Characterizing Transport Layer Behavior in the MeshTest Wireless Testbed

Jessica Seastrom
University of Maryland-College Park
Computer Science Department
College Park, MD, USA
jkseast@cs.umd.edu

***Abstract*—The MeshTest testbed enables real wireless nodes, real RF, and programmable attenuators to simulate controllable and repeatable mobile wireless experiments. MeshTest provides realistic wireless conditions with variable link states and interference that are difficult to model in software based wireless simulators. In addition, MeshTest reduces the complexity and lack of control introduced by wireless field tests with actual wireless devices. Previous work provides validation of the testbed's ability to accurately simulate propagation loss through RF comparisons with actual wireless devices arranged at similar distances. Similarly, this paper focuses on a comprehensive analysis and validation of the testbed's transport layer behavior. In this paper, we conduct experimental scenarios which help to characterize TCP and UDP in the testbed. The result is a higher degree of confidence in the overall accuracy of the MeshTest wireless testbed. In addition, this work contributes to a more comprehensive DTN convergence layer design optimized for mobile wireless networks.**

I. Introduction

A store-and-forward network with support for high delay, intermittent, or non- existent end-to-end connectivity is classified as a Delay-Tolerant Network (DTN). The DTN protocol is designed to provide connectivity in adverse and challenged network conditions, conditions where TCP often fails [1]. The DTN reference implementation (DTNRI) released by the DTNRG [2] is an implementation of the DTN protocol [3] for use in heterogeneous networks including mobile wireless networks. A DTMN is a DTN network of mobile nodes where end-to-end connectivity may never exist between any two nodes in the network [4]. In such a network, a node may transmit and then disconnect from the network but still require some form of reliable delivery guarantee. Designing such a network that is robust and reliable under a wide variety of mobile scenarios, with the potential for extreme network partitioning, requires an extensive amount of research and testing.

As in [4], such testing in the DTMN area has mainly relied upon simulation. This is due in part to the time, difficulty, and expense of running live experiments with real devices. Field tests can be complex and difficult to coordinate over large geographic areas. In addition, they are often only marginally reproducible. On the other hand, simulators such as ns2 [5], OPNET [6], and GloMoSim [7], which base their network modeling on RF propagation theory, do not capture the dynamic link conditions and real world hardware interactions reflected in field tests.

MeshTest is a laboratory-based, mobile wireless testbed that offers a hybrid between field testing and simulation [8]. The testbed features wireless Orbit [9] nodes placed in shielded enclosures with their RF wired into a matrix switch of programmable attenuators. By dynamically adjusting the attentions between the nodes, MeshTest can effectively simulate arbitrary physical arrangements of nodes and mobile scenarios. The testbed allows a diverse variety of experiments to be run with real hardware and real implementations while maintaining close control and monitoring of the nodes. The network can be configured with mutlti-hop and point-to-point links.

MeshTest provides an ideal environment for testing the mobile wireless devices running the DTN protocol itself as well as convergence layers that bridge the transport and DTN layers. In [10], MeshTest was used to experiment with the data MULE scenarios presented in [11]. The experiments used the DTNRI (DTN2 version) running on Orbit nodes, and the testbed provided a unique ability to reproduce these experiments without introducing uncontrollable variables from the environment. One observation that resulted from these experiments was inconsistent data exchanges between the MULEs and the nodes across experiments. While the same amount of data was available for transfer at each hand-off opportunity, the amount of actual data transferred often varied, yet the amount of data generated and the mobility scenario remained constant. Due to the complexity of multiple layer interactions, the reason for this behavior is not easily understood.

[12] and [8] validate, both in theoretical and actual performance terms, the simulation accuracy of the testbed. However, in light of the data MULE experiments, characterization of the testbed transport layer behavior also seems appropriate. Thus, this paper provides an analysis of the MeshTest transport layer in order to experimentally validate its behavior. In particular, we focus on the transmission and throughput range for both UDP and TCP as a function of distance and examine the throughput variability. Through this work, we also establish a foundation for future DTNMN transport layer research and

contribute to the development of a robust convergence layer for DTN in a mobile wireless network. Our work provides an initial analysis of conditions where using an unreliable transport layer such as UDP provides greater link utilization while deferring reliability to the convergence and DTN layers.

This paper is organized into several sections. Section II discusses the DTN convergence layer (CL) and the need for a CL optimized for mobile wireless environments. Section III-A provides an overview of the MeshTest testbed. Section IV-A details our transport layer experiments and section IV-B presents our findings. Finally, our conclusions and suggestions for future work are presented in section V and VI.

II. The DTN Convergence Layer

One of the more interesting aspects of the DTN protocol is the convergence layer. The convergence layer provides an interface between the transport layer and the DTN layer. This layer bridges the gap between the “bundle” at the DTN layer and segments at the transport layer. In fact, a DTN can provide services over any transport layer provided an appropriate convergence layer with the necessary interface. Depending on the bundle size and the size of the underlying transport layer maximum transmission unit (MTU), a bundle may be subdivided into smaller units for transmission into the network. This subdivision introduces fragmentation of the bundles prior to transmission. The DTN specification defines the concepts of proactive, pre-convergence layer bundle fragmentation, and reactive, post-convergence layer bundle fragmentation. However, details of DTN fragmentation and a fragmentation implementation are as yet unspecified by the DTNRG. Despite the lack of a concrete specification, the DTNRI implements fragmentation using TCP at the transport layer, however we have observed that this fragmentation does not reactively fragment and retransmit only missing bundle segments, instead entire bundles are retransmitted.

Custody transfer is a method for acknowledging that a bundle has been received by potentially many nodes in this store-and-forward network. Any node accepting custody can be designated as a “custodian” in a DTN, and a bundle may traverse multiple hops before reaching a custodial node. Custody transfer provides a mechanism for reliable delivery at the DTN layer, which is distinct from transport layer reliability in use over any hop. Custody transfer provides end-to-end reliability whereas a reliable transport layer can only provide hop-by-hop reliability in a DTN.

The introduction of reliability via custody transfer and reactive fragmentation in the convergence and DTN layer provides an opportunity for discussion regarding the need for reliability at the transport layer as well. With reactive fragmentation and custody transfer reliability, using an unreliable transport layer such as UDP over point-to-point wireless links would provide both reliability and efficient link utilization. This approach is adopted by Saratoga, a convergence layer protocol designed for file transfer between satellites and ground stations [13]. Another convergence layer using UDP is the Licklider (LTP) protocol [14]. The main difference between these convergence layers is the type of data being transmitted; however, their main objectives remain the same- efficient link utilization. (In the case of Saratoga, another goal is efficient link utilization during small transmission windows.) Ultimately, determining the best convergence and transport layer for mobile wireless DTNs will depend on the scenario (e.g. multi-hop or point-to-point transmissions, the level of end-to-end connectivity between the sender and receiver, etc.) and the application using DTN.

The MeshTest testbed provides an ideal environment for research and development of a mobile wireless convergence layer implementation for DTN. MeshTest provides a realistic framework for understanding whether UDP or TCP has better link utilization and under which scenarios and what types of network topologies. Thus, in order to make use of the MeshTest testbed and improve our understanding of DTN convergence layer issues, this paper focuses on the behavior of UDP and TCP in general in the MeshTest wireless mobile testbed.

III. The MeshTest Testbed

A. *MeshTest Design*

MeshTest consists of a rack of 12 Orbit nodes running Debian Linux in shielded enclosures, an RF matrix switch, and a server that provides experiment control, as depicted in Figure 2. The RF from each computer’s Atheros WiFi card is cabled through the enclosures and into the matrix switch. The enclosures prevent inadvertent cross-talk between the computers, and the matrix switch allows us to arbitrarily control the attenuation between the devices. This modification of signal attenuation simulates internode distances.

The switch attenuation settings are computed based on node coordinates. Repeatedly updating these coordinates simulates node movement in the testbed. The graphical user interface depicted in figure 3 is one method for generating a mobility scenario. Additionally, node coordinate updates can be sent via xml directly to the MeshTest simulation daemon. Attenuation based on node distances is computed using any path loss model, however currently only free space path loss is implemented.

Figure 1 is a simplified diagram of an $n \times b$ switch for $n = 3$, $b = 2$. It has 3 inputs that connect through 6 digital attenuators to 2 buses. Each bus has a direct, unattenuated, external connection. The current MeshTest switch is actually a 16×4 switch and an 8×2 switch has been recently added. In [15] we present possible ways for connecting these two switches to address testbed scalability.

The nodes are connected to the switch inputs, and the outputs are left unterminated. Signals are reflected back through the combiners to all the inputs. The amount of components the signal must pass through leads to considerable insertion loss that is dependent on the frequency used. However, this loss is approximately the free space path loss over 2m and thus does not significantly limit the types of scenarios that can be simulated [15].

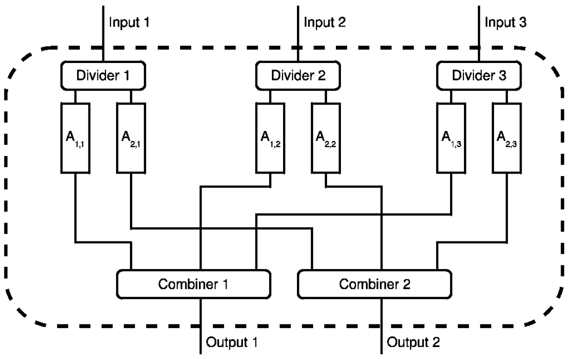


Fig. 1. Simplified RF matrix switch diagram, showing three inputs and two busses, each with an unattenuated output. The boxes labeled $A_{i,j}$ represent the digital attenuators with ranges 0-127 dB

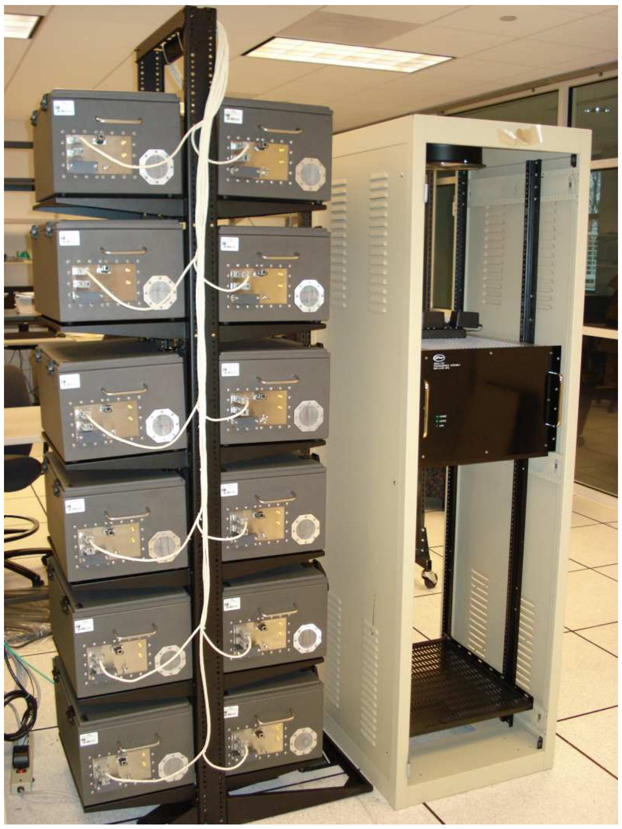
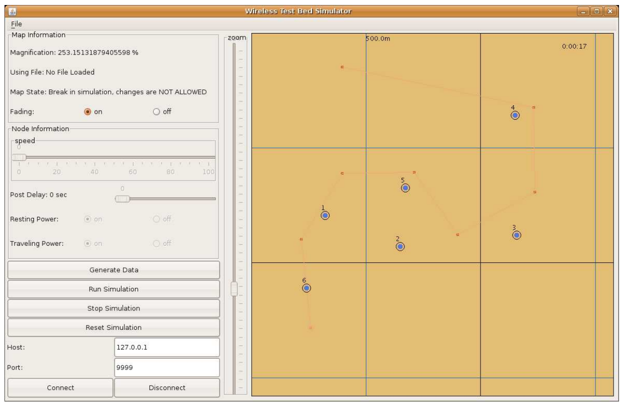


Fig. 2. The MeshTest testbed. The shielded enclosures contain wireless nodes and the RF matrix switch controls the attenuation experienced between the nodes.



Wireless Test Bed Simulator

Map Information
Magnification: 253.15131879405598 %
Using File: No File Loaded
Map State: Break in simulation, changes are NOT ALLOWED
Fading: [✓ on] off

Node Information
Post Delay: 0 sec
Resting Power: [✓ on] off
Traveling Power: [✓ on] off

Host: 127.0.0.1
Port: 9999

Fig. 3. A screenshot of the simulation daemon GUI. This gui can be used to control placement and movement of the nodes. Each grid square represents $1km^2$

It is not possible to exactly compute switch attenuation settings equal to the 16×16 path loss matrix represented by an arbitrary scenario. Thus, a simulated annealing algorithm is used to approximate the path loss and to find the appropriate switch settings. The accuracy of this solution is established in [16]. Given a scenario with 12 nodes placed randomly in a $1\text{km} \times 1\text{km}$ square, on average this solution finds attenuator settings that come within about 1dB of the desired path loss on each link.

Modification of signal attenuation simulates propagation loss due to node distance. However, propagation delay is not modeled in the testbed. Since the nodes do not physically move, propagation delay remains constant regardless of attenuation changes. Although the TCP retransmission mechanism makes use of the round-trip time (RTT) in order to establish a retransmission timeout (RTO) value, we contend that the propagation delay between nodes accounts for a negligible amount of true network round-trip time estimates. Given the propagation delay of the 802.11 medium is the speed of light ($3 \times 10^8 m/s$) and the granularity of the RTO timer is on the order of $500ms$ and is updated once per RTT in most Berkeley-derived systems [17], the distance at which propagation delay of 802.11 would become measurable is calculated as:

$$500ms \times 3 \times 10^8 m/s = 15 \times 10^7 m = 15 \times 10^4 km$$

Thus, at distances less than $15 \times 10^4 km$, propagation delay contributes far less to the round-trip delay as compared to delay introduced by processing at intermediate routers or by traversing multiple hops. Because this intermediate delay is retained within the testbed, the TCP RTO variable is updated based on the dominating delay, the delay introduced by processing at intermediate nodes.

On the other hand, constant propagation delay can have affects on the physical layer. Given the fact all nodes within range receive the same signal, it is impossible to delay the signal according to the propagation delays of multiple re-

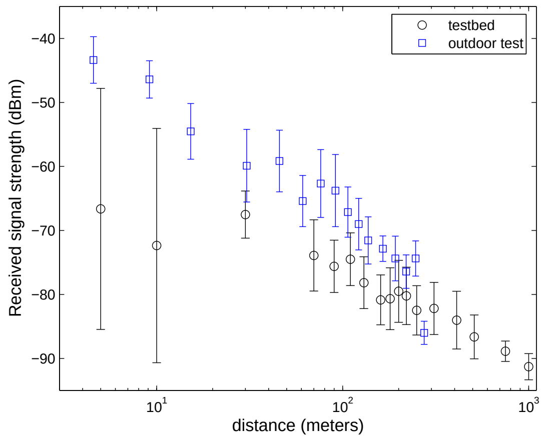


Fig. 4. The stationary measurement results from [12]. RSSI measurements at various distances for both the testbed simulations and actual wireless devices.

ceivers. Delaying the signal for one receiver could be done by transmitting the signal at a later time based on this receiver's expected propagation delay. However, all other nodes would also receive the signal at the same time, which may not may not match their expected propagation delays. Delaying transmission, therefore, does not solve this problem since the signal has already affected the shared medium at this point. In [12], the affect of this constant propagation delay was studied. It was shown in shared medium tests that this constant propagation delay did not have significant affects on throughput.

B. *MeshTest Validation Studies*

In [12] the authors perform several validation studies of MeshTest. The results of stationary measurements, "drive by" experiments, and shared medium testing showed that MeshTest is able to accurately simulate path loss experienced by real-world 802.11 devices. The stationary measurement testing compared the RSSI (received signal strength) between a pair of MeshTest nodes at simulated distances and a pair of laptops configured as an 802.11 sender and receiver pair. This field test evaluated the accuracy of the MeshTest path loss algorithm. The results of this experiment are depicted in figure 4. MeshTest's simulated results are reasonably consistent with the wireless laptop results, however a high degree of variability exists when the nodes are at close distances. The authors attribute this variability to the switch insertion loss which can not produce attenuation settings less than -45dB. In section IV-B, we find that this variability translates to the transport layer as well.

IV. TCP and UDP Throughput Characterization

A. *Experiment Description*

Our objectives were to experimentally validate the transport layer protocols in the MeshTest testbed and to establish the range where a particular transport protocol might perform especially better than its counterpart in a mobile wireless scenario. We conducted two types experiments which we call the "walkaway" and the "drive-by" experiments. The nodes are placed in a 2-D grid and for both experiments we varied only the $x - coordinate$, thereby introducing a 1-D mobility pattern.

In the walkaway experiment, two nodes, a sender, designated s, and receiver, r, begin transmitting when they are $x_s = x_r = 0$, distance, $d = 0$ apart, at time $t = 0$. S moves at a constant rate of 4m/sec while r remains stationary. The experiment ran for 300 secs at which time the nodes achieved a maximum separation of 1.2km. Through this experiment we are able to characterize the effect of distance on throughput when the nodes started at a minimum distance of $d = 0$.

The drive-by experiment begins with the sender and receiver nodes 1.2km apart, s is at $x_s = -1.2km$ and r is located at $x_r = 0$. S moves in a straight line toward r at a rate of 4m/sec. $t = [0, 300)$ is designated the "approaching path" and $t = [300, 600)$ is designated as the "parting path." Exactly halfway through the experiment, $t = 300$, $x_s = x_r = 0$, thus $d = 0$. S continues moving away from r at a rate of 4m/sec until they are separated by $d = 1.2km$. S has traveled a total distance of 2.4km in a straight line.

We used the same experimental scenarios for both UDP and TCP using iperf [18] to generate the traffic at a constant bit rate of $29Mbps$ for UDP and with a receive window size of $85.3kB$ and a sender window size of $16kB$ for TCP. The MSS was 1450 and the UDP packet size was 1470 bytes. Iperf defaults to the window sizes of the particular Linux defaults and in all experiments the defaults selected by iperf were used. We conducted 100 individual runs of TCP and UDP for each experiment.

Since the walkaway experiment begins with the nodes at $d = 0$, there is no issue with establishing an immediate TCP or UDP connection between the sender and the receiver. However, measuring the time at which a route between the nodes is established in the TCP drive-by experiment required repeated attempts to establish a successful connection. We achieved this via a timer inserted at the start of the client code to measure the time between the start of the experiment until a successful TCP connection was established. The time at which a route was established for the UDP drive-by experiment was easily measured by the number of lost packets transmitted until the nodes began successfully transmitting (a statistic reported without modification by iperf).

In section III-A, we describe the two methods for updating switch settings, via the gui or direct xml updates. Since these experiments required precise coordinate updates, the test code directly updated the simulation daemon with new node coordinates via xml every second.

B. *Results and Analysis*

Figure 5 and figure 6 depict the mean throughput for both UDP and TCP in the walkaway and the drive-by experiments. In both the walkaway and the drive-by experiments, UDP achieves greater mean throughput than TCP. This result is in line with the findings in [19] that UDP throughput in WLAN's

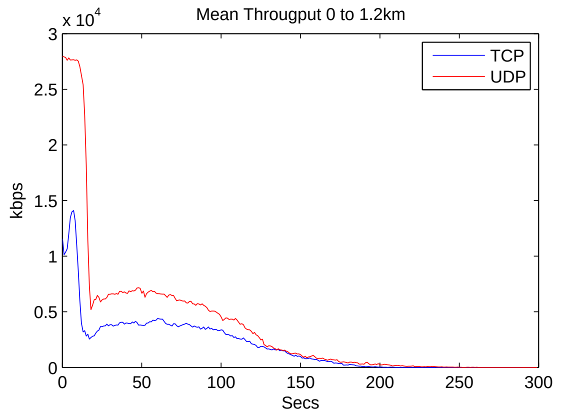


Fig. 5. The walkaway experiment mean throughput for TCP and UDP.

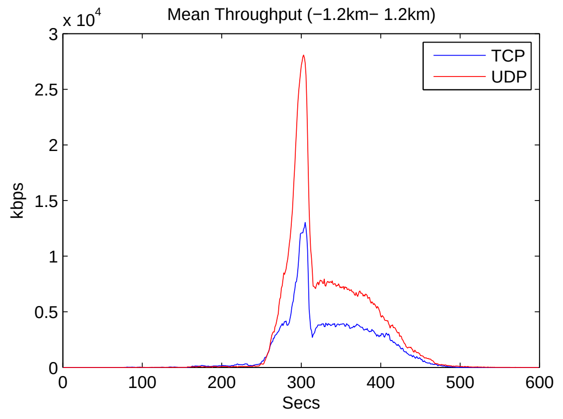


Fig. 6. The drive-by experiment mean throughput for TCP and UDP.

in close range outperforms TCP. The free space model most closely models close range scenarios, thus even though our nodes are at large distances, our simulation model does not capture this type of propagation loss (i.e. Two-ray or Flat Earth). It is therefore reasonable to compare our results with [19].

These throughput measurements provide a characterization of the performance of both TCP and UDP in the MeshTest testbed and establish the testbed transmission ranges along a straight line path. Interestingly, the drive-by experiment clearly shows that connection distances are asymmetric. The connection remains established longer on the parting path than on the approaching path. While delay introduced by failed TCP SYN-ACK handshakes and early losses in the TCP "slow start" phase seem likely to introduce an amount of early delay, the fact that UDP fares no better than TCP seems to contradict any transport layer protocol issues as the main cause for this asymmetry. (Recall that the testbed nodes are not physically moving, therefore antenna direction is not a variable in our experiments.) Delayed MAC layer associations prevent ARP table updates and are a likely cause for a portion of this delay on the approaching path. It is found in [12] that the RSSI is primarily a function of distance, therefore it appears that the RSSI threshold for 802.11 association is higher than the threshold for disassociation.

Fgures 7 and 8 illustate the throughput distribution at 50 sec intervals for both the walkaway and driveby experiments. For a more representative distribution, the interval, i, consisted of 5 seconds, $[i-2, i+2]$, in order to clearly show the throughput probability within the given time interval.

The walkaway distributions, depicted in figure 7, indicate an "all-or-nothing" throughput state for UDP while TCP throughput appears more distributed within the interval. At $t = 198$, the probability of any non-zero throughput is approximately $p = .03$, however the probability for non-zero throughput during this interval for UDP is $p = .5$. By $t = 248$, the probability of any throughput for TCP is 0, however UDP has a probability of approximately $p = .07$. UDP's probability of throughput is the same as TCP's 50 seconds later. After $t = 252$, $d > 1008m$, the probability of any throughput for either TCP or UDP is 0.

In the drive-by experiment, figure 8 presents only the time intervals of any noticeable throughput. In the interval, $t = [148, 152]$, TCP had a small probability of throughput ($p < .01$) whereas UDP's remained 0. For compactness, this figure is not presented. Results beyond 302 seconds resembled those already captured by the walkaway experiment and are thus also omitted. As a result of the drive-by experiment, we also find that TCP begins successfully transmitting sooner than UDP. It appears that slow start and the initial TCP SYN-ACK handshake do not significantly delay initial TCP connection setup. The $t = [198, 202]$ plot shows a roughly equivalent probability of throughput, and TCP slow start appears to limit throughput whereas UDP, which is not subjected to such congestion control, achieves greater throughput.

Throughput variability is represented in figures 9, 11, 10, and 12. For clarity, 10 secs intervals are plotted. Throughput variability is greatest when node distances are least, particularly at $d = 0$. The drive-by experiments illustrate that this variability is roughly symmetric. Comparing these results to figure 4, we find that variability in RSSI at the physical layer due to some error in path loss simulation calculations at small distances appears to affect the transport layer as well. An unexpected result is the high variability of UDP in the drive-by experiments, figure 11 at $t = +/- 10$, or $d = +/- 40m$. Since figure 4 only includes three data points between $d = [10, 100]$, and two of these points are at $d > 40m$, it is difficult to conclude whether this variability is due to path loss simulation error. An interesting result might be that the path loss simulation error at small distances affects the transport layer at greater distances. Further transport and physical layer experiments at small distances would provide more data points and an ability to understand better the root causes of this variability.

V. Conclusions

Our results provide us with an interesting insight into the behavior of TCP and UDP in the MeshTest testbed. The

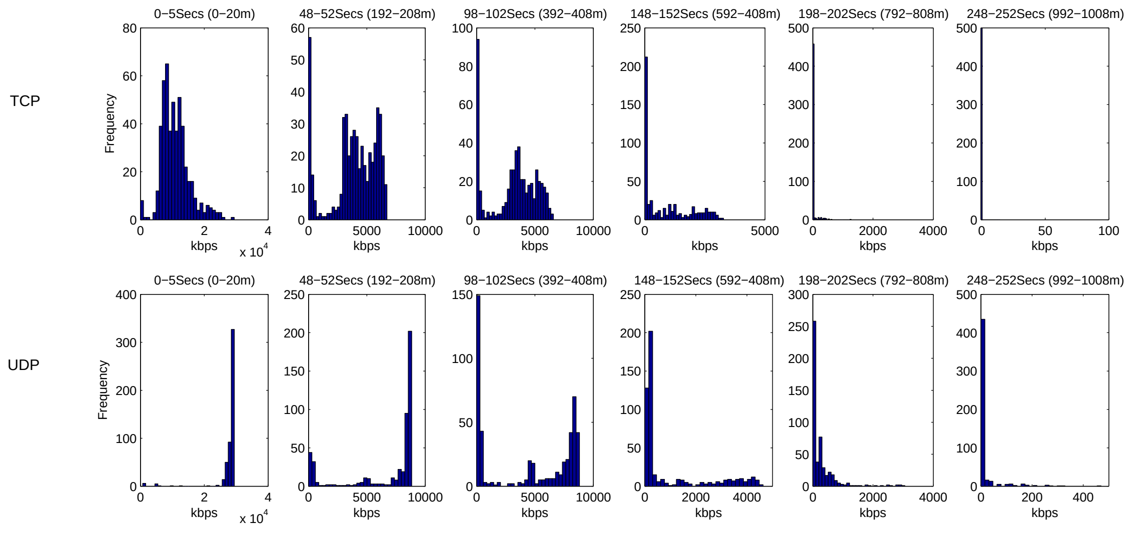


Fig. 7. Distribution per 50 second interval (0-252 secs) in the walkway experiment for TCP and UDP.

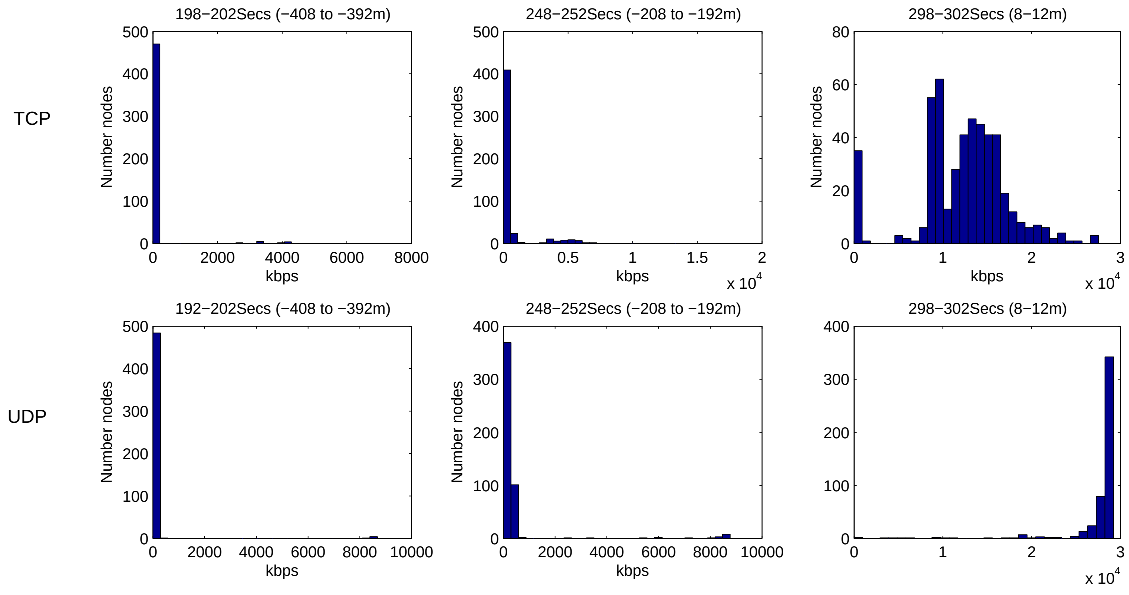


Fig. 8. Distribution per 50 second interval (198-302 secs) in the drive-by experiment for TCP and UDP.

main reason for studying the throughput distances for mobile wireless networks is to understand how to achieve maximum throughput over the link. With this quantitative analysis we are better able to predict the probability of successful throughput at a given distance. Our results also show that simulation models based on distance alone do not capture the asymmetry observed in the drive by experiment. This result provides more insight into the behavior of wireless mobile nodes than can be modeled with traditional simulation tools.

Although UDP appears to have a throughput advantage over TCP in these point-to-point experiments, we can not generalize our findings to other network topologies. In comparing the performance of TCP and UDP in multi-rate, multi-hop networks, the authors in [20] find that UDP throughput decreases more than TCP as hop count increases. This is because the probability of packet error increases with the number of hops. Thus, our findings generalize to the behavior of UDP and TCP over point-to-point links.

From our results, it is likely that UDP over point-to-point links might provide better link utilization. Using reactive fragmentation and custody transfer would provide overall protocol reliability. While such a design is successfully used

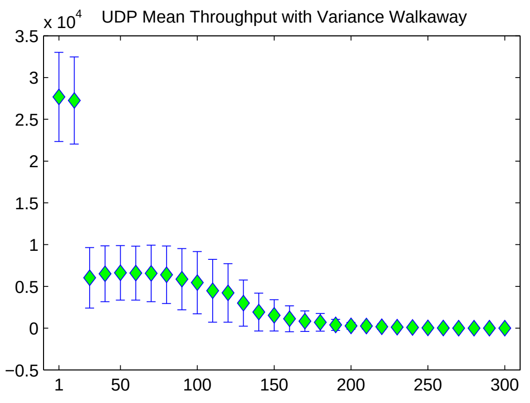


Fig. 9. The walkaway experiment UDP mean throughput and variance.

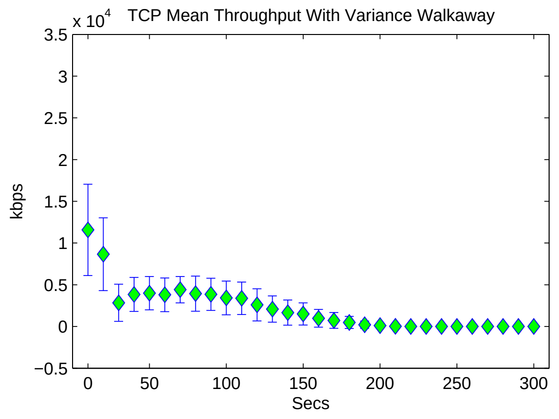


Fig. 10. The walkaway experiment TCP mean throughput and variance.

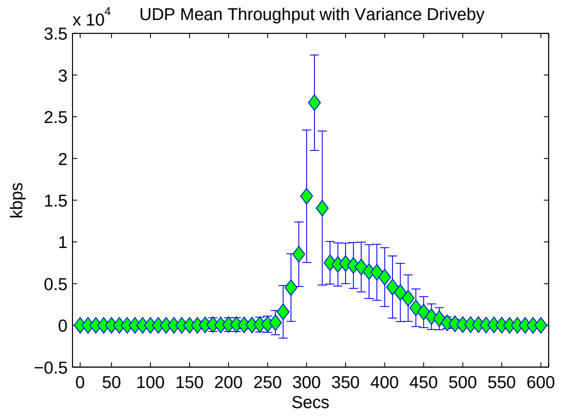


Fig. 11. The drive-by experiment UDP mean throughput and variance.

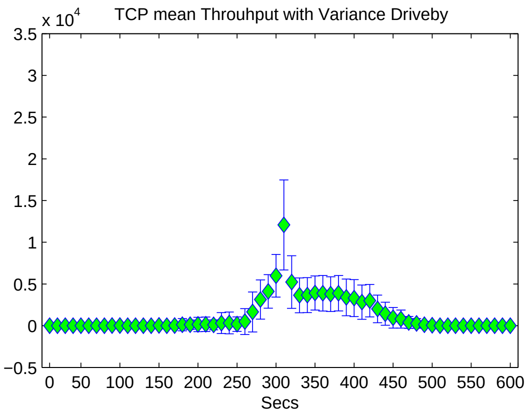


Fig. 12. The drive-by experiment TCP mean throughput and variance.

in the convergence layers previously mentioned, further study is required to determine the suitability of UDP in general in a complex mobile wireless environment.

Finally, our experiments point to transport layer throughput variability at distances less than $40m$. This is a potential cause of the variability observed in the the data MULE experiments in [10].

VI. Future Work

The testbed continues to evolve, and as it does, further areas of research both on the testbed itself and ways in which it might be used are presented. Throughout this paper, ideas about future work have already been suggested, including further investigation into the cause of initial transport layer throughput variability. The feasibility of incorporating propagation delay at the MAC layer was discussed in section III-A.

Additionally, during the experiments presented in this paper, we observed that at times during the drive-by experiment a TCP sender failed to find a route to the receiver. This occurred even at tested distances where the nodes previously were able to find a route. Adding an ICMP “ping” prior to attempting to connect resolved this issue; however, it is unclear why this was necessary. One theory is that the “ping” packet forced an ARP update thus allowing the nodes to connect. Future work includes evaluating the number of failed and successful drive-by connections without the use of the “ping” and whether this is a transient or persistent behavior.

Currently only the free space path loss model is implemented in the simulation daemon. We plan in the future to include additional path loss models such as the Two-Ray model. This model uses a path loss exponent of 2 (free space model) for near sight and 4 (flat earth model) for far sight [21]. Inclusion of this model will enable more comprehensive simulation experiments.

VII. Acknowledgments

The author wishes to thank Professor Jeffrey Hollingsworth and Professor Howard Elman for their support of this research. In addition, this work would not have been possible with out the assistance of Brenton Walker and Matt Seligman, two of the three MeshTest creators. Finally, Dr. Lloyd Wood's assistance with dtnd and related DTN topics is greatly appreciated.

REFERENCES

[1] S. Farrell and V. Cahill, *Delay and Diruption Tolerant Networking*. Artec House, 2006.

[2] "DTN Research Group." http://www.dtnrg.org/.

[3] V. Cerf et al, "Delay Tolerant Network Architecture," *Internet Draft draft-irtf-dtnrg-arch-07*, October 2006.

[4] K. A. Harras and K. C. Almeroth, "Transport layer issues in delay tolerant mobile networks.".

[5] "The network simulator - ns-2." htpp://www.isi.edu/nsnam/ns.

[6] "Opnet technologies." www.opnet.com.

[7] "Glomosim." htpp://pcl.cs.ucla.edu/projects/glomosim.

[8] T. C. Clancy and B. D. Walker, "Meshtest: Laboratory-based wireless testbed for large topologies," in *IEEE TridentCom 2007*, pp. 1–6, 2007.

[9] "Orbit- trac.".

[10] M. Seligman, B. D. Walker, and T. C. Clancy, "Delay-tolerant network experiments on the meshtest wireless testbed," in *CHANTS '08: Proceedings of the third ACM workshop on Challenged networks*, (New York, NY, USA), pp. 49–56, ACM, 2008.

[11] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the 2003 IEEE Workshop on Sensor Network Protocols and Applications*, pp. 30–41, 2003.

[12] B. Walker and C. Clancy, "A quantitative evaluation of the meshtest wireless testbed," in *TridentCom '08: Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*, (ICST, Brussels, Belgium, Belgium), pp. 1–6, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[13] W. I. J. M. Lloyd Wood, Wesley M. Eddy and C. Jackson, "*Saratoga*: a delay-tolerant networking convergence layer with effieicent link utilization," in *Delay Tolerant Networking Session, Third International Workshop on Space and Satellite Communications*, 2007.

[14] M. Ramadas, et. al, "Licklider Transmission Protocol - Specification," *work in progress as an IETF internet draft*, April, 2007.

[15] B. Walker and J.Seastrom, "Addressing scalability in a laboratory-based multihop wireless testbed." Under Submission TridentCOM 2008, 2008.

[16] T. Clancy and B. Walker, "Meshtest: Laboratory testbed for large wireless topologies," IEEE TridentCOM 2007, 2007.

[17] W. R. Stevens, *TCP/IP Illustrated Volume 1 The Protocols*. Addison Wesley Longman, 1994.

[18] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf - the tcp/udp bandwidth measurement tool." http://dast.nlanr.net/Projects/Iperf/.

[19] G. Xylomenos and G. C. Polyzos, "Tcp and udp performance over a wireless lan," pp. 439–446, 1999.

[20] R. S. Sorev Bansal and A. A. Kherani, "Performance of tcp and udp protocols in multi-hop multi-rate wireless networks," in *IEEE WCNC 2004*, 2004.

[21] H. L. Bertoni, *Radio Propagation in Modern Wireless Systems*. Prentice Hall, 1999.