A further comparison of different TCP/IP and DTN protocols over the D-STAR Digital Data mode

John Ronan, EI7IG and Cathal O'Connor
Telecommunications Software &
Systems Group
Waterford Institute of Technology
Cork Road, Waterford, Ireland
{jronan,coconnor}@tssg.org

Abstract

This work examines the performance of the Digital Smart Technologies for Amateur Radio - Digital Data mode with various IP and non-IP based protocols. A throughput comparison was performed between TCP/IP and two DTN convergence Layers. The experimental results show that the DTN NORM Convergence Layer exhibits better performance than TCP/IP and TCP based convergence layers, and, furthermore appears to be more suited for use on difficult radio links.

Index Terms

Disruption-tolerant networking, Internetworking; TCPIP; Transport Protocols; D-STAR

I. Introduction

The Icom Digital Smart Technologies for Amateur Radio (D-STAR [1]) family of transceivers and the use of the D-STAR protocol is becoming more and more an integral part of the toolbox used by Amateur Radio operators for emergency communications activities. The D-STAR Digital Data (DD) mode (in the Icom ID-1 transceiver) is of interest as the radio transceiver presents an ethernet interface, and thus any protocol that can be transmitted over ethernet can be sent between any pair of ID-1 transceivers.

In the event of more than two transceivers operating on a single channel, which is likely in an emergency communications scenario, there would likely be a lot of traffic on a channel all in contention for the same bandwidth. This would be necessary in the early stages of an incident until normal communications links were restored.

Most D-STAR deployments include a DD Gateway which act as a repeater and also Internet Gateway for deployed ID-1 transceivers in an area. However we have taken the approach of experimentally measuring the impact on the TCP/IP suite of protocols (TCP in particular) and some Disruption-tolerant networking (DTN) protocols, of operating multiple transceivers on a single channel in this type of environment. Previously [2], some initial results of experiments with DTN and IP networking using Icom ID-1 transceivers in Digital Data mode were presented. These results were limited to a control or "ideal" test set-up, and testing over a Non-line-of-sight (NLOS) link. As the "ideal" test results were available, the approach taken in this work was to construct "real" network of 4 nodes all operating on a single channel. This was considered to be typical of an *ad hoc* network, rapidly put together for in response to an incident or other event.

II. Background

The authors interest in DTN stems from the potential of DTN to be used to support emergency communications activities, especially where multiple different network types converge i.e. AX.25 [3], D-STAR and the set of 802.11 standards [4] that make up what is commonly referred to as "WiFi".

In this paper we compare the performance of the TCP/IP protocols, TCP [5], [6], versus two DTN Convergence Layer implementations namely TCP-CL [7] and NACK-Orientated Reliable Multicast Transport Protocol (NORM) [8].

A. Disruption/delay tolerant networking

Disruption or Delay Tolerant Networking (DTN), is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity or other extreme environments. Some issues to be addressed include large delay for transmissions resulting from either physical link properties or extended periods of network partitioning, routing capable of operating efficiently with frequently-disconnected, pre-scheduled, or opportunistic link availability, high per-link error rates making end-to-end reliability difficult, heterogeneous underlying network technologies (including non-IP-based internets). The DTN architecture [9] uses in-network or node-level storage to provide an overlay network over various types of network infrastructures. This node-level storage allows application messages (*bundles* in the DTN architecture) to be stored on DTN gateways (or nodes) for arbitrary lengths of time, while waiting for a forward path to become available. This clearly differs from the IP model where IP packets must be forwarded immediately, or dropped. The Delay-Tolerant Networking Research Group (DTNRG)[1] has a reference implementation of the protocol [10] available for experimentation, extension and real-world deployment. See [11] for more information on DTNs.

B. Digital Smart Technologies for Amateur Radio (D-STAR)

Digital Smart Technologies for Amateur Radio, commonly known as D-STAR, is a digital voice and data protocol specification, published in 2001, which was developed as the result of research funded by the Japanese government and managed by the Japan Amateur Radio League [12]. The purpose of the research was to investigate digital technologies for amateur radio. While there are other digital on-air technologies being used by amateurs that have come from other services, D-STAR is one of the first on-air and packet-based standards to be widely deployed and sold by a major radio manufacturer that is designed specifically for amateur service use.

The D-STAR system supports two types of digital data streams. The Digital Voice (DV) stream used for example on 430-440 MHz contains both digitised voice (3600 bps including error correction) and digital data (1200 bps). Using a DV radio is like having both a packet link and FM voice operating simultaneously. The Digital Data (DD) stream, used only on 1200MHz, is entirely data with a bit rate of 128k bps. An Ethernet connection is used as the interface for high-speed D-START Digital Data.

This work is solely concerned with the Digital Data mode available on the Icom ID-1 transciever.

III. Experimental Network

Figure 1 shows the area where the experiments were conducted and the location of the nodes. Figure 2 shows the experimental network used to measure the system performance. Each node in the network consisted of an Icom ID-1 transceiver and a Linux PC. In our testing, both the DTN reference implementations TCP Convergence Layer (TCP-CL) and the NORM Convergence Layer (NORM-CL) were used to investigate DTN performance. NORM was chosen for examination as previous research [13] suggests that NORM would be suited for use in networks that are bandwidth constrained, or networks that suffer from high levels of packet loss. The Iperf [14] and Wget [15] tools were used to test TCP. While Iperf is more of a "network test tool". Wget is effectively, in this case just a http client and it attempts to pull down a file from a web server. Several separate network configurations were examined:

[1] www.dtnrg.org

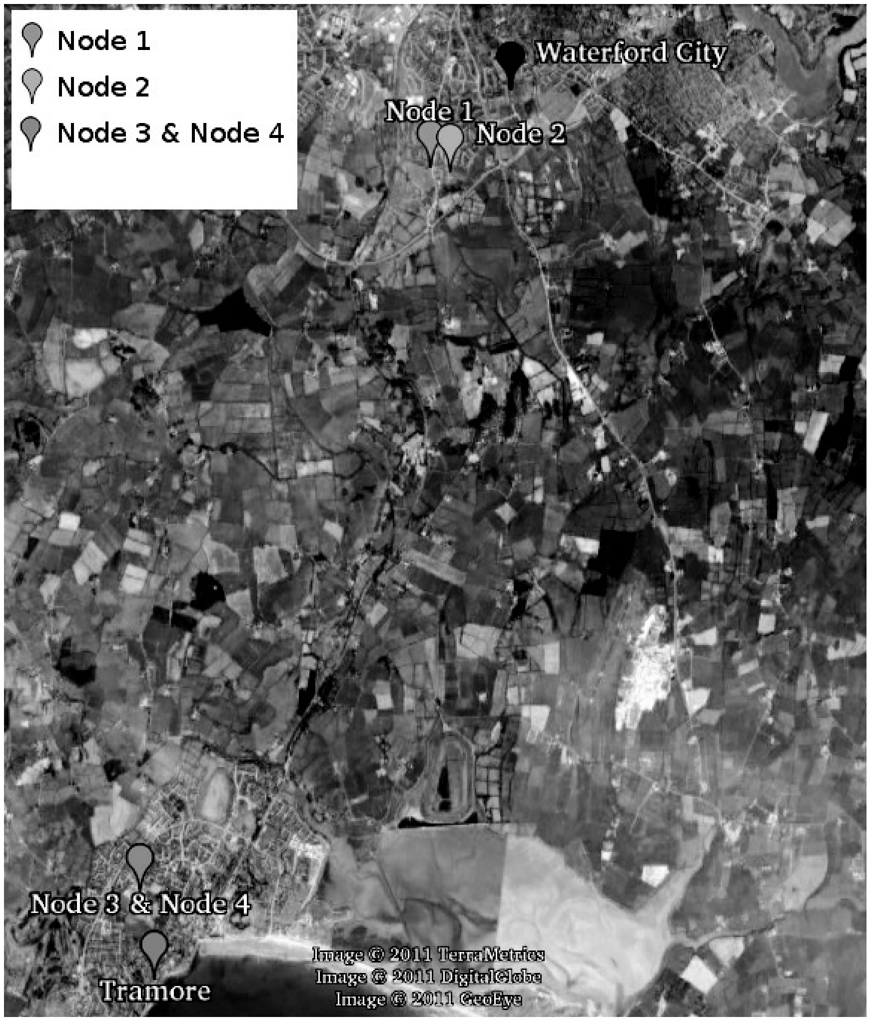

Fig. 1. Map of nodes

Control

This entailed placing two radios in close proximity on the bench using dummy loads for aerials.

Point-to-Point

This was a 220m link (approx.), from Node 1 to Node 2.

Single Hop

This included the link between Node1 and Node 2 and added a 8.5km hop (approx.), from Node 2 to Node 3.

Double Hop

This included both links above with a short hop from Node 3 to Node 4.

Single Hop with interfering node

This final test was simply where Node 2 was transferring data to Node 3. Every 2 minutes, Node 3 would also receive data from Node 4. Node 4 could not be heard by Node 2 and thus was effectively causing deliberate interference to Node 2.

The "control" configuration was investigated with both radios operating indoors in an ideal environment. For this point-to-point test, no discovery or routing mechanisms were needed.

For the tests involving just TCP/IP all routing was configured manually. For the DTN convergence layers, the discovery mechanisms were not used, however *dtlsr* the DTN routing mechanism was

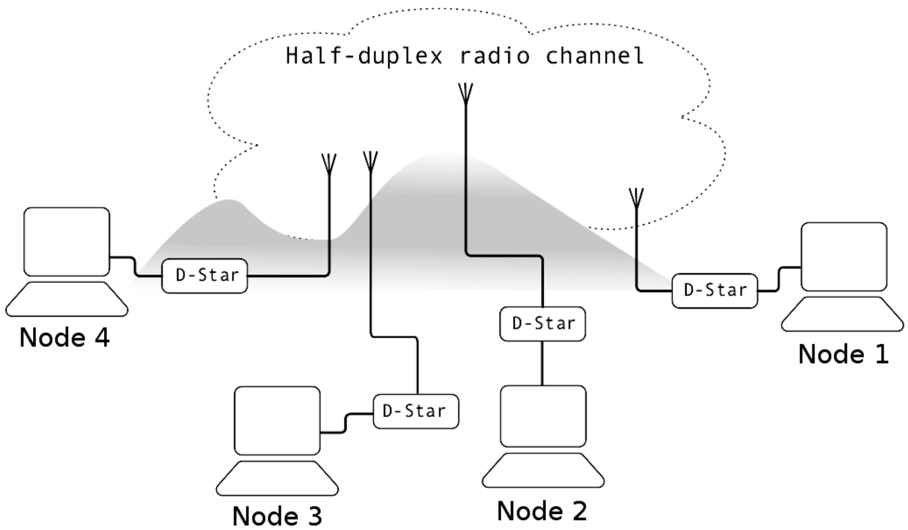

Fig. 2. Experimental network

configured with defaults which meant that each node broadcast a route "announcement" once per hour.

As per figure 2, the testbed was configured with Linux nodes and Icom ID-1 transceivers at 3 separate locations.

Location 1
: Node 1, a GuruPlug [16] Server running Debian GNU/Linux 6.0, Icom ID-1 transceiver and a Diamond X5000 aerial.

Location 2
: Node 2, an Intel Atom based Notebook running Ubuntu 10.04 LTS, Icom ID-1 transceiver and a Diamond X5000 aerial.

Location 3
: Node 3, an Athlon based Laptop running Ubuntu 10.04 LTS, Icom ID-1 transceiver and a Diamond X5000 aerial.

Location4
: Node 4, an Intel Atom based Notebook running Ubuntu 10.04 LTS, Icom ID-1 transceiver, and a tri-band Magmount aerial. Co-located with Node 3.

Node 1 could occasionally be heard by Node 3 and vice-versa, initially the signal was not strong enough for a reliable connection, or even successful packet decodes (tcpdump). However, later on in the testing it was noticed that the occasional ARP [17] request, and response, made it across the link between nodes 1 and 3, the machines immediately attempted to communicate directly, but the link seemed unable to carry full IP packets. Once the issue was understood, static ARP mappings were put in place and the nodes were configured through the Linux sysctl interface to ignore ICMP [18] redirection messages. Nodes 2 and 3, while not quite line-of-sight, were always a good connection with a ping time in the order of 64ms. Node 4 was co-located with Node 3, with Node 4 connected to a magnetic antenna and running low power so that it could not be heard by nodes 1 or 2. This would have been done with Node 1, however, Node 1 developed a cooling problem and had to be "retired".

The following tests were done in the *Point-to-Point*, *Single Hop* and *Double Hop* network configurations:

- Iperf
- Wget
- TCP Convergence Layer
- NORM Convergence Layer

Wget was not run in the *control* tests [2], and, for the final test, *Single Hop with interfering node* it was deemed unnecessary to run three independent TCP based protocols, so only NORM and Wget were used.

Iperf was developed by National Laboratory for Applied Networking Research//Distributed applications Support Team (NLANR/DAST) as a tool for measuring maximum TCP and UDP bandwidth performance.

GNU Wget is a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it is easily called from scripts.

Each test was repeated 25 times to get an average throughput figure for that particular protocol. Care was taken to run the tests under similar atmospheric conditions. The Iperf tool was used to test TCP only, Wget was used to approximate a HTTP connection. The results for Iperf were generated with the following command run in a loop 25 times:

```
iperf -c 192.168.2.11 -t 600 -i 10
```

Where *192.168.2.11* was the IPv4 address of the destination node (Node 2) and *192.168.2.10* was the source addresses.

The result was a report, with a summary line similar to the following:

```
[  3]  0.0-605.7 sec  2.28 MBytes  31.6 Kbits/sec
```

To do a full HTTP/TCP test, the Wget utility was used to retrieve a 6MB file, the following command was, again run in a loop, with the start and end times being recorded:

```
wget 192.168.2.11/ftp_file_6mb
```

The result was a report, similar to the following:

```
###########  25 ############
Sun Jul 17 03:59:13 UTC 2011
Sun Jul 17 04:40:03 UTC 2011
```

To test the DTN Convergence Layers the *dtnsend* utility was used to send the same 6MB file as used in the TCP tests, across the link. *dtnsend* was configured to ask for a delivery receipt, thus confirming reception of the file at the destination node.

```
dtnsend -e 21600 -w -D -s dtn://node1.dstar.dtn/me \
-d  dtn://node2.dstar.dtn/hitme -t f -p ftp_file_6mb
```

and the result of this was a report similar to the following:

```
got 33 byte report from [dtn://node1.dstar.dtn/]: time=639445.0 ms
```

From these results a spreadsheet was compiled and all results were then converted into kilobits per second.

IV. Results & Discussion

Looking at Table I, “Control” results seem in line with general expectations. The DTN TCP-CL average throughput is slightly less than IPv4, i.e. the DTN overhead on an IPv4 packet is increasing the overall duration of the transfer. The NORM result is interesting, it is using "unreliable" UDP, yet it performs significantly (almost 15%) better. Though NORM is intended for reliable multicast delivery of file or stream objects, it is being used here for unicast delivery. NORM's ability to function with much less end-to-end interactivity than TCP alllows for more efficient use of wireless links [13].

> *The NORM protocol is designed to provide end-to-end reliable transport of bulk data objects or streams over generic IP multicast routing and forwarding services. NORM uses a selective, negative acknowledgement (NACK) mechanism for transport reliability and offers additional protocol mechanisms to conduct reliable multicast sessions with limited "a priori" coordination among senders and receivers.*[2]

The † in Table I indicates that NORM's rate control mechanism was configured to use a transmission rate of 84kbps. In previous work [2] it was determined that the best performance from NORM over a D-STAR link could be achieved by configuring NORM with this transmission rate.

TABLE I
CONTROL & POINT-TO-POINT RESULTS

Control			
Protocol	Min (kbps)	Max (kbps)	Average (kbps)
Iperf	66.4298	67.5769	67.0002
TCP-CL	63.3455	64.8974	64.3969
NORM-CL (84)†	76.7097	77.5527	77.1651

Point-to-point			
Protocol	Min (kbps)	Max (kbps)	Average (kbps)
Iperf	67.3000	69.0000	68.0000
Wget	55.7911	66.7826	63.0611
TCP-CL	52.0730	67.6054	64.3216
NORM-CL	74.0174	79.0396	78.0389

In the point-to-point results in Table I, it can be seen that on "real-world" links, Iperf, Wget and TCP-CL seem to perform similarly, with slight reductions in throughput compared to the "control". NORM however seems to have increased its throughput compared to the "control" tests. This is interesting considering the TCP-CL has remained approximately the same. One possible explanation for this is that the linux-based computers used to generate the "control" were much older and may not have been as efficient at processing UDP datagrams as the computers in use for this work.

TABLE II
D-STAR PERFORMANCE ON A SINGLE-HOP LINK

Protocol	Min(kbps)	Max (kbps)	Average (kbps)
Iperf	9.9200	34.9000	27.2808
Wget	19.9157	29.5385	26.0009
TCP-CL	15.3129	38.2490	25.1616
NORM-CL	37.4994	38.4632	38.1211

In Table II, where a single hop is introduced we can see that the Iperf throughput has dropped by $\approx 60\%$, Wget by $\approx 59\%$, TCP-CL by $\approx 60\%$, NORM by $\approx 51\%$. Note the approximate 10% advantage that NORM has over TCP based protocols.

[2]http://cs.itd.nrl.navy.mil/work/norm/

TABLE III
D-STAR PERFORMANCE ON A TWO-HOP LINK

Protocol	Min (kbps)	Max (kbps)	Average (kbps)
Iperf	4.5900	18.1000	12.3292
wget	2.0192	19.6216	11.9851
TCP-CL	3.5409	32.3209	13.8637
NORM-CL	22.7116	25.4541	24.9753

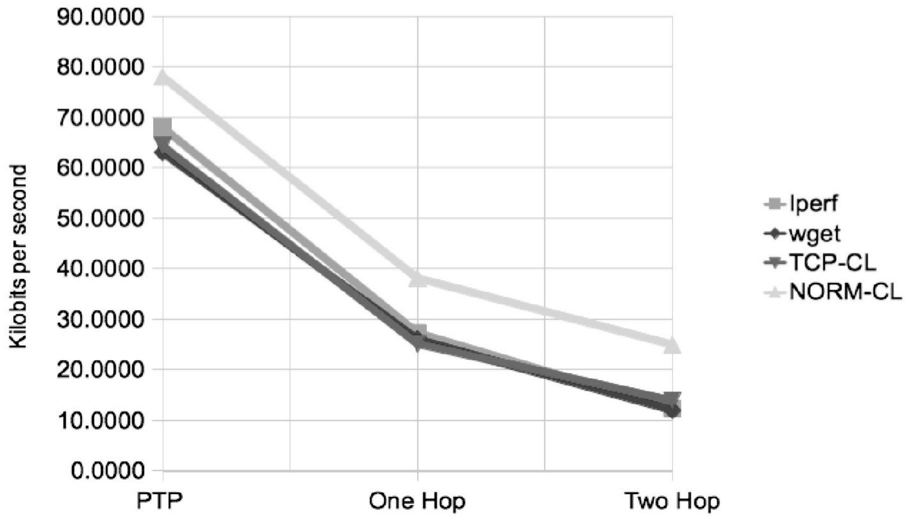

Fig. 3. Summary of results

In Table III, where a second hop is introduced we can see that the performance of Iperf has dropped a further $\approx 55\%$, wget by $\approx 54\%$, TCP-CL by $\approx 45\%$, NORM by $\approx 35\%$. That gives a total degradation of $\approx 82\%$ for Iperf, $\approx 81\%$ for Wget, $\approx 78\%$ for TCP-CL and finally $\approx 68\%$ for NORM. These results are graphed in Figure 3.

TABLE IV
TCP AND NORM PERFORMANCE — SINGLE HOP WITH INTERFERING NODE

Protocol	Min (kbps)	Max (kbps)	Average (kbps)
Wget	59.5782	62.1391	60.7134
NORM-CL	66.3910	71.7619	69.5582

Finally Table IV shows the figures for throughput from Node 2 to Node 3, in parallel with the 6MB file transfers, Node 4 transfers a 64kbyte file over TCP to Node 3 every 2 minutes. It can be seen that Wget shows only a 4% degradation, while NORM shows an 11% degradation in throughput. This is interesting of itself and will probably require further investigation. In spite of this, NORM still maintains an almost 13% advantage over Wget.

V. CONCLUSION

From previous work, it was seen the DTN NORM Convergence Layer showed signs of being more efficient than the TCP/IP protocol over DD mode D-Star radio links. A 12% to 15% improvement using NORM over TCP is significant enough, what was not expected was a dramatic difference between the robustness of NORM vs TCP. In this work we attempted to do an evaluation of NORM versus TCP in a more “real world” scenario. The locations for the nodes were chosen in the hope that they would cause difficulty, which indeed they did. Looking back at the results, the TCP tests they appear to be broadly in line with what would be expected, in that the throughput is best in the Iperf, then Wget, then TCP-CL due to the extra overhead imposed. On Icom ID-1 transceivers, NORM appears to have a optimal transmission rate of 84kbps which gives anywhere from 12 to 15% improvement over TCP in our testbed.

For future work, it would be useful to compare two other DTN Protocols, Saratoga [19], developed by Surrey Satellite Technology Ltd and NASA Glenn Research Centre, and the Licklider Transmission Protocol [20], while also looking at the work being done in the High-Speed Multimedia (HSMM) area, and performing a useful comparison.

VI. Acknowledgements

The authors would like to thank Nicky Madigan, EI3JB, for allowing us to place a node at his house for the duration of our experiments. This work was partly funded by the HEA Research Facilities Enhancement Scheme, 2008.

References

[1] Icom America, “D-STAR General Information,” accessed on 2010-08-10. [Online]. Available: http://www.icomamerica.com/en/products/amateur/dstar/dstar/default.aspx

[2] J. Ronan and C. O’Connor, “A comparison of different TCP/IP and DTN protocols over the D-Star Digital Data mode,” in *Proceedings of 29th ARRL and TAPR Digital Communications Conference*. ARRL, September 2010, pp. 134–138.

[3] W. A. Beech, D. E. Dielsen, and J. Taylor, “AX.25 Link Access Protocol for Amateur Packet Radio,” AX.25 Link Access Protocol for Amateur Packet Radio, version 2.2 Revision July 1998, 1998, accessed on 2010-08-10. [Online]. Available: http://www.tapr.org/pdf/AX25.2.2.pdf

[4] IEEE, “802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” accessed on 2010-07-07. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[5] J. Postel, “Internet Protocol,” RFC 791 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFC 1349. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[6] K. Nichols, S. Blake, F. Baker, and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474 (Proposed Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 3168, 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2474.txt

[7] M. Demmer, “Delay Tolerant Networking TCP Convergence Layer Protocol,” Internet Engineering Task Force, Nov. 2008. [Online]. Available: http://tools.ietf.org/id/draft-irtf-dtnrg-tcp-clayer-02.txt

[8] B. Adamson, C. Bormann, M. Handley, and J. Macker, “NACK-Oriented Reliable Multicast (NORM) Transport Protocol,” RFC 5740 (Proposed Standard), Internet Engineering Task Force, Nov. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5740.txt

[9] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-Tolerant Networking Architecture,” RFC 4838 (Informational), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[10] K. Scott and S. Burleigh, “Bundle Protocol Specification,” RFC 5050 (Experimental), Internet Engineering Task Force, Nov. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[11] F. Brickle, “A brief introduction to delay tolerant networking,” in *27th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, 2008, pp. 6–8.

[12] “Japan Amateur Radio League,” accessed on 2009-07-20. [Online]. Available: http://www.jarl.or.jp/English/

[13] C. Rigano, K. Scott, J. Bush, R. Edell, S. Parikh, R. Wade, and B. Adamson, “Mitigating naval network instabilities with disruption tolerant networking,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008, pp. 1–7.

[14] National Laboratory for Applied Network Research, “Iperf,” http://iperf.sourceforge.net/, accessed on 2010-08-10.

[15] Free Software Foundation, “Gnu wget,” http://www.gnu.org/s/wget/, accessed on 2011-08-04.

[16] “GuruPlug Server,” accessed on 2011-07-31. [Online]. Available: http://www.globalscaletechnologies.com/t-guruplugdetails.aspx

[17] D. Plummer, “Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware,” RFC 826 (Standard), Internet Engineering Task Force, Nov. 1982, updated by RFCs 5227, 5494. [Online]. Available: http://www.ietf.org/rfc/rfc826.txt

[18] J. Postel, “Internet Control Message Protocol,” RFC 792 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 950, 4884. [Online]. Available: http://www.ietf.org/rfc/rfc792.txt

[19] L. Wood, W. M. Eddy, W. Ivancic, J. Mckim, and C. Jackson, “Saratoga: a delay-tolerant networking convergence layer with efficient link utilization,” in *Third International Workshop on Satellite and Space Communications (IWSSC 07*, 2007.

[20] M. Ramadas, S. Burleigh, and S. Farrell, “Licklider Transmission Protocol - Specification,” RFC 5326 (Experimental), Internet Engineering Task Force, Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5326.txt