

Cross-Layer MANET Routing Algorithm Based on Radio Link Stability

Kambiz Homayounfar

PHYBIT, INC.

AM Building 5F, 2-3-3 Higashi Gotanda, Shinagawa, Tokyo 141-0022, Japan

E-mail: kambiz@phybit.com

Abstract Routing is one of the critical functions in a Mobile Ad Hoc Network (MANET). It enables unreachable mobile nodes to communicate via intermediate nodes. The lack of fixed coordination in MANET compels mobiles to cooperate in order to implement routing, and this cooperation needs simple and robust metrics to indicate which routes are best. The challenge of MANET routing is that node mobility creates rapidly varying links in terms of bandwidth and interference. This paper introduces radio link stability and shows how cross-layer cooperation leads to higher throughput and lower delays by avoidance of unstable links.

Key words Routing, Mobile Ad Hoc, Link Quality, End-to-End Delay, Packet Delivery Ratio, Packet Overhead

1 Introduction

Routing is the function of finding a path for a Protocol Data Unit (PDU) to go from a source to a destination. For wired networks, this function is performed at Layer 3 (Network Layer) by routers whose job is to find the best path for the PDU to travel along. Here, the metric that characterizes the selection of the best path is the geographic distance between the nodes, or the number of intermediate nodes (hops) that necessary for the packet to reach its destination. This approach works well and much of the research activity in wired routers focuses on higher speeds at lower costs. In mobile (wireless) networks, the situation is different. If the mobile network is cellular, the router function remains in the core network and the mobile devices are not expected to route any packets.

In a MANET, the mobile nodes are the routers. They must cooperate in order to pass packets around and they must work efficiently, with lowest delay and smallest possible energy consumption. The routing algorithms used in a MANET are therefore very different to wired networks. To begin with, in a MANET routers move. The meaning of geographic distance in mobile networks as it pertains to routing algorithms becomes questionable. The routing algorithms conceived for wired networks never considered the possibility that the boxes would be free to move randomly and fit into the teenager palms. Also, MANET routers can organize themselves as they wish, with no centralized coordination between them. MANETs may operate in a stand alone fashion, as in military or in emergency situations, or several amongst them may be connected to the Internet, providing web access for the rest.

Surveys of routing algorithms for MANET [1] report that three categories have emerged: on-demand (such as AODV [2] and DSR [3]), proactive such as DSDV [1], and hybrid methods (also covered in [1]). These algorithms share common features. When a source has a packet to send, it first checks to see if itself knows of a path. If so, then it goes ahead and sends out the packet hoping a neighbor would pick it up and forward it. If the source does not know of a route, it sends out a route request. The intermediate nodes help out by forwarding the packets until the request eventually reaches its intended destination. (if it does not, then the source sends another request, and yet another until the packet lifetime expires in which case it is dropped). Once the intended destination receives a request, it answers with a reply packet. Again the intermediate nodes cooperate to this time, deliver the reply to the source. Along the way, all nodes keep track of where reply messages came along and tag the packet with their own identification (ID) numbers so that the source can know how to route the actual data-bearing packet (the PDU). As simple and as sensible this algorithm sounds, when it comes to MANETs, it is deeply flawed. The problem is that by the time a reply comes back to the source, the path that was suggested as the best may no longer be so. This turns MANET routing into a formidable challenge. Independent studies like [4] and [5] report that MANETs barely work. If they do work, it is because a path was found. Take it or leave it. And most sources do the sensible thing and take it.

This paper describes an algorithm that helps MANET routing in two ways. First, it provides a metric that by its nature warns of the possibility that links can break. This metric, which can be considered a link stability index, accumulates at each node to form a path stability index. Therefore, the algorithm enables inter-

mediate nodes to balance stability of the route with end-to-end delay. Its principle is simple: intermediate nodes must wait before they re-broadcast a request they just picked up from a neighbor. This waiting mechanism has, in turn, two advantages. First, in case a better link comes along, there is no need for re-broadcast. This reduces overhead of redundant broadcasts. Second, by using a simple waiting mechanism that depends on link stability, end-to-end delay reduces.

2 Basics of Cross-Layer Design

The algorithm proposed here uses cross-layer design concepts. This approach, in its general form, is depicted in Figure 1 and enables protocol layers to work together. The purpose of cross-layer cooperation is adaptation to channel conditions. Layer 1, the physical layer or PHY, measures the quality of a radio link and forms the metric CQI, the Channel Quality Indicator. Many methods to measure CQI are available, mostly based on measuring the signal-to-noise ratio in a link. The CQI, once used within PHY, enables adaptive modulation (and coding). It helps the transmitter and the receiver to decide which modulation and coding scheme works best for the link within a particular time-frame. Layer 2, the media access control (or MAC) layer is in charge of scheduling access to the channel by the users. MAC can benefit from CQI measurements to figure out which users are best scheduled now (if their channels are good) or later. Layer 3, or the network layer, is responsible for routing. The concept of using PHY metrics for adaptive routing is not new. In fact, significant results based on physical layer constraints have been reported in [8] and [9]. What is new, and the intention of this paper, is a simple and effective method for mapping physical layer measurements to link stability (or rather instability) and then including link stability in routing algorithms.

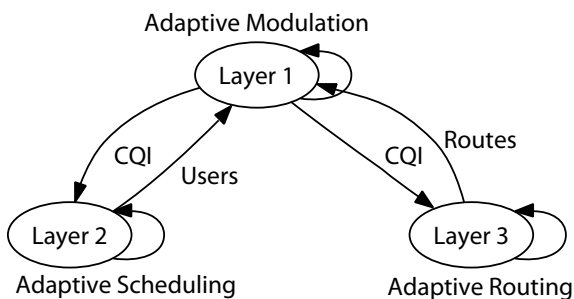


Figure 1: Outline of Cross-Layer Design

3 CLMRA: A Cross-Layer MANET Routing Algorithm

A Cross-Layer MANET Routing algorithm (CLMRA) is now described. Similar to AODV and DSR, CLMRA is an on-demand routing protocol. When a source wants

to send a data packet PDU to a destination, it first checks its own routing table to see if it has a valid path, in which case the source uses the valid path to setup the path for the . If not, the source emits a Route Request (RREQ) packet. Any node along the way to the destination who hears this RREQ must act according to the rules of CLMRA described below. Central to the operation of CLMRA is the requirement for any node to update a path metric field in the RREQ that it emits. This metric is a measure of the cumulative stability of the path as far as a given intermediate node is concerned. In addition, CLMRA features a waiting mechanism that permits a degree of self-control for each intermediate node. In the essence, a node does not simply re-broadcast an RREQ just because it has a valid node to send it to. Instead it waits for a period in case better links show up. The duration of this waiting time depends on the stability of the link where RREQ came from.

3.1 Radio Link Stability

CLMRA uses two metrics to achieve efficient routing. First is denoted ψ by and is a measure of how unstable the link is. It can be called a Link Instability Index. The second is denoted as Ψ , and is an accumulation of ψ for an entire path, so it can be considered as a Path Instability Index.

ψ is calculated from physical layer measurements as follows. The channel decoder must be a soft-decision decoder, such as a Viterbi or turbo decoder. For every decoded bit b_m , the soft channel decoder outputs a corresponding Bit Error Probability p_m . An efficient method to map output of a soft channel decoder to the bit error probability is described in [7]. In effect, p_m is a time series and its variation over an entire packet is used as an indicator of link instability.

Three levels of link instability were used to measure link stability, characterized by three thresholds $Th1 = 0.125$, $Th2 = 0.25$ and $Th3 = 0.375$. Observe that p_m must be significantly less than 0.5, otherwise there is a high probability that an erroneous bit would be present and the entire packet could not be correctly decoded. In other words, the fact that a packet is received indicates that all $p_m \ll 0.5$. The three limits on p_m help to determine the degree of link instability, as shown in Table 1.

Table 1: Conditions for Link Instability

Instability	Condition	Index
Low	All $p_m < Th1$	$\psi = 0$
Medium	Any $p_m > Th2$	$\psi = 1$
High	Any $p_m > Th3$	$\psi = 2$

Figures 2 to 4 show how link stability works. For example, Figure 2, all values of $p_m < Th1$, so the link is indexed as having a low risk of un-stability and $\psi = 0$. On the other hand, as Figure 4 shows, for a few values of

$p_m > Th3$. So this link has a high risk of becoming unstable and $\psi = 3$. Figure 3 shows a link with a medium risk of turning un-stable.

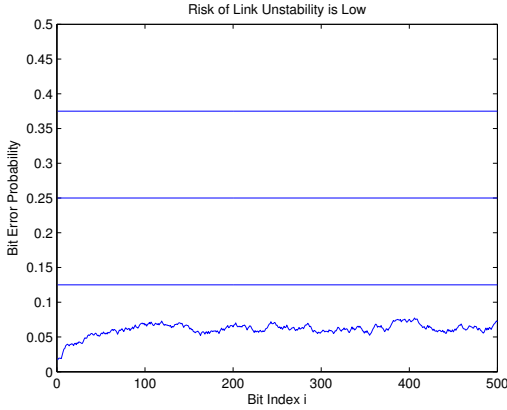


Figure 2: Stable Link

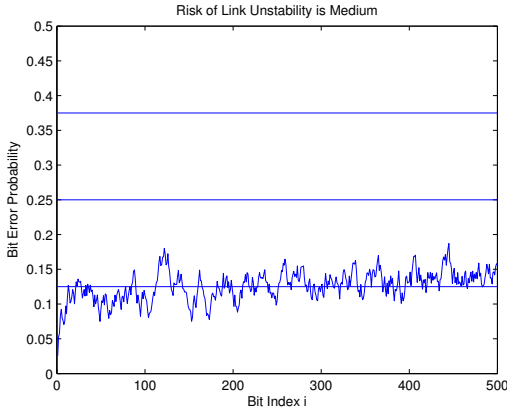


Figure 3: Medium Risk of Link Unstability

3.2 CLMRA Specification

Based on definitions of link instability given above, the CLMRA can be specified as in Algorithm 3.1. Let us consider the operation from the perspective of an intermediate node v_i . Assume, without any loss of generality, that the nodes are $v_0, v_1, \dots, v_{i-1}, v_i, \dots, v_n$, with v_0 as the source, and v_n as the destination (i.e. a total of $n+1$ nodes). Three events can take place for v_i .

3.3 Event 1: $v_i \leftarrow PDU$

In this case, if v_i can reach another node (say v_{i+1}), then v_i simply forwards the *PDU* it received to v_{i+1} . However, if v_i cannot reach any nodes, then it must take custody of *PDU* and become the source itself. So v_i buffers the just received *PDU* and initiates a new RREQ.

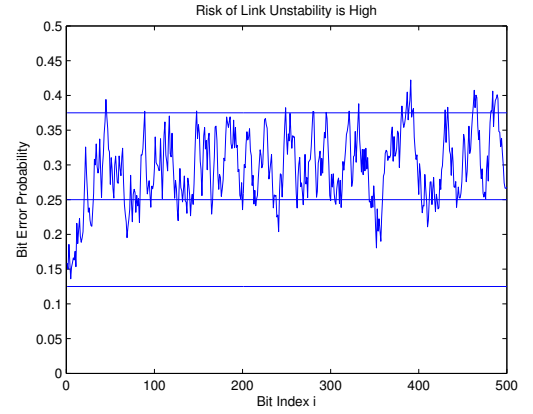


Figure 4: High Risk of Link Unstability

3.4 Event 2: $v_i \leftarrow RREP$

The operation of v_i in this event is also simple. It updates its routing table (assuming that v_i is not the destination node) and then follows a similar procedure to Event 1 to either forward the RREP or initiate a new RREQ.

3.5 Event 3: $v_i \leftarrow RREQ$

The heart of CLMRA is here. First we label the node that sent out the RREQ as v_{i-1} (again no generality is lost here). Instead of simply following similar procedures to Events 1 or 2, v_i must check several conditions first before forwarding the RREQ. The most obvious is to see if v_{i-1} has a better (in terms of path stability) than v_i . This can be checked by the condition $\Psi_{0,i-1} < \psi_{i-1,i}$. If so, why add instability to the path of v_{i-1} ? In fact, v_i should drop RREQ here, in case another node can hear v_{i-1} and possibly cause less instability. Otherwise, v_i puts CLMRA to work some more. The link instability index is computed as described in 3.1. Specifically, v_i can use information from its PHY to calculate the index $\psi_{i-1,i}$, and also use this index to update the path instability index $\Psi_{i-1,i}$.

$$\Psi_{0,i} = \Psi_{0,i-1} + \psi_{i-1,i} \quad (1)$$

Recall that v_i must get $\Psi_{0,i-1}$ as a field within the RREQ packet. Equation ?? also shows how v_i prepares its very own Path Instability Index $\Psi_{0,i}$ in case it decides to forward RREQ.

Before forwarding RREQ, v_i waits to see if nodes other than v_{i-1} can do better. Suppose during this waiting time, another node v_j sends out an RREQ that v_i can overhear. If the condition $\Psi_{0,j} < \Psi_{0,i}$ then the node v_j has a better path to v_n and therefore v_i had better drop the packet. (v_i should not interfere with other nodes along the path from v_j to v_n who can possibly do a better job). Also, during the waiting time, if the accumulated path from any node, say v_k , turns out

to have higher stability than the path through (v_{i-1}, v_i) , then RREQ should be dropped.

If none of the conditions above hold true, then v_i can safely add v_{i-1} to its routing table and reset the waiting timer.

3.6 Reducing Overheads: Why Wait?

In CLMRA, like other routing algorithms, RREQ packets create overhead traffic for the network and ought to be avoided. By including the waiting mechanism above, CLMRA provides a degree of self-control that helps to reduce overhead. The idea behind waiting before forwarding an RREQ is simple. If the node from which RREQ came to v_i has a good link, then the waiting time should be short. Otherwise, it ought to be longer to give chance to other nodes with potentially better link stability. So the simplest way to set the waiting timer ω is through:

$$\omega_i \propto \psi_{i-1,i} \quad (2)$$

In the simulations ω was set to 1 millisecond (ms).

Table 2: CLMRA Simulation Parameters

Parameter	Setting
Simulator	GloMoSim [10]
Network Area	1 km x 1 km
Mobile Population	50 Nodes
MAC Protocol	802.11
Packet Size	64 Bytes
Packet Rate	10 Packets/s
Node Range	250 m
Mobility Model	Random Waypoint [11]
Node Speed	0 to 72 km/h
Traffic Type	Constant Bit Rate (CBR)

4 Simulation Results

To evaluate the performance of CKMRA, and compare it to AODV, system-level simulations were performed based on the setup shown in Table 2. Figure 5 shows the delivery ratio for all the packets exchanged during a simulation run. Because CLMRA tries to use stable paths (ones which are less likely to break) the algorithm can deliver more packets to the destination than AODV. The improvement over AODV is particularly noticeable in high mobility region. As the mobility decreases, we approach the static case for which AODV routine metrics catch up with CLMRA. The former simply drops data packets whenever routes break. CLMRA also experiences packet loss. Less stable paths may break due to mobility, or can disappear during the route reply phase as neighbors move away from each other. Collisions and contention can also cause packet loss. End-to-end delay is shown in Figure 6. The lower delays associated

Algorithm 3.1 Cross-Layer MANET Routing Operations at Node v_i

```

1:  $v_0$ : Origination (Source) Node
2:  $v_n$ : Destination Node
3:  $v_i$ : Intermediate Node
4: RREQ: Route Request Packet
5: RREP: Route Reply Packet
6: PDU: Data Packet
7:
8: EVENT 1:  $v_i \leftarrow$  PDU {If a next hop is available,
   call it  $v_{i+1}$ }
9: if  $v_{i+1}$  is available then
10:   Forward PDU to  $v_{i+1}$ 
11: else
12:   Buffer PDU
13:    $v_i = v_0$  { $v_i$  now takes custody of PDU}
14:    $v_i$  sends out a new RREQ
15: end if
16:
17: EVENT 2:  $v_i \leftarrow$  RREP
18: if  $v_i \neq v_n$  then
19:    $v_i$  updates its routing table
20:    $v_i$  sends a RREP
21: end if
22:
23: EVENT 3:  $v_i \leftarrow$  RREQ
24: Assume RREQ came from  $v_{i-1}$ 
25: Compute  $\psi_{i-1,i}$ 
26: Compute  $\Psi_{0,i}$  as in Eq. 1
27: Set timer  $\omega$  based on Eq. 2
28: while  $\omega$  ticks do
29:   if  $\Psi_{0,i-1} < \psi_{i-1,i}$  then
30:     Drop RREQ
31:     break
32:   end if
33:   if  $\exists [v_j | j \neq i-1 \wedge \Psi_{0,j} < \Psi_{0,i}]$  then
34:     Drop RREQ
35:     break
36:   else if  $\exists [v_k | k \neq i-1 \wedge \Psi_{0,k} + \psi_{k,i} < \Psi_{0,i-1} + \psi_{i-1,i}]$ 
   then
37:     Drop RREQ
38:     break
39:   else
40:      $v_i$  adds  $v_{i-1}$  to its path
41:      $v_i$  updates routing table
42:     Forward RREQ
43:     Reset the timer  $\omega$ 
44:   end if
45: end while

```

with CLMRA can be attributed to the waiting mechanism employed in the algorithm. This may appear ironic, yet it stems from cooperation between nodes. Through waiting, nodes enable others with potentially better paths to perform the forwarding instead of forwarding redundant packets that can cause interference and contention. Packet overhead is shown in Figure 7. The y-axis shows how overhead packets are on average transmitted for every successful packet. In the case of AODV, it takes more than 4 overhead packets to deliver one PDU successfully. For CLMRA, the ratio is substantially less. Here we see how the simple waiting mechanism pays benefits as a ratio of almost 0.5 (1 overhead for every two successful PDUs) for high mobility case, down to almost nil with for low mobility (pause time of 250s).

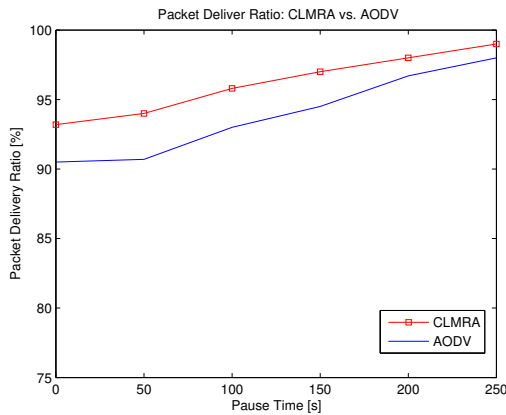


Figure 5: CLMRA Packet Delivery Results

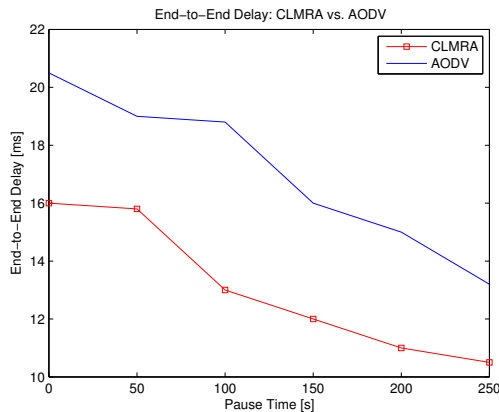


Figure 6: CLMRA Delay Results

5 Conclusions

This paper introduced a cross-layer MANET routing algorithm, CLMRA, with two features. First, a metric for radio link stability was introduced based on physical layer measurement of bit error probability as a time series for an entire packet. This metric included a sim-

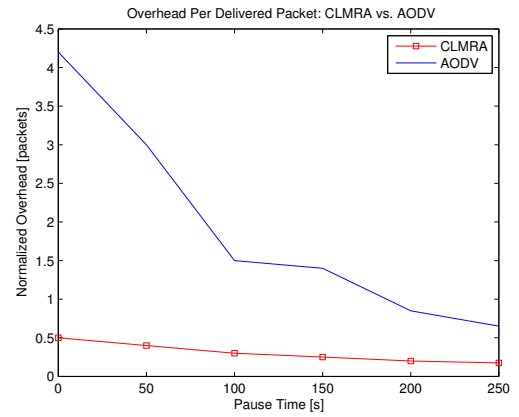


Figure 7: CLMRA Overhead Results

ple mapping to degree of instability in a link based on how often bit error probability exceeded certain thresholds. Second, the algorithm is equipped with a waiting mechanism that depends on the magnitude link stability. By waiting for better route request packets to arrive, the overhead in forwarding redundant packets was lowered. The performance of CLMRA compared to AODV showed noticeably better throughput, as indicated by higher packet deliver ratio. Yet the most significant advantage of CLMRA turns out to be the reduction of overhead packets.

References

- [1] M. J. Lee, J. Zheng, X. Hu, H. Juan, C. Zhu, Y. Liu, J. S. Yoon, and T. N. Saadawi, A New Taxonomy of Routing Algorithms for Wireless Mobile Ad Hoc Networks: The Component Approach, *IEEE Communications Magazine*, November 2006.
- [2] C. E. Perkins, E. M. Belding-Royer, and S. Das, Ad Hoc on Demand Distance Vector (AODV) Routing, IETF Internet draft, June 2002, draft-ietf-manet-aodv-11.txt
- [3] D. Johnson, D. A. Maltz, and J. Broch, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, IETF Internet draft, April 2003, draft-ietf-manet-dsr-09.txt
- [4] K. W. Chin, J. Judge, A. Williams and R. Kermode, Implementation Experience with MANET Routing Protocols, *ACM SIGCOMM Computer Communications Review*, Volume 32, Number 5: November 2002.
- [5] D. S. J. De Couto et al., A High-Throughput Path Metric for Multi-Hop Wireless Routing, *Proc. 9th ACM Int. Conf. Mobile Comp. and Net. (MobiCom 2003)*, San Diego, CA, Sept. 2003.
- [6] S. J. Lee and M. Gerla, AODV-BR: Backup Routing in Ad hoc Networks, *Proceedings of IEEE WCNC 2000*, Chicago, IL, September 2000.

- [7] J. Ch. Fricke, M. I. Rafique, and P. A. Hoeher, Routing metrics based on soft-output decoding, in Proc. IEEE 66th Vehicular Technology Conference (VTC 2007-Fall), Baltimore, MD, USA, Sep.-Oct. 2007.
- [8] G. Ferrari, S. A. Malvassori, M. Bragalini, and O. K. Tonguz, Physical layer-constrained routing in ad-hoc wireless networks: A modified AODV protocol with power control, in Proc. Int. Workshop on Wireless Ad-hoc Networks 2005 (IWWAN 2005), London, UK, May 2005.
- [9] I. Rubin and Y. Liu, Link stability models for QoS ad hoc routing algorithms, in IEEE 58th Vehicular Technology Conference (VTC), vol. 5. IEEE, 2003, pp. 3084-3088.
- [10] University of California Los Angeles Computer Science Department Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems. Available at: <http://pcl.cs.ucla.edu/projects/domains/glomosim.html>.
- [11] D.B. Johnson and D.A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, In Mobile Computing, Kluwer Academic Publishers, 1996, pp. 153-181.