

# A COMPARATIVE STUDY OF AODV AND OLSR ON THE ORBIT TESTBED

Devashish Rastogi<sup>¶</sup>, Sachin Ganu<sup>¶</sup>, Yanyong Zhang<sup>¶</sup>, Wade Trappe<sup>¶</sup>, and Charles Graff<sup>§</sup>

<sup>¶</sup>WINLAB, Rutgers University  
Technology Center of New Jersey  
North Brunswick, NJ 08902-3390

<sup>§</sup>US Army RDECOM  
CERDEC-STCD  
FT. Monmouth, NJ 07703

## ABSTRACT

*Though ad hoc network routing protocols, such as AODV and OLSR, have been extensively studied through simulations, there have been fewer evaluations of their software protocol implementations on real network testbed deployments. Consequently, validating the protocol functionality and performance on an actual experimental platform is imperative in order to understand relative merits or limitations under different network conditions. In this paper, we present our initial experimental evaluations of the publicly available AODV and OLSR implementations on the national Open Access Research Testbed (ORBIT) for Next Generation Wireless Networks. In particular, we look at maximum achievable throughput using these protocols under different topologies and dynamics. Our preliminary observations indicate that in relatively low mobility scenarios, these protocols can achieve comparable throughput, with AODV offering better stability than OLSR.*

## INTRODUCTION

In the last decade, wireless communication networking has played an important role in scenarios such as tactical situations, emergency and rescue operations, sensor networks as well as more commercial deployments to provide community-wide wireless access to the Internet. These networks can be classified as infrastructure-less mobile ad-hoc networks (MANETs) and wireless mesh networks (WMNs). A mobile ad-hoc network (MANET) consists of no stationary infrastructure like base stations, or fixed routers. Nodes form arbitrary dynamic networks using wireless links and create multihop paths between end points. Nodes may be mobile, links between nodes can change, new nodes can join the network and existing nodes can leave. WMNs, on the other hand, consist of a mesh of relatively static routers and portals to extend the coverage of conventional hotspot based deployments. A client typically connects to a mesh AP and the data traffic is routed by the mesh network to and from the Internet portal and the end client.

For both these kinds of network deployments, finding high quality paths for traffic delivery directly impacts the network performance. Traditional shortest path based routing protocols (such as OSPF, RIP) may be inadequate to han-

dle the dynamics of the network attributed to user mobility as well as wireless links. A number of routing protocols have been designed for MANETs, and these protocols can be broadly categorized into two classes: proactive routing protocols (e.g. DSDV [1], OLSR [4]) and reactive routing protocols (e.g. AODV [2], DSR [3]) More recently, these protocols have been extended to use cross layer information about link error rates (ETX [5]), bandwidths (WCETT [6]) and medium access delay (PARMA [7]) to yield high quality routes between communicating nodes.

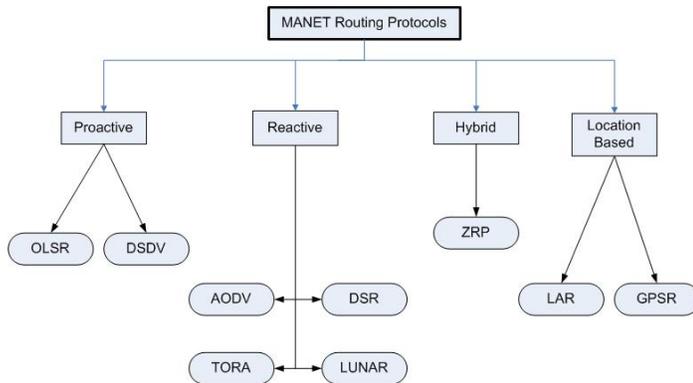
Regardless of the metric used, each protocol has to keep up with frequent, unpredictable changes in link quality and network topology caused by harsh environment or mobility. Depending on their specifications, every protocol reacts to network changes differently, and can lead to considerable variations in network performance due to the associated control overhead. It is therefore important to understand the relative merits and limitations of the protocols under different operating conditions in controlled realistic network environments that are often overlooked or oversimplified in simulation based evaluations.

Towards this end, some experiment-based studies have been conducted [8, 9] using simple linear chain topologies. Though, tests conducted during a recent study [10], did use a large number of nodes, their tests focused on the effects of unstable links on routing protocols. The goal of this study is to test the network performance in terms of achievable system throughput when using popular implementations of ad hoc routing protocols on a test bed to emulate real world large scale networks. For this purpose, AODV and OLSR have been selected, since these are the most popular and widely tested protocols. The ORBIT (Open-Access Research Testbed) testbed [13] is used to conduct the experiments.

Our experiments have shown that the throughput performance of the network with two important protocols – AODV and OLSR – is comparable up to channel saturation. Noted difference is seen their behavior beyond saturation. There is also significant difference in the stability of both protocols under the experimental conditions of the ORBIT testbed.

In the rest of this paper, we first give a brief taxonomy of routing protocols and explain the two protocols under study: AODV and OLSR in some detail. The following section describes our experimental setup, topology and measurements. We further evaluate the system throughput achieved using each routing protocol under identical topologies and network conditions. Next, we discuss our experimental results in detail. Finally, we provide the concluding remarks of the study.

## ROUTING PROTOCOLS



**Figure 1. Classification of routing protocols**

There exists a large number of MANET routing protocols [11]. They can be broadly classified into four categories as shown in Figure 1. In this study, we focus on two types of protocols:

- **Proactive (table-driven) routing.** In these types of routing protocols, each node maintains a table of routes to all destination nodes in the network at all times. This requires periodic exchange of control messages between nodes. Since the route to every destination already exists, there is little or no initial delay when first sending data. However, periodic control traffic competes with data transfer to gain access to the channel.
- **Reactive (on-demand) routing.** In reactive routing protocols, the route is calculated only when a node needs to send data to an unknown destination. Thus, route discovery is initiated only when needed. This saves overhead in maintaining unused routes. However, this may lead to larger initial delays. During route discovery, the query is flooded into the entire network and the reply from the destination (or intermediate nodes) sets up the path between the source and destination.

In this study, we specifically look at two popular protocols: AODV and OLSR. In order to understand their operation, we briefly describe both the protocol specifications and relevant implementation details.

**AODV:** Ad-Hoc On Demand Distance Vector Routing combines mechanisms like hop-by-hop routing, periodic beacons and sequence numbers to guarantee loop free routing of DSDV, on-demand route discovery, and maintenance of DSR. AODV minimizes the number of required broadcasts, making it suitable for large MANETs. AODV has two important phases: route discovery and route maintenance.

In the route discovery phase, nodes exchange periodic HELLO messages with their neighbors, which are used to establish a list of neighbors at each node. If a valid route exists between two nodes, AODV route discovery is not initiated (e.g. in single hop flows). When a route does not exist between two nodes, i.e., when a route is required to a new destination, a link has broken, or a route has expired, the source node broadcasts Route Request (RREQ) messages to its neighbors to find the destination node. As the RREQs are forwarded, each node builds a *reverse path* to the source node. This process of forwarding RREQs continues till the destination node or a node with a valid route to the destination is discovered. In order to guarantee loop free routing, the RREQs contain two counters - sequence number and broadcast ID. These two counters ensure that a node ignores multiple RREQs with the same sequence number and that the route to the destination is the most recent one. When the destination or a node with a route to it receives the RREQ message, it sends a unicast Route Reply (RREP) message containing the number of hops and latest destination sequence number. The RREP uses the reverse path built by the RREQs and builds the *forward path* to the source node. Due to the hop-by-hop nature of AODV, each node stores only the next hop information in its routing table for both forward and reverse routes. The forward path is associated with a lifetime called ACTIVE\_ROUTE\_TIMEOUT. If no packets are sent over the route during this period, the forward route is invalidated. The route is then deleted if there is no communication within the DELETE\_PERIOD. Route discovery has to be initiated again for further communication between the same source-destination pair

In the route maintenance phase, each node surveys the link status to its neighboring nodes. HELLO packets are used to maintain connectivity between neighboring nodes. A link is assumed to be broken if no HELLO packet is received within ALLOWED\_HELLO\_LOSS\*HELLO\_INTERVAL msec. This may happen due to node movement or link failure. The node detecting the failure sends Route Error (RERR) messages to its upstream neighbor on the active route. The RERRs are thus propagated upstream till the source node is reached. The source node then restarts route discovery to find a new route to the destination. Link failure can be detected either using the mechanism of HELLO packets as described above or using link

layer notification. Use of HELLO packets enables detection of link failure before data packets are forwarded, but it suffers from the disadvantage of bandwidth wastage. The default values suggested in RFC 3561[18] are as shown in the Table 1 below

**Table 1. AODV default parameters**

ACTIVE_ROUTE_TIMEOUT (this is the default initial value, it is refreshed whenever a new broadcast packet is received)	3 msec
ALLOWED_HELLO_LOSS	2
HELLO_INTERVAL	1 msec
DELETE_PERIOD	$k \cdot \max(\text{ACTIVE\_ROUTE\_TIMEOUT}, \text{HELLO\_INTERVAL})$ $k = 5$ (recommended)

We have used the AODV-UU [17], a Linux based user space implementation from Uppsala University, compliant with RFC 3561 [18]. AODV-UU runs as a user-space daemon, maintains the kernel routing table and uses kernel modules to utilize Netfilter [20] hooks to capture data packets. Routing protocol logic is in user space facilitating easier implementation. However, this approach has overheads related to context switches that require CPU intervention.

**OLSR:** Optimized Link State Routing is a proactive routing protocol based on the following three mechanisms: neighbor sensing using HELLO messages, efficient control traffic flooding using multipoint relays (MPRs), and optimal path calculation using shortest path algorithm. OLSR is independent of the underlying link layer.

Each node sends periodic HELLO messages to discover neighbors. The neighborhood of a node A contains all those nodes which are directly linked to A. The links may be symmetric or asymmetric. OLSR also uses a concept called a two-hop neighbor. A node, C, is a two-hop neighbor of A if a node B is a symmetric neighbor of A and C is a symmetric neighbor of B, but C is not a neighbor of A. The HELLO packet contains the node's own address, a list of its neighbors and the status of the links of all its neighbors. These HELLO packets are used by the nodes to generate the immediate and two-hop neighborhoods as well as to determine the quality of links in the neighborhood. This information is stored for a limited time in each node and needs to be refreshed periodically.

Flooding HELLO packets across an arbitrarily-sized MANET is costly due to the presence of multiple duplicate retransmissions. In order to avoid this, OLSR uses the concept of multipoint relay (MPR) flooding instead of full flooding. Each node uses its two-hop neighborhood infor-

mation to select a minimal set of MPRs such that all the nodes in its two-hop neighborhood are reachable. Each node maintains a list of nodes, called the MPR selector set, for which it is an MPR. The node then retransmits only those messages received from nodes which have selected it as an MPR.

The MPR flooding mechanism is also used to spread topology information throughout the MANET. All nodes with a non-empty MPR selector set periodically send out a topology control (TC) message. This message contains the address of the originating node and its MPR selector set. Thus, each node announces reachability to its MPR selectors. Since every node has an MPR selector set, effectively, the reachability to all the nodes is announced. Thus, each node receives a partial topology graph of the entire network. The shortest path algorithm is then used on this partial graph to calculate optimal routes to all nodes. The topology information is maintained only for a specific period of time and needs to be refreshed periodically. Table 2 lists the default parameters used for our evaluation.

**Table 2. OLSR default settings**

HELLO_INTERVAL	2 seconds
REFRESH_INTERVAL	2 seconds
TC_INTERVAL	5 seconds
NEIGHB_HOLD_TIME	3 x REFRESH_INTERVAL
TOP_HOLD_TIME	3 x TC_INTERVAL

For OLSR, we used the implementation from olsr.org [12], which is compliant with RFC 3626 [4]. This implementation is also based on user space routing protocol logic with *ioctl* and *proc* based interaction with the kernel routing tables.

## EXPERIMENTAL SETUP

The tests were conducted using the ORBIT testbed [13], hosted at WINLAB, Rutgers University. ORBIT is an indoor grid based wireless network emulator consisting of 400 802.11a/b/g radio nodes, which can be dynamically interconnected into specified topologies. Each node on the grid consists of a 1 GHz VIA C3 processor with 512MB RAM, 20GB hard drive, two wireless mini-PCI 802.11 a/b/g cards and two 100BaseT Ethernet ports for transfer of experimental data, control and management information. All the nodes run Debian GNU/Linux with the 2.6 kernel.

All the experiments were conducted using the ORBIT Testbed Software Architecture. The testbed provides a system consisting of a central experiment controller called Node Handler and a local client called Node Agent running on each node. The Node Handler processes the experiment script, sends commands to the Node Agents via

multicast and receives the state of command execution from each node. The Node Agent executes commands received from the Node Handler, runs applications on the node, passes parameters to applications, and reports the state of execution to the Node Handler.

Measurements are collected using the ORBIT Measurement Library (OML) framework. A Collection Server collects statistics and results that are measured by each node and sent over a multicast channel. OML provides data structures and functions to be used by the Collection Server to handle these statistics. All the results are stored in a central MySQL database, which can be accessed later for appropriate post-processing.

In this study, the ORBIT Traffic Generator (OTG) [15] and Iperf [16] traffic generators are used to generate UDP flows. OTG is a software tool designed to work with OML and generate traffic for network testing. The traffic generator system consists of two parts -- the ORBIT Traffic Generator (OTG) and the ORBIT Traffic Receiver (OTR). OTG can be used to generate bulk TCP or UDP data traffic

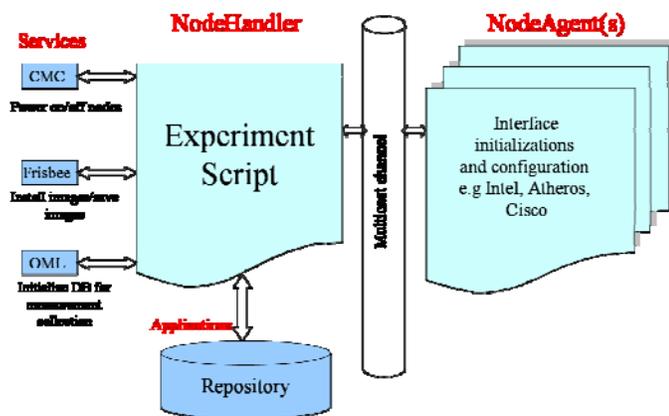


Figure 2. Node Handler – Node Agent Framework

with various characteristics including packet size, offered load, and traffic distribution (CBR or exponential traffic). Measurements like throughput, packet loss, delay, etc are directly reported by OTG to the OML library. Iperf is a popular open source traffic generator which can generate the traffic with similar characteristics as OTG. UDP traffic was used to conduct the experiments.

Popular implementations of AODV and OLSR in the public domain, as described in the earlier section, were adopted for our tests. OML functionality was integrated directly into the AODV-UU code to leverage the ORBIT measurement framework for collection of statistics. OTG was used to generate traffic for AODV tests.

The OLSR protocol does not specify a broadcast address to be used for control traffic. OLSRd implementation uses the interface broadcast address for this purpose. For exam-

ple, for an interface with address 192.168.17.16, OLSRd uses 192.168.17.255 for broadcasting control packets. This causes a problem if all the nodes are not on the same sub-network, i.e., 192.168.17.XX. This was fixed so OLSRd uses 255.255.255.255 for all interfaces. Identical traffic characteristics were used in both sets of tests.

### Emulating Multi-hop Behavior

To emulate a multihop network among closely located nodes, a MAC filter called MACKill [14] was used. The MACKill software performs MAC level filtering. A list of MAC addresses to be filtered is sent to each node. The filter rejects all the packets from the specified nodes within this list. MACKill filters 100% of the packets received from certain nodes and using this utility, we generate the topology as shown in Figure 5.

Since the aim of this study is to determine situations where certain protocols perform better than others, both AODV and OLSR were subjected to the same tests under identical conditions. More specifically, the following configurations were adopted throughout the experiments:

- Wireless card in 802.11a mode using Channel 36
- Wireless interface in Ad-Hoc mode
- Varying channel rates - 6Mbps, 24Mbps and 54Mbps
- Packet size is 1024 bytes

The Node Handler - Node Agent system (Figure 2) is used to set up these configurations on all the nodes in the experiments.

## EXPERIMENT RESULTS

### Baseline for benchmarking

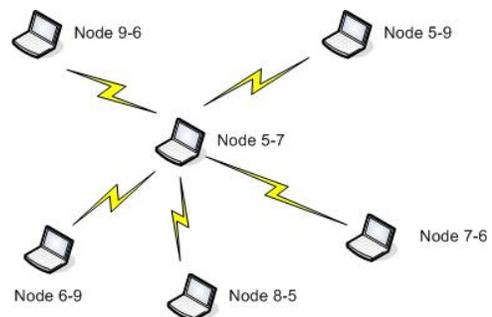


Figure 3. Baseline test Scenario

Since all the nodes in the network were configured to operate on the same channel, a set of baseline experiments were first performed in order to evaluate the raw one hop throughput of the channel at various channel rates. Five one-hop flows were created in these experiments. In this set of experiments, for each channel rate, the offered load was increased until the channel was saturated. The aggreg-

gated throughput at the central receiver node is measured every second.

**Table 3. One hop throughput results**

Channel Rate (Mbps)	Max Theoretical Throughput (Mbps)	Max. Experimental Throughput (Mbps)
54	31.2	30
24	17.8	17
6	5.41	5

Table 3 summarizes the total throughput values measured from the experiments which are in agreement with the predicted theoretical throughput, calculated as follows (using the parameter values are taken from the IEEE 802.11a standard):

Maximum throughput =  $L_{\text{payload}} / T_{\text{total}}$

$T_{\text{total}} = T_{\text{DIFS}} + T_{\text{SIFS}} + T_{\text{ACK}} + T_{\text{Backoff}} + T_{\text{Data}}$

$T_{\text{ACK}} = T_{\text{OFDM}} + L_{\text{ACK}} / r$ , where  $r$  is the channel rate

$T_{\text{Data}} = (L_{\text{Data}} + L_{\text{MAC Header}}) / r + T_{\text{OFDM}}$

Example: For channel rate,  $r = 54$  Mbps:

$T_{\text{OFDM}} = 20\mu\text{s}$

$L_{\text{ACK}} = 20$  bits

$T_{\text{ACK}} = 20 + 216/54\text{M} = 24\mu\text{s}$

$L_{\text{Data}} = L_{\text{Payload}} = 1500$  bytes

$L_{\text{MAC Header}} = 36$  bytes

$T_{\text{DATA}} = 20 + 1536*8/54\text{M} = 248\mu\text{s}$

$T_{\text{DIFS}} = 34\mu\text{s}$

$T_{\text{SIFS}} = 16\mu\text{s}$

$T_{\text{Backoff}} = 63\mu\text{s}$

$T_{\text{total}} = 34+16+248+24+63 = 385\mu\text{s}$

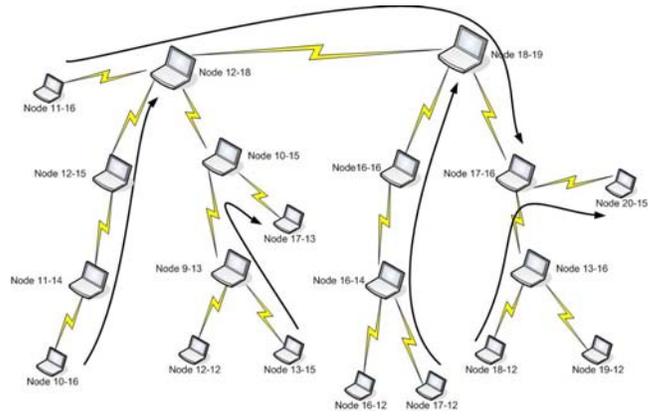
Maximum throughput =  $1500*8/385 = 31.2\text{Mbps}$

These throughput values can be used as upper bounds when conducting tests in multi-hop scenarios.

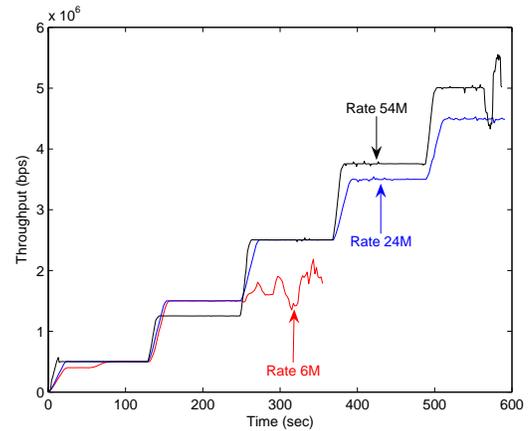
### Effect of Rate Diversity

**AODV** – In this experiment, we created a network of 20 ORBIT nodes. The topology as shown in Figure 4 has five flows, and each flow with 3 hops. We gradually increased the input offered load of each flow and conducted the experiments for 100 seconds at each setting to obtain steady-state statistics.

Figure 5 plots the total throughput for different channel rates of 6Mbps, 24 Mbps and 54 Mbps, when the offered load (from all five flows) is increased. The plot shows that the total throughput for AODV is rather stable before satu-



**Figure 4. Topology for experimental evaluation**



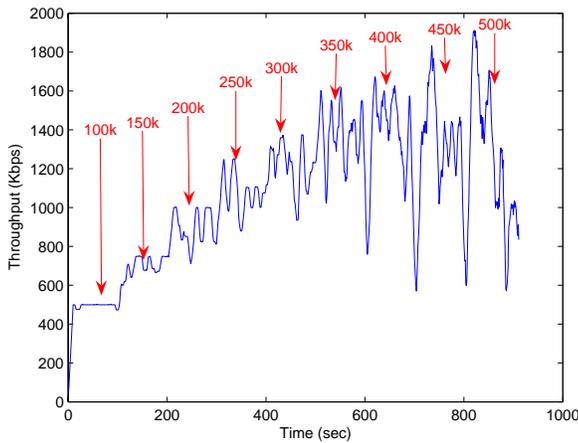
**Figure 5. Total throughput with AODV at rates 6M, 24M and 54M**

ration. In order to quantify the overhead introduced by having multiple hops, we first look at the 6 Mbps case by comparing the throughput values in Figure 5 (representing a multi-hop case) with those in Table 3 (representing a one-hop case). At the channel rate of 6 Mbps, the maximum one-hop throughput is 5 Mbps (Table 3). Since there are fifteen active links sharing the same channel, the maximum throughput per flow/link is 0.33 Mbps (5Mbps/15=0.33Mbps). In Figure 5, we observe that the per-flow throughput is 0.3Mbps (1.5/5=0.3Mbps), which corresponds to 90% of the maximum one-hop throughput. Similarly, for channel rates 24 Mbps and 54 Mbps, the saturation per-flow throughput values in these two cases are 0.9Mbps and 1Mbps, corresponding to 77.6% and 50% of one-hop per-flow throughput. These results show that channel utilization is higher for lower channel rates. This can be attributed to the fact that when the channel rate increases, the ratio of  $T_{\text{data}}$  to  $T_{\text{total}}$  decreases and hence the utilization decreases.

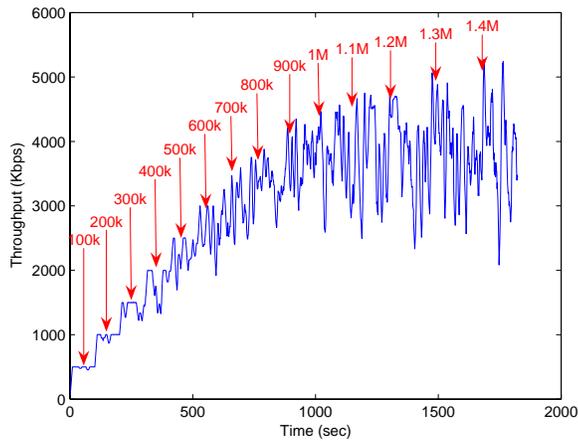
The variation in throughput at loads above saturation can be explained by the observation in [19]: as the offered load increases, the amount of control traffic, i.e., RREQ packets, increases, while the throughput goes up only by a

small amount. The increased contention for the available bandwidth by data and control traffic causes temporary breakdowns and frequent invocation of route recovery, leading to higher variation.

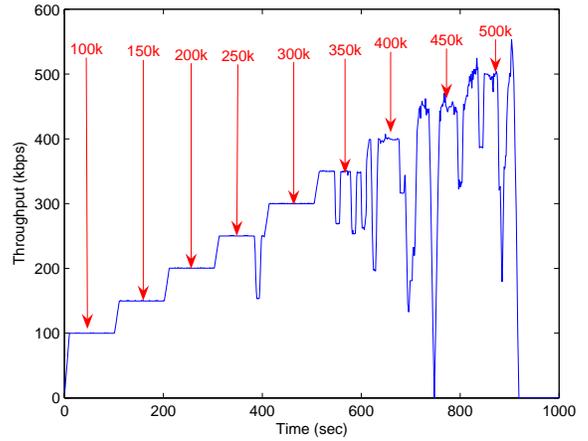
**OLSR** – Throughput experiments for OLSRd were conducted using similar configurations as those for AODV. For each channel rate, the offered load gradually increases until saturation. Figure 6 shows how the throughput changes with time when the offered load increases at the channel rate of 6Mbps, and Figure 7 shows the corresponding throughput at the channel rate of 24Mbps. Please note that we do not include the results for the case of channel rate of 54Mbps in this paper because the experiments could not successfully complete. In all the trials, one of the five flows crashed when the offered load started increasing. We are still investigating the reason for this behavior and will include the results in the final version of the paper



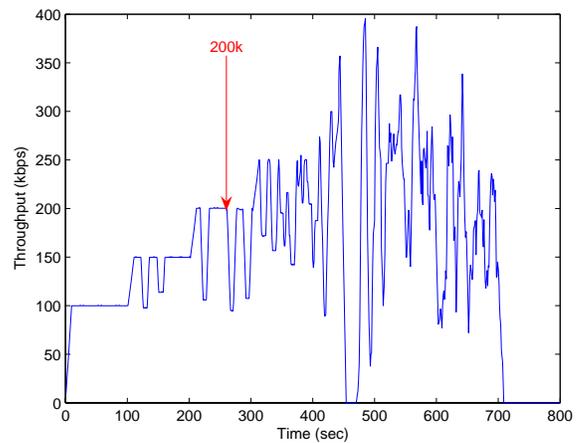
**Figure 6. Total throughput with OLSR at rate 6 Mbps**



**Figure 7. Total throughput with OLSR at rate 24 Mbps**



**Figure 8. OLSR Throughput at Node 12-18 at rate 6Mbps**



**Figure 9. OLSR Throughput at Node 18-19 at rate 6 Mbps**

In both plots, we mark the total input data rate (offered load) when there is a change in the value. Compared to AODV experiments, we ran the OLSR experiments over a much longer duration because they exhibit higher variation.

These results show that the saturation throughput levels for OLSRd are comparable to those of AODV (the saturation throughput of OLSRd is slightly lower than that of AODV). The major difference, however, lies in the fact that OLSRd is less stable and shows considerable variations in throughput. A more in-depth investigation reveals that the large variation at high offered loads is due to channel contention among the nodes. All the nodes contest for the available bandwidth causing a large number of collisions, resulting in multiplicative backoff, which tends to lead to large variations in throughput. Another factor is that the data traffic occupies all the available bandwidth, preventing control traffic like HELLO and TC packets from getting through the network. This can cause temporary breakdown of a route, which can in turn lead to the large variation as observed in Figures 6 and 7. The spikes are

due to the initial burst of traffic caused by queued packets, waiting for the route to be established.

In addition to studying the total throughput in the network, we also collected statistics to look at the performance of individual flows under both protocols. Unlike in the case of AODV, where the throughputs for all the flows behave similarly beyond saturation, in the case of OLSRd, flows exhibit varying behaviors beyond saturation. For example, in our experiments, one flow (whose destination node is 12-18) continues to access the channel for packet transfers beyond saturation (shown in Figure 8) at the expense of throughput loss for another flow whose destination node is 18-19 (Figure 9). This is due to the multiplicative backoffs induced by collisions caused by increased volume of traffic effect mentioned above, which likely causes one flow to dominate the other, resulting in the behavior seen in Figures 8 and 9. This suggests that the particular implementation of OLSR used in this study may not be suitable for applications that have rigid QoS requirements.

### CONCLUSION

In our study, we compared the throughput performance using AODV and OLSR using experimental evaluations and relatively low mobility scenario. Based on the maximum throughput results from the one-hop tests, each protocol was tested for maximum possible stable throughput. Our initial observations indicate that AODV performs better than OLSR in terms of stability. This is due to the overhead of periodic control traffic involved in OLSR. However, OLSR allows throughput to increase beyond saturation for some flows, at the expense of degradation of other flows. The throughput beyond saturation is very jittery for OLSR. AODV does not allow throughput to increase above saturation and maintains it fairly constant at that level. Thus, if the offered load is increased further, stable throughput can be obtained, but with large packet loss.

These experiments show that it is important to study the behavior of ad-hoc routing protocols on a large testbed to study the effects of various physical factors on performance. Using these results, the correct protocol can be chosen depending on the tactical scenario. We further plan to investigate the resiliency and adaptability of these protocols under more dynamic link conditions.

### REFERENCES

[1] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, August, 1994, pp. 234-244.

[2] C. E. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Sys-*

*tems and Applications (WMCSA) '99*, New Orleans, FL, Feb. 1999, pp. 90-100.

[3] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks," Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer Publishers, 1996, ch 5, pp. 153-181.

[4] T. Clausen, P. Jacquet, A. Laotiti, P. Minet, P. M. and A. Qayyum, and L. Viennot, "Optimized link state routing protocol," IETF RFC 3626, Oct. 2003.

[5] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of The Ninth International Conference on Mobile Computing and Networking (MOBICOM) '03*, San Diego, Sept 2003.

[6] R. Draves, J. Padhye and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM) '04*, Philadelphia, Sept. 26 - Oct. 1, 2004, pp. 114-128.

[7] S. Zhao, Z. Wu, A. Acharya and D. Raychaudhuri, "PARMA: a PHY/MAC aware routing metric for ad-hoc wireless networks with multi-rate radios," in *Proc. of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, (WoWMoM) '05*, June 13-16, 2005, pp. 286-292.

[8] S. Desilva and S. R. Das, "Experimental evaluation of a wireless ad hoc network," in *Proc. 9th Int. Conf. on Computer Communications and Networks IC3N'00*, Las Vegas, Oct. 2000.

[9] K. Kuladinithi, A. Udugama, N. A. Fikouras, and C. Görg, "Experimental performance evaluation of AODV implementations in static environments," ComNets, Universität Bremen, Bremen, Germany, Tech. Rep.

[10] E. Borgia and F. Delmastro, "Effects of Unstable Links on AODV Performance in Real Testbeds," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Article ID 19375, 14 pages, 2007.

[11] List of ad hoc routing protocols. Wikipedia. [Online]. Available: [http://en.wikipedia.org/wiki/Ad\\_hoc\\_routing\\_protocol\\_list](http://en.wikipedia.org/wiki/Ad_hoc_routing_protocol_list)

[12] A. Tønnesen. (2004) OLSR.ORG. OLSRd. [Online]. Available: <http://www.olsr.org>.

[13] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)'05*, vol. 3, Mar. 13-19, 2005, pp. 1664-1669.

[14] H. Lundgren. Mackill - tool to filter packets at link layer. [Online]. Available: <http://user.it.uu.se/henrik/aodv/>

[15] ORBIT Traffic Generator (OTG). [Online]. Available: <http://www.orbitlab.org/wiki/Documentation/OTG>

[16] M. Gates, A. Tirumala, J. Dugan, and K. Gibbs. (2005, May) Iperf: The TCP/UDP Bandwidth Measurement Tool. [Online]. Available: <http://dast.nlanr.net/Projects/Iperf>

[17] E. Nordström. (2004) AODV implementation on Linux. AODV UU. [Online]. Available: <http://core.it.uu.se/core/index.php/AODV-UU>

[18] C. Perkins, E. Royer, and S. Das, "Ad hoc on-demand distance vector routing," IETF RFC 3561, 2003.

[19] S. Desilva and R.V. Boppana, "Sustaining Performance Under Traffic Overload," in *Proc International Conference on Wireless Networks*, 2004, pp. 3-8.

[20] Netfilter, Packet filtering framework, <http://www.netfilter.org/>