Signal Strength Based Link Sensing for Mobile Ad-Hoc Networks*

Husnain Mansoor Ali
University Paris XI
IEF - CNRS UMR 8622
Centre Scientifique d'Orsay
91405 Orsay - France
husnain.ali@ief.u-psud.fr

Amina Meraihi Naimi
University Paris XI
IEF - CNRS UMR 8622
Centre Scientifique d'Orsay
91405 Orsay - France
amina.meraihi@ief.u-psud.fr

Anthony Busson
University Paris XI
IEF - CNRS UMR 8622
Centre Scientifique d'Orsay
91405 Orsay - France
anthony.busson@ief.u-psud.fr

Veronique Veque
University Paris XI
IEF - CNRS UMR 8622
Centre Scientifique d'Orsay
91405 Orsay - France
veronique.veque@ief.u-psud.fr

November 30, 2007

Abstract

Mobility causes frequent link failures in ad-hoc networks. This results in a severe degradation of performance specially in case of high mobility of nodes. This is because the routing protocols for ad-hoc networks are not equipped to handle high mobility. In this paper, we have presented a new link management algorithm to locally manage

*This work is supported by the French government funded project ANR RNRT R2M (Reseaux Mesh et Mobilite - Mesh network and mobility)

links. This new mechanism is based on signal strength measurements. Researchers over the years have presented approaches which use signal strength measurements but their focus has been on re-active protocols while our algorithm is aimed at pro-active protocols. Pro-active protocols are used since they provide greater flexibility to take advantage of the mesh configuration. We develop the hysteresis mechanism provided by OLSR, based on hello packets, to include signal strength measurements. The mechanism in OLSR uses Hello packets received/lost to decide to establish link or not. The problem with this approach arises when there is high mobility in which case the time to break the link and use a new path becomes significant. To overcome this, we propose to use signal strength to determine if the link-quality is improving or deteriorating. This combination of the two mechanisms, makes the link management more robust and also helps in anticipating link breakages thereby greatly improving performance.

1 Introduction

There has been a phenomenal growth in wireless communication in recent years. Devices have become smaller with longer battery life and communication protocols have become more robust. Thus now it is the fastest growing segment of the communications industry and has even surpassed wired communications in many areas. This has led to a situation in which more and more tasks are being done using wireless technology giving users the advantage of mobility.

Wireless networks can be classified according to two main types: Infrastructure based e.g. Global Standard for Mobile communications (GSM),Wi-Fi etc. or Infrastructure less which are called ad-hoc networks. Infrastructure based networks use an approach where mobile nodes communicate directly with some centralized access point. These type of networks demand centralization for configuration and operation. Thus they can not be used in all situations e.g. military networks, disaster relief operations etc.

An ad-hoc network, as the name suggests, is a network formed by nodes connected arbitrarily for some temporary time. A Mobile Ad-hoc NETwork (MANET) is a kind of wireless ad-hoc network having arbitrary topology (no fixed infrastructure) with mobility. It consists of mobile routers connected wirelessly to each other where each node is free to move. This results in a continuously changing topology.

Mesh networking is a way of interconnecting nodes by having multiple paths present between them. Thus *"wireless mesh networks are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad-hoc network and maintaining the mesh connectivity"* [6]. Hence it allows for continuous connections and reconfiguration around blocked paths by hopping from node to node until a connection can be established. As there are multiple paths, the network can still operate even when a node breaks down or a connection goes bad. Thus the performance depends highly on how the protocol handles link management. Also with mobility, there is a continuous change of topology, so the importance of link management increases greatly.

1.1 Motivation and related works

In 802.11 infrastructure based networks, association management is done usually by detecting frames lost and monitoring Signal-to-Noise Ratio(SNR). The clients start scanning for new Access Points(APs) when SNR passes a minimum threshold [15]. The APs typically emit a beacon packet every 100ms. The clients use this beacon to find an AP having highest SNR with which to associate. The change is thus transparent to user and is efficient in the sense that very few packets are lost during the handoff.

In case of ad-hoc networks, link management is done locally by each node for all nodes within its radio range. Also, this is done at the network layer by the routing protocol. The routing algorithms are thus equipped with mechanisms that aim to manage mobility, i.e. changes in the topology and routes. Both local and global approaches are used. Locally for link breakage detection, different mechanisms are employed, for example consecutive hello losses used in the OLSR [3] and AODV [13] as are the link expiration timers. Globally, it is done by periodically sending control packets such as TC messages in OLSR to inform the entire network of topology changes and make it possible for nodes to recompute routes, or in a reactive way, sending a route error message to inform the source of route loss. Usually this does not involve using signal power measures obtained from lower layers since this information is unavailable. This leads to poor performance in terms of Packet Delivery Ratio(PDR) as it requires more time at this layer to manage links. In [14], the authors discuss the benefits and disadvantages of using cross-layer feedbacks.

Recently a number of studies have been done in which measures from lower layer have been used for link management. Different protocols have proposed and used different approaches. In Associativity Based Routing (ABR) [17], temporal stability of a link is used to classify it as stable or unstable. This is determined by keeping track of beacons received from each node which are sent at regular intervals. The stable links are preferred when selecting a route when a route request originates even though they may be longer. In [5], the authors propose a protocol based on signal strength and stability of hosts. It classifies links as strongly connected or weakly connected based on signal strength and measures the time interval during which neighbors are strongly or weakly connected to determine route stability. [1] uses an affinity parameter based on signal strength which is used in conjunction with path length to select routes. In [12], the nodes estimates the worst case life time of link by calculating the trajectory of nodes based on the signal strength, and use this estimate for route selection. The received transmission power is used in [9] to estimate when a link will break and the mechanism is applied to DSR [11] and AODV. On the other hand, [4] uses periodic connectivity information(position and mobility pattern) to predict link breakage in AODV to prompt local repair.

The inconvenience with above and other such approaches is that they all have been developed for on-demand protocols. Thus they assume that either, we already have an established link and based on signal strength try to avoid link breakage by employing different mechanisms, or use the information to select a path with stable links when there is a request. In both cases, there is substantial initial delay in connection establishment which makes them inefficient for high mobility scenarios since due to high mobility, the link quality changes rapidly. Also for mesh networks, where part of the network has very little mobility, using pro-active protocols instead of on demand protocol offers greater flexibility since we can take better advantage of mesh connectivity.

In this paper, we focus on local link management for efficiently managing mobility. For implementation we have used the Optimized Link State Routing(OLSR) protocol. OLSR is one of the protocols proposed by Internet Engineering Task Force(IETF) for MANET's. Being pro-active, it is well suited to mesh networks since a part of a network is always fixed. But the OLSR link management is not adapted to extremely mobile networks. In fact, the problem with link breakage detection as proposed in OLSR hysteresis based on hello losses is that the former needs time, at-least 2 hello intervals (see figure 1), during which packets destined to that node are simply lost. In a highly mobile network, link breakages are very frequent, leading to a very short link validity time (not exceeding some seconds) [16], and waiting for two packet loss can lead to a very bad packet delivery ratio.

To solve the above problem, one could think of increasing Hello frequency (reducing hello interval time) [2]. This would limit the detection time but incurs too much overhead making this solution inappropriate. However, we think that for obtaining maximum packet delivery ratio, the most suitable approach is to be able to anticipate breakages instead of waiting for link breakage to be detected.

We first describe in brief OLSR and its link management in the next section. Then we propose our algorithms in section 3 and discuss its pros and cons. In section 4, we describe the simulations scenarios and discuss the results. Finally we conclude in section 5 giving some future work directions.

2 Optimized Link State Routing(OLSR) Protocol and Hysteresis overview

In this section, we describe in brief the working of OLSR and look at the Hysteresis mechanism provided by it.

2.1 OLSR functioning

OLSR as it's name suggests is an optimization of the Link State Protocol for ad hoc networks and like other link state protocols, nodes construct map of the whole network. Thus the whole network should, in theory, be known to all nodes. OLSR is a table-driven pro-active protocol i.e. each node maintains information about all the other nodes at all times. This is unlike other reactive protocols which initiate route discovery on demand. This results in no initial delay in communication but requires constant overhead of routing traffic.

It is called Optimized because firstly, OLSR minimizes flooding of control traffic by using only selected nodes, called multipoint relays MPR (1-hop neighbor nodes which re-transmit all broadcast messages), to diffuse its messages. Secondly, it reduces the size of the control packets by declaring only a subset of links with its neighbors instead of all the links. It only declares the links to those nodes which are its multipoint relay selectors (nodes which has selected its 1-hop neighbor, as its multipoint relay, are called a multipoint relay selector). By using the above techniques, it significantly reduces the number of retransmissions in a flooding or broadcast procedure and also reduces packets size, minimizing overhead.

The protocol is designed to work in a completely distributed manner and thus does not depend on any central entity. The protocol does not require reliable transmission for control messages: each node sends control messages periodically, and can therefore sustain an occasional loss of some packets.

Being a proactive table-driven protocol, OLSR operation mainly consists of updating and maintaining information in a variety of repositories. The data in these repositories is updated based on received control traffic. The control traffic in turn is generated by each node based on the data present in its data tables. There are two main types of messages generated by OLSR.

HELLO - HELLO messages are transmitted to all neighbors. These messages are used for neighbor sensing and MPR calculation and are broadcasted periodically depending on hello interval.

TC - Topology Control messages are the link state signaling done by OLSR. This messaging is optimized by the use of MPRs.

In the following section, we describe the link management mechanism present in OLSR.

2.2 Link Hysteresis

Link hysteresis is used to make links robust by slowing down link establishment so as not to consider transient connectivity between nodes. This means that we are interested in making sure that a newly registered link is not just a node passing by or a node that is at the border of radio range and alternates between residing just outside and just inside radio range. Hysteresis, thus provides more robust link-sensing at the cost of some delay in establishing new links which, in our opinion, becomes necessary for providing better performance when considering high mobility scenarios.

The strategy suggested in OLSR [3] is based upon two functions, stability rule and instability rule, and uses two link-quality thresholds (HYST_THRESHOLD_HIGH and HYST_THRESHOLD_LOW). Hysteresis requires a node to maintain a link-quality value for every link. OLSR RFC specifies hysteresis based on HELLO packets. The hello packets are thus used to update the link status when it crosses one of the defined thresholds. The stability rule is applied on a registered link every time a hello packet is received at that link. The instability rule is applied to a registered link every time a hello packet is lost which is detected by using timers to determine if packets are received within the specified interval and also by tracking packet sequence numbers to find missing packets.

Using hysteresis, the status of a link is only changed if the link-quality crosses one of the two threshold:

- A link is set to be symmetric if it is currently asymmetric and value of link-quality is larger than the upper threshold (HYST_THRESHOLD_HIGH).
- A link is set to asymmetric if it is currently symmetric and value of link-quality is smaller than the lower threshold (HYST_THRESHOLD_LOW).

In figure 1, we plot the trajectory of link-quality parameter when a node receives five consecutive hello messages followed by three consecutive losses. When a hello message is received on a link for the first time, the link-quality is initialized to 0.5. On reception of next four hello messages, the stability rule is applied increasing the value of link-quality. Then for the loss of next three messages, the instability rule is applied. As can be seen from figure, it takes three hello messages to establish a link while it takes two losses to take the link into pending state which can then be broken after link timeout if there are no further reception of messages.

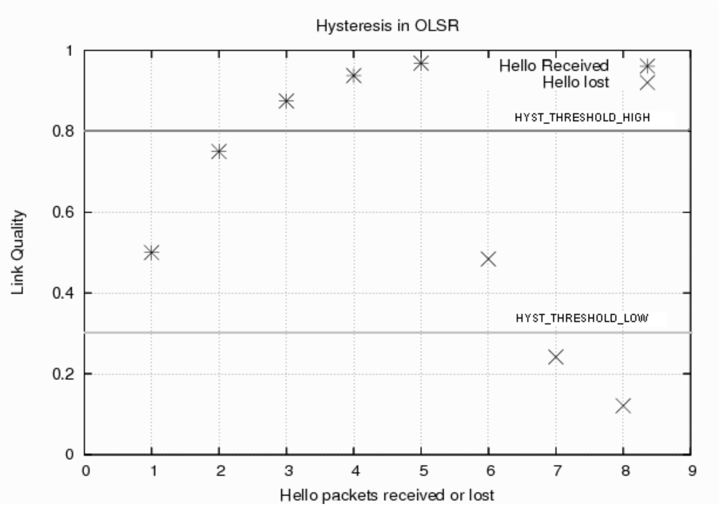


Figure 1: Hysteresis mechanism of OLSR

The above strategy while beneficial at low speeds and in low mobility scenarios, doesn't work well when there is high mobility and the speed of nodes are high. It fails since the nodes are moving at such high speeds that the time taken to establish and break links becomes significant. To overcome this problem, we modify the Hysteresis such that instead of being based on Hello losses, its now based on received signal power. By using signal power, instead of messages lost, we also gain the advantage of anticipating link breakages.

3 The proposed signal strength based link hysteresis algorithm

As mentioned above, hello loss hysteresis only detects breakage. We propose an algorithm for link hysteresis based on the received power measures.

3.1 Assumptions

In outdoor environment, where wireless mesh networks are usually deployed, the received power depends on the distance from which the packet was transmitted. The further the transmitter is, the lower is the received power. This assumes we are in a free space (without obstacles) (for dense urban areas, the parameters of the protocol will need to be adjusted since shadowing plays an important role as discussed in [8]). The received power measure is delivered by the wireless interface, and should be transmitted to the routing protocol. Thus a cross-layer approach is used. Also, in a mesh network, it is assumed that the fixed nodes/routers are deployed such that the mobile nodes always find a fixed node in its neighborhood with good enough signal strength for communication to be possible.

3.2 Signal Strength(ss) based hysteresis algorithm

The underlying idea behind the algorithm is to progressively increase the link-quality metric when a node's signal strength is increasing (node is approaching) and to decrease the metric when signal strength decreases (node is moving away). For this, we use two thresholds *ss_threshold_low* and *ss_threshold_high* "linked" to the signal strength of the received hello packets. Since the dimensioning of the network allows a mobile node to always find at least one fixed router with a good signal in its neighborhood, only the links for which the power is above *ss_threshold_low* are considered by the routing protocol. So, when the signal strength of the hello packets are above *ss_threshold_high*, the hellos are considered as received and stability rule of OLSR is applied. When the signal strength is below *ss_threshold_low*, the hello packet is supposed to be lost and the link quality is decreased by applying instability rule. When the received power is in between the two thresholds, the nodes may be moving away or approaching. In this case, we compare the signal strength of the successive hellos to anticipate link breakage or link establishment. In an ideal situation, the link-quality metric should cross the HYST_THRESHOLD_HIGH at the same time the signal power crosses the minimum reception power required for good reception i.e. *ss_threshold_high*. Also the link should be broken at the time before the communication becomes impossible.

To achieve above goals, we developed an algorithm which is detailed below. For each hello packet received, the received signal strength is measured and transmitted to the OLSR daemon. If the signal strength is greater than *ss_threshold_high*, it is considered as received. Else if the signal strength is less than *ss_threshold_low*, it is considered as lost. Considering "bad" received packets as loss (packets with low signal strength) while they have been considered as received at the lower layers, provides us with the means to anticipate link breakage. On the other hand if signal strength is in between *ss_threshold_high* and *ss_threshold_low*, the decision depends on the link status and signal strength values obtained previously from that link. When an incoming packet is considered as a reception, the link-quality is *rewarded* (like the stability rule) whereas it is *punished* when the packet is considered to be lost (like instability rule).

In figure 2 we trace the signal strength as a function of distance. The approximation of *Signal Strength (ss)*$=\frac{1}{d^4}$, where *"d"* is the distance between the nodes, is used (for simulations, however, Two-ray ground propagation model is used with free space model used for small distances; for further details, see [7]). The figure also plots the the signal strength of hello packets received when a node is approaching. Figure 3 plots the corresponding link-quality values for the hello packets received.

As can be seen from the figures 2 and 3, when the first hello packet with signal strength below *ss_threshold_low* is received, the packet is not considered as reception. Thus link-quality is not initialized. Only the reception of a packet with strength above *ss_threshold_low* initializes the link-quality

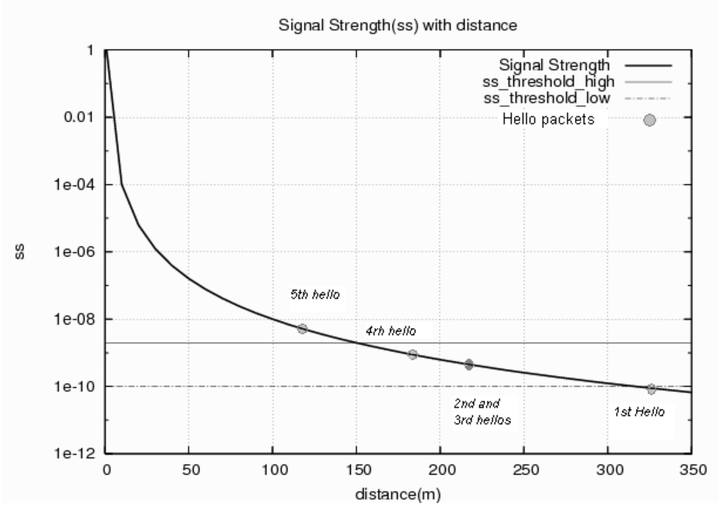


Figure 2: Signal Strength with distance

parameter. The second hello received is above the *ss_threshold_low* and thus initializes the link-quality but the link status is still pending(therefore, it can not be used to transmit messages yet). When the node receives another packet at about the same power level, there is no effect on link-quality. But whenever the difference in signal strength, which is cumulated, crosses a certain value Δ, the link-quality is *rewarded* or *punished* based on its improvement or deterioration respectively. The use of Δ allows us to reward only approaching nodes and not nodes which would stay at the same distance between successive hellos.

Here in the figure 3, it is rewarded as it is improving for the fourth hello packet. Rewarding and punishing between the thresholds for every Δ change is done to make the links more robust and reduce the time required in changing link status. Finally, when a packet with signal strength above *ss_threshold_high* is received, the link-quality is again rewarded and since it crosses the threshold, the link becomes active (link pending becomes false).

To implement the above mechanism, we made changes to the way OLSR handles the message processing which is detailed in the algorithm below.

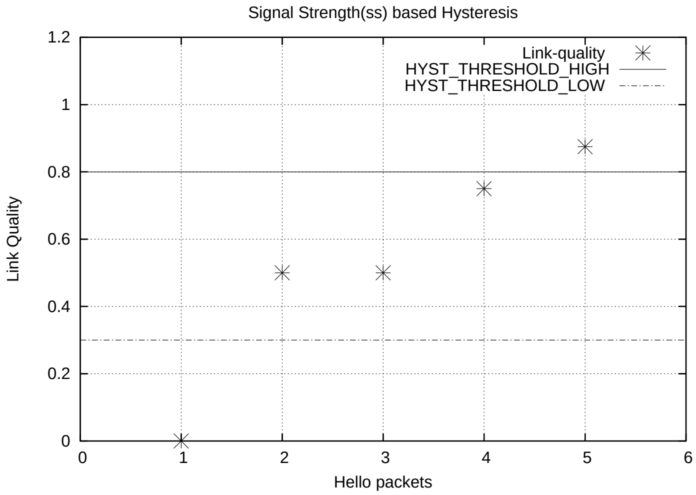


Figure 3: Hysteresis based on Signal Strength

First when a node receives a message from a neighbor for the first time, it checks the signal strength and based on it assigns it a link-quality parameter.

On receiving further hello messages from the same neighbor, it always checks the signal strength. If it is above *ss_threshold_high*, link-quality is rewarded. If it is between the thresholds, then depending on signal strength variation, it rewards or penalizes the link-quality. The status of a link is only changed if the link-quality crosses one of the two threshold. Note that punishing the link when its signal strength is deteriorating changes its link status to pending before it becomes impossible to communicate with it; thus our algorithm in fact anticipates link breakage.

Signal Strength(ss) based Hysteresis Algorithm. .

```
IF there does not exist an entry in neighbor table
    IF ss > ss_threshold_high
        Link_quality ← 1-Hyst_ss_scaling
    ELSE
        Link_quality ← Hyst_ss_scaling
    ENDIF
    Link_pending ← true
```

```
  Sum_sig_var = 0 ; (to cumulate the signal variation)
ELSE
  if there is already an entry
  IF ss > ss_threshold_high ; (good reception; we reward)
    Link_quality ← (1 - Hyst_ss_scaling ) * Link_quality + Hyst_ss_scaling
  ELSE
    (punish, reward or do nothing based on ss)
    IF Link_pending = false AND (Sum_sig_var + = Last_ss − ss) $\geq \Delta$
      (we punish since signal strength has deteriorated more than $\Delta$)
      Link_quality = Hyst_ss_scaling $*$ Link_quality
      Sum_sig_var = 0
    ENDIF
    IF Link_pending = true AND (Sum_sig_var + = ss − Last_ss) $\geq \Delta$
      (we reward since signal strength has improved by $\Delta$)
      Link_quality = min(HYST_THRESHOLD_HIGH , (1 − Hyst_ss-
      _scaling) $*$ Link_quality + Hyst_ss_scaling )
      Sum_sig_var = 0
    ENDIF
  ENDIF
  Change the status of Link if Link_quality crosses any of the two thresholds
ENDIF
```

where *ss* is the current value of signal strength; *Last_ss* is signal strength value of last hello packet; *Hyst_ss_scaling* is like HYST_SCALING parameter of OLSR but for signal strength; *ss_threshold_low* and *ss_threshold_high* are the two signal strength thresholds; *Sum_sig_var* cumulates the signal variation and compares it with Δ which is an arbitrary value used to measure change.

3.3 Hybrid hysteresis algorithm

The above described algorithm does not consider packets which are really lost (e.g. due to collision, error in Frame Check Sequence etc.) since it can measure signal power only from received packets. That means that when a packet is lost, the corresponding link-quality is not affected leading to a delay (have to wait until the next packet) in punishing the link if the later is deteriorating. Also if a node suddenly disappears (is switched off), then we will have to wait until the link expires to invalidate the link and packets sent during the time will be lost. It is also important to note that hello messages can also be lost due to collision, as 802.11 doesn't have reliable transmission of broadcast messages. Knowing that packets sent on links which are further from the node are more likely to be lost (more probability of moving out of range and collision in a given time), the punishment becomes inefficient.

The main idea of the hybrid algorithm is to combine both Hello loss and signal strength based hysteresis. If a hello packet is lost, the algorithm acts as in link hysteresis proposed in the OLSR, i.e. instability rule is applied. On the other hand, for each hello packet received, the algorithm acts as the proposed signal strength based hysteresis. Packet has to be considered as a reception or as a loss depending on its quality (signal strength value). Note that in this case, *Hyst_ss_scaling* and HYST_SCALING could equal or different. By differentiating the two, we could introduce the idea of 'weak punishment', 'weak reward' and 'strong punishment', 'strong reward'. They can be used to micro-manage the network based on the network load and mobility.

4 Simulation results

For simulations, we have used an implementation of OLSR on the network simulator version 2(ns-2). Ns-2 [7] is an open source, discrete event simulator

Table 1: Simulation Parameters

Parameter	Value
Simulation time	200 [sec]
Terrain range	594 by 594 [m]
Number nodes	29
Propagation model	Two-ray ground
Bandwidth	11 Mbps
Mobility model	Random way point
Pause time	0 sec
MAC protocol	IEEE802.11
MAC queue size	50
Queue type	Drop Tail / Priority Queue
Traffic type	Constant Bit Rate(CBR)
Antenna model	Omni-Antenna
Hyst_ss_scaling	0.5
ss_threshold_low	-9.3 (logarithmic value)
ss_threshold_high	-8.9 (logarithmic value)
Δ	0.2

which is widely used for research purposes. It has an excellent implementation of the 802.11 standards at physical, Data link and higher layers. The parameters were configured such that the physical interface corresponded to the 914MHz Lucent WaveLAN DSSS radio interface. Other parameters used for the simulation are given in table 1 with simulation time, terrain and number of nodes varying according to the scenarios. The parameters *Hyst_ss_scaling*, *ss_threshold_low*, *ss_threshold_high* and Δ have been selected such that they mimic the default OLSR behavior i.e. we require three hello messages to establish a link and two to break it. This is done so that the two approaches can be compared objectively. Also, all the given plots are plotted with a confidence interval of 95%.

4.1 Chain scenario

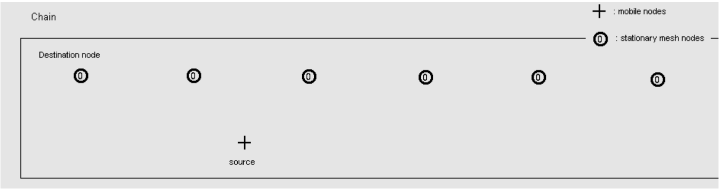
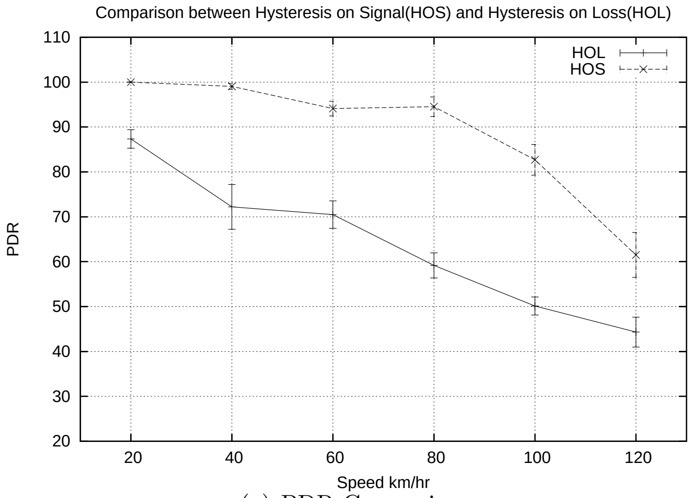


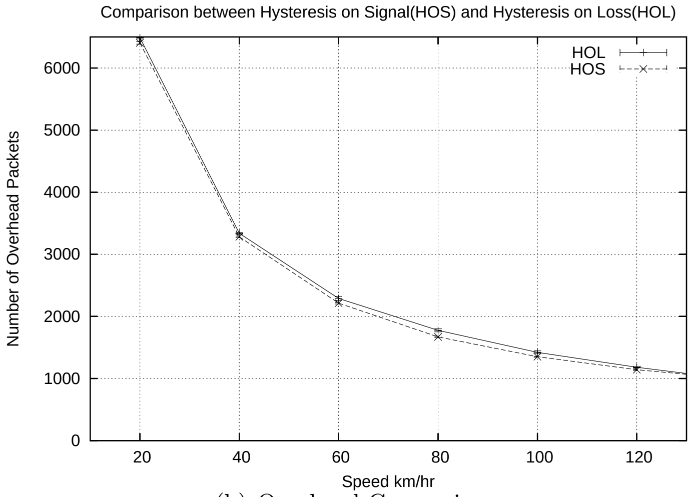
Figure 4: Chain

First of all, to validate the connectivity, we made simulations such that we placed stationary nodes in a straight line, forming a chain, while a node moved passed them at different speeds. The moving node was made the source which transmits data to the node which is located at the beginning of the chain (first node). As the source moves away from the destination, it has to switch the nodes for transmitting data. OLSR being a shortest path algorithm, switches to a new link only if a new shorter path is discovered or the older path is no longer valid. So it tries to keep the old link in use as long as possible before switching to the new link as the older path is always the shortest. Also, since there exists only a single path (in the chain) for information exchange, the loss of control packets is an important factor here. So to minimize packet loss due to collision, constant bit rate traffic with only 2 packets of 512 bytes each per second were used.

The topology consisted of an area of about 1500 * 300 with 10 stationary nodes, each placed at a distance of 130m forming a chain. The mobile node was placed at a distance of 100m from the chain and its speed was varied from 5 m/sec to 30 m/sec. The source started moving and sending data after 50 sec and the simulation time was adjusted according to the time needed by the mobile node to traverse the topology.



(a) PDR Comparison



(b) Overhead Comparison

Figure 5: Results of Chain Scenario

In the above scenario, the performance metric used is Packet Delivery Ratio(PDR). PDR is the ratio between the number of packets delivered to the destination to that sent by the source. The results for this scenario is shown in figure 5(a). As can be seen from the figure, the hysteresis on signal strength (also referred to as Hysteresis on Signal) performs much better than Hysteresis on hello Loss (referred as Hysteresis on Loss). This is because Hysteresis on Loss waits for at least 2 packets loss, before breaking the link. On the other hand, Hysteresis on Signal has nearly 100% PDR since it breaks and establishes the links quickly. Figure 5(b) compares the overhead packets sent when using Hysteresis on Signal and Hysteresis on Loss. As can be seen from the figure, there isn't much difference between the two. In short it can be said that our approach doesn't cause any substantial increase in number of overhead packets.

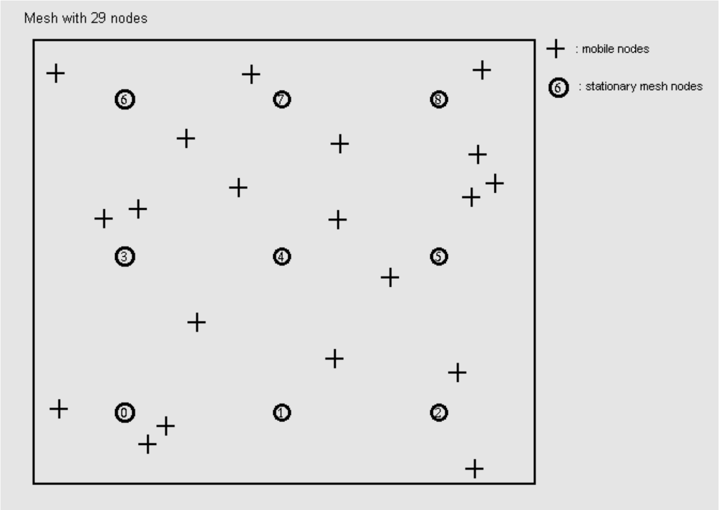


Figure 6: Mesh structure

4.2 Grid mesh network scenario

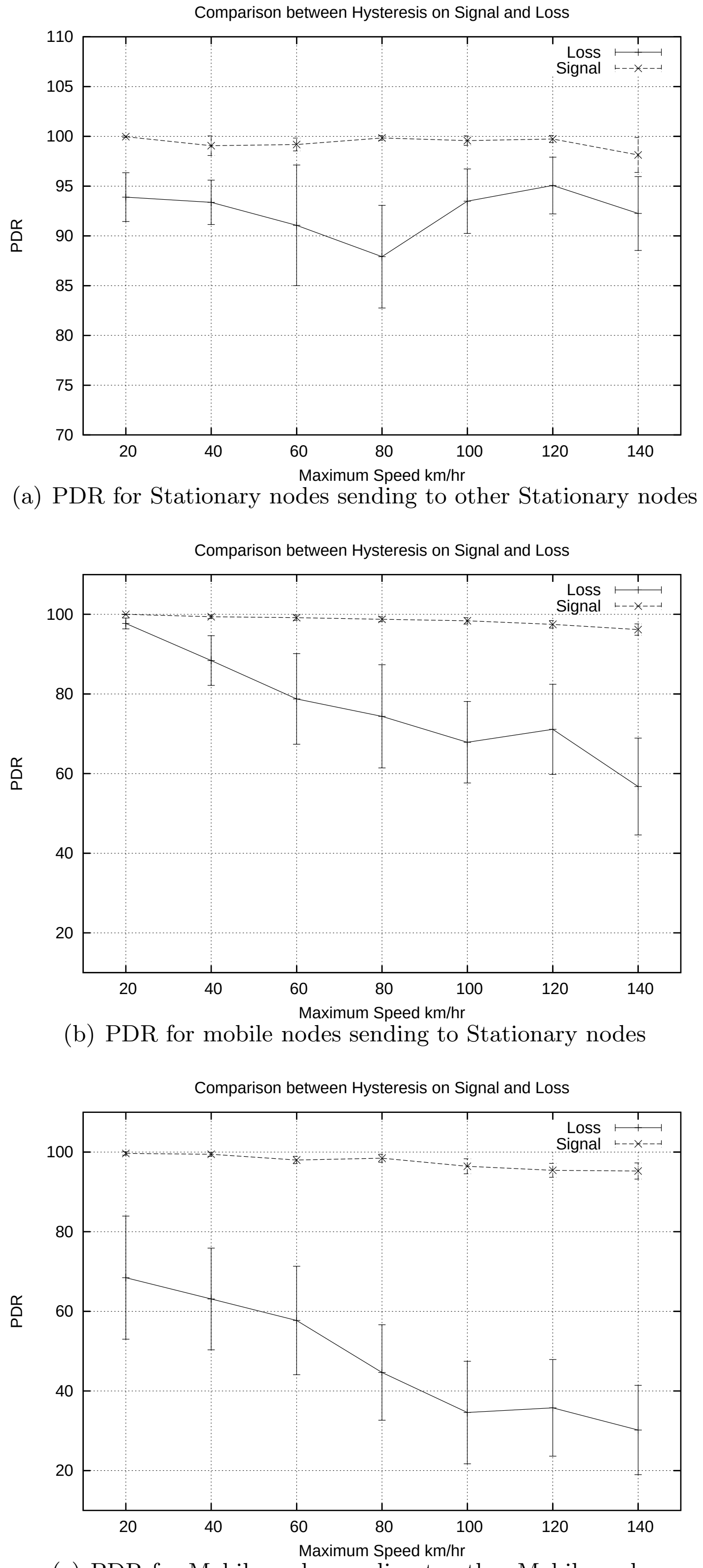
Then we considered a mesh scenario with 29 nodes in which 9 of the nodes were stationary(henceforth referred to as mesh nodes) and placed such that they formed a mesh (figure 6). A mesh node in this case is connected to its horizontal and vertical neighbors only and not to the diagonal neighbor. The mobile nodes, on the other hand, were placed using uniform random distribution. Also, their movement, speed and direction was based on the Random way point model. Random Waypoint (RWP) model is a commonly used mobility model in Ad Hoc networks. Briefly, in the RWP model [10] a node moves directly towards the next waypoint (destination) at a certain velocity v selected using uniform distribution from $[0, v_{max}]$. Once the node reaches the waypoint the next waypoint is drawn randomly from the uniform distribution. Furthermore, it is possible to introduce "thinking times" when the node reaches each waypoint. The scenario is depicted in figure 6.

Other parameters used for the simulations are detailed in the table 1.

We simulated first with a single traffic with both Hysteresis based on Loss and Hysteresis based on Signal. The sources and destinations were varied to cover all possible scenarios. Three cases were considered and results plotted

- a stationary node sending to another stationary node
- a mobile node sending to a stationary node, and finally
- a mobile node sending to another mobile node.

The data used for each of the above scenario consisted of 2 packets of 512 bytes per second. The results for each of the above three scenarios are shown in figures 7(a), 7(b) and 7(c) respectively. As can be seen from the figures, Hysteresis on Signal easily outperforms hysteresis based on loss. This is because of the anticipated link breakage which makes it use only those path which are stable. In the case when stationary nodes are sending to other stationary nodes, we don't have a 100% PDR, since OLSR being a shortest path algorithm, often selects mobile nodes to forward data. This causes link breakage when nodes move out of range and hence some packets are lost before the network reconfigures itself.

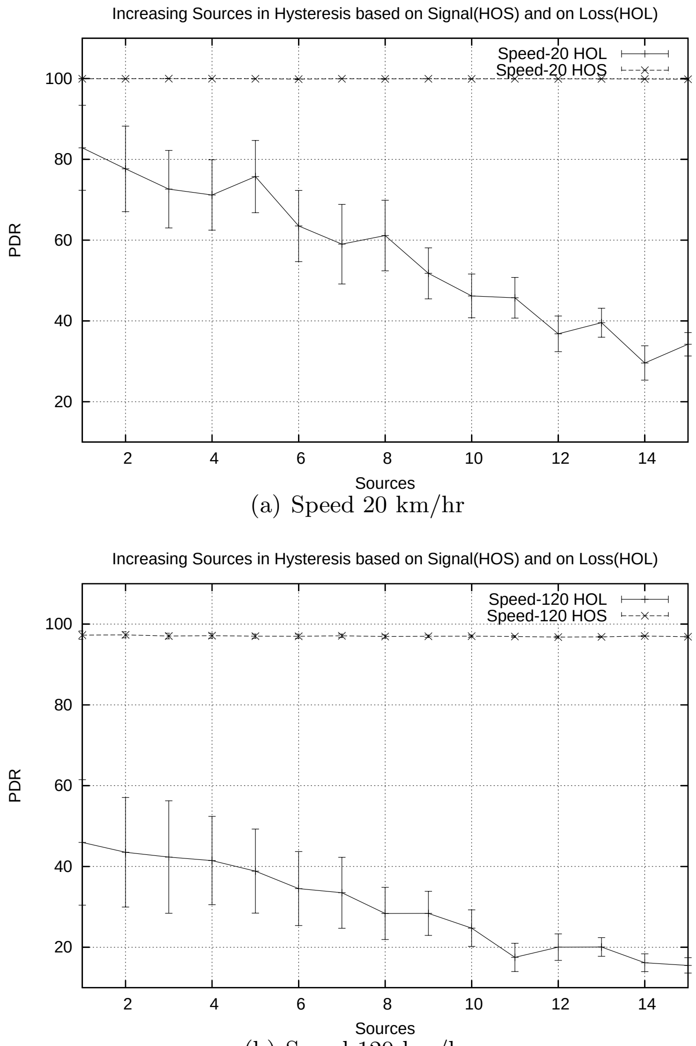


(a) PDR for Stationary nodes sending to other Stationary nodes

(b) PDR for mobile nodes sending to Stationary nodes

(c) PDR for Mobile nodes sending to other Mobile nodes

Figure 7: Grid Mesh Networking Scenario



(a) Speed 20 km/hr

(b) Speed 120 km/hr

Figure 8: Multiple Sources with Hysteresis on Loss and Signal

Then, we increased the sources from 1 to 15 and saw how the protocol scales. The results are shown in figure 8(a) and figure 8(b) for Hysteresis on Loss and Hysteresis on signal with speed 20 and 120 respectively. As can be seen from the figures, our approach not only scales well but outperforms Hysteresis on signal by a big margin. This is because in Hysteresis on Loss, sometimes a node breaks its link with its MPRs. When this occurs, its TC messages are not diffused through out the network and some nodes in the network are left with invalid topology. This creates routing loops in the network resulting in continous exchange of RTS/CTS messages for sending cbr data until Time To Live(TTL) of data expires. This continous exchange of RTS/CTS messages, not only reduces the throughput of the network, but also causes a chain reaction since because of it other TC messages are also prevented from disseminating the valid topology information. On the other hand, by anticipating breakages in Hysteresis on Signal and thus changing MPRs in time, we seldom loose links and thus have improved performance. It was seen that there were almost thrice as many RTS/CTS messages in Hysteresis on Loss as compared with Hysteresis on Signal.

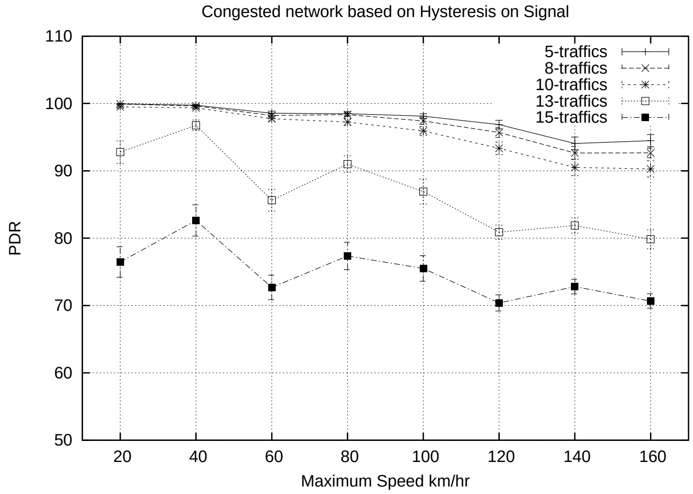


Figure 9: PDR with high traffic based on Hysteresis on Signal

Finally, we increased the data rate progressively to determine how the proposed changes to protocol performs under a congested network. The results with the above scenario but with traffics between 5 to 15 and with data rates of 20 packets of 512 bytes each per second are given in the figure 9. There is a graceful degradation of performance in case of congested network as can be seen from the figure 9.

5 Conclusion

In this paper, we have presented a link management technique for OLSR to locally manage links. This new mechanism is based on signal strength hence cross layer approach is used. The hysteresis mechanism provided by OLSR is improved upon by using signal strength in combination with the hello loss based hysteresis. The signal power is used to determine if the link-quality is improving or deteriorating while packet losses are handled through the hysteresis mechanism specified in OLSR RFC. This not only makes the link management more robust but also helps in anticipating link breakages thereby greatly improving the performance. Our future work will focus on trying to optimize the thresholds based on network configuration. Also, comparison of the above mechanism in terms of overhead and performance with other related protocols is also planned.

References

[1] S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey. Route-lifetime assessment based routing (RABR) protocol for mobilead-hoc networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 3, pages 1697–1701, New Orleans, LA, USA, 2000.

[2] M. Benzaid, P. Minet, and K. Agha. Integrating fast mobility in the olsr routing protocol, 2002.

[3] T. Clausen (ed) and P. J. (ed). Optimized Link State Routing protocol (OLSR), October 2003. RFC 3626, Experimental.

[4] S. Crisostomo, S. Sargento, P. Brandao, and R. Prior. Improving AODV with preemptive local route repair. In *Wireless Ad-Hoc Networks, 2004 International Workshop on*, pages 223–227, May/June 2004.

[5] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks, 1997.

[6] I. F.Akyildiz and X. Wang. A survey on wireless mesh networks. *IEEE Communications magazine*, 43(9):S23–S30, September 2005.

[7] K. Fall and K. V. Editor. *The ns Manual.* UC Berkeley and LBL and USC/ISI and Xerox PARC., http://www.isi.edu/nsnam/ns/ns-documentation.html, January 2005.

[8] G. Gaertner, E. ONuallain, A. Butterly, K. Singh, and V. Cahill. 802.11 link quality and its prediction - an experimental study. In *PWC*, pages 147–163, 2004.

[9] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *Mobile Computing and Networking*, pages 43–52, 2001.

[10] E. Hyytia and J. Virtamo. Random waypoint mobility model in cellular networks, 2005.

[11] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The Dynamic Source Routing protocol for mobile ad hoc networks (DSR), February 2007. RFC 4728, Experimental.

[12] B. S. Manoj, R. Ananthapadmanabha, and C. S. R. Murthy. Link life based routing protocol for ad hoc wireless networks. In *Computer*

Communications and Networks, 2001. Proceedings. Tenth International Conference on, pages 573–576, Scottsdale, AZ, USA, 2001.

[13] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-demand Distance Vector (AODV) routing, July 2003. RFC 3561, Experimental.

[14] V. T. Raisinghani and S. Iyer. Cross-layer design optimizations in wireless protocol stacks.

[15] I. Ramani and S. Savage. Syanscan: Practical fast handoff for 802.11 infrastructure networks. In *INFOCOM'05*. IEE Communications Society, March 2005.

[16] P. Samar and S. B. Wicker. Link dynamics and protocol design in a multihop mobile environment. *IEEE Transactions on Mobile Computing*, 5(9):1156–1172, 2006.

[17] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wirel. Pers. Commun.*, 4(2):103–139, 1997.