alpha=\beta=2$) and TCP AP (weighting factor $\alpha=0.7$ and history size $N=50$ [7]).

A. Chain5 Scenario

In the chain 5 scenario, five static mobile nodes (0 to four) are placed on a straight line at a distance of 200 m each. We have added a fixed node directly connected to the gateway and the capacity of the link connecting the gateway with the fixed node was set to 100 Mbps at a delay of 2 ms (simulating a broadband connection to a local LAN). The middle node number 3 plays the role of the gateway towards the internet. We create four different scenarios: one TCP flow from node 0 to node in the internet, 1 TCP flow from node 0 to internet competing with one flow from node 4 to node 0, 2 TCP flows from node 0 to the internet and from node 4 to the internet (partially overlapping in time) and simultaneous.

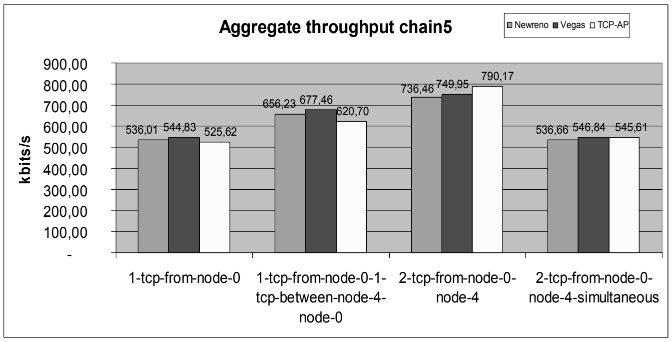

Figure 3: Aggregate throughput Chain5

As can be seen from Figure 3, the throughput is similar for all three TCP variants. However, TCP AP has a lower average throughput than the others. Vegas has a higher average throughput than TCP AP, of around 20 Kbits/s. NewReno has higher variability in throughput over time (not shown but verified from the traces). This is due to the congestion control mechanism as NewReno increases *CWnd* until packet drops occur and then reducing *CWnd*, see Figure 5, and thus throughput, which leads to bandwidth oscillations.

The aggregate throughput is in general higher for two flows than if only one flow is present. This is due to the traffic distribution as the two flows only have the gateway in common and start at different ends of the forwarding chain.

B. Chain 5+1 Scenario

In the chain 5+1 scenario (Figure 4), five static mobile nodes are placed on a straight line at a distance of 200 m each. The node in the middle runs the gateway functionality. In addition, one mobile node moves constantly from left to right and returns when reaching the end of the simulation area at a distance of 200m at speeds varying between 1.5 and 60 m/s with varying pause times at the end sides. To clarify the results of this simulation, Table 1 summarizes the speed for the sending node over time.

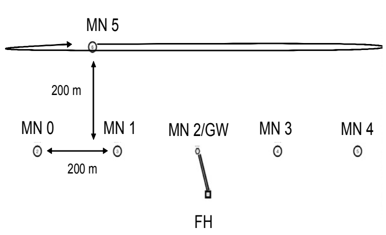

Figure 4. Chain 5+1 scenario

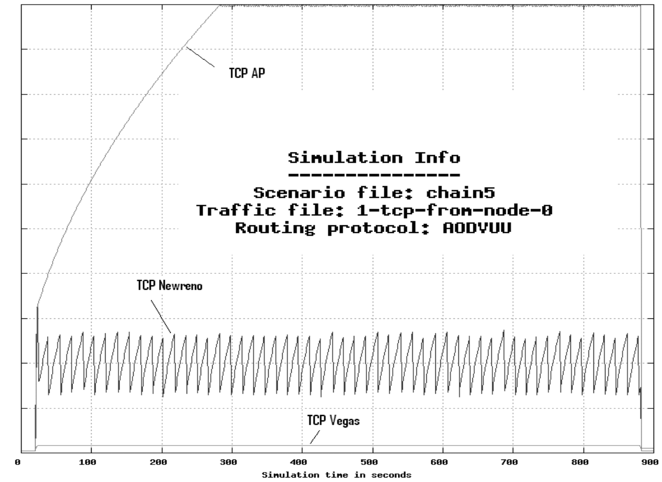

Figure 5 Cwnd in scenario Chain 5 with 1 flow

TABLE 1 SPEED OF SENDING NODE SCENARIO CHAIN 5+1

Start time (s)	Speed (m/s)	Arrival time (s)	Pause (s)
10	**20**	**50,00**	**5,00**
55	**30**	**81,67**	**8,33**
90	**10**	**170,00**	**0,00**
170	**20**	**210,00**	**10,00**
220	**5**	**380,00**	**20,00**
400	**40**	**420,00**	**10,00**
430	**60**	**443,33**	**6,67**
450	**1,5**	**983,33**	**-**

Consequently the route to the gateway changes together with the number of hops, from three to one, between the gateway and the mobile node. We have also added a fixed node directly connected to the gateway and we varied the capacity of the link connecting the gateway with the fixed node between 100 Mbps at a delay of 2 ms (simulating a broadband connection to a local LAN) and 756 Kbps at a delay of 25 ms (simulating an ADSL uplink of the gateway).

In scenario 1-tcp-from-node-5, 1 TCP flow was created from node 5 towards the fixed host (during 20 s and 880 s). In 2-tcp-from-node-0-node-5 there were 2 competing flows overlapping in time, one from node 0 (during 20 and 440 s) and one from the moving node 5 towards the fixed host (during 220 s and 880 s). In 1-tcp-between-node-5-node-0 we established one MANET only flow between nodes 5 and 0 (during 20 s and 880 s) and in 2-tcp-from-node-0-node-5-simulatneous there were 2 competing flows (during 20 s and 880 s) between node 0 and the fixed host and between node 5 and the fixed host. As can be seen in Figure 6, for every route change there is a drop of the congestion window because of lost packets and timeouts. Noteworthy is that the impact of Vegas is less, mainly because the congestion window of Vegas is much smaller then for Newreno and TCP AP.

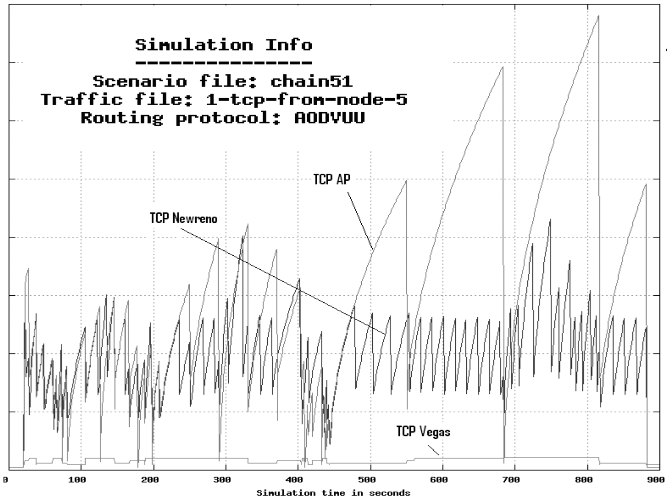

Figure 6 Cwnd Scenario chain 5+1 with 1 flow

As can be seen from Figure 7, Vegas outperforms all other TCP versions. The impact of uplink wired bandwidth and delay (indicated as 756 in the figure for the slower ADSL) on NewReno and Vegas is around 10%, whereas the performance of TCP-AP is almost reduced by 50 % if the uplink bandwidth is reduced from 100 mbps to 756 kbps. This is due to the bad estimation of 4 hop bandwidth availability as in all except one our scenarios we had at most three wireless hops followed by a fixed link from the gateway to the fixed host. Therefore TCP-AP wrongly assumes that also the last hop is a wireless hop and thus scheduled the packets for transmission accordingly leading to suboptimal performance.

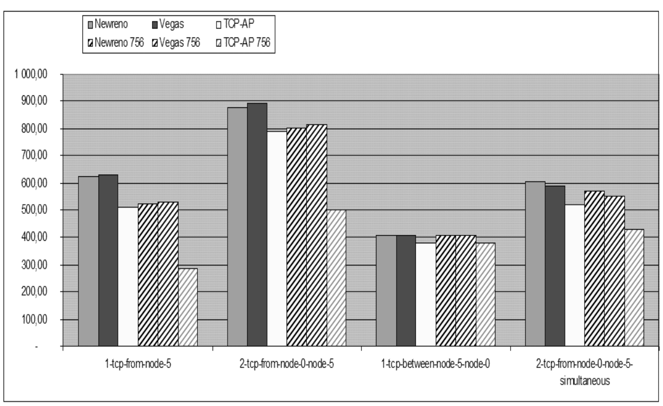

Figure 7. Throughput (kbps) Chain 5+1 - different uplink characteristics

Interestingly, the performance of the pure MANET flow is worse than the performance of the 1-tcp-from-node-5 towards the fixed host. This can be explained by looking at the number of hops. Whereas in the pure MANET flow setting the maximum number of hops is 5, in the setting where the flow between node 5 and the fixed node passes through the gateway the maximum number of wireless hops are 3. This reduces contention at the MAC layer and therefore increases throughput and goodput. Finally, the throughput where we had two competing flows simultaneously during the full duration of the simulation is even lower than for the single flow. This is due to the higher network congestion and the adverse interactions between the MAC, routing and transport layer.

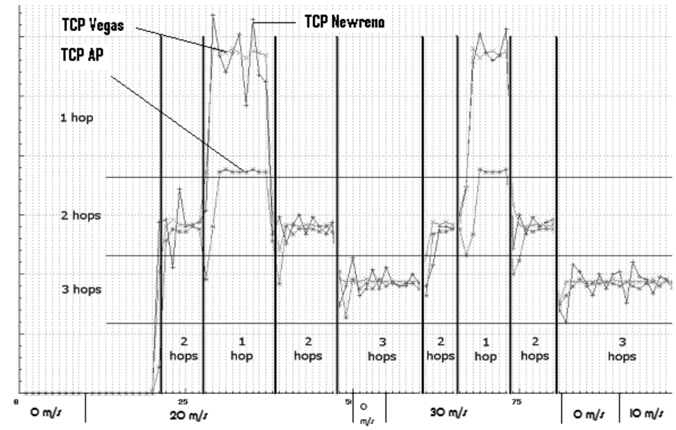

Figure 8. Goodput vs. number of hops for 1 flow from node 5 towards the fixed host over time (s)

Figure 8 shows the goodput for the flow from node 5 towards the fixed host over time (100mbps/2ms) for the first 100 second when there was no competing flow. There are also notations in figure 8 of the speed of the mobile node and the number of hops to the gateway over time. Table 2 also shows the number of hops for that node towards the gateway for TCP Vegas during the same period of time.Clearly, the higher number of hops, the lower the goodput, which is inline with the standard MANET behavior of TCP. What is interesting is that during the First 100 seconds of simulation, of which 80 seconds where used for traffic, the number of hops to the gateway was changed by AODV-UU 8 times (for Vegas and TCP AP) but 26 times for New-Reno, for the same mobility and traffic scenario. This is due to the influence of the more aggressive bandwidth probing of New-Reno that provokes packet loss as part of its congestion control algorithm. When packets cannot be delivered at the MAC-layer, it returns an error to the routing layer after it gives up re-transmitting. AODV-UU then triggers unnecessary route error messages and a re-routing procedure which results in reduced performance. In Figure 8 it is clearly shown that movement and number of hops severely affect throughput. This influence is higher than the difference between the TCP variants. The largest gap between TCP AP throughput and the others is when the gateway is only one hop away.

TABLE 2. NUMBER OF HOPS FOR MOBILE NODE 5 FOR TCP VEGAS

Time (s)	Nr of Hops	Time (s)	Nr of Hops	Time (s)	Nr of Hops	Time (s)	Nr of Hops
20.3	**2**	105.2	**2**	250.2	**2**	433.1	**2**
27.6	**1**	125.7	**1**	290.1	**1**	435.9	**1**
38.1	**2**	145.2	**2**	330.6	**2**	439.4	**2**
47.7	**3**	165.6	**3**	370.6	**3**	443.1	**3**
60.2	**2**	177.7	**2**	404.0	**2**	550.6	**2**
66.7	**1**	188.6	**1**	409.4	**1**	683.4	**1**
73.6	**2**	197.7	**2**	414.0	**2**	817.3	**2**
80.2	**3**	207.8	**3**	419.0	**3**		

This is a consequence of TCP AP´s pacing mechanism that assumes a standalone multihop environment. However, TCP AP’s goodput is more stable than NewReno’s and Vegas, because it underutilizes the network as indicated by lower throughout compared to the other simulations.

C. Random Mobility

We used a random mobility scenario using 47 mobile nodes and one gateway (at position 400x400) to evaluate performance in a more challenging environment. Simulation area was 1200x1200m. Nodes move with a speed varying between 1 and 20 m/s pausing randomly for 5 seconds and picking a new random destination. This results in an average speed of 7.36 m/s. We use the following traffic scenarios: node zero is uploading a large file to the internet, node 0 uploading file to Internet while starting to download from node 4, nodes 0 and 4 start in parallel to upload a file to the internet, node 0 starts to upload a file to the Internet and later node 4 competes with another upload. Regarding mobility, node 0 changes its routes 2194 while node 4 changes 2027 times. In total there are 54650 route and 15860 link changes during the simulation; 372 destinations can not be reached at all times.

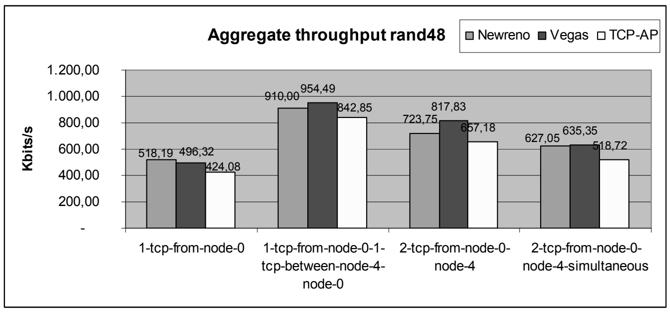

Figure 9 Aggregate throughput Random mobility Scenario

Figure 9 shows aggregate throughput in random mobility scenario, where the link between the gateway and the fixed network was simulating a broadband connection. In this simulation the difference between NewReno and TCP AP is smaller than in the other scenarios, which is due to the fact that higher fraction of lost packets is due to mobility and route breaks and not due to congestion. As in all simulations, Vegas keeps a low and stable congestion window in all scenarios (average congestion window size for 1-tcp-from-node-0 was 3.39, for second scenario 3.08, for third scenario 3.24 and for forth scenario it was 2.75). This is also beneficial as it reduces the number of packets in flight and thus less packets are competing within interference range for MAC layer transmission.

It is interesting to note that in the scenario where nodes 0 and 4 start in parallel to upload a file to the internet, the performance is better than in the Chain5+1 scenario. This is due to the reason that those two flows do not interfere so much as in Chain5+1 (where they mostly share a common path). Also, nodes might have a direct connection to the gateway sometimes due to their random mobility, whereas this is not the case in the Chain5+1 scenario. With only one flow the throughput is as expected lower than in the chain5+1 scenario due to more frequent route changes and packet loss.

IV. CHANGING THE ROUTING PROTOCOL

We used several simulations to determine the dependence of TCP performance on the routing protocol using DSDV [16] and compared results with AODVUU.

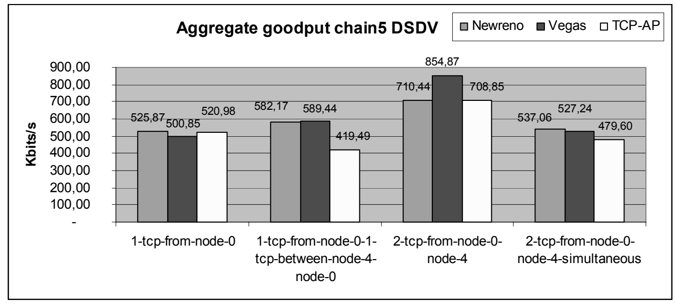

Figure 10 Aggregate goodput Chain 5 DSDV

Aggreagte goodput for DSDV is generally lower compared to AODVUU showing similar trend compairing TCP variants. TCP Vegas shows slightly better aggregate goodput in most cases except in the scenario where 1 flow has been used. Compared to the values for AODVUU, DSDV values are in average of all traffic scenarios roughly 4 % lower for Newreno, 2 % lower for Vegas and 13 % lower for TCP AP. The difference between the TCP variants are however more or less consistent with the values for AODVUU, with a slight advantage for Newreno except in the traffic scenario with two flows see Figure 10 where Vegas outperforms the others.

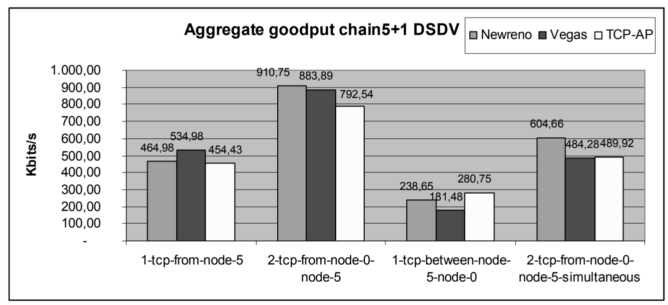

Figure 11 Aggregate goodput Chain5+1 DSDV

Figure shows goodput for DSDV and as can be seen from Figure , using DSDV instead of AODVUU different variants perform better confirming the dependency of TCP throughput on the routing protocol for the same traffic scenarios. Vegas performance suffers more from longer hops and more competing flows then the other variants when DSVD was used. Newreno actually increses performance in traffic scenarios with 2 flows. The movement is also making a difference, as the the goodput values for DSDV is beginning to decline in almost all traffic scenarios in comparison to AODVUU under mobility of nodes.

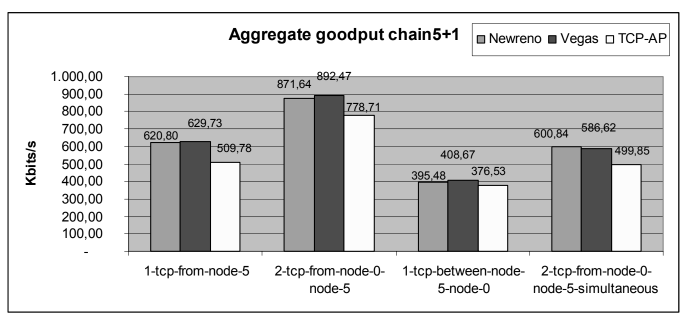

Figure 12 Aggregate goodput Chain5+1 AODVUU

The goodput for using DSDV is in average around 12 % lower for Newreno, 21 % lower for Vegas and around 4 % lower for

TCP AP then when AODVUU was used. The benefit that Vegas had of its modified congestion detection, which reduced the amount of traffic and false route updates compared to Newreno, is decreasing. More and more route updates are not caused by the transport layer and Newreno seams to benefit from being able to increase the sending rate faster then the other TCP variants. Simulations not shown here, further emphasize that if DSDV is used the performance of Vegas is very dependent of the traffic scenario used.

V. CONCLUSIONS

In this paper we have looked in to several aspects of TCP performance in an internet connected MANET using a simulation based approach. In our scenario, mobile nodes use multi hop paths to connect to the public Internet. This vision is interesting for 4G operators as it allows extending hotspot coverage. Providing satisfactory TCP performance is very important as TCP is the de-facto standard for reliable data delivery in the Internet. This will be even more important in future wireless meshed networks, which will provide a true wireless internet part being able to rapidly deploy community or urban networks. We compared performance of different TCP variants, namely NewReno, Vegas and TCP-AP based on AODV-UU as routing protocol in the MANET. Internet connectivity was achieved through half-tunneling towards the gateway.

The key findings are that it is beneficial to use TCP-Vegas as it keeps the congestion window low, a key enabler for high throughput in multi-hop environment with intra-flow interference. This reduces bad interactions with the routing layer leading to more stable network behavior when AODVUU is used. When DSDV is used as routing protocol, one can argue that there is an advantage of using TCP Newreno as TCP Vegas has been shown in our simulations to experience performance drops when the traffic was increased and the route consisted of more hops. Table 3 shows a taxonomy of the TCP variants used in the TCP comparison, the remarks are based on the simulation of a Internet connected MANET and may be different under other environments.

TABLE 3. TAXONOMY OF TCP VARIANTS

	TCP NewReno	TCP Vegas	TCP AP
General	Yes	Yes	No (ad hoc)
Crosslayer information	No	No	Yes (on the source node, nr of hops between sender and receiver)
Network influence	Severe	Mild	Mild
Performance	Medium	Medium	Low
Long term Fairness	Close to none	Low	Medium
Hop influence	High	High	Less high (for long routes)

As future work, we identify many different areas. An interesting idea is to make the network and link layer in the MANET more TCP friendly by hiding as much the MANET specific problems towards TCP. More simulations runs are required to find out the exact reason why Vegas performance suffers under certain scenarios when DSDV is used. Also the impact of different packet scheduling techniques to increase fairness is one option that needs to be explored further. Another direction is to work more on adaptive strategies for ACK treatment within the network, which is crucial for TCP performance. Even if network and link layer could be made fairer to TCP another issue is network asymmetry, which naturally arises in a multi hop environment. Therefore, strategies to cope with different path characteristics need to be researched for TCP in such environment.

REFERENCES

[1] Jun-Zhao Sun. Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing. In: *Proceedings of International Conferences on Infotech & Infonet*, Beijing, China, C:316 – 321

[2] IST-Daidalos project homepage at http://ist-daidalos.org

[3] Kitae Nahm, Ahmed Helmy, C.-C. Jay Kuo. TCP over Multihop 802.11 Networks: Issues and Performance Enhancement. In *Proceedings of ACM MobiHoc*, Urbana-Champaign, IL, May, 2005.

[4] Ahmad Al Hanbali, Eitan Altman, Philippe Nain. A survey of TCP over Mobile Ad Hoc Networks. *INRIA Technical Report Nr. 5182*, May 2004.

[5] Erik Nordström, Per Gunningberg, Christian Tschudin. Comparison of Forwarding Strategies in Internet Connected MANETs. In *Mobile Computing Communications Review*, Volume 1, Number 2, 2004.

[6] L.S. Brakmo, L.L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. In *Journal on Selected Areas in Communications*, Volume: 13, Issue: 8, pp. 1465-1480, 1999.

[7] S.M. ElRakabawy, A. Klemm, C. Lindemann. TCP with Adaptive Pacing for Multihop Wireless Networks. In: *Proceedings of the 6th ACM international symposium on Mobile Ad Hoc Networking and Computing*, pp. 288 – 299. 2005.

[8] Pedro M. Ruiz, Francisco J. Ros, Antonio F. Gomez Skarmeta. Internet Connectivity for Mobile Ad Hoc Networks: Solutions and Challenges. In *IEEE Communications Magazine*, vol.43, no.10 pp. 118-125.

[9] X. Kaixin, B. Sang, L. Sungwook, M. Gerla. TCP Behavior across Multihop Wireless Networks and the Wired Internet. In *Proceedings of WoWMoM'02*, New York. ACM press, 2002.

[10] Luqing Yang, Winston K.G. Seah, Qinghe Yin. Improving Fairness among TCP Flows crossing Wireless Ad Hoc and Wired Networks. In *Proceedings of ACM MobiHoc 2003*, Annapolis, USA.

[11] S. Floyd, T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm, *RFC 2582*, April 1999.

[12] S. Low, L. Peterson, L. Wang. Understanding TCP Vegas: A Duality Model. In *Journal of the ACM,* New York: ACM press, 2002. Volume 49, Issue 2, pp. 207 – 235.

[13] O. Ait Hellal, E. Altman. Analysis of TCP Vegas and TCP Reno. In *Telecommunication Systems*. Volume 15, Numbers 3-4, pp. 381 – 404.

[14] S.M. ElRakabawy, C. Lindemann, M.K. Vernon. Improving TCP Performance for Multihop Wireless Networks. In *Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pp. 684 – 693.

[15] D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Ya, and H. Yu. Advances in Network Simulation. In *Computer*, 33(5):59-67, May 2000.

[16] C.E. Perkins, P. Bhagwat, Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, ACM SIGCOMM'94. London:University College, 1995. pp. 234-244.