
TCP over Geo-routing for High Mobility: Vehicle Grids and Airborne Swarms

Jiwei Chen, Mario Gerla and Yeng-Zhong Lee

University of California, Los Angeles, CA 90095, USA
cjw@ee.ucla.edu, {gerla,yenglee}@cs.ucla.edu

Summary. Ad hoc wireless networks have become the architecture of choice for peer to peer communications in areas where the telecommunications infrastructure is inadequate or has failed. A major challenge is the reliable delivery of data when nodes move. The reliable Internet protocol is TCP. However, TCP performs poorly in mobile ad hoc networks, mainly because of route breakage. To overcome this problem, a robust routing protocol must be used. To this effect, Geo-routing has recently received attention in large scale, mobile systems as it does not require end-to-end path establishment and pre-computed packet forwarding routing structure at nodes. These properties make Geo-routing robust to highly dynamic route changes. For best performance, however, several parameters must be carefully tuned.

In this paper we study the joint optimization of TCP and Geo-routing parameters to handle high speeds. We first introduce two highly mobile ad hoc scenarios that require reliable delivery, namely the vehicle urban grid and the airborne swarms. Then, we study the impact of critical system parameters (e.g., hello message exchange rate, delay timer in TCP for out-of-order delivery, etc) on the performance of both UDP and TCP. We improve hello message efficiency in Geo-routing by using an adaptive hello exchange scheme. Then, we fix the out-of-order problem in TCP by using a receiver-side out-of-order detection and delayed ack strategy. We show that these parameter adjustments are critical for efficient TCP over Geo-routing in highly mobile applications. With these enhancements our TCP with Geo-routing solution easily outperforms TCP over traditional ad hoc routing schemes, such as AODV.

1 Introduction

Transport Control Protocol (TCP) is unquestionably one of the most widely used protocols in the Internet. TCP was originally designed for a wired network where congestion and buffer overflow account for most packet losses. Unlike wired networks, TCP performance in wireless ad hoc networks is affected by several new factors. These factors include unpredictable channels error, medium contention which may lead to capture, and frequent route breakage

due to node mobility. All these factors challenge TCP to provide efficient and reliable end-to-end communications in mobile ad hoc networks.

Two recent developments have stimulated great interest for TCP in highly mobile scenarios: car to car communications, file sharing and content distribution during highway driving; and tactical communications between Apache helicopters and UAVs (manned and unmanned). The latter “airborne” situation is common in battlefield, homeland defense and search and rescue scenarios. In these scenarios, mobility can be characterized by the inverse of “contact time” with neighbors. The lower the contact time, the more challenging the routing. For example, an aircraft flying at 360Km/hr with transmission range of 300m will maintain contact with neighbors flying in random directions for an interval of the order of seconds. Thus, neighbor acquisition and route computation must be completed well below the second. Notice that “contact time” is proportional to the ratio of transmission range over speed, thus, what counts is the “relative” speed. For instance, people walking in a shopping mall (at 3m/s) with Bluetooth or Zigbee radios with 10 m range, say, would also lead to “high mobility” ad hoc net scenarios.

To overcome the problems posed by vehicular and aircraft speeds, a robust routing protocol must be used. To this effect, Geo-routing has recently received attention in large scale, mobile systems as it does not require end-to-end path establishment and pre-computed packet forwarding routing structure at nodes. These properties make Geo-routing robust to highly dynamic route changes. For best performance, however, several parameters must be carefully tuned.

In this paper we study the joint optimization of TCP and Geo-routing parameters to handle high speeds. We first introduce two highly mobile ad hoc scenarios that require reliable delivery, namely the vehicle urban grid and the airborne swarms. Then, we study the impact of critical system parameters (e.g., hello message exchange rate, delay timer in TCP for out-of-order delivery, etc) on the performance of both UDP and TCP. We improve hello message efficiency in Geo-routing by using an adaptive hello exchange scheme. We also fix the out-of-order problem in TCP by using a receiver-side out-of-order detection and delayed ack strategy. We show that these parameter adjustments are critical for efficient TCP over Geo-routing in highly mobile applications.

The remainder of this paper is organized as follows. Section 2 describes the urban vehicle ad hoc wireless environment. Next, Section 3 describes the use of clusters of small unmanned aircrafts for search, rescue and scouting applications. Section 4 reviews prior work on TCP for mobile ad hoc scenarios. Following this, Section 5.1 presents an impact study of various Geo-routing parameters and comparison of UDP and TCP over Geo-routing versus traditional ad hoc routing in a carefully controlled, deterministic motion scenario. An adaptive hello exchange scheme for GPSR is introduced. In Section 5.2 we consider the more general scenario with random motion. With random motion we show that out-of-order detection and recovery is required for efficient TCP performance. We conclude our work in Section 6.

2 The Vehicle Grid

2.1 Background

Safe navigation support through wireless car to car and car to curb communications has become an important priority for Car Manufacturers as well as Municipal Transportation Authorities and Communications Standards Organizations. New standards are emerging for car to car communications (DSRC and more recently IEEE 802.11p). There have been several well publicized testbeds aimed at demonstrating the feasibility and effectiveness of car to car communication safety. For instance, the ability to rapidly propagate accident reports back to oncoming cars on the highway, the awareness of unsafe drivers around you and the prevention of intersection crashes. The availability of powerful radios on board of vehicles, and of abundant spectrum (when not used for emergencies) will pave the way to a host of new applications for the “vehicle car to car grid”.

These emerging applications span many fields: from office-on-wheels to entertainment, mobile internet games, mobile - shopping, crime investigation, civic defense, etc. Some of these applications are conventional “mobile internet access” applications, say, downloading files, reading e-mail while on the move. Others involve the discovery of services in the neighborhood (e.g., restaurants, movie theaters, etc) using the vehicle grid as an ad hoc network. In addition, a brand new type of applications can be envisioned which will involve a much closer “cooperation” among cars including maintenance of distributed indices, creation and “temporary” storage of sharable content, “epidemic” distribution of content and index. Examples include the collection of “sensor data” by cars as seen as “mobile sensor platforms”, the sharing and streaming of files in a Bit-torrent fashion, and the creation/maintenance of massively distributed data bases with locally relevant commercial, entertainment and culture information (e.g., movies, hotels, museums, etc). Typically, these applications are totally distributed and follow a P2P collaboration model among cars. Fig.1 depicts a possible Urban Vehicle Grid scenario.

2.2 VANET Design

In designing protocols for the next generation vehicular network, we recognize that nodes in these networks have significantly different characteristics and demands from those in traditional wireless ad hoc networks deployed in infrastructureless environments (e.g. sensor field, battlefield, etc). Speed is one such difference. Automobiles travel at speeds up to one hundred miles per hour, making sustained vehicle-to-vehicle communication difficult. However, existing statistics of vehicular motion, such as traffic patterns during commute hours, can be used to develop sophisticated mobility models much more realistic than the current random waypoint models. By accurately characterizing

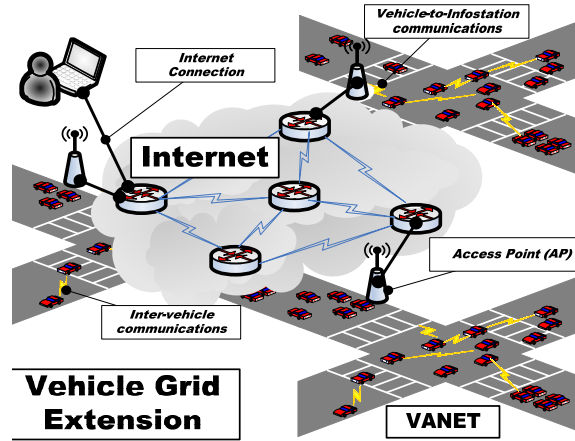


Fig. 1. A Urban Vehicle Grid scenario

vehicles' tendencies to travel together, these models can help maintain connectivity across mobile vehicular groups. Another important departure of vehicle networks from conventional ad hoc networks is the opportunity to deploy, in addition to traditional applications, a broad range of innovative content sharing applications (typically referred to as Peer-to-Peer applications). While their popularity has been well documented, they have been thus far confined to the fixed Internet (e.g., Bit Torrent, etc). The storage and processing capacity of VANET nodes make such applications feasible. Moreover, the fact that car passengers are a captive audience provides incentive for content distribution and sharing applications that would be unsuitable to other ad hoc network contexts. Examples include locality-aware information (map based directions) and content for entertainment (streaming movies, music and ads) [1, 2, 3]. These applications require high throughput network connectivity and fast access to desired data. Vehicles can also be producers of content. Examples include services that report on road conditions and accidents, traffic congestion monitoring, and emergency neighbor alerts. These applications require real-time and location-aware data gathering and dissemination [4].

The demands of these applications give us a list of requirements and challenges for vehicular applications including:

- Location awareness: both data gathered from vehicles and data consumed by vehicles are highly location-dependent.
- Time-sensitive dissemination of data to and from vehicles.
- Reliable communications in the presence of high vehicular mobility, intermittent connectivity and lossy channels.

The Geo-routing solution discussed in this paper and the associated Geo Location Service are ideally suited to support location awareness. The proposed

robust TCP solution is a good match for the time-sensitive, reliable communications requirements in high mobility.

3 Airborne Swarms

Another example of highly mobile ad hoc network is the swarm of unmanned aircraft. A possible application scenario is a disaster area that requires the intervention of police, firemen, paramedics etc, but where the unfriendly environment bars direct access. The swarm operates in a completely distributed, autonomous manner, establishing a communications network between the rescue teams and all critical fixed and mobile sensors and actuators in the disaster area. It allows the police to “see”, probe and manipulate the environment remotely before they can safely enter. Possible emergency scenarios include: chemical, nuclear plant disaster/sabotage; fire on a ship; explosion/fire on high rise building, etc. Other non-emergency application domains can benefit from the proposed swarm technology. For instance, space and planetary exploration (e.g., Mars), collection of scientific data in remote, sparsely instrumented regions, etc.

In the aftermath of a disaster we may assume that some “networked islands” of sensors, monitors and actuators have survived in the plant. For example, sensors may have been installed in strategic locations in the plant, building, ship etc; they had been preplanned for such an emergency. However, full sensor coverage and networking are generally not available after the accident, either because it was not practical or too costly to establish a complete infrastructure or because the infrastructure has been partially destroyed. Consequently, the entire area becomes partitioned into islands.

The rapid deployment of a swarm of air/ground agents will reestablish network connectivity, restore access to critical sensor probes, install new probes as necessary and help the collection and filtering of relevant data. Fig.2 shows a mobile backbone network of UAVs that reestablishes connectivity.

Airborne swarms find important applications also in the battlefield. Autonomous agents such Unmanned Airborne Vehicles (UAVs) are projected to the forefront for intelligence, surveillance, strike, enemy antiaircraft suppression, damage assessment, search and rescue and other tactical operations [5]. UAVs are organized in clusters to launch complex missions that include: coordinated aerial sweep of vast urban/suburban areas to track suspects; search and rescue operations in unfriendly areas (e.g., chemical spills, fires, etc), exploration of remote planets, and; reconnaissance of enemy field in the battle theater. The successful, distributed management of the mission will require efficient, reliable, low latency communications within members of each team, across teams and to a manned command post. Again, an efficient TCP implementation capable to survive aircraft mobility is essential. In the sequel we review current ad hoc TCP implementations and then propose a robust solution that will work up to vehicle and aircraft speeds.

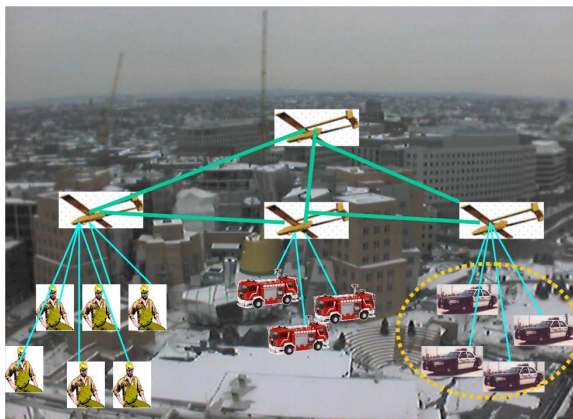


Fig. 2. Maintaining connectivity with a mobile backbone

4 Related Work

Many research efforts in recent years [6, 7, 8] have directed to making TCP robust to all sorts of wireless network disruptions including mobility. In mobile ad hoc networks, in fact, most of the packet losses are due to route breakage [7]. Mobility causes frequent route interruptions. If the routing algorithm cannot track node motion and packets enroute cannot be salvaged until a new route is found, TCP goes into exponentially increasing timeout intervals with severe performance hit.

How to improve TCP performance in such mobile networks has been a hot area for years. Routing of course is one key factor, thus the interaction between TCP and routing has been thoroughly investigated in the past. Unfortunately, traditional on-demand routing schemes, such as DSR and AODV, cannot efficiently address the frequent route breakage and packet loss due to high mobility. These schemes pre-compute the route at call setup time. Every node has a predefined next-hop for the designated destination. When this next-hop node moves away (or dies), the routing scheme must find another path. Unfortunately, finding another path takes time (and generally leads to TCP to time out). To avoid this delay, multi-path routing could be adopted, allowing multiple candidate next hops for packet forwarding. However the overhead of multi-path routing grows fast and becomes intolerable as node mobility increases [9]. Another solution is to exploit routing layer feedback (eg, route broken/route repaired) to TCP. In [10, 11], TCP's state is frozen when the sender receives the route failure signal from intermediate nodes. TCP exits the frozen state when route is re-established. Fixed-RTO was proposed in [12] with selective and delayed ack to help constraining the damage when packet loss is known to come from path breakage.

Even Fixed-RTO has only limited effect (as we will show in our experiments). When nodes are moving very fast, no traditional (proactive or on

demand) routing structure can adjust rapidly enough. For these extreme situations, Geo-routing [13, 14] has recently shown remarkable promise. Geo-routing uses the destination location as the “routable” address, and forwards packets (when possible) in a greedy manner towards it. Geo-routing is highly scalable, as nodes only keep geo-locations for their local neighbors. No explicit end to end route establishment is required. Since there is no pre-computed next hop to destination at set up time (as in all traditional schemes), rather the next hop is selected opportunistically “on the fly”, Geo-routing promises to be robust to path breakage and short term channel failure if the network is sufficiently dense (ie, there are always nodes in the “right” direction).

Geo-routing also places some extra costs on the network. It relies on information that is not needed in conventional routing schemes (eg, GPS positioning, Geo Location Server, accurate knowledge of neighbor locations). Moreover, if the basic Geo-routing “greedy” approach fails when the packet is trapped in a “cul de sac” (sort of a local “maximum” in the greedy search), it adopts perimeter (face) mode to go around the void area.

Many of the above issues (eg, location determination without GPS, Geo Location server, perimeter routing to circumvent local maxima) have already been studied extensively in the literature [15, 16, 17] and will not be addressed here. However, previous studies were mostly based on UDP and lightly loaded networks. We recall that Geo-routing uses hello messages to update neighbor information. In light load, the issues of hello message O/H and of interference between hellos and data packet did not emerge. TCP is rather aggressive in increasing network load, thus, it is important to “tune” the hello message rate taking into account not only speed but also load. Regarding hello messages, the careful reader will recall that some schemes [18] discover the best next hop dynamically, with an election and thus are not encumbered with background hello message maintenance all together. However, such schemes require a change in the MAC protocol (and thus in 802.11 firmware) which we exclude in this study. Moreover, they introduce the extra election overhead. Thus, one of the important contributions of this study is the hello rate optimization. Other useful contributions are the analysis and solution of the out of order delivery problem in the specific Geo-routing context.

5 TCP Performance over Geo-routing in Mobile Ad Hoc

In this section, we analyze GPSR [13], the most popular implementation of Geo-routing. We first study the impact of high mobility on UDP and TCP over GPSR and tune GPSR parameters to optimize performance. Then, we compare UDP and TCP performance on GPSR with AODV. We do not compare GPSR with other routing schemes (eg, DSR, DSDV, OLSR etc) for lack of space. Besides, the latter schemes tend to perform worse than AODV in mobile scenarios [8].

5.1 Case Study: Deterministic Motion

The first motion scenario, deterministic motion model, is carefully crafted to allow high mobility in a controlled way, yet maintaining end to end topology connectivity all the time. In this scenario, shown in Fig.3, the sender and the receiver are fixed. A total of 12 intermediate nodes arranged in 3 columns are moving vertically up and down in constant but opposite velocity. We place the nodes at 200m interval of each other in each column. Neighboring columns have opposite moving directions so the relative motion is twice the node speed. Recalling that the transmission range is 250m, in this scenario a path is always available from the sender to the receiver in spite of motion throughout the experiments. This is a very important detail of this experiment that we will exploit later.

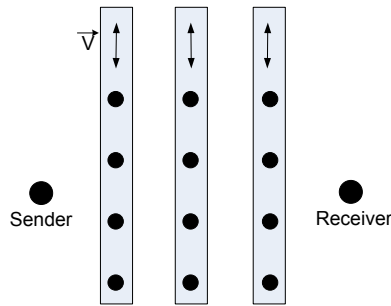
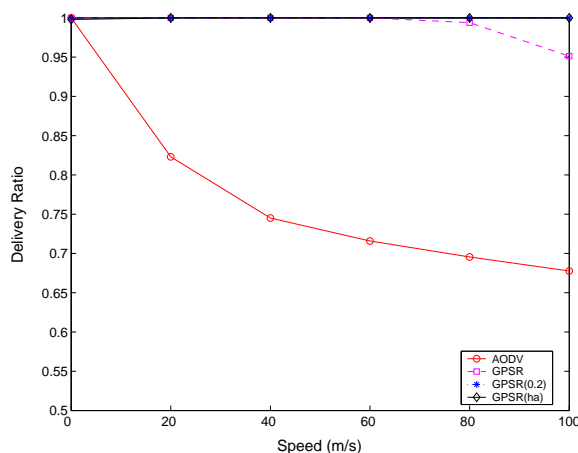


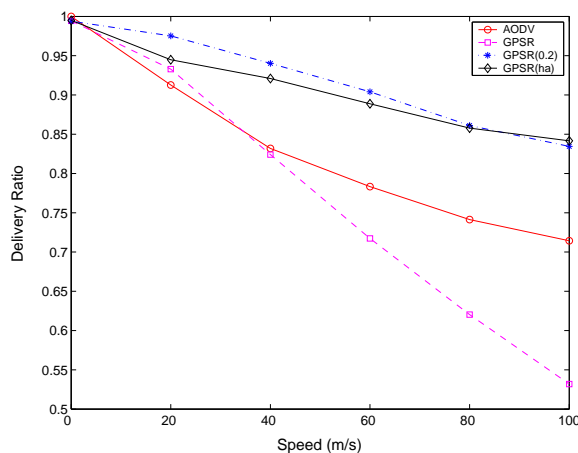
Fig. 3. Deterministic Movement Topology

In our experiments, radio rate is set to 2Mbps. Standard TCP (TCP NewReno) is used. Data packet size (for both UDP and TCP traffic) is 1000 bytes. GPSR hello packet refresh interval is initially set to 1s. All the results are averaged over 5 simulation runs with different random seeds. We vary mobility speed from 0m/s to 100m/s.

In the first experiment, UDP delivery ratio is presented in Fig.4. Fig.4(a) shows the delivery ratio with low CBR data rate (1 packet per second). The UDP delivery ratio in GPSR is quite good, almost 100%. This performance is indeed remarkable given the relatively high speed. Eventually, at top speed (100m/s) some packets are lost because of lack of a forwarding neighbor. This problem is easy to explain. Simple geometry shows that two nodes in neighbor columns moving at relative speed 200m/s are in contact at most for 1.5s. Recalling that hello refresh rate is 1s, and that some hellos may be lost because of interference, it is very possible that for some small fraction of the time a node has no forwarding neighbors. The packet is then lost! AODV in contrast does not work well, and the UDP delivery ratio deteriorates monotonically with the increased mobility, with more than 30% loss at 100 m/s . This is expected because of repeated path breakage and failure to find a route.



(a) UDP with low rate



(b) UDP with high rate

Fig. 4. UDP performance over deterministic motion

Fig.4(b) shows the delivery ratio for high rate UDP (40 packets per second). Surprisingly, GPSR collapses! The delivery ratio is even worse than in AODV for speed larger than 40m/s. From simulation results we find that the major reason for this problem is the loss of hello packets due to interference. Since hello packets are broadcast with an “unreliable MAC” (no RTS/CTS/ACK), when the UDP rate is high and congestion builds up, hello packet mortality is high. This leads to inaccurate neighbor information. If GPSR finds no neighbors in the forward directions, it initiates “perimeter routing” which can lead to loops, and to hop count timeout. In fact, in this experiments the lost packets all had very high hop count except packet losses at

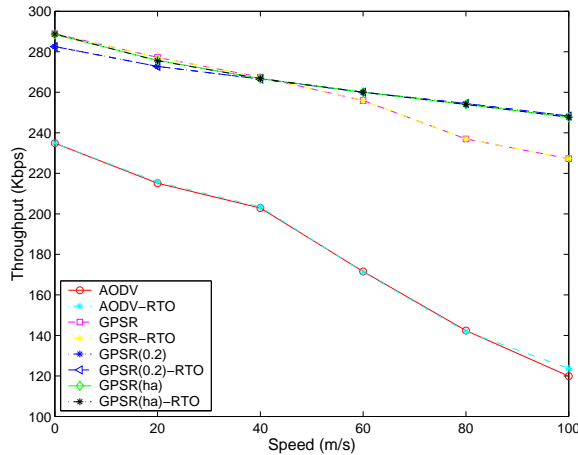


Fig. 5. TCP performance over deterministic motion

the source! Another experimental observation is that, when congestion builds up, most of the packets are dropped at the source node. This information could be useful to adjust (ie. reduce) CBR rate. We will further discuss this property later. To improve the chance that some hello messages are received, we increase the hello exchange rate from 1 hello per second in the initial GPSR to 1 hello per 0.2 second (named as GPSR(0.2)). Fig.4 shows that delivery ratio is significantly improved, especially for high CBR rate. However, high rate hello exchange brings more overhead. In Fig.4(b), the delivery ratio at speed 0m/s in GPSR(0.2) is slightly less than that for GPSR(1.0). From these results, high hello exchange rate is most effective when mobility speed is high.

Suspecting that in general there will be a trade off between routing efficiency and extra network O/H in the hello rate selection, we propose an adaptive hello interval scheme that increases hello rate based on mobility, also taking into account that hello packets will be lost due to interference. Specifically, we select the adaptive hello interval according to the following formula:

$$I = \frac{R}{k \times speed} \quad (1)$$

Where I is the hello interval, R is the transmission range and k is a tunable parameter. The rationale behind the formula is the following. On average two randomly moving neighbors see each other in a window equal to R meters. Thus, the contact interval is $R/speed$. During the contact interval, a node needs to send several hellos to announce its presence. The factor k should be adjusted to balance the overhead and effectiveness and to account for hello loss. In our simulation, k is set to 16. Additionally, there are two limits for hello interval, an upper limit of 2 second and a lower limit of 0.1 second. Thus, the adaptive hello interval used in the paper is $max(min(I, 2), 0.1)$.

The results for GPSR with adaptive hello exchange rate, named as GPSR(ha), are also shown in Fig.4. Equipped with adaptive hello rate in GPSR, UDP delivery ratio keeps around 100% for low rate UDP, and it is much better than AODV in high rate UDP as well. For high rate UDP, GPSR(ha) is slightly less than GPSR(0.2) for speeds from 20m/s to 60m/s. A more aggressive choice of parameters should be explored. In all, however, the difference ($< 3\%$) is quite small.

After studying UDP performance, we turn to TCP performance in Fig.5. Since TCP consists of two way traffic and a data packet can collide with an ack packet, TCP traffic press much more stress on the routing. From Fig.5, TCP over GPSR is only moderately affected when nodes moves, even at very high speed, due to the GPSR robustness to mobility. The throughput of TCP over GPSR only drops 21% from 0m/s to 100m/s and still achieves high throughput (around 230k at 100m/s), while TCP throughput over AODV drops by 50% and degrades fast for high mobility, just about 120k at 100m/s. Due to the frequent route breakage and more packet losses from two way traffic, TCP never has the chance to perform well and the performance degrades quickly with mobility in AODV. We recall that Fixed-RTO was proposed in TCP as a remedy to path breakage. We note in Fig.5. that Fixed-RTO only has minimal impact on performance over AODV and GPSR (the Fixed-RTO curves are practically overlapped with original TCP curves).

We note that, as a difference from UDP, TCP does not perform too bad with GPSR(1.0). This is due to the fact that TCP does congestion control and thus limits the interference onto hellos. Nevertheless, a reduction from hello interval from 1s to 0.2s brings significant benefits for high mobility as shown in 5, however the performance is worse in low mobility. This calls for adaptation. GPSR with adaptive hello (GPSR-ha) eliminates the overhead problem at low mobility. TCP over GPSR(ha) traces the upper envelope of performance of both GPSR(1.0) in low mobility and GPSR(0.2) for high mobility. Therefore, it satisfies our quest to improve TCP performance under varying mobility.

5.2 Random Movement

In what follows, we study TCP performance in a general case where nodes move randomly. The *random waypoint mobility model* [20] is used in the simulation. Fig.6 illustrates the random moving topology. The sender and receiver are kept at fixed positions, while the remaining 40 nodes are moving randomly in a region of 1000m \times 1000m. The speed of the nodes ranges from 10m/s to 90m/s.

The performance of UDP over GPSR and TCP with Fixed-RTO in this scenario is pretty similar to that observed in deterministic motion, so we omit their results here. Fig.7 presents TCP results over AODV and GPSR. As expected, TCP over GPSR still outperforms TCP over AODV. However, its performance is by far worse than in the deterministic motion case. We discover that TCP over Geo-routing has a serious out-of-order(OOO) delivery problem

in the random movement. In contrast, TCP over AODV suffers no significant OOO problems.

In GPSR, a route is selected on a packet by packet basis and thus can change very rapidly. If a new route is shorter or has lower delivery delay due to lighter load, data packets on the new route could arrive before packets on the old route. OOO packets cause throughput degradation. In fact, TCP receiver responds to OOO data packets with duplicate acks which potentially trigger fast retransmits leading to congestion window reduction and extra inefficiency. This OOO problem was discussed in [21] and some approaches for OOO detection and response were presented. However, the approaches proposed in [21] require the modifications of packet header and cooperations at TCP sender and receiver. In the paper we propose a novel approach, which only involves TCP receiver without modifying the packet header format and TCP sender. TCP receiver determines if a non-in-order packet has come from a different route by simply checking the TTL value in the packet header. If the packet is from a path different from that of the latest in-order packet, there was a route change and the OOO event is detected. The missing packets between this OOO packet and in-order packet could arrive latter. The receiver could wait for some time before issuing a duplicate ack. A challenge here is to estimate the “optimal” waiting window at the receiver. We propose to let the receiver passively estimate RTT and use it to decide the period for the waiting timer. The receiver could use the TCP timestamp option in the packet header to estimate RTT, though such an estimate may be inflated if the sender does not send data packets immediately after receiving an ack [22]. However, the waiting timer is only a coarse timer for predicting when the missing packets will arrive, thus RTT inflation errors can be tolerated.

The receiver computes the waiting time based on this RTT measurement. The following formula is used:

$$\begin{aligned} SRTT_{k+1}^r &= \frac{7}{8}SRTT_k^r + \frac{1}{8}RTT_{k+1}^r \\ RTT_{k+1}^{var} &= \frac{3}{4}RTT_k^{var} + \frac{1}{4}|RTT_{k+1}^r - SRTT_{k+1}^r| \\ RTO_{k+1}^r &= SRTT_{k+1}^r + 4 \times RTT_{k+1}^{var} \end{aligned}$$

Where RTT^r is the RTT estimation at the receiver, $SRTT^r$ is the smoothed RTT. RTT_{var} and RTO^r are RTT variance and waiting timer period at the receiver. This ack waiting timer is started after detecting OOO event, and is canceled after all missing packets arrived.

As shown in Fig.7, the OOO packet handling strategy (GPSR-OOO) enhances TCP performance by about 10%. Incidentally, we also tested OOO delivery in deterministic motion, however we did not find significant OOO delivery effects because all direct paths have the same length. Next, we enhance TCP over GPSR by adjusting hello intervals. Fig.8 presents the TCP performance over GPSR with fast and adaptive hello exchange. Note: we only show results with OOO response. As expected, GPSR(0.2) only provides per-

formance gain for high mobility, while GPSR(ha) integrates the advantages of GPSR in low mobility and GPSR(0.2) in high mobility. TCP over GPSR with adaptive hello interval is considerably better than TCP over AODV. The throughput of TCP over GPSR(ha) only drops from 315kbps at 10m/s to 225kbps at 90m/s (about 28% performance degradation), while AODV drops from 260kbps at 10m/s to 95kbps at 90m/s (about 63% performance degradation).

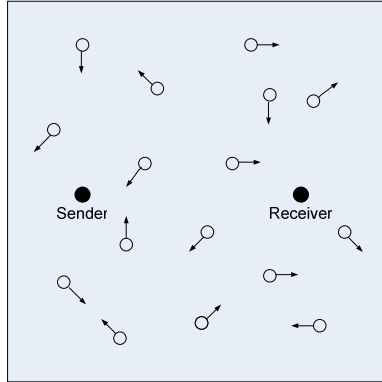


Fig. 6. Random Movement Topology

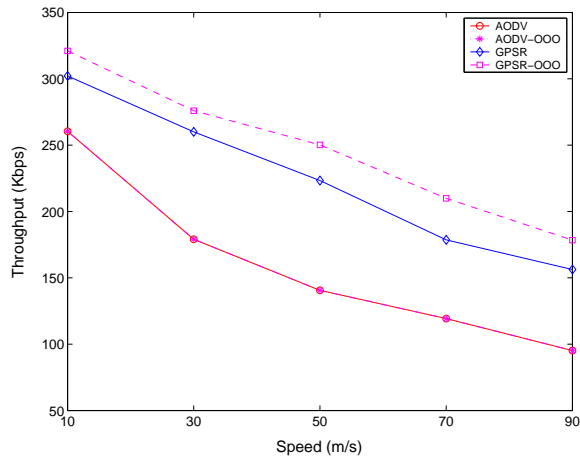


Fig. 7. TCP performance over random motion

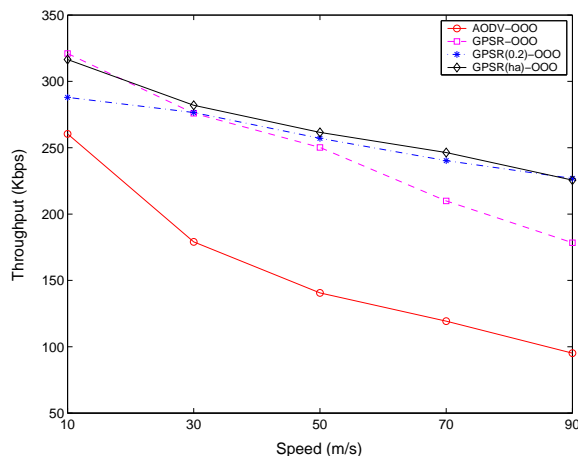


Fig. 8. TCP performance over random motion with OOO response

6 Conclusion

We have studied TCP and UDP performance over Geo-routing in highly mobile ad hoc network. As expected, Geo-routing introduces substantial benefits. Several Geo-routing parameters have been carefully tuned to achieve such benefits. First, we have proposed a hello scheme adaptive to the mobility. Second, we have proposed a novel scheme for handling out-of-order delivery which requires TCP receiver-only modification. These two enhancements can improve TCP performance by 50%, from 150kbps to 225kbps at aircraft speed (90m/s) when Geo-routing is used. In contrast, TCP over AODV delivers less than 100kbps at this speed.

References

1. A. Nandan, S. Das, B. Zhou, G. Pau, and M. Gerla, (2005). AdTorrent: Digital BillBoards for Vehicular Networks. Vehicle-to-Vehicle Communications Workshop (V2VCOMM 2005), San Diego, CA, USA, July 2005.
2. A. Nandan, S. Das, S. Tewari, M. Gerla, and L. Kleinrock, (2006). AdTorrent: Delivering Location Cognizant Advertisements to Car Networks. The Third International Conference on Wireless On Demand Network Systems and Services(WONS 2006), Les Menuires, France, January, 2006
3. A. Nandan, S. Das, G. Pau, M.Y. Sanadidi, and M. Gerla, (2005). Cooperative downloading in Vehicular Ad Hoc Networks. Wireless On-Demand Network Systems and Services (WONS 2005), St Moritz, Switzerland, 2005.
4. U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista and A. Corradi, (2006). Efficient Data Harvesting in Mobile Sensor Platforms. IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2006), Pisa, Italy, March, 2006.

5. Mario Gerla, Kaixin Xu, and Allen Moshfegh, (2002). Minuteman: Forward Projection of Unmanned Agents Using the Airborne Internet. IEEE Aerospace Conference 2002, Big Sky, MT, Mar. 2002.
6. Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, (2003). The impact of multihop wireless channel on TCP throughput and loss, IEEE INFOCOM 2003.
7. X. Yu (2004). Improving tcp performance over mobile ad hoc networks by exploiting cross-layer information awareness, MobiCom.
8. A. A. et al., (2000). Performance of tcp over different routing protocols in mobile ad-hoc networks, IEEE VTC.
9. P. Pham and S. Perreau, (2003). Performance analysis of reactive shortest path and multipath routing mechanism with load balance, IEEE INFOCOM.
10. G. Holland and N. H. Vaidya, (1999). Analysis of TCP performance over mobile ad hoc networks, MobiCom.
11. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, (2001). A feedback-based scheme for improving TCP performance in ad hoc wireless networks, IEEE Personal Communications Magazine.
12. T. D. Dyer and R. V. Boppana, (2001). A comparison of tcp performance over three routing protocols for mobile ad hoc networks, MobiHoc.
13. B. Karp and H. T. Kung, (2000). GPSR: Greedy perimeter stateless routing for wireless networks, MobiCom.
14. I. S. P. Bose, P. Morin and J. Urrutia, (2001). Routing with guaranteed delivery in ad hoc wireless networks, ACM/Kluwer Wireless Networks.
15. J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, (2000). A scalable location service for geographic ad hoc routing, MobiCom.
16. D. Niculescu and B. Nath, (2001). Ad hoc positioning system, IEEE GLOBECOM.
17. S. Capkun, M. Hamdi, and J. P. Hubaux, (2001). Gps-free positioning in mobile ad-hoc networks, Proceedings of HICSS.
18. M. Zorzi and R. R. Rao, (2003). Geographic random forwarding (gegraf) for ad hoc and sensor networks: energy and latency performance, IEEE Transaction on Mobile Computing.
19. C. E. Perkins and E. M. Royer, (1999). Ad-hoc on-demand distance vector routing, Proceedings of IEEE WMCSA.
20. C. Bettstetter, H. Hartenstein, and X. Prez-Costa, (2004). Stochastic properties of the random waypoint mobility model, ACM/Kluwer Wireless Networks: Special Issue on Modeling and Analysis of Mobile Networks.
21. Y. Zhang and F. Wang, (2004). Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response, MobiHoc.
22. V. Jacobson, (1992). Tcp extensions for high performance, RFC 1323.