Comparative Study of Routing Protocols for Mobile Ad Hoc Networks

Thomas Clausen, Philippe Jacquet, Laurent Viennot

Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks

Thomas Heide Clausen*, Philippe Jacquet and Laurent Viennot

INRIA Rocquencourt, Projet Hipercom,
Domaine de Voluceau, B.P.105, 78153 Le Chesnay cedex, France
Telephone: +33 1 3963 5363 Fax: +33 1 3963 5363
Email: {Thomas.Clausen,Philippe.Jacquet,Laurent.Viennot}@inria.fr

***Abstract*— In this paper, we describe the Optimized Link State Routing Protocol (OLSR) [19],[20], a proactive routing protocol for Mobile Ad-hoc NETworks (MANETs). We evaluate its performance through exhaustive simulations using the Network Simulator 2 (ns2) [1], and compare with other ad-hoc protocols, specifically the Ad-hoc On-Demand Distance Vector (AODV) [4] routing protocol and the Dynamic Source Routing (DSR) [5] protocol. We study the protocols under varying conditions (node mobility, network density) and with varying traffic (TCP, UDP, different number of connections/streams) to provide a qualitative assessment of the applicability of the protocols in different scenarios.**

I. Introduction

A mobile ad-hoc network (MANET) is a collection of nodes, which are able to connect on a wireless medium forming an arbitrary and dynamic network. Implicit in this definition of a network is the fact that links, due to node mobility and other factors, may appear and disappear at any time. This in a MANET implies that the topology may be dynamic - and that routing of traffic through a multi-hop path is necessary if all nodes are to be able to communicate.

A key issue in MANETs is the necessity that the routing protocols must be able to respond rapidly to topological changes in the network. At the same time, due to the limited bandwidth available through mobile radio interfaces, it is imperative that the amount of control traffic, generated by the routing protocols is kept at a minimum.

Several protocols exist, addressing the problems of routing in mobile ad-hoc networks. Such protocols are, traditionally, divided into two classes, depending on when a node acquires a route to a destination. *Reactive* protocols are characterized by nodes acquiring and maintaining routes on-demand. In general, when a route to an unknown destination is required by a node, a query is flooded onto the network and replies, containing possible routes to the destination, are returned. Examples of reactive protocols include the "Ad Hoc On Demand Distance Vector Routing Protocol" (AODV) [17] and "Dynamic Source Routing" (DSR) [9].

Proactive protocols are characterized by all nodes maintaining routes to all destinations in the network at all times. Thus using a proactive protocol, a node is immediately able to route (or drop) a packet. Examples of proactive protocols include the "Topology Broadcast based on Reverse-Path Forwarding" routing protocol (TBRPF) [14] and the "Optimized Link State Routing Protocol" (OLSR) [19].

In this paper, the Optimized Link-State Routing Protocol will be presented. We will describe the protocol, as well as expose some of the protocol's basic characteristics through simulations. We will then compare OLSR with two reactive protocols, AODV and DSR. In particularly, we will focus on establishing the fact that OLSR as a proactive protocol complements the reactive protocols.

A. Paper Outline

The remainder of this paper will be organized as follows: in section II, we describe the OLSR protocol as a proactive routing protocol for MANETs, emphasizing the three independent components making up the protocol. We also provide some insights, gained through exhaustive experiments and simulations, on the design of the protocol. In section III, we introduce the reactive protocols in general, and describe AODV and DSR in some detail. We then proceed, in section IV, by describing our simulation environment, including the scenarios we use. Through providing a large set of randomly generated scenarios, all conforming to some general scenario specifications, we aim at providing unbiased simulations. Section V presents and discusses selected results of our simulations, and the paper is concluded in section VI.

*Thomas Heide Clausen may also be contacted at MindPass Center for Distributed Systems, Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7E 9220 Aalborg Ø, Denmark

II. The Optimized Link-State Routing Protocol

The Optimized Link-State Routing Protocol (OLSR) is a proactive link-state routing protocol, employing periodic message exchange to update topological information in each node in the network. While having some commonalities with OSPF [13], OLSR is specifically designed to operate in the context of MANETs, i.e. in bandwidth-constrained, dynamic networks.

In this section, we present details of the protocol, as well as describe some of the operational experiences that have influenced the development of the protocol design.

A. Protocol Details

Conceptually, OLSR contains three generic elements: a mechanism for neighbor sensing, a mechanism for efficient diffusion of control traffic, and a mechanism for selecting and diffusing sufficient topological information in the network in order to provide optimal routes. These elements are described in details in the following.

1) Neighbor Sensing:
Basically, neighbor sensing is the process through which a node detects changes to its neighborhood. The neighborhood of a node, *a*, contains the set of nodes with which there exists a direct link over which data may be transmitted (in either or both directions). Further attributes can be associated with such a link, depending on the direction(s) in which communication is possible. If traffic can only flow in one direction (e.g. if the nodes have asymmetric transmitters), the link is said to be *asymmetric*. If traffic can flow in both directions, the link is said to be *symmetric*. If there exist a symmetric link between node *b* and node *a*, node *b* is said to be a *symmetric neighbor* of node *a* (and vice versa).

In OLSR, the concept of a *two-hop neighbor* is introduced. A two-hop neighbor of node *a* is simply a node which has a symmetric link to a symmetric neighbor of node *a* AND which is not node *a* itself (i.e. node *a* can not be a two-hop neighbor of itself).

A prime goal for OLSR is to be completely independent of the underlying link-layer being used. While additional information from the link layer, such as information about existence of links to neighbor nodes and link quality, may be utilized by the protocol, care is taken such that the protocol can function without. The advantages are, that the protocol immediately can be deployed on most existing and anticipated wireless network interfaces and operating systems.

The neighbor sensing mechanism in OLSR is designed to operate independently in the following way: each node periodically emits a HELLO-message, containing the node's own address as well as a the list of neighbors known to the node, including the status of the link to each neighbor (e.g. symmetric or asymmetric).

Upon receiving HELLO-messages, a node can thus gather information describing its neighborhood and two-hop neighborhood, as well as detect the "quality" of the links in its neighborhood: the link from a node *a* to a neighbor *b* is symmetric if the node *a* sees its own address if in the HELLO-message from *b* (with any link status) - otherwise the link is asymmetric.

Each node maintains an information set, describing the neighbors and the two-hop neighbors. Such information is considered valid for a limited period of time, and must be refreshed at least periodically to remain valid. Expired information is purged from the neighbor- and two-hop neighbor sets.

2) Generic Message Diffusion:
HELLO-messages are exchanged between neighbors only. They provide each node with topological information up to two hops away. However, since MANETs can be of arbitrary size, a method is required for diffusing topological information into the entire network. In OLSR, this is introduced in form of a generic way of efficiently diffusing arbitrary control traffic to all nodes in the network. While being directly used in OLSR for diffusion of topological information, the mechanism is build as an independent and efficient MPR-flooding mechanism, and may thus be used to carry other types of control traffic (e.g. for service discovery protocols etc). Indeed, in an operational context, this mechanism was found to be an easy way for a node, connected to both a MANET and a wired network, to announce its "non-manet" routing capabilities to MANET nodes.

One of the prime requirements, stated in section I was, that due to limited bandwidth resources, the overhead from control traffic should be kept at a minimum. This, for a control message destined to all nodes in the network, implies that (i) all nodes ideally receive the message, while (ii) that not too many duplicate retransmissions of the message occurs.

A simple pure flooding strategy, where all nodes forward a flooded message if they have not previously forwarded the message meets the first part of the requirement: that all nodes, ideally, receive a copy of the message. The second part is only partially met through "transmission duplicate elimination", eliminating the situation where two copies of the same message are transmitted from a given node[1]. However, a given node might be receiving the same message from two neighboring nodes. This is illustrated in figure 1a.

The fact that a message is likely to be received by a node more than once is a problem: using pure flooding,

[1] This is achieved through maintaining a duplicate table, recording, for each received message, the originator address and a sequence number (generated by the source and transmitted in the message)

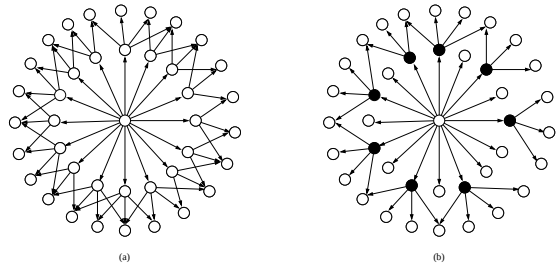

Fig. 1. *Example of pure flooding (a) and diffusion using Multipoint Relays (b). The source of the message is the node in the center. Each arrow pointing to a node, indicates that a node receives a copy of the message. The filled nodes are selected by the center node as Multipoint Relay.*

when a message is transmitted over the wireless medium, all other nodes within radio range of the transmitting node will either have to remain silent, or may experience message loss due to collisions.

The OLSR protocol applies an optimized flooding mechanism, called MPR-flooding, to minimize the problem of duplicate reception of message within a region. The optimization is performed in the following way: a node selects a subset of its symmetric neighbors, called the nodes MultiPoint Relays (MPR's). Each node thus has a (possibly empty) set of MPR selectors (neighbors, which have selected the node as MPR). A node, selected as MPR, has the responsibility of relaying flooded messages from its MPR selectors. A message emitted by node *a* is thus only retransmitted by node *b* if node *a* is in the MPR selector set of node *b*. As illustrated in figure 1b,"careful" selection of MPRs (the filled nodes) may greatly reduce duplicate retransmissions.

While selecting MPRs, a node utilizes information describing the two-hop neighbors, as acquired from the neighbor sensing process. All nodes select their MPRs independently, possibly choosing different algorithms or heuristics for selecting a "minimal" MPR set. The invariant for the algorithms is, that a message, emitted by the node and relayed by its MPRs, would reach all the node's two-hop neighbors. [3] presents an analysis of MPR selection algorithms.

A node is informed of its MPR selector set through information piggybagged to the HELLO-messages.

Thus when using MPR-flooding, the forwarding rule for handling flooded control messages in each node being:

1) the message must be meant to be forwarded (indicated by information in the header of the message),
2) the message must not have been received by the node before, and
3) the node must have been selected as MPR by the node, from which the message was received

The OLSR protocol specification [19] defines a generic message format and an algorithm for processing such messages. This includes time-to-live considerations, sequence numbers *etc.*, out of scope for this description.

3) Topology Information:
The MPR flooding mechanism is directly used by OLSR for diffusing topological information to the network.

In OLSR, all nodes with a non-empty MPR selector set periodically generate a topology control message (TC-message). This TC-message is diffused to all nodes in the network, using MPR flooding. A TC-message contains the address of the node generating the TC-message, as well as the addresses of all the MPR selectors of that node. Thus through a TC-message, a node effectively announces reachability to all its MPR selectors. Since all nodes have selected an MPR set, reachability to all nodes will be announced through the network. The result is that all nodes will receive a partial topology graph of the network, made up by all reachable nodes in the network and the set of links between a node and its MPR selectors. Using this partial topology graph, it is possible to apply a shortest path algorithm for computing optimal routes from a node to any reachable destination in the network [19]. A noticeable result is that the shortest path obtained from the partial topology yielded by the TC-messages have the same length as the shortest path from the full topology [8].

The topological information in each node is valid for a limited period of time, and must be refreshed at least periodically to remain valid. To improve reactiveness to network dynamics, additional TC-messages may be generated. Expired information is purged from the topology graph.

B. Operational Experiences

OLSR has been the subject of exhaustive studies through analysis, simulations and experiments, and the resulting protocol is largely a product of operational experiences and requirements. In this section, we outline some of these experiences.

1) External Route Injection:
While the ability to route data within a MANET is, indeed, the priority task for a MANET routing protocol, we found that the ability to inject external routes into the network was a practical requirement. MANET routing is based on the information acquired through TC messages which, in principle, advertise a list of host-routes.

External routes would typically be either sequences of IP-addresses or gateways to the whole Internet, making TC messages impractical for distributing such information.

Utilizing the existing generic mechanisms for diffusing control traffic, the requirement of a way of injecting routes to continuous sequences of addresses was satisfied

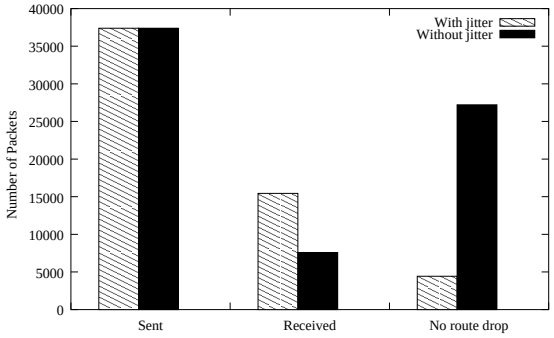

Fig. 2. *The average number of packets sent, received, and dropped because of route unavailability. The number are averages from test sets with and without enforced jitter on transmission of control packets.*

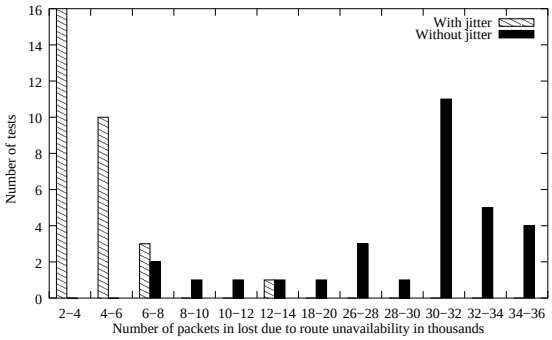

Fig. 3. *The number of tests as a function of number of packets lost due to route unavailability. The number of packets are showed in intervals of thousands.*

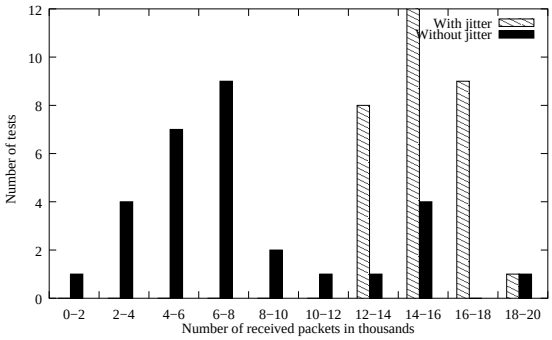

Fig. 4. *The number of tests as a function of number of packets received. The number of packets are showed in intervals of thousands.*

simply through introduction of a new type of message. Potentially emitted less frequently than the TC-messages, this extension was thus able to take direct advantage of the optimizations of MPR flooding.

2) Jitter:

Practical experiments showed that even an otherwise static network in continuous operation would suffer from transient loss of routes to parts of the network. It was observed that the cause for this was in the fact that the periodic emission of control messages from the individual nodes had become synchronized: When a node reports a change to its MPR-set (in a HELLO message), emission of a TC-message may be simultaneously triggered in a set of neighboring nodes. Such TC messages would collide at the receiving nodes, where neither of the messages would be received (and potentially not forwarded into the network). TC-messages are emitted periodically by nodes which experience no changes to their MPR selector set. Thus, until e.g. clock drift in the nodes un-synchronize the emission of TC-messages, collision - and loss - of topology information would occur. This, in turn, would cause routes to time out and disappear from parts of the network - despite physical connectivity.

Experiments indicated that enforcing jitter (a small delay, randomly picked for each packet from the interval $[-0.5; 0.5]$ sec.) on sending of each control messages resulted in fewer control message collisions - and hence in more stable routes. Figure 2 and shows the number of packets sent, received and dropped due to “no route to host” in the same scenario with and without jitter.

It can be observed, that roughly twice as many data packet reach their destination with jitter enforced than without. The remaining sent, but not received, data packets are dropped mainly due to collisions, drops in interface queues etc. In particulary when using jitter, we observe that a large amount of packets are dropped for other reasons than “no route to host”, whereas without jitter, almost all drops are due to “no route to host”. This is a natural concequence: more packets being routeable implies more packets inserted into interface queues and more packets that can be transmitted on the media. This increases the likelyhood of congestions and drops, both in queues and in the media.

Figure 3 and figure 4 show the dispersion of the packet drops (due to lack of topology information) and the dispersion of the number of received packets. It is observed, that the dispersion with jitter is smaller than without. This means that the performance with enforced jitter is not only better, but also more stable than without jitter.

III. Reactive Protocols

OLSR is a proactive protocol, maintaining information about routes to all destinations at all times. The consequence of this approach is that the amount of control traffic is independent of the actual traffic and mobility patterns in the network. An alternative approach is that of reactive protocols. Basically, a reactive protocol reacts to the traffic in the network, and constructs routes when needed - and only to those nodes to which a route is needed.

The common element in reactive protocols is the mechanism used for discovering routes. The source node emits

a request message, requesting a route to the destination node. This message is flooded, i.e. relayed by all nodes in the network, until it reaches the destination. The path followed by the request message is recorded in the message, and returned to the sender by the destination, or by intermediate nodes with sufficient topological information, in a reply message. Thus multiple reply messages may result, yielding multiple paths - of which the shortest is to be used.

In this section, we introduce two reactive protocols, AODV and DSR, with the purpose of providing a context in which we can evaluate the characteristics of OLSR.

A. AODV

In the Ad Hoc On-Demand Distance Vector protocol (AODV), when a source requires a path to the destination, a *route request* message is flooded in the network. Upon receiving such a message, a node examines its local route-cache to check if a fresh route to the required destination is available. If so, the node unicasts a *route reply* message to the source with information about the route. Otherwise, the *route request* is retransmitted using a pure flooding mechanism with local duplicate elimination. As an optimization, AODV employs an "expanding ring" flooding, where a *route request* is issued with a limited TTL. If no *route reply* message is received within a certain time, the message is issued again with a larger TTL. If still no reply, the TTL is increased in steps, until a certain maximum value.

While this route discovery is performed, any IP-packets to the destination are buffered in the source node. When a route is established, the packets are transmitted. If no route can be established, the packets are dropped.

When a link is detected to be broken (either through a neighbor discovery protocol, as in OLSR, or through a link-layer notification), the detecting node issues a *route error* message to those neighbors who have been using a route over the now broken link. These nodes will then have to issue new *route requests* to repair the broken routes.

B. DSR

The Dynamic Source Routing protocol employs the same basic mechanism of on-demand flooding a *route request* and awaiting that the destination node, or an intermediate node with verified valid information, replies with a *route reply*.

DSR employs source routing, both as a way of obtaining loop freedom and as a way of "sharing" a nodes route cache with other nodes in the network: since each data packet contain routing information, nodes along its path, as well as nodes which overhear the transmissions, may collect and cache the route information for later use.

Route maintenance is based on each hop receiving an acknowledgment for a packet being forwarded (either through link-layer notification, through overhearing the next-hops forwarding of the packet or through requesting a DSR-specific acknowledgment). If a node thus detects a broken route, a *route error* is returned to the source. Upon receiving a *route error*, the source removes the broken route from its routing cache. If an alternative route is available, it may be used for remaining data to the destination - alternatively, a new route discovery is initiated.

Like AODV, DSR buffers IP packets in the source node while route discovery is performed.

IV. Scenarios

We conduct our simulations using the network simulator ns2 [1]. We use a physical layer, simulating the behavior of IEEE 802.11[2] as included with ns2: each node has a radio range of 250 meter, when no obstacles are present, and a nominal bandwidth of 2 Mbit/s. The MAC scheme is an implementation of that specified by IEEE 802.11.

The purpose of our simulations is to uncover in which situations the individual protocols have their strengths and weaknesses, rather than to promote one protocol as generally "better" than the others. Thus, in order to avoid getting results which favor either of the protocols, we apply a strategy of specifying a set of parameters (number of nodes, node mobility, traffic characteristics etc), from which a large number of scenarios are randomly generated. These scenarios will be different, yet have the same overall characteristics.

We base all our scenarios on the following basic parameters:

- 50 nodes
- 1000 x 1000 m^2 field
- 250 seconds simulation time
- 1-5$\frac{m}{s}$, 0-5s rest time, 1000 m. distance, random-waypoint model.
- 25 CBR streams, 0.1 sec. packet interval, 64 bytes/packet, 10 sec stream duration

Unless otherwise stated when describing the simulation results, the simulations are conducted with scenarios conforming to the above parameters.

Each sample point, represented in the simulation results in section V is the mean taken over 30 different scenarios, conforming to the same parameter set. We emphasize, that the set of 30 scenarios per sample point are the same for all the three tested protocols. I.e. for a given sample point, the each of the protocols are tested with the same 30 scenarios.

V. Simulation results

Simulations have been conducted with varying mobility and varying number of traffic streams to examine the

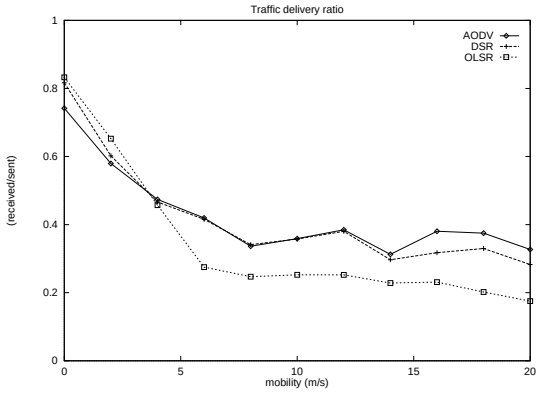

Fig. 5. *Data packet delivery ratio with varying mobility.*

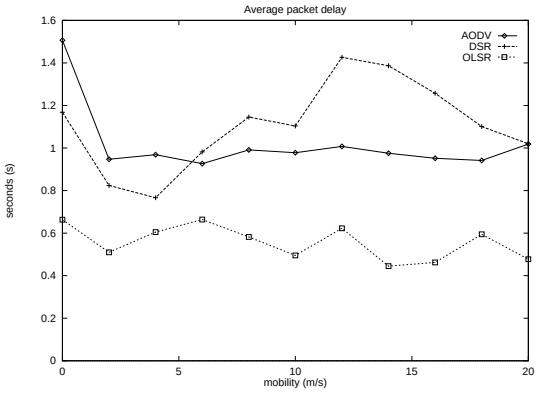

Fig. 7. *Packet delays with varying mobility.*

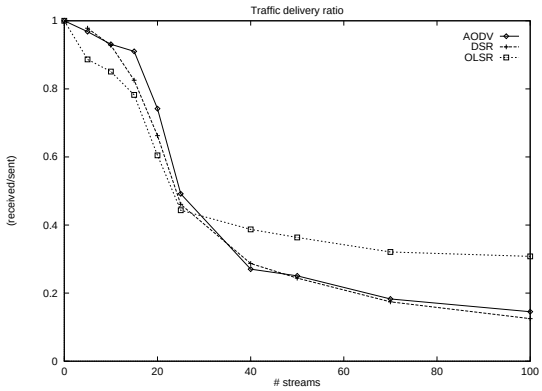

Fig. 6. *Data packet delivery ratio with number of traffic streams.*

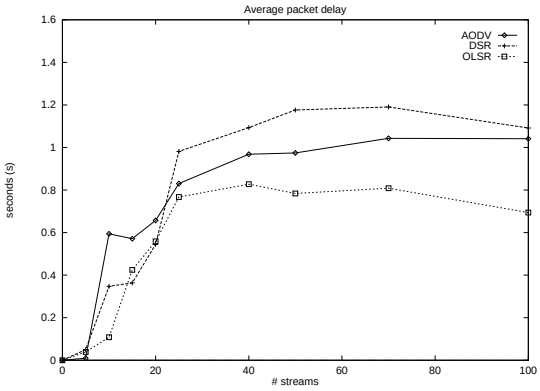

Fig. 8. *Packet delays with varying number of traffic streams.*

protocols in different contents. Comparisons have been done on the following: packet delivery rate, control traffic overhead, route length, and performance under TCP traffic.

A. Packet Delivery Rate and Packet Delay

The prime property for a routing protocol is to provide routes between sources and destinations. Thus one measure of success for a routing protocol is the fraction of data packets being successfully delivered to the destinations.

In figure 5 and figure 6, we present the traffic delivery ratio (i.e. number of received/number of sent using the three protocols under various mobility scenarios and traffic scenarios, respectively.

Figure 5 shows that the two reactive protocols perform roughly equivalent and manage to deliver about the same amount of data packets. A slight advantage to DSR is noticed in static networks, while AODV has a slight advantage in largely mobile networks. We also notice the effect of buffering packets in the reactive protocols, in case a route is not available: the delivery rate of AODV and DSR is slightly higher in scenarios with high mobility than that of OLSR - which in the current implementation does not employ buffering of IP packets.

Figure 6 confirms that the two reactive protocols exhibit similar performance. However it also shows that for more than approximately 35 concurrent traffic streams, the delivery rate of OLSR is slightly higher than that of both AODV and DSR. With a large number of concurrent traffic streams, extra control-traffic overhead is consumed by route maintenance in the reactive protocols (as showed in figure 10). This leaves less available bandwidth for data traffic and increases chances of loss due to collisions and interface queue overflows. These effects counter the positive effects of buffering of IP packets.

Figure 7 and figure 8 present the average packet delay, i.e. the delay from a packet has been transmitted until it is received.

We observe that OLSR consistently presents the lowest delay, regardless of mobility. This may be explained by the fact that OLSR is a proactive protocol: when a packet arrives at a node, it can immediately be forwarded or dropped. In reactive protocols, if there is no route to a destination, packets to that destination will be stored in a buffer while a route discovery is conducted. This may (in case a route is actually discovered) cause longer delays

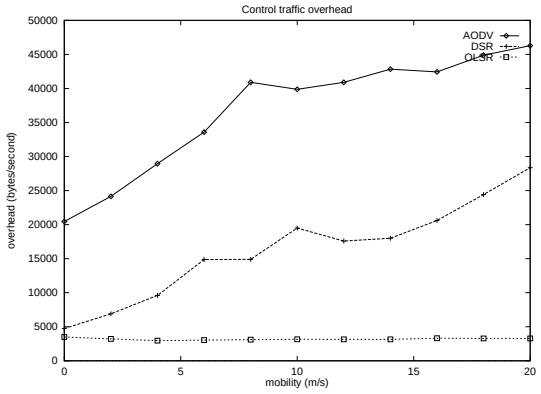

Fig. 9. *Total control traffic overhead with varying mobility.*

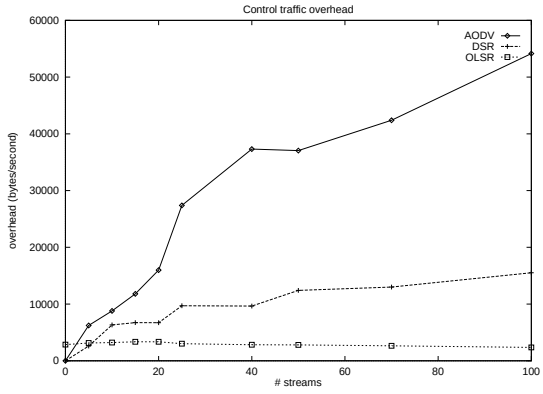

Fig. 10. *Total control traffic overhead with varying number of traffic streams.*

but also ensure a higher delivery rate. Due to the buffering, the expectation would also be that the packet delivery ratio would be higher for the reactive protocols. This is, as can be seen from figure 5, the actual case. We also notice that OLSR, in the cases where the protocol provides higher delivery rate than the reactive protocols, still provides a lower packet delay.

B. Control Traffic Overhead

One of the contributions to routing protocol overhead in a network is the overhead from control traffic. Figure 9 shows the total control traffic generated by all nodes as a function of the node mobility. We observe, that the amount of control traffic generated by OLSR remains constant, while that of the two reactive protocols changes, depending on mobility (and, hence, the degree of topological changes in the network). This corresponds with the observation that the described and simulated version of OLSR does not react explicitly to link-breaks, whereas AODV and DSR do. Figure 10 shows the total overhead from the control traffic generated by all nodes as a function of the number of concurrently active data streams in the network. We observe that the control traffic of OLSR exhibits the expected characteristics of being independent of the traffic pattern, while the control traffic, generated by the reactive protocols, increases with an increased number of active streams.

Our simulations reveal that in networks where the topology and the traffic patterns are relatively static, the reactive protocols introduce less control traffic overhead than OLSR. On the other hand, in networks with relatively dynamic traffic and mobility patterns, the control traffic from the repeated route discovery procedures in the reactive protocols introduces a large control traffic overhead.

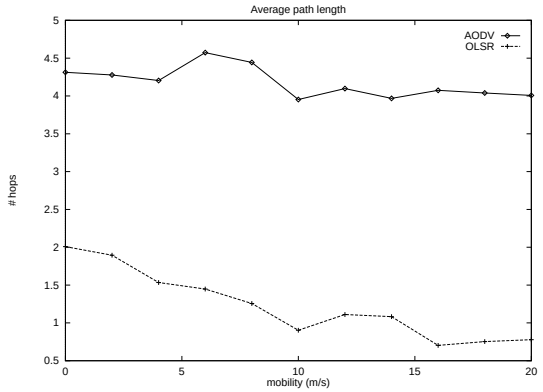

Fig. 11. *Average path length with varying mobility.*

C. Route length

The second major contribution a routing protocol can bring to overhead in the network is that of providing suboptimal routes, i.e. routes which are longer than the shortest path. This contributed both to the overhead (in terms of "wasted transmissions") and the delay, experienced in the network.

In figure 11 and figure 12, we present the average path lengths for successfully delivered data packets, obtained using AODV and OLSR in scenarios with a varying mobility and traffic patterns respectively[2].

We observe that the route lengths obtained by OLSR and AODV are very different. AODV produces significantly longer paths than OLSR. We also observe, that the average path length of OLSR seems to drop when the mobility and the traffic increases. There are several explanations for this observation. Firstly, with increased mobility and increased traffic, TC-messages, employed by OLSR for diffusing topological information in the network, may be lost or may not be diffused frequently enough to track

[2]Due to differences in the trace file format yielded by ns2 for DSR and for the other protocols, route length measurements for DSR are not included.

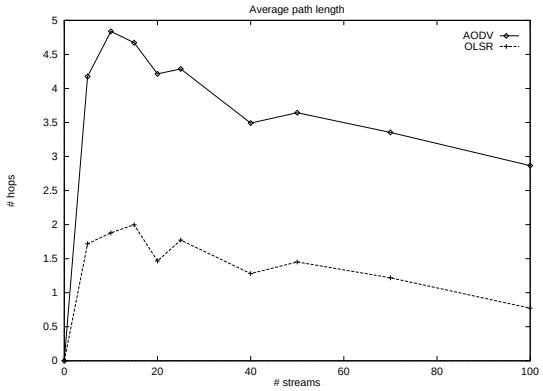

Fig. 12. *Average path length with varying number of traffic streams.*

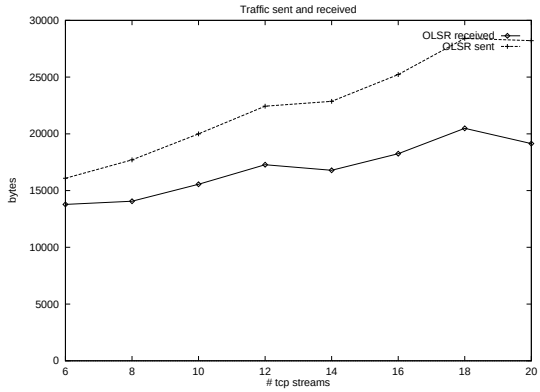

Fig. 13. *Number of sent and received packets using OLSR as the routing protocol with varying number of TCP streams.*

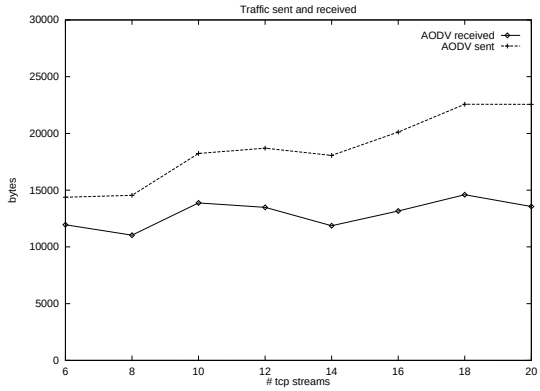

Fig. 14. *Number of sent and received packets using AODV as the routing protocol with varying number of TCP streams.*

the network dynamics. This leads to a situations where an OLSR node lacks sufficient topological information to construct a route to a given destination, and hence choose to drop the packet (the packet not being included in route length calculations). This results in nodes being able to deliver packets to nodes in its proximity (up to a few hops away), while traffic to nodes farther away would not be delivered.

Considering both figure 11 and figure 5, we observe that the data packet delivery rate of OLSR is lower than that of AODV in scenarios with a high mobility rate. In such high-mobility scenarios, routes to "far away" nodes are more likely to be absent in OLSR than routes to "close" nodes. This results in an average shorter path for those packets which are successfully delivered. Packets that would follow a "longer" path are in higher risk of being dropped. We also observe, however, that at mobility rates where OLSR has a higher delivery rate than AODV, the average route length of OLSR is still significantly lower.

Considering both figure 12 and figure 6, we observe that, when varying the number of traffic streams in the network, OLSR consistently has a shorter route length - even at a large number of traffic streams where OLSR has a higher delivery rate than AODV. Notice, that for the simulations depicted in these figures, the mobility characteristics are as described in section IV.

In AODV, if a route is not available, data packets are buffered and route discovery is initiated. Flooding of route requests may, e.g. due to heavy traffic in parts of the network, arrive at the destination node through a path, longer than the shortest possible. Hence, using AODV, it is in some situations possible for a data traffic to flow through a longer path - as an alternative to being dropped.

D. TCP Performance

Comparisons of MANET protocols based on TCP are rare. This is in part due to both the generally unsuitability of TCP in the wireless domain and in part also due to the fact that TCP is a *conforming* protocol: the protocol adjusts its data rate according to feedback from the network. This means e.g. that when packets are dropped due to collisions and interference, TCP assumes that the reason for dropping packets congestion and lowers the data rate. Thus, for stress-testing a MANET protocol, TCP is not a good choice. Fact remains, though, that approximately 95% of the traffic on the Internet today carries TCP [6], [12]. It is thus appropriate to study how well the different routing protocols support TCP.

In figure 13, figure 14 and figure 15 we present the number of sent and received packets in OLSR, AODV and DSR, respectively, with varying number of TCP streams. The number of packets sent is the number of data packets that leave the node. Hence, packets that are not send due to TCP congestion handling is not included in the graph. The figures show that OLSR sends and receives more packets than the reactive protocols in all cases.

Figure 16 shows the control traffic overhead in the three protocols, when varying the number of TCP streams. As expected, the overhead of OLSR remains constant just as

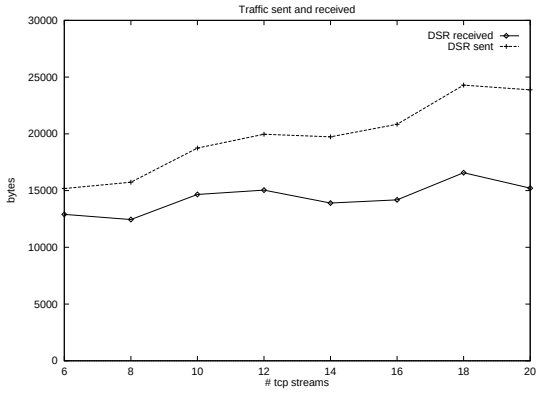

Fig. 15. *Number of sent and received packets using DSR as the routing protocol with varying number of TCP streams.*

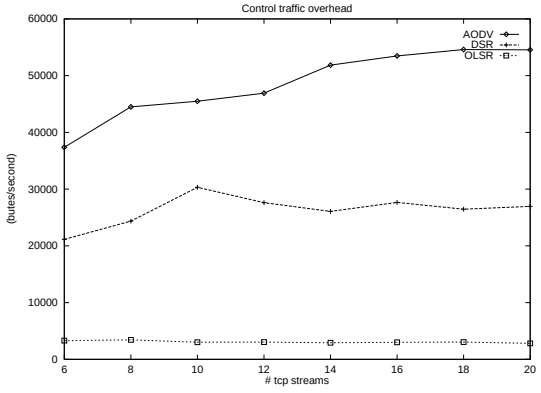

Fig. 16. *Control traffic overhead with varying number of TCP streams.*

in the case with CBR traffic. The overhead of the reactive protocols increases as the number of TCP streams increases, AODV with a higher growth rate than DSR. The overhead of OLSR is in all cases substantially lower than that of the reactive protocols. Notice that, contrary to CBR-traffic, a TCP stream requires data to flow in both directions between the source and the destination node. While routes in both direction are immediately available to OLSR, reactive protocols need to explicitly maintain two routes for each TCP-flow (one from source to destination, one from destination to source).

VI. Conclusion

Our experiments and simulations have shown that the Optimized Link-State Routing Protocol (OLSR) for MANETs is a viable routing protocol, showing performance characteristics complementing AODV and DSR.

OLSR is a proactive protocol and, as such, utilizes periodic message exchange to maintain topological information in the network. A mechanism, MPR-flooding, for efficiently diffusing such topological information to the entire network, is a core element of the protocol, significantly receding the control traffic overhead. Operational experiences and simulations indicated, that a significant performance gain, measured in terms of delivery ratio, was achievable through introduction of a small random jitter on transmission of control messages.

We notice that AODV and DSR both provide buffering of undeliverable data packets while route discovery is ongoing. This is a feature not present in OLSR. In scenarios with a high rate of mobility, we observe a positive effect of this: the delivery rate of both AODV and DSR becomes higher than that of OLSR. For low mobility rates, the performance of OLSR exceeds that of AODV while performs roughly equivalent to DSR.

Observing the packet delivery rate for scenarios with a varying number of communicating pairs, we observe that for few communicating pairs, the reactive protocols have an advantage. However when the number of communicating pairs increases, OLSR keeps higher the delivery rate. This indicates that proactive protocols outperform reactive protocols in heavy load conditions.

Overall, we observe that OLSR offers the lowest delay and the shortest path length of the three protocols. However this fact might be accentuated with the fact that, for some scenarios, the data delivery rate of OLSR is less than that of the reactive protocols: without buffering, a data packet either a packet is dropped immediately, or it is delivered with a low delay. However we find, that even in situations where the delivery rate of OLSR is higher than that of the reactive protocols, OLSR consistently provides shorter paths. We also find, that the average path length provided by AODV is surprisingly high.

The control traffic overhead is significantly lower for OLSR than for AODV and DSR, except for situations with almost no traffic. A way of increasing OLSRs reactiveness to link changes, and thereby the performance under high-mobility scenarios, could therefore be to increase the rate of control traffic. We observe as expected, that the control traffic of AODV and DSR increase with both traffic and mobility. The two reactive protocols perform roughly identical.

For TCP-traffic, we observe that OLSR as a proactive protocol performs significantly better than both the reactive. We attribute this to two main factors: TCP-traffic requires routes both from source to destination and from destination to source, demanding more from the reactive protocols, while such routes by default are provided by OLSR. The other factor is, that TCP is network performance conformant: the route discovery control traffic and the buffering of IP packets while route discovery is ongoing collides with TCP flow control initialization.

In conclusion, we find, that OLSR, as a good candidate for a proactive protocol, performs comparatively to the reactive protocols. We find, that the two classes of protocols complement each other, providing advantages in different domains. It is clear, that neither of the two

protocol classes outperform the other in every domain, and that there, therefore, is a need to keep both solutions available.

REFERENCES

[1] Network Simulator - ns - 2. Available at http://www.isi.edu/nsnam/ns/.

[2] Wireless lan medium access control (mac) and physical layer (phy) specifications. ISO/IEC Std. 8802-11, ANSI/IEEE Std 802.11, 1999.

[3] Laurent Viennot Amir Qayyum and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical report, Project HiPERCOM, INRIA Rocquencourt, 2000. INRIA research report RR-3898.

[4] Charles E. Perkins, Elizabeth M. Belding-Royer, Samir R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. Technical report, Nokia Research Center; University of California, Santa Barbara; University of Cincinnati, November 2001. draft-ieft-aodv-09.txt - work in progress.

[5] David B. Johnson, Yih-Chun Hu, David A. Maltz, Jorjeta G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Technical report, Rice University, AON Networks, Carnegie Mellon University, November 2001. draft-ieft-dsr-06.txt - work in progress.

[6] Kevin Jeffay F. Donelson Smith, Felix Hernandez Campos and David Ott. What TCP/IP Protocol Headers Can Tell Us About the Web. In *SIGMETRICS*, October 2001.

[7] J.J. Garcia-Luna-Aceves and M. Spohn. Efficient routing in packet-radio networks using link-state information. In *Proc. IEEE WCNC 99*, August 1999.

[8] Philippe Jacquet, Pascale Minet, Paul Muhlethaler, and Nicolas Rivierre. Increasing reliability in cable-free radio LANs: Low level forwarding in HIPERLAN. *Wireless Personal Communications*, 4(1):65–80, January 1997.

[9] J. G. Jetcheva, D. Johnson, D. Maltz, and Y.C. Hu. Dynamic source routing (DSR). Internet Draft, draft-ietf-manet-dsr-06.txt, November 21 2001, Work in progress.

[10] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 5:153–181, 1996. Kluwer Academic Publishers.

[11] Scott Corson Joseph Macker. Mobile adhoc networking and the ietf. *ACM Mobile Computing and Communications Review*, 1998.

[12] Gregory J. Miller Kevin Thompson and Rick Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, November/December 1997.

[13] J. Moy. Ospf version 2. Internet Standard, Request For Comments 2328, April 1998.

[14] Richard G. Ogier, Fred L. Templin, Bhargav Bellur, and Mark G. Lewis. Topology broadcast based on reverse-path forwarding (tbrpf). Internet Draft, draft-ietf-manet-tbrpf-03.txt, November 28 2001, Work in progress.

[15] V. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. IEEE INFOCOM '97*. Kobe, Japan, 1997.

[16] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proceedings of the IEEE International Conference on Communications*, 2000.

[17] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing. Internet Draft, draft-ietf-manet-aodv-09.txt, November 9 2001, Work in progress.

[18] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999. New Orleans, LA.

[19] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot and Thomas Clausen. Optimized Link-State Routing Protocol. Technical report, Project HiPERCOM, INRIA Rocquencourt, March 2001. draft-ieft-olsr-04.txt - work in progress.

[20] Thomas Clausen, Gitte Hansen, Lars Christensen and Gerd Behrmann. The optimized link state routing protocol - evaluation through experiments and simulation. In *Proceeding of Wireless Personal Multimedia Communications*. MindPass Center for Distributed Systems, Aalborg University, Fourth International Symposium on Wireless Personal Multimedia Communications, September 2001.