

Experimental Evaluation of Peer to Peer Applications in Vehicular Ad-hoc Networks

Eugenio Giordano *
Univ. di Bologna, WiLab, DEIS
Bologna, Italia
&
UCLA - CSD
Los Angeles, California
giordano@cs.ucla.edu

Abhishek Ghosh
University of California Los
Angeles
Computer Science
Department
Los Angeles, California
aby@cs.ucla.edu

Giovanni Pau
University of California Los
Angeles
Computer Science
Department
Los Angeles, California
gpau@cs.ucla.edu

Mario Gerla
University of California Los
Angeles
Computer Science
Department
Los Angeles, California
gerla@cs.ucla.edu

ABSTRACT

Peer to Peer (P2P) applications are already very popular in the wired internet environment. They can prove advantageous in vehicular ad-hoc networks as well. However existing P2P applications are not well suited for wireless ad-hoc networks. Therefore it is crucial to better understand the limitations of P2P applications in wireless ad-hoc environment, investigating the reasons for their non optimal performance. With this study we attempt to assess the underlying flaws with P2P applications in ad-hoc networks. In order to do so we present the results obtained with our experimental campaign for both static and mobile cases.

¹ ²

1. INTRODUCTION

Military tactical and other security-sensitive operations are some of the main applications of ad-hoc networks. However, there is a trend to adopt ad-hoc networks for commercial uses due to their unique properties. By placing special electronic devices in vehicles, we can provide ad-hoc connectivity between a set of closely grouped cars. Each vehicle equipped with this device will act as a node in the

*This work was supported by Microsoft Research through its PhD Scholarship Programme.

¹Conference name: ISVCS 2008, July 22 - 24, 2008, Dublin, Ireland.

²Copyright C 2008 ISBN 978-963-9799-27-1

ad-hoc network and can receive and relay others messages through the wireless network. By enabling vehicles to communicate with each other as well as with roadside base stations, these vehicular networks can aid in providing safer and more efficient roads. Vehicular Ad-hoc Networks can implement a variety of wireless technologies such as Dedicated Short Range Communications (DSRC), Cellular, Satellite, and WiMax. Vehicular ad-hoc Networks form a component of the Intelligent Transportation Systems (ITS). By merging ad-hoc connectivity with Peer to Peer networking, we can develop a vehicular network which can feature self organisation, scalability, fault tolerance, sharable resources and services and cost efficiency. With this paper we attempt to state the underlying flaws with P2P applications in ad-hoc networks as they were designed for the wired internet. The paper is organized as follows: In section 2 we describe the equipment we used as well as the experimental scenarios we set up. In section 3 we present the results obtained with our experimental campaign for both static and mobile cases. In the experiments conducted, we attempted to observe how some of the Peer to Peer applications scale with regard to some popular TCP applications, in a wireless ad-hoc vehicular network.

2. EXPERIMENT SETUP

2.1 Equipment

In our experiments we used 6 Latitude D-600 laptops all equipped with a belkin 802.11 b/g MIMO wireless card. We also used 2 Microsoft webcams to stream the video from the cars. Each laptop runs Windows XP Professional. The multi-hop connectivity is achieved using the OLSR Daemon application [1]. In table 1 we show the parameters we used for the OLSR daemon for both the static and the mobile experiments we performed. In order to test the maximum throughput in a multi-hop scenario we used Iperf [2], that

Table 1: OLSR parameters

| Parameter | Static Experiments | Mobile Experiments |
|----------------|--------------------|--------------------|
| Hello Interval | 1.00 sec | 1.00 sec |
| Hello Hold | 25 sec | 5 sec |
| TC interval | 3 sec | 1 sec |
| TC Hold | 375 sec | 5 sec |

can generate both UDP and TCP flows. Additionally we investigated the behavior of Peer to Peer (P2P) protocols like BitTorrent [5] in ad-hoc networks with Azureus [4] that is a well-known open source BitTorrent [5] client. Azureus also provides the possibility to use a decentralized tracker for the seeding of the shared files, this allows to share files among the nodes that are in the ad-hoc network with no need of an internet connection. Additionally we used Video Lan Client (VLC) [7] to transfer a video stream from a webcam mounted on a car.

2.2 Static Experiments

In our first experiment we tried to establish what are the throughput limits for multi-hop data transfer. To do so we arranged the nodes in a linear configuration, as shown in figure 1, where the single-hop connections are represented by the light blue bold lines. We tested the maximum possible throughput with Iperf [2]. Iperf creates a single end to end TCP or UDP connection between source and destination (represented by the arrows in figure 1), but the packets are actually routed through the middle nodes. With the



Figure 1: Iperf Connections (Bold: single hop scenario; Single Dashed: 2 hops scenario; Dashed-Dotted: 3 hops scenario)

same experiment setup we tested the maximum download speed achievable with Azureus. To investigate the impact of multi-hop on P2P protocols we put into practice three different scenarios with increasing number of nodes trying to download the same 3.19GB file from a single source placed at one end of the network. In figure 2, we show the end to end connections created by the nodes participating in the P2P network.

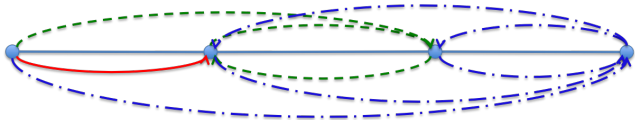


Figure 2: Azureus Connections (Bold: single node scenario; Single Dashed: 2 nodes scenario; Dashed-Dotted: 3 nodes scenario)

2.3 Mobile Experiments

In the second set of experiments, we investigated the impact of mobility on P2P ad-hoc networks. In order to do so we placed four nodes on the four corners of Engineer IV building at UCLA. In this configuration each node could reach only the two nodes that are next to it. This means that in order to reach the farther node, 2 hops are required. With this setup we performed 3 different trials:

2.3.1 2 Fixed Sources and 2 Cars

In this trial we had 2 cars moving around the Engineer IV building, as shown in figure 3. In this experiment we used Azureus to test the download speed in a mobile scenario. We placed the same 50 MB file on 2 source nodes at 2 opposite corners, then we set the remaining 4 nodes (2 mobile and 2 fixed) to download the file. In this scenario the downloading nodes are always one hop away from at least one source.

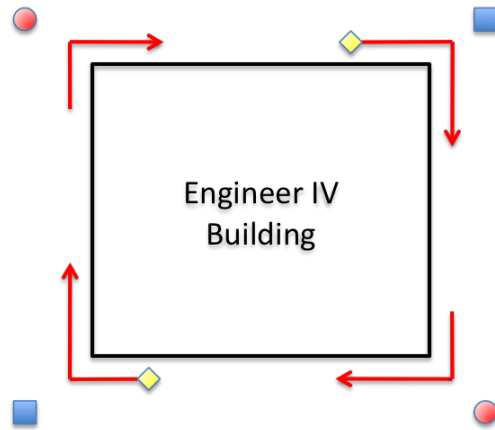


Figure 3: Mobility Experiment: 2 fixed sources (Circles), 2 fixed downloading nodes (Squares) and 2 moving downloading nodes (Diamonds)

2.3.2 1 Fixed Source and 1 Car

In this trial we had only one car moving around the Engineer IV building as shown in figure 4. In this experiment we again used Azureus. We placed a 100 MB file on a fixed source node and downloaded it from the other 3 fixed nodes and from the car. With this setup 2 nodes will be at 1 hop distance from the source but they will have to act as relays for the node placed in the opposite corner, that is 2 hops away from the source. The moving node instead will alternate periods of one hop distance with periods of 2 hops distance. The car will most likely not act as a relay for the fixed downloading nodes, because for the fixed nodes a path to the source is always present (through the other fixed nodes) and will be used to route the packets.

2.3.3 Video Streaming

In this trial we used the same setup as the one described in section 2.3.2. As shown in figure 5 we placed a webcam in the moving car and used VLC to stream the video to one of the fixed nodes. With this setup the car is always connected to the network and at most 2 hops away from the receiving node. The webcam was generating a video stream at resolution of 176x144 pixels at 15 frames per second. Thus

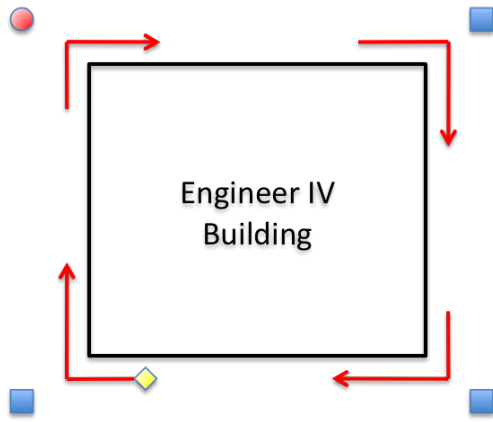


Figure 4: Mobility Experiment: 1 fixed source (Circle), 3 fixed downloading nodes (Squares) and 1 moving downloading node (Diamond)

the stream was generating an average of 128 Kbps (since the codec used was DIV3 the bitrate was not constant due to dynamic compression). The video was streamed using UDP so the lost frames were not retransmitted. The VLC server was set with a cache of 200ms.

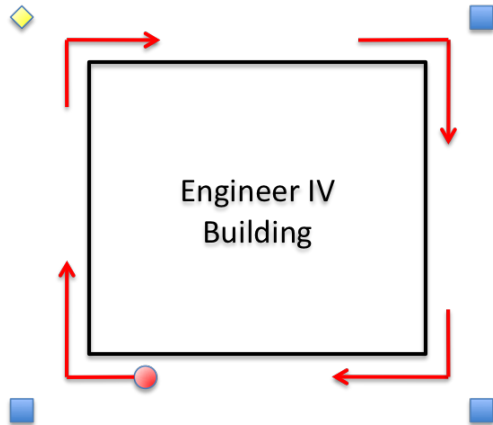


Figure 5: Video Streaming Experiment: 1 moving video source (Circle), 3 fixed nodes providing connectivity (Squares) and 1 fixed receiving node (Diamond)

3. RESULTS

In this section we show the results we obtained in the several scenarios described in section 2.

3.1 Static Experiments

In figure 6 we show the minimum, average and maximum throughput obtained using iperf in the static linear scenario for both UDP and TCP traffic. As shown in figure 6 UDP performs better than TCP in all cases. Moreover, regardless of the transport protocol used, the throughput drastically decreases as the number of hops increases. In figures 7,

8 and 9 we show the results obtained in the linear static scenario with Azureus. We use iperf as a term of comparison in order to have an idea of the maximum throughput that can be achieved. However iperf establishes only an end to end connection, Azureus instead, as shown in figure 2, establishes many more connections. Moreover Azureus is transferring the data to all the destination nodes, while iperf is only transferring the data to a single destination. Thus, iperf tends to overestimate torrent performance (unless the torrent manages to download from separate sources). For another upperbound, we introduce the concept of "Ideal Transfer Rate". The "ideal transfer rate" is the transfer rate we would obtain if the the transfer would be managed in the ideal manner (i.e. move the file 1 hop at a time). In figure 7 is shown the average download speed for a node placed just one hop away from the source, and it is comparable with the TCP iperf transfer rate that in this case is equal to the "ideal transfer rate". In figure 8 we show the average download speed for a 2 hop scenario. In this scenario, with Azureus, the node at 1 hop distance from the source is both downloading and acting as a relay for the node placed 2 hops away. As shown in figure 8 the download speed is much lower for the 1 hop node but it is slightly higher for the node placed 2 hops away as it is downloading from both the source and the 1 hop node. Furthermore the overall download time is comparable with the time needed to simply transfer the file to the first node and then transfer it again to the second node as shown by the "ideal transfer rate". In figure 9 we show the same results for a 3 hops scenario. In this case however the number of connections generated by Azureus is much higher, as shown in figure 2. Thus the overall download speed decreases drastically due to MAC collisions. From this results we can conclude that in multi-hop scenarios P2P protocols perform best within 2 hops. Thus P2P protocols designed for ad-hoc wireless networks should establish connections more than 2 hops away only with very low probability. BitTorrent [5] and thus Azureus applies the policy "rarest first" in selecting the chunks to download. That is one of the causes of the frequent multi-hop connections. Torrent protocols designed on purpose for ad-hoc networks, such as CarTorrent [3], use a "rarer-closer" that trades off rarity with hops and was proven to improve the performance. We have used Azureus in this experiment for because Azureus is a published protocol and is thus more stable than CarTorrent. Future extensions of this work will explore CarTorrent and the "rarer-closer" heuristics.

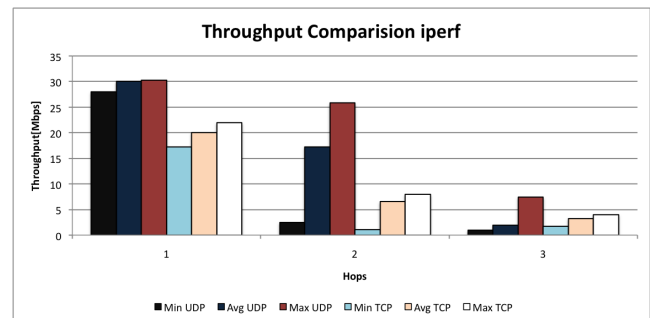


Figure 6: Iperf Throughput

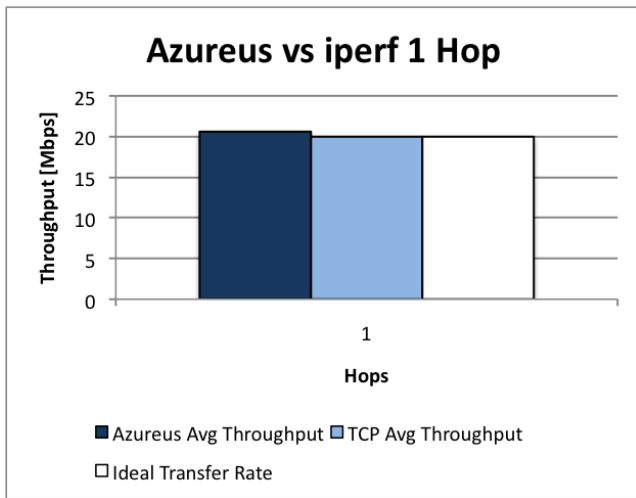


Figure 7: Comparison between Azureus and Iperf in a 1 hop scenario

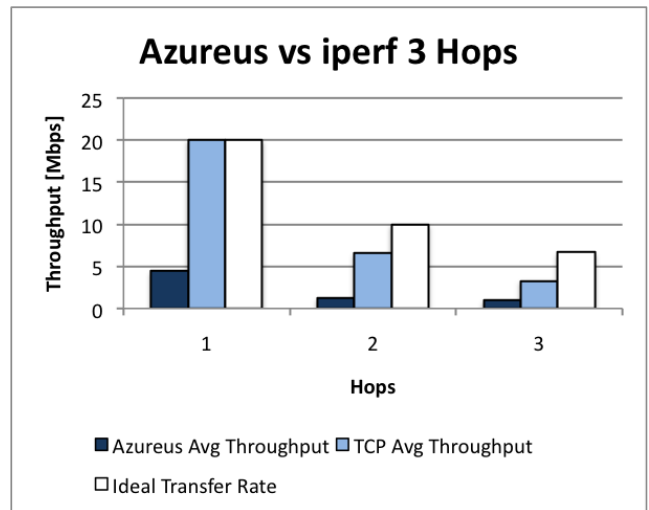


Figure 9: Comparison between Azureus and Iperf in a 3 hop scenario

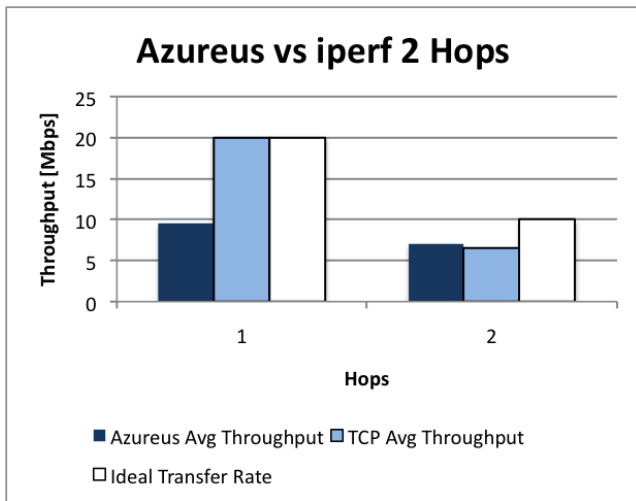


Figure 8: Comparison between Azureus and Iperf in a 2 hop scenario

3.2 Mobile Azureus Experiments

Figure 10 refers to the scenario described in section 2.3.1 and shows the average download speed for both the fixed nodes and the moving nodes. The average download speed is much higher for the fixed nodes. In fact the fixed nodes are always 1 hop away from both the sources, instead the mobile nodes are always one hop away from one source and 2 hops away from the other.

Figure 11 refers to the scenario described in section 2.3.2 and shows the average download speed for both the fixed nodes and the moving node. The average download speed is much higher for the moving node. This happens because one of the fixed nodes is 2 hops away from the source and the other two fixed nodes act as relays hence decreasing their download speed. On the other hand, the mobile node is 1 hop away from the source for half of the time and most likely never acts as a relay for the reasons explained in section 2.3.2;

3.3 Mobile Video Streaming

On the receiving node we were both saving and displaying the video. In the real time video the missing frames were much more than 10% but since we were saving the raw data received from the source we were able to reconstruct and re-encode the video received. In figure 12 we show the percentage loss for the video after the reconstruction. As shown in figure 12 the percentage of loss for both frames and blocks is approximately 10%. Such a loss still grants the possibility of actually displaying the video. In real time then the video was not displayable due to fact that the frames were not always delivered in order (due to UDP), this issue could be solved using a more reliable transport protocol or implementing a longer buffer for the frames. Another important detail is when the frame losses occurred. In fact they occurred when the mobile node was swapping from one relay to another. This means that the refresh of the route is not

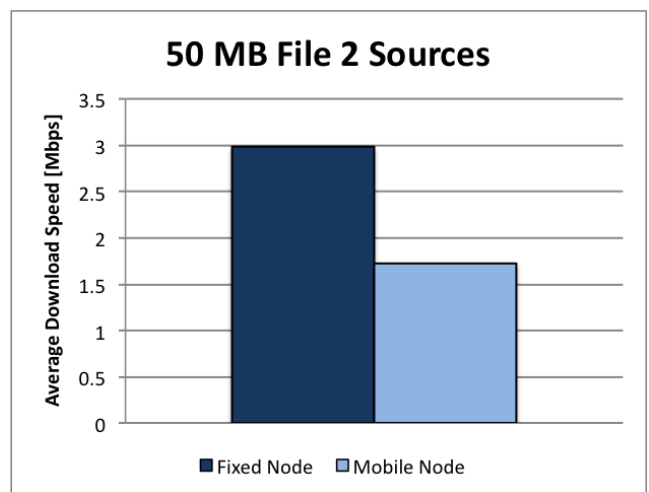


Figure 10: Average download speed in the 2 sources scenario

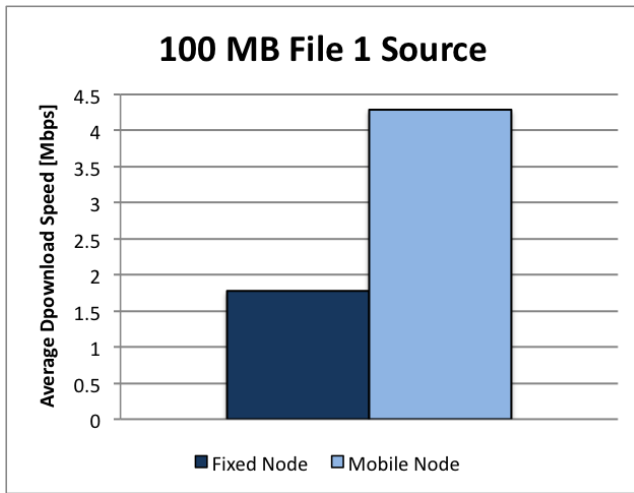


Figure 11: Average download speed in the 1 source scenario

fast enough to be transparent for the video stream.

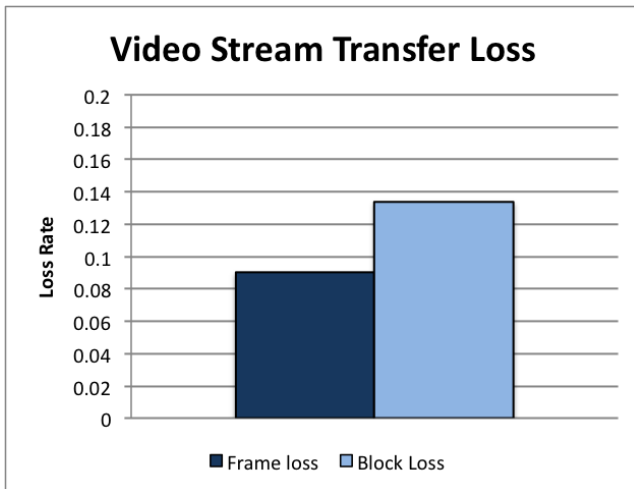


Figure 12: Loss Rate for the video stream

4. CONCLUSIONS AND FUTURE WORK

In this report we presented the results of our set of experiments. We can conclude that P2P protocols are extremely useful in ad-hoc networks. However the performance of these P2P protocols decreases with increasing number of hops. For this reason, P2P protocols should be specifically designed for ad-hoc networks in order to prevent the use of paths having a large number of hops. Next we plan to extend our experiments to a wider set of protocols and applications for example CarTorrent [3] and SopCast [6]. Additionally we would like to port our framework to a linux environment and use a larger number of fixed and mobile nodes. Further we would like to test several other configurations such as static star structure, and finally analyze the performance of a hybrid network (both ad-hoc and infrastructured).

5. REFERENCES

- [1] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). *RFC3626*, 2003.
- [2] <http://dast.nlanr.net/Projects/Iperf/>.
- [3] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. *Wireless On-demand Network Systems and Services*, 2005.
- [4] www.azureus.com.
- [5] www.bittorrent.com/protocol.html.
- [6] www.sopcast.org.
- [7] www.vlc.org.