European Collaborative Innovation Centres
for Broadband Media Service

Place for logotype

Tactical Ad Hoc Networks
Case Study Final Report

EUROPEAN UNION
European Regional
Development Fund

Investing in the future by working together for a sustainable and competitive region

The Interreg IVB
North Sea Region
Programme

Abstract - Tactical networks are used in military and rescue operations to provide timely and accurate information to operating teams. One of the most common use cases for tactical networks are short text or voice messages. Most of the research has therefore been focused on UDP and VOIP. Due to the increasing demand for e.g. digital maps, there are increasingly important to support time elastic traffic, i.e. TCP. Most of the research on MANETs has also focused on Carrier Sense Multiple Access (CSMA) networks, i.e. IEEE 802.11 based. In tactical networks where Quality of Service and bounded delays are crucial, Time Division Multiple Access (TDMA) can provide better service than CSMA. Traditionally, tactical networks require a fixed or semi-fixed communication infrastructure, which can be difficult to supply in an emergency scenario. As an alternative that can work without a infrastructure, is Mobile AdHoc Networks (MANETs). In MANETs a importent consideration is the routing algoritm. The purpose of this report is to assess and compare the performance of three state-of-the-art TCP versions and two routing protocols over TDMA based MANETs.

I. INTRODUCTION

During a disaster or a military campaign it can be anticipated that the regular wired infrastructure is at best semi-functional. Therefore, relying on solely an infrastructure-based system is in most cases not possible. Prime examples of this are disaster areas, e.g. earthquakes, tsunamis and nuclear disasters, which can render an infrastructure unusable. Therefore a tactical network should be able to be resilient to node failures and operate with little or no backbone infrastructure.

One network type that promises to do this is Mobile Ad Hoc Networks or MANETs. In a MANET, a node functions both as a host and as a router and can automatically organize and form a network. However, due to a limited transmission range, intermediate nodes need to forward packets to form a connected network topology. This creates contention and multiple points of failure. Node mobility also leads to a network topology that is highly dynamic and prone to frequent changes and errors. Due to this dynamicity, routing protocols which work well in fixed networks do not show the same performance in MANETs. Two state-of-the art routing protocols for MANETs, that we evaluate in this paper, are AODV and OLSR. AODV is a reactive routing protocol that determines routes only when needed [1]. OLSR is a table driven proactive protocol that regularly exchanges topology information with its neighbors [2].

Although voice and short text messages are the main applications for tactical networks it is also important to support standard applications used on todays Internet, to e.g. provide maps or other content. The obvious solution to support standard applications would be to treat MANETs as a general Internet system and use standard TCP/IP. However, TCP was designed as a reliable end-to-end connection-oriented protocol for data delivery over somewhat unreliable wired networks. In a MANET, which have a substantially higher packet loss rate and jitter compared with a wired network, the performance of TCP dramatically degrades.

Research have mainly focused on CSMA based MANETs, i.e. using IEEE 802.11 radio cards. However, using a TDMA scheme has several advantages, e.g. QoS, bounded delays and a stable network under heavy traffic loads. The purpose of this paper is to assess and analyze the problems and weaknesses three different TCP versions in a TDMA based MANET using state-of-the-art reactive and proactive routing protocol. The rest of the paper is structured as follows. Section II describes the related work and motivation for this paper. Section III describes the experiment setup and the motivation for the chosen topologies. Finally section IV and V describe the experiment results and our conclusions.

II. BACKGROUND

MANETs are wireless networks which do not require any infrastructure for its operation. Intermediate nodes must therefore participate in the route discovery and packet forwarding to other nodes. As nodes are mobile, the network topology and link status is constantly changing. This makes routing a challenging task and routing protocols needs to quickly respond to topology changes.

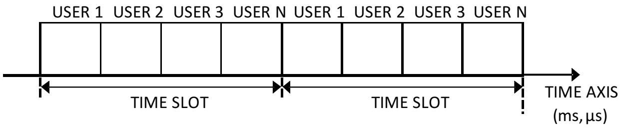

Fig. 1. TDMA CHannel access Figure 7

Routing protocols in MANETs are commonly divided into two categories based on how and when routes are discovered and maintained: proactive and reactive routing protocols. Reactive or on-demand routing protocols, establish the route to a destination only when there is a demand for it. This ensures a low routing overhead when there is little traffic. Although certain caching mechanism normally is implemented, the routing protocol need to do route lookup on all new transmissions [1]. This can lead to a increase of routing overhead in high traffic conditions. A proactive routing protocol on the other hand, maintain a consistent and updated view of network by periodically propagating route updates in the network [2]. This is costly when there is limited amount of traffic. The routing overhead of a proactive approach is not influenced by the traffic volume, i.e. the overhead low when the traffic volume is large. Therefore, even if a proactive routing protocol might have a slightly higher average overhead, it can in high traffic situations maintain a stable(r) network than a reactive routing protocol. In this paper we have chosen two state-of-art routing protocols to represent these two categories, proactive OLSR and reactive AODV.

Another important consideration in MANETs are the Media Access Control (MAC) layer. The MAC layer emulates a full-duplex logical communication channel for the upper (network) layer and is responsible for terminal addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multi-point network.

A TDMA channel access method allows several stations to share the same medium by dividing real time into different time slots, see Figure 1.

The nodes transmit in rapid succession one after the other, each using its own time slot without interfering with its neighbors' transmissions. By its design TDMA has a number of advantages over contention-base approaches, such as:

- Fairness a node cannot suffocate other nodes.
- Bounded delays making throughput more predictable and facilitating Quality of Service (QoS).
- Asymptotic behavior under heavy traffic loads the networks remains stable.

Due to the requirements of fairness and bounded delays in tactical networks we in this paper have chosen to focus on a TDMA based MAC layer.

The critical component of a TDMA protocol in MANET is to assigning different time slots to any two conflicting nodes in a distributed way. The slot assignment problem is a direct

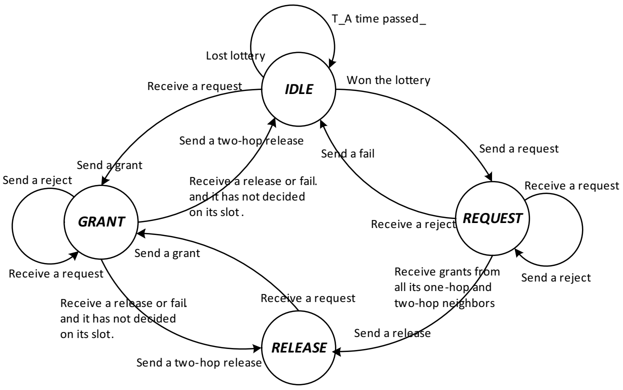

Fig. 2. DRAND State Diagram

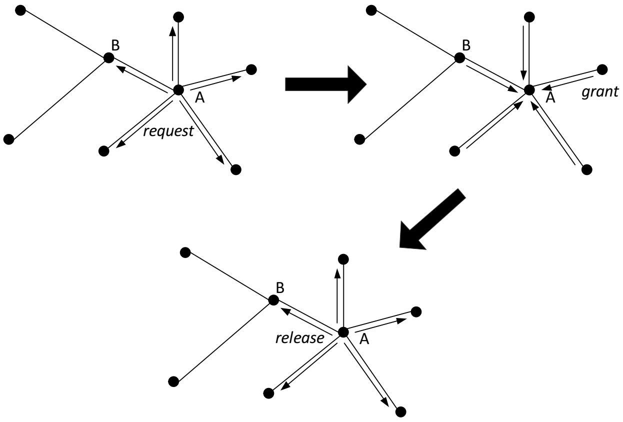

Fig. 3. DRAND successful round

extension of the graph coloring problem in graph theory. Let the network be represented by a graph $G = (V, E)$ where V is the set of nodes (or vertexes), and E is the set of edges. An edge $e = (u, v)$ exists if and only if u and v are in V and nodes u and v can hear each other. The problem is to find the minimum number of colors (timeslots) and assign them to the vertices of the graph in such a way that no two adjacent nodes have the same color (timeslot). As the graph scales bigger and more complex, the time it takes to calculate the coloring increases dramatically. One solutions that we evaluate in this paper is DRAND [3]. DRAND was introduces in MANETs as means of providing a better interoperability between radio transmitting nodes, to minimize collisions and promote bounded delay for the purpose of real-time communication (e.g. voice communication). DRAND differs from the traditional graph coloring problem in that it performs distance-2 coloring i.e. it assumes that any two interfering nodes are at most two hops way from each other. Although this scheme require more slots than distance1 coloring, it is motivated by the fact that the interference range is usually larger than the communication range, hence nodes may not be able to communicate directly, but they can still influence each other's reception. On of the benefits of DRAND is does not require each node to have synchronized clocks. In the following we will give an overview of DRAND, for a detailed description see [3]

There are four states that a node maintains: IDLE, REQUEST, GRANT, and RELEASE. Figure 2 shows the state diagram for the DRAND implementation.

To be assigned a timeslot a node plays in a lottery. The chance for node to win the lottery is inversely proportional to the number of one hop and two hop neighbors, who have not yet decided on a timeslot. If a node wins the lottery it moves to the Request state, in which it broadcasts a request message to its one hop neighbors. When a certain neighbor receives the request message from the initial node, it send back a grant message and moves to the Grant state, but only if it has been in the Idle or Release state. The neighbor can be in any of these two states if:

- None of its neighbors has sent a request so far.
- It hasn't sent a grant message to anyone yet.
- It has received a fail or release message from another node.

With the grant message, the neighboring node sends back a list of the occupied timeslots by its one-hop neighbors, which is provided by their release messages. If the same neighboring node is already in the Request or Grant state, it sends back a reject message to the request originator. When the original node receives the reject message, it broadcasts a fail message to all of its one hop neighbors. When the neighbor receives a fail from a node whose request turned the neighbors current state to GRANT, the same neighbor returns to the Idle state if it has not decided on its slot already or to the RELEASE state if it has decided on its timeslot. If a node does not receive any grant or reject messages from a one hop neighbor within a predefined time, it retransmits request to those nodes that it has not received a grant from. When a node receives a request from another node for which it has already sent a reject, then it retransmits the reject to that node. Whenever a node receives a grant or reject message, it estimates the message delay by taking differences of time stamps of request and the received message. If the new estimate is larger than the current value, the value is updated to the new estimate. As a node receives a grant from its entire one hop neighbors, it decides on its time slot to be the minimum of the time slots that have not been taken by its two hop neighbors before this round. Then the node enters the Release state and broadcasts a release message containing information about its selected time slot to its one hop neighbors. Figure 3 illustrates a successful round.

DRAND handles packet loss via retransmission. To handle sever packet loss or node failing, the algorithm offers nodes the possibility of "giving up". Since only requests and grants require any response, when a node does not receive any response to these messages from a one hop neighbor for a fixed number of retries, then it removes the neighbor from its neighbor list. This allows the node to make progress with a response from the removed neighbor. Because of its inheritably probabilitybased nature, DRAND requires time to adapt with topology changes and are expected to perform worse when the node mobility increases.

When evaluating a MANET using simulation there is a need to have models for the nodes motion pattern. Some application areas such as Vehicular Ad Hoc Networks, battlefield communication and urban mesh networks have received significant attention in generation of realistic models. For example on a battlefield, the movement of the soldiers/nodes will be influenced by the commander. In an urban, the node movement is restricted by obstacles like buildings. There are various mobility models trying to cater for these situations such as Random Waypoint (RWP), Reference Point Group Mobility Model (RPGM), Manhattan Mobility Model, Freeway Mobility Model, Gauss Markov Mobility Model [4][5]. In the investigated scenarios communication are often among groups which tend to coordinate their movements, i.e. a rescue team. We have therefore focused on using RPGM.

Due to the node mobility there will always be a certain amount of indirect packet loss in a MANET due to route breaks. However, TCP assumes that all packet losses are due to congestion of the network infrastructure. Another aspect of TCP performance is the retransmission timer (RTO). It keeps track of how long TCP wait for an acknowledgement and when TCP should retransmit. The RTO value is doubled each time the RTO expires. With route re-computations and link layer retransmission causing jitter, this timer can be inflated leading to a long delay before a packet loss is detected which makes TCP performance worse.

A great amount of research has been invested in dealing with TCPs performance issues in MANETs, e.g. [6] [7]. Most proposals are based on the idea of changing the functionality and/or the behavior of TCP to adapt it to the new network environment. In this paper we focused on three TCP variants that represent three different design choices; baseline, general but tailored to better sustain packet loss and a cross-layer approach designed for wireless networks. TCP New Reno [8] is one of the most common TCP variants. We have therefore chosen this to be our baseline for comparison.

The biggest difference between TCP Reno and New Reno is the Fast Recovery algorithm that begins when three duplicate acknowledgments are received and ends when either a retransmission timeout occurs or an acknowledgment arrives that acknowledges all of the data in the transmission windows instance before the Fast Recovery procedure began. During Fast Recovery the Congestion Avoidance takes place instead of the Slow Start, as it is with Reno, but for every duplicate acknowledgment that arrives to the sender, a new unsent packet is transmitted. The purpose of this action is to keep the transmit window full. When an acknowledgement (ACK) arrives, confirming only part of the packets in the congestion window, New Reno assumes this ACK points to a loss hole in the sequence space and a new packet beyond the confirmed sequence number is resend. This allows New Reno to fill holes in the sequencing space while effectively maintaining the high throughput.

TCP Westwood+ [9] is a sendersideonly modification of TCP New Reno that is intended to better handle large bandwidth delay product paths (large pipes) with potential packet loss due to transmission or other errors. TCP Westwood+ relies on monitoring the ACK stream for information to help it better set the congestion control parameters, i.e. Slow Start Threshold (ssthresh) and Congestion Window (cwnd). In TCP Westwood+, an ”Eligible Rate” is estimated and used by the sender to update ssthresh and cwnd upon loss indication. Significant efficiency gains can be obtained for both wireless and wired networks, while maintaining fairness.

TCP ELFN (Explicit Link Failure Notification) [10] is a cross layer approach between the routing and the transport (TCP) layer. This interaction aims to inform the TCP agent about route failures when they occur. The proposal uses an ELFN message, which is transported by or piggy backed on routing messages to the sender. The ELFN message is essentially similar to ICMP [11] Destination Unreachable - Source Route Failed message, which contains the sender, receiver addresses and ports, as well as the TCP packet sequence number. On receiving the ELFN message, the source responds by disabling its retransmission timers and enters a ”frozen” state. During the ”frozen” period, the TCP sender probes the network to check if the route is restored. When the route is restored, i.e. the sender starts to see acknowledgment of the probe packets, the TCP sender leaves the ”frozen” state and resumes its state as before the freeze event. The probe interval is a crucial parameter and in the original paper the authors propose to use a probe interval of 2 sec. Upon route restoration, the proposals use the values of RTO and cwnd prior to the route failure. In the original paper the authors used DSR [12], which is a reactive routing protocol. In this paper we are evaluating TCP ELFN with both proactive and reactive routing protocol.

To the best of our knowledge this is the first paper that evaluate the performance of three well established TCP versions using both reactive and proactive routing protocols in a TDMA based network.

III. EXPERIMENTS

To compare TCP performance, we used the ns 2.26 simulator [13], with TCP-NewReno, -ELFN and -Westwood+. The routing protocols are slightly modified versions of OLSR and AODV as described in next section.

The original ELFN was designed and tested with DSR as a routing protocol. In short, whenever DSR detects a route break, it generates a Route Error message which traverses back to the traffic source. With the ELFN modification, DSR generated an additional ICMP (Destination Unreachable Source route failed) message to the traffic source and intercepted by the TCP agent which reacts adequately. For the purpose of the current study we modified AODV so that when it detects a route break and generates a Route Error message, it is preceded by an ICMP message which is only sent back to the traffic source.

When using the pro-active OLSR, the ELFN operation becomes much more peculiar. With OLSR, a node regardless of whether it is participating in a data forwarding process is notified about any topology change (route break). The implementation has been changed to reflect this by removing the explicit ICMP message from the node that detects the route break. Instead, whenever the OLSR agent at the traffic source detects a route change/break to the destination it sends an ELFN message to itself.

The implementation of DRAND that we used is the implementation done by the Computer Science department of the North Carolina State University, United States. It is important to stress the choice of using DRAND, is not for optimality to the evaluated scenarios. The choice should rather be seen as one viable way of implementing a TDMA scheme that works both for stationary and mobile networks.

The following parameters describe the full specification of the experiment scenarios:

- Simulation area is 1000 meters by 1000 meters area without any obstacles or heterogeneity.
  - Radio communication distance is 150 meters and carrier sensing distance is 300 meters.
  - We used two node densities, consisting of 3 and 16 groups of 6 nodes thus forming scenarios with 18 and 96 nodes.
  - Group radius is set to 250 meters to ensure that innergroup communication is forwarded by at least one intermediate node.
- We used the RPGM mobility model
  - Mobility speed is random in the range of 1,5 - 5,0 m/s.
- Simulation time is 600 seconds and traffic flows start at 60 sec, giving time for the routing protocols to converge.
- Physical layer speed is set to 1 Mbps.
- In each scenario there are two independent TCP flows between two different node pairs.
- Each experiment is performed both with and without background traffic.
  - The background traffic was sent between 10% of the nodes in a scenario.
  - Each flow was a 20 kbit/s UDP flows of consisting of 100 byte packets. Making the total traffic volume between 20 kbit/sec (18 nodes scenario) and 180kbit/s (96 nodes secenario).

IV. RESULTS/ANALYSIS

In this section we will describe the results from our experiments. In each of the presented results, we show the aggregate average TCP throughput using either OLSR or AODV. Throughput is marked on the y-axis and is measured in kilobits per seconds (Kbit/s).

The results showed in Figure 4 is from a scenario with 18 nodes in 3 groups and with no background traffic. The network density is low which causes occasional route breaks and network divisions between nodes and groups. However, due to the low node density, the routing overhead and contention in this scenario is also low giving the possibility to achieve a high TCP throughput. However, both TCP NewReno and TCP Westwood+ cannot utilize this possibility as the constant route breaks limits the contention window. Therefore both of the approaches have similar performance. TCP ELFN on the other hand freezes its contention window during the route breaks and can therefore maintain a high contention window. Due to the low contention the impact of the routing layer is limited.

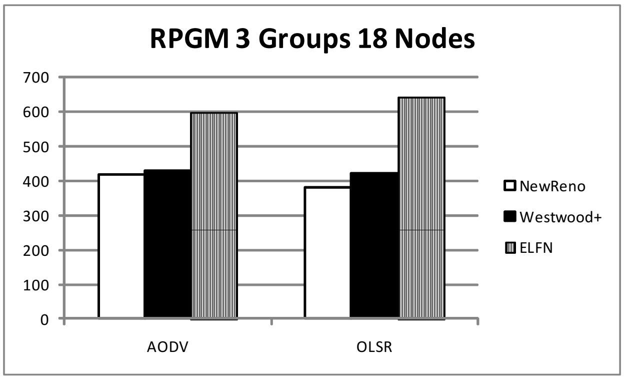

Fig. 4. Low node density with no backgound traffic

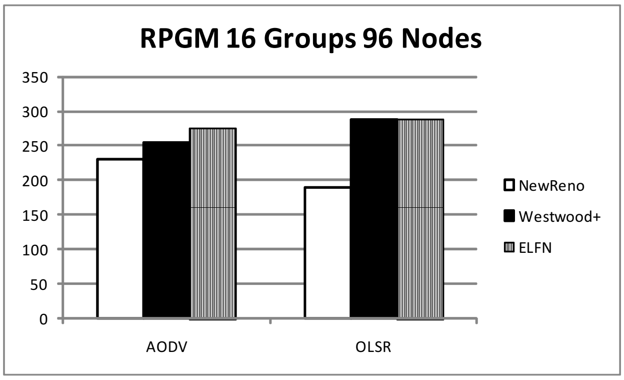

Fig. 5. High node density with no background traffic

Figure 5 refers to a high node density scenario with a 96 nodes divided on 16 groups. In this scenario the connectivity is high but with an increased stress on both routing layer and TDMA scheduling. Due to this the overall throughput is less. This leads to a smaller avergae contention window for all TCP versions and consequently TCP ELFN has less benefit of freezing the contention window. What also can be seen is that in this scenario the routing layer starts to impact the results. The slightly lower overhead of AODV improves TCP NewRenos performance compared to using OLSR. On the other hand due to the higher amount of delay jitter when using AODV, TCP Westwood+ have difficulties to determine the available bandwidth.

Figure 6 shows results for a 18 node density scenario with a low amount of background traffic. Once again, with low contention, TCP ELFN outperfomes both TCP Westwood+ and TCP NewReno.

The last Figure 7 refers to a scenario with a high amount of background traffic and 96 nodes. In this scenario the effect of the background traffic is clearly seen. All TCP proposals

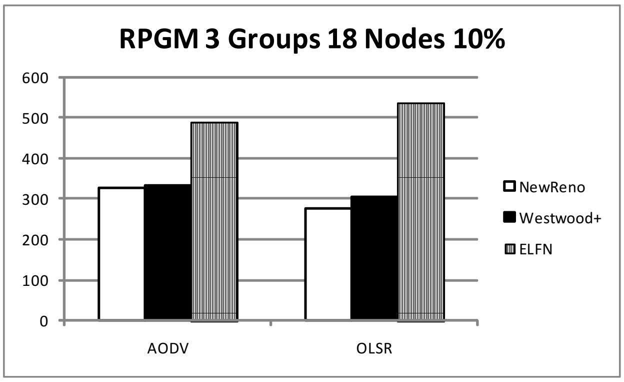

Fig. 6. Low node density with low amount of backgound traffic

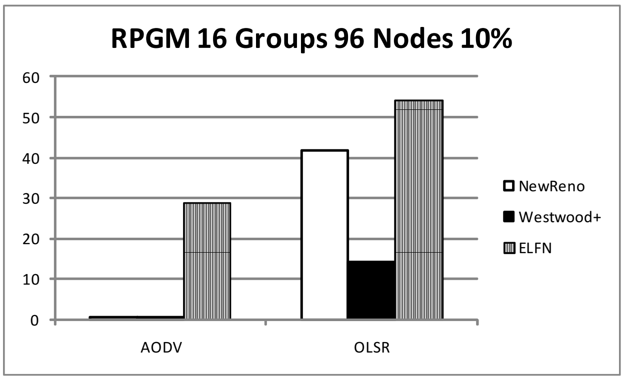

Fig. 7. High node density with a hi amount of backgound traffic

have very low throughput. The high amount of contention also leads to a high amount of jitter, which reduces TCP Westwood+ performance compared to TCP NewReno. The performance gains shown by TCP ELFN in this scenario is harder to analyze but is most likely a combination of the freeze mechanism maintaining a higher contention window and the reduced amount of stress on the network during the probing phase. However, when comparing the TCP proposals, although the relative difference is large in percentage, in real numbers the difference between the TCP proposals are relatively small.

Nevertheless, this is clearly a scenario where it is beneficial with a proactive routing approach as with OLSR all TCP protocols achieve higher throughput.

V. CONCLUSION

Despite the large volume of research activities and rapid progress made in the MANET technologies in the past few years, almost all research areas still impose many open issues.

In this paper we have investigated the TCP performance of three TCP versions in a tactical MANET using both OLSR and AODV as routing protocols. TCP ELFN achieves the overall highest throughput. The gains are however reduced in low throughput scenarios as the benefit of freezing the contention window is less. We showed that the impact of the routing protocols is dependent on which TCP version is used. TCP NewReno performed slightly better with AODV in low traffic scenarios. While in high traffic scenarios or combined with TCP Westwod+ the proactive OLSR protocol had a clear advantage. The performance of TCP ELFN was similar regardless of which routing protocol was used.

REFERENCES

[1] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “Ad hoc on-demand distance vector (aodv) routing,” Published Online, Internet Engineering Task Force, RFC Experimental 3561, July 2003. [Online]. Available: http://rfc.net/rfc3561.txt

[2] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR) RFC 3626,” *Acessado em*, vol. 10, 2009.

[3] I. Rhee, A. Warrier, J. Min, and L. Xu, “Drand: distributed randomized tdma scheduling for wireless ad-hoc networks,” in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc ’06. New York, NY, USA: ACM, 2006, pp. 190–201. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132927

[4] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[5] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, “A survey on mobility models for performance analysis in tactical mobile networks,” *Journal of Telecommunications and Information Technology*, vol. 2, pp. 54–61, 2008.

[6] F. Wang and Y. Zhang, “Improving tcp performance over mobile ad-hoc networks with out-of-order detection and response,” in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2002, pp. 217–225.

[7] K. Chen, Y. Xue, and K. Nahrstedt, “On setting tcp’s congestion window limit in mobile ad hoc networks,” in *Communications, 2003. ICC’03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1080–1084.

[8] S. Floyd, T. Henderson, and A. Gurtov, “RFC 3782-The NewReno modification to TCP’s fast recovery algorithm,” IETF, 2004.

[9] L. Grieco and S. Mascolo, “Tcp westwood and easy red to improve fairness in high-speed networks,” in *Protocols for High Speed Networks*, ser. Lecture Notes in Computer Science, G. Carle and M. Zitterbart, Eds. Springer Berlin / Heidelberg, 2002, vol. 2334, pp. 130–146.

[10] G. Holland and N. Vaidya, “Analysis of tcp performance over mobile ad hoc networks,” *Wireless Networks*, vol. 8, pp. 275–288, 2002, 10.1023/A:1013798127590. [Online]. Available: http://dx.doi.org/10.1023/A:1013798127590

[11] J. Postel, “Rfc 792: Internet control message protocol.” 1981.

[12] D. Johnson, Y. Hu, and D. Maltz, “Rfc 4728: The dynamic source routing protocol for mobile ad hoc networks for ipv4,” 2007.

[13] S. McCanne, S. Floyd, K. Fall, K. Varadhan *et al.*, “Network simulator ns-2,” *The Vint project, available for download at http://www.isi.edu/nsnam/ns*.