Evaluating OLSR and B.A.T.M.A.N over D-STAR

John Ronan, EI7IG
Telecommunications Software & Systems Group
Waterford Institute of Technology
Cork Road, Waterford, Ireland
jronan@tssg.org

Darren Long, G0HWW
Bury St. Edmunds
England
Email: darren.long@mac.com

***Abstract*—This work examines the performance of both the Optimized Link State Routing (OLSR) protocol and the Better Approach To Mobile Ad hoc Networking (B.A.T.M.A.N.) protocol over the Digital Smart Technologies for Amateur Radio - Digital Data mode (D-Star DD). A comparison was performed with differing parameters to evaluate what impact, if any, they would have on overall goodput, over and above the default settings. In this scenario, with multiple nodes, the experimental results show that the housekeeping data being transmitted by both protocols can significantly and adversely impact the available bandwidth on the channel.**

***Index Terms*—MANET; 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. This allows for approximately 68kbps [2] of goodput between any pair of transceivers, much higher than via 1200 or 9600 AX.25 [3] baud packet radio.

The authors interest in the use of mesh protocols over D-STAR Digital Data mode comes from the potential of mesh networking 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”.

Due to the terrain around the area (East County Waterford, Ireland) where the nodes were deployed, it quickly became obvious that some sort of mesh network protocol would be required in order that the network be as self-configuring as possible, otherwise mistakes could be made in configuration that would render the equipment inoperable, or, at best, delay proper operation.

As OLSR [5] is widely used, it seemed a good candidate to investigate. However the default OLSR parameters appeared to be very aggressive and it was suspected that they use more bandwidth than is really necessary for our purposes.

On reviewing some of the literature in the area, we found some work being done in the areas of Vehicular and Mobile ad hoc Networks (VANETs & MANETs). Specifically these works related to tuning of OLSR parameters and the impact of the routing component of OLSR leading to more efficient data transport.

The authors of [6] were mostly concerned with optimizing for Route Change Latency (RCL) i.e. the time needed to determine a new route after a link failure, and its dependence on routing protocol settings. They performed some tests with several OLSR configurations, and then derived an analytical model for evaluating the impact on end-to-end connectivity on an ad hoc network. More importantly, the configuration parameters they used seem to be a basis for later works to evaluate themselves against.

In contrast to [6], [7] analyzes the impact of the hello and topology control intervals to ascertain their impact on overhead and route convergence time. [8] present the use of a meta-heuristic algorithms [9] as efficient techniques to solve this optimization problem. Usefully, the output of their solutions are compared against the parameters used in [6], and also defined in the OLSR RFC [5], but also it provides us with a maximum and minimum range of parameter values to investigate.

The authors in [7] suggest that the point that the topology control (TC) timer has a much higher impact on overheads then the hello timer, and in certain scenarios increasing the TC timer can significantly lower overheads while only marginally increasing the route convergence time, [10] and other works largely agree, though they do also mention that for a network with lots of mobile nodes, lower values mean faster route convergence time. In our case it was envisaged that it should be possible to increase the TC interval for a significant resource saving with a marginal increase in the "settling" or route convergence time, so that the network can begin to support traffic subsequent to deployment or potentially a network reconfiguration.

B.A.T.M.A.N [11] was borne from experience with OLSR, but (as the name hints), attempts to take the experience and knowledge gained with large OLSR deployments and improve on it. The approach of the B.A.T.M.A.N algorithm is to divide the knowledge about the best end-to-end paths between nodes in the mesh to all participating nodes. Each node perceives and maintains only the information about the best next hop towards all other nodes. Thereby the need for a global knowledge about local topology changes becomes unnecessary. This is quite different to OLSR.

In order to investigate the issue, a network was constructed of three fixed nodes and one mobile node, all using omni-directional aerials. Due to unexpected delays, the fourth node was never used for experiments.

The rest of this paper is organized as follows: in §2 we briefly explain the D-STAR, OLSR and B.A.T.M.A.N. concepts, §3 gives an overview of the test scenarios, §4 presents our results, §5 our discussion and §6 our conclusions.

II. BACKGROUND

A. 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 digitized 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-STAR Digital Data.

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

B. Optimized Link State Routing Protocol (OLSR)

Optimized Link State Routing Protocol (OLSR) is a routing protocol for mobile ad hoc networks. It runs on community wireless mesh networks with such as the German FreiFunk.net [13], and also the HSMM-MESH™ [14].

OLSR is an optimization of the classical link state algorithm tailored to the requirements of a mobile wireless Local Area Network. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes which forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. In OLSR, link state information is generated only by nodes elected as MPRs. Thus, a second optimization is achieved by minimizing the number of control messages flooded in the network. Essentially each node behaves like a smart APRS digipeater, in that MPRs are like "elected", wide area digipeaters. There is also a third optimization, where an MPR node may chose to report only links between itself and its MPR selectors. So, contrary to the classic link state algorithm, partial link state information is distributed in the network. This information is then used for route calculation. OLSR provides optimal routes (in terms of number of hops).

The olsr.org [15] OLSR daemon is is an implementation of the Optimized Link State Routing protocol. Hence it allows for mesh routing to take place over for any network device supported by the underlying operating system.

C. Batman

As stated above, B.A.T.M.A.N "comes from" OLSR[1]. Its development was driven due to limitations that became apparent with OLSR once deployed in large networks (hundreds of nodes). Due to the constant growth of existing community mesh networks and because of the inherent requirement of a link-state algorithm to recalculate the whole topology-graph (a particularly challenging task for the limited capabilities of embedded router HW), the limits of this algorithm have became a challenge. Recalculating the whole topology graph once, in an actual mesh with several hundred nodes, can take several seconds on a small embedded CPU. Though, it has to be noted that this was not a particular problem in our test environment.

The approach of the B.A.T.M.A.N algorithm is to divide the knowledge about the best end-to-end paths between nodes in the mesh, to all participating nodes. Each node perceives and maintains only the information about the best next hop towards all other nodes. Thereby the need for global knowledge of local topology changes becomes unnecessary. Additionally, an event-based but timeless (timeless in the sense that B.A.T.M.A.N never schedules nor timeouts topology information for optimizing it's routing decisions) flooding mechanism prevents the build-up of contradictory topology information (the usual reason for the existence of routing loops) and limits the amount of topology messages flooding the mesh (thus avoiding the extra overhead of control-traffic). The algorithm is designed to deal with networks that are based on unreliable links.

[1]http://www.open-mesh.org/projects/open-mesh/wiki/The-olsr-story

The protocol algorithm of B.A.T.M.A.N can be described (simplified) as follows. Each node transmits broadcast messages (called originator messages or OGMs) to inform the neighboring nodes about it's existence. These neighbors are re-broadcasting the OGMs to inform their neighbors about the existence of the original initiator of this message. Thus the network is flooded with originator messages. OGMs contain at least the address of the originator, the address of the node transmitting the packet, a TTL and a sequence number.

OGMs that follow a path where the quality of wireless links is poor or saturated will suffer from packet-loss or delay on their way through the mesh. Therefore OGMs that travel on "good" routes will propagate faster and more reliably.

In order to tell if a OGM has been received once or more than once it contains a sequence number, given by the originator of the OGM. Each node re-broadcasts each received OGM at most once and only those received from the neighbor which has been identified as the currently best next hop (best ranking neighbor) towards the original initiator of the OGM.

In this manner, the OGMs are flooded selectively through the mesh and inform the receiving nodes about other node's existence. A node X will learn about the existence of a node Y in the distance by receiving it's OGMs, when OGMs of node Y are rebroadcasted by it's single hop neighbors. If node X has more than one neighbor, it can tell by the number of originator messages it receives more quickly and more reliable via one of its single hop neighbors, which neighbor it has to choose to send data to the distant node.

The algorithm then selects this neighbor as the currently best next hop to the originator of the message and configures its routing table respectively.

Due to the "linear" layout of our test network, most of this functionality was not really of importance, however, it was noticed that B.A.T.M.A.N. never "lost" a route to another host, while OLSR frequently did.

III. Experimental Network

Fig. 1. Map of nodes, EI7IG/M is co-located with EI7IG

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. The Iperf [16] tool was used to generate TCP test traffic.

Several separate network configurations were examined:

Control (static routing)
: This was a 8.5km link (≈ 5 miles), from EI3JB to EI7IG.

Point-to-Point
: This was also the 8.5km link from EI3JB to EI7IG.

Relay
: This included the link between EI3JB and EI7IG and added a short hop $\approx 10-15$m, from EI7IG to EI7IG/M. EI7IG/M was also running low power and a magnetic antenna (as though operating from a vehicle).

For the control point-to-point and relay tests, all routing was statically configured. EI7IG was the traffic generator for the former, EI7IG/M for the latter.

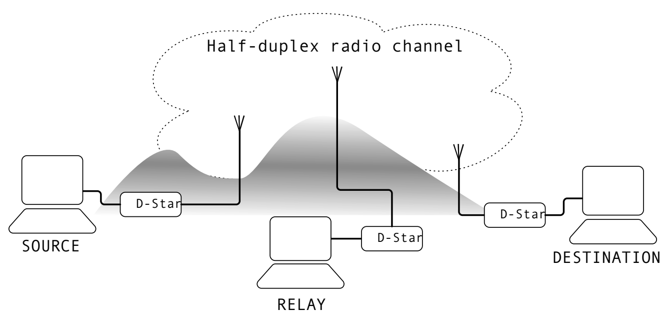

Fig. 2. Experimental network

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

Location 1- Destination
: Node 1 (EI3JB), a Laptop running Ubuntu 12.04 LTS, Icom ID-1 transceiver and a Diamond X5000 aerial.

Location 2 - Relay
Node 2 (EI7IG), an Intel Atom based Mini-ITX running Ubuntu 12.04 LTS, Icom ID-1 transceiver and a Diamond X5000 aerial.

Location 3 - Source
Node 3 (EI7IG/M), a Laptop running Ubuntu 12.04 LTS, Icom ID-1 transceiver and a Diamond magmount aerial.

Node 3 was co-located with Node 2, with Node 3 connected to a magnetic antenna and running low power so that it could not be heard by Node 1.

As a starting point, the parameters defined in the RFC [17] were used, as per tables I and II.

TABLE I
EMISSION INTERVALS

Parameter Name	Time (Seconds)
HELLO_INTERVAL	2
TC_INTERVAL	5
MID_INTERVAL	TC_INTERVAL
HNA_INTERVAL	TC_INTERVAL

TABLE II
HOLDING TIME

Parameter Name	Time (Seconds)
NEIGHB_HOLD_TIME	3 x REFRESH_INTERVAL
TOP_HOLD_TIME	3 x TC_INTERVAL
MID_HOLD_TIME	3 x MID_INTERVAL
HNA_HOLD_TIME	3 x HNA_INTERVAL

The emission interval parameter REFRESH_INTERVAL, was not obviously changeable from the configuration file and was thus ignored. The Holding time parameter, DUP_HOLD_TIME, was also not obviously changeable from the configuration file and was ignored.

The following tests were done in the *Point-to-Point* configuration:

- No OLSR
- OLSR
  - Hello - 2, TC/MID/HNA 5 (Default)
  - Hello - 4, TC/MID/HNA 5
  - Hello - 6, TC/MID/HNA 5
  - Hello - 8, TC/MID/HNA 5
  - Hello - 10, TC/MID/HNA 5
  - Hello - 10, TC/MID/HNA 10
  - Hello - 10, TC/MID/HNA 20
  - Hello - 10, TC/MID/HNA 30
  - Hello - 30, TC/MID/HNA 30
- Batman
  - 1 Second OGM interval
  - 2 Second OGM interval
  - 4 Second OGM interval
  - 8 Second OGM interval
  - 10 Second OGM interval
  - 30 Second OGM interval

The following tests were done in the *Relay* configuration:

- No OLSR
- OLSR
  - Hello - 2, TC/MID/HNA 5 (Default)
  - Hello - 10, TC/MID/HNA 30
  - Hello - 30, TC/MID/HNA 30
- Batman
  - 2 Second OGM interval
  - 10 Second OGM interval
  - 30 Second OGM interval

A. IPERF

Iperf [16] 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.

Each test was repeated a minimum of 5 times in order to get an average throughput figure for that particular protocol and configuration. Care was taken to run the tests under similar atmospheric conditions. The Iperf tool was used to test TCP only. The results for Iperf were generated with the following commands (run 5 times):

```
iperf -c <destination> -t 300
sleep 30
iperf -c <destination> -t 300 -i  -d
```

Where *destination* was the IPv4 address of the destination node, 44.155.6.228. From these results a spreadsheet was compiled and all results were then converted into kilobits per second.

IV. RESULTS & DISCUSSION

The results of the “point-to-point” tests contain nothing unexpected and are all broadly in line with the results from a previous paper [2] for Iperf TCP

TABLE III
POINT-TO-POINT RESULTS

Control			
Protocol	Min (kbps)	Max (kbps)	Avg. (kbps)
Static	66.4	69.6	67.96

OLSR			
Settings	Min (kbps)	Max (kbps)	Avg. (kbps)
Hello 2,5	60.30	64.40	62.08
Hello 4,5	65.30	66.20	65.78
Hello 6,5	64.90	66.70	66.00
Hello 10,5	64.30	67.10	66.14
Hello 10,10	64.40	67.30	66.12
Hello 10,20	66.90	67.70	67.32
Hello 10,30	66.30	67.60	67.10
Hello 30,30	67.20	70.10	68.76

Batman			
OGM Interval	Min (kbps)	Max (kbps)	Avg. (kbps)
1	57.70	58.80	58.42
2	63.10	64.30	63.74
4	65.20	66.80	65.96
6	66.80	67.70	67.44
8	67.80	68.70	68.36
10	68.30	69.20	68.68
30	67.50	68.90	68.20

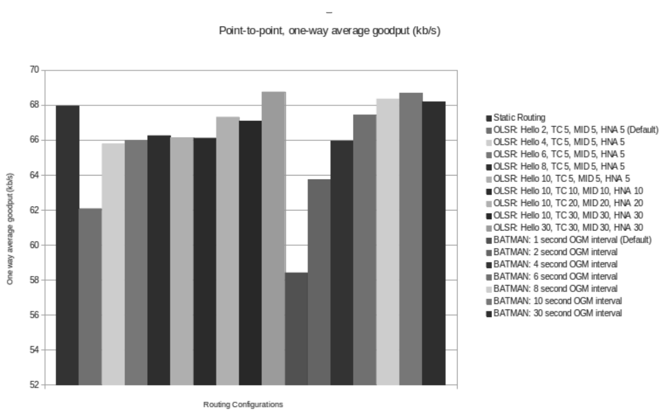

Fig. 3.

transfers. The bi-directional Iperf TCP transfers, on average, show transfers from Node 2 to Node 1 contributing slightly more to the aggregate goodput than the competing transfers from Node 1 to Node 2, with an overall aggregate goodput contribution averaged across all routing strategies of $\approx 61.3\%$ suggesting some asymmetry in the link.

In the two-hop (relay) tests, the goodput achieved varies significantly between the different routing configurations. Static routing achieves broadly expected results for unidirectional goodput

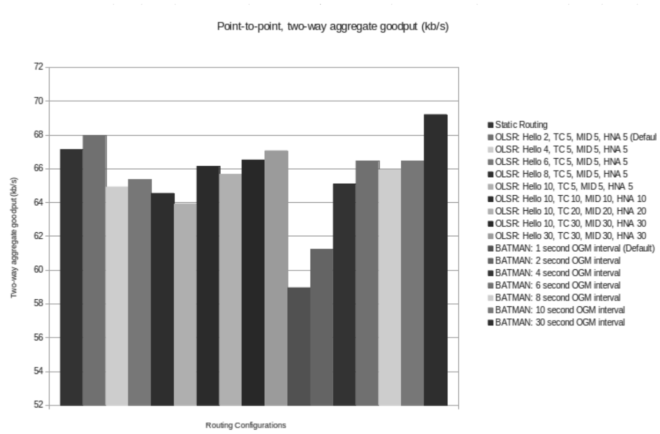

Fig. 4.

and bidirectional aggregate goodput, i.e. roughly half of the rates achieved for the single hop topology and is consistent with the expected doubling of the channel utilization.

TABLE IV
RELAY

Control			
Protocol	Min (kbps)	Max (kbps)	Avg. (kbps)
Static	27.00	37.5	34.86

OLSR			
Settings	Min (kbps)	Max (kbps)	Avg. (kbps)
Hello 2,5	0.45	2.05	1.07
Hello 10,30	2.88	16.8	9.39
Hello 30,30	2.71	7.77	5.59

Batman			
OGM Interval	Min (kbps)	Max (kbps)	Avg. (kbps)
2	0.49	0.54	0.52
10	3.30	21.3	9.02
30	30.7	36.6	34.56

For both the OLSR and B.A.T.M.A.N. routing methods in the "relay" topology, we see a considerable drop in one-way goodput and aggregate bi-directional goodput that improves a little with reduced routing state distribution overhead.

The topology is essentially one with mutually hidden transmitters affecting end-to-end transfers in both directions. Of note is that for the two-hop bidirectional tests, transfers from Node 3 to Node 1 yield an overall aggregate goodput contribution of $\approx 89\%$. It is thought that this can be explained by the link margin advantage of roughly 40dB (estimated) for the Node-3/Node-2 link over the

TABLE V
RELAY - BIDIRECTIONAL AGGREGATE GOODPUT

Control			
Protocol	Min (kbps)	Max (kbps)	Avg. (kbps)
Static	31.2	40.09	37.0

OLSR			
Settings	Min (kbps)	Max (kbps)	Avg. (kbps)
Hello 2,5	10.48	13.60	12.45
Hello 10,30	12.7	21.32	17.68
Hello 30,30	6.46	16.74	13.21

Batman			
OGM Interval	Min (kbps)	Max (kbps)	Avg. (kbps)
2	13.54	14.76	14.23
10	22.21	26.70	23.49
30	22.12	45.09	33.31

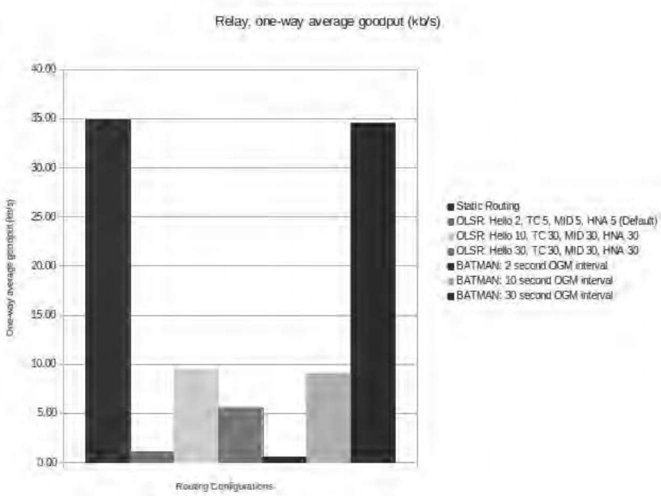

Fig. 5.

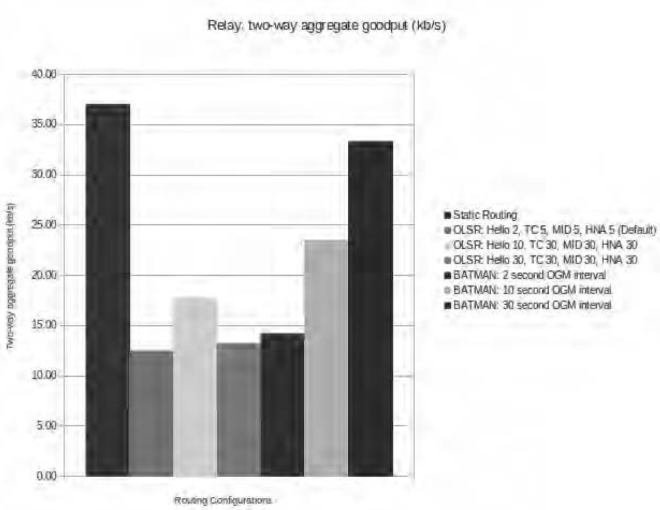

Fig. 6.

Node-1/Node-2 link. Ostensibly then, the network allows for traffic from the Node-3 to Node-1 flow, traversing the Node-3 to Node-2 link, to hog the Node-2 relay and consequently the shared channel. TCP ACK traffic from the same flow traversing the two-hop path in the reverse direction competes with a 40dB (estimated) disadvantage in the hidden transmitter "competition" and always loses, causing congestion avoidance mechanisms to back off, reducing the goodput achieved by the Node-3 to Node-1 TCP flow. During these back off periods some goodput is achieved by the competing Node-1 to Node-3 flow, but these opportunities are scarce and very little results.

The aggressive, higher overhead OLSR and B.A.T.M.A.N. configurations achieve poor goodput even in the unidirectional tests on the two-hop topology, as the routing state distribution traffic is enough to cause collision loss induced congestion avoidance in TCP on its own. The bidirectional tests show better results as the advantaged link provides a more significant contribution.

If the links were more evenly matched, the results would be expected to be worse.

V. CONCLUSION

In this paper, we set out to evaluate OLSR and B.A.T.M.A.N. for use over D-STAR's Digital Data (DD) mode. We constructed a small, three node network with mutually hidden transmitters. We began by using the standard parameters (from RFC3626) [5] for OLSR, then increased them in order to achieve better "goodput". In light of the experimental results we can conclude that:

- For a point-to-point, or indeed point-to-multipoint configuration, where all nodes can see one another OLSR would appear to operate reasonably well.
- Once any relaying is introduced, it appears that B.A.T.M.A.N. performs better than OLSR. Based on our results, we would not expect to see any multi-hop scenario where OLSR would outperform B.A.T.M.A.N. in a DD mode network. More testing in larger networks would assist in validating or invalidating our opinion.
- In a relaying scenario, B.A.T.M.A.N. begins to perform comparably to a static configu-

ration once OGM intervals of $\approx$30 seconds are used. This obviously impacts on the convergence time of the network. Consequently as the network grows, there is a trade-off to be considered between the limited bandwidth available with DD mode, and how quickly the network is required to converge.
- On the B.A.T.M.A.N. website [11], as part of the description of the protocol concept, the following sentence is used "The algorithm is designed to deal with networks that are based on unreliable links.". Our evaluation appears to validate this claim.

Far from being complete, this paper only gives a limited snapshot of the abilities of both protocols. The test network is small, and was deliberately chosen to be "difficult", though not outside the realms of possibility. For any Amateur Radio operators attempting to use a mesh protocol with DD mode, we would strongly suggest to look at B.A.T.M.A.N. first to address your particular network idiosyncrasies.

Looking to the future, and the imminent release of North West Digital Radio's UDR56k-4 [18], it would be interesting to re-do these tests over a more intelligent link layer, and see if better efficiencies could be achieved.

VI. Acknowledgements

The authors would like to thank Nicky Madigan, EI3JB, for assisting with the experiments. This work was partly funded by the HEA Research Facilities Enhancement Scheme, 2008.

References

[1] Icom America, "D-STAR General Information," accessed on 2013-07-31. [Online]. Available: http://www.icomamerica.com/en/products/amateur/dstar

[2] J. Ronan and C. O'Connor, "A further comparison of different TCP/IP and DTN protocols over the D-Star Digital Data mode," in *Proceedings of 30th ARRL and TAPR Digital Communications Conference*. 225 Main Street, Newington, CT 06111-1494, USA: ARRL, September 2011, pp. 72–79.

[3] W. A. Beech, D. E. Dielsen, and J. Taylor, "AX.25 Link Access Protocol for Amateur Packet Radio, version 2.2 Revision July 1998," 1998, accessed on 2013-07-31. [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] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[6] C. Gomez, D. Garcia, and J. Paradells, "Improving performance of a real ad-hoc network by tuning olsr parameters," in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, june 2005, pp. 16 – 21.

[7] S. Demers and L. Kant, "Manets: Perfrormance analysis and management," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, oct. 2006, pp. 1 –7.

[8] J. Toutouh, J. Garcia-Nieto, and E. Alba, "Intelligent olsr routing protocol optimization for vanets," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 4, pp. 1884 –1894, may 2012.

[9] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003. [Online]. Available: http://doi.acm.org/10.1145/937503.937505

[10] J. Hosek and K. Molnar, "Investigation on olsr routing protocol efficiency," in *Proceedings of the 2011 international conference on applied, numerical and computational mathematics, and Proceedings of the 2011 international conference on Computers, digital communications and computing*, ser. ICANCM'11/ICDCC'11. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2011, pp. 147–153. [Online]. Available: http://dl.acm.org/citation.cfm?id=2047950.2047972

[11] "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)," http://www.opeon-mesh.org/, accessed on 2013-07-31.

[12] "Japan Amateur Radio League," accessed on 2013-07-31. [Online]. Available: http://www.jarl.or.jp/English/

[13] "Freifunk," http://www.freifunk.net/, accessed on 2013-07-31.

[14] "HSMM-MESH - The broadband hamnet," http://hsmm-mesh.org/, accessed on 2013-07-31.

[15] "olsrd - and adhoc wireless mesh routing daemon," http://www.olsr.org/, accessed on 2013-07-31.

[16] National Laboratory for Applied Network Research, "Iperf," http://iperf.sourceforge.net/, accessed on 2013-07-31.

[17] R. Gellens and H. Garudadri, "The QCP File Format and Media Types for Speech Data," RFC 3625 (Informational), Internet Engineering Task Force, Sep. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3625.txt

[18] "North West Digital Radio," http://www.nwdigitalradio.com/, accessed on 2013-07-31.