A Load aWare Routing (LWR) Based on Local Information

Yunjung Yi, Taek Jin Kwon and Mario Gerla
University of California, Los Angeles
{yjyi, gerla}@cs.ucla.edu
*Telcordia Technologies, Applied Research
tkwon@research.telcordia.com

Abstract - This paper suggests a new load aware routing (LWR) enhancement that improves on-demand routing protocols significantly in ad hoc networks. This paper describes deterministic characteristics of ad hoc networks and suggests improved routing strategies that can sustain the network functionality even under highly stressful data traffic. LWR evenly distributes the load throughout the networks. It also reduces redundant packet flooding which is major limitation of conventional ad hoc routing algorithms. Simulation results illustrate the performance improvements obtained with LWR by using new flooding control over conventional on-demand routings.

I. INTRODUCTION

An ad hoc network is a fast deployable wireless network that is typically subject to dynamic network topology change due to node mobility.

An important performance issue in an ad hoc network is the interference among neighbors. As the interference increases, the performance of system degrades dramatically. One of the most effective ways of dropping such interference is reducing redundant transmissions e.g. unnecessary flooding relays. In fact, ad hoc network routing protocols are after designed based on overhead reduction criteria.

There are two leading ad hoc routing schemes. One is proactive routing (e.g. OSPF), and the other is the reactive (on-demand, e.g. AODV) routing. Both routing schemes have drawbacks. Proactive routing usually generates a large and uniform number of control packets to maintain routes no matter how frequently the routes are used. On the other hand, reactive routing finds a route only when it is needed. This method is effective when there are only a few communication pairs, but it is very ineffective when many connections are active.

In this paper, we focus on the reactive routing and provide a solution that mitigates drawbacks of reactive routing.

Basically the reactive (on-demand) routing has two phases to establish a route: the Route Request phase and the Route Reply phase. In route request phase, flooding is used to propagate a route request packet, and in route reply phase the min hop path selected. However the min hop path is not always the best path in ad hoc networks.

Sometimes, a detour through idle links can be the better choice.

This paper suggests an improved reactive routing scheme named LWR (Load aWare Routing) that chooses better routes by considering local information such as current queue size, and channel utilization.

Basic idea of LWR is to drop route request packets at heavily loaded intermediate nodes. This provides an efficient way of establishing an alternate detour when the shortest path is congested.

II. RELATED RESEARCH

In the past decade, much research on congestion control has been done. There are two major classes of congestion algorithms: TCP and Packet Scheduling scheme [1,2]. They are known to be effective and widely accepted in wired networks. There is an on-going effort to extend them to ad hoc networks [3,4,5]. However ad hoc networks have two major differences in respect to wired networks: node mobility and dynamic bandwidth fluctuation due to interference.

LWR is different from existing schemes in that it is a routing algorithm with congestion control. Intermediate nodes drop route request packets when they are overloaded. As the result, congestion control is a by-product of routing.

There are also load aware routing schemes [6,7]. Those works focus on the route reply phase to choose the proper path based on load awareness, while LWR focuses on the route request phase to prevent unnecessary flooding packet forwarding.

LWR has unique characteristic. It prevents route request packets from being propagated through heavily loaded nodes. This improves network performance because it reduces not only congestion but also interference caused by redundant control packets.

III. LOAD AWARE ROUTING (LWR)

As we mentioned, the key idea behind LWR is to reduce unfavorable routes traversing heavy loaded intermediate nodes. In figure 1, we show an example. The shortest path between source and destination goes through congested...
nodes by conventional reactive routing. By our scheme, an alternate detour path is established instead.

LWR has several advantages compared to conventional reactive routing. First, LWR performs call admission control – heavily loaded nodes do not participate in a new route discovery. Second, it does reduce redundant network traffic. Those two are the key mechanisms of LWR that sustain the network functionality in a highly stressful condition. As a consequence, data delivery ratio will be higher than for conventional reactive routing even in a saturated network. Best of all, this scheme decreases interference dramatically. It helps improving delivery ratio of both data packet and route control packet. It solves the saturation problem of reactive routing. LWR is self-adaptive – in a sparse and lightly loaded situation, the protocol utilizes the efficiency of conventional reactive routing, and in a dense and heavily loaded situation, it controls unfavorable route establishment.

The heuristic of LWR is the load level, which can be determined by local information such as channel utilization, queue size, number of active neighbors and value of backoff timer. We exploit all possible local information to estimate accurate load level. Besides queue size, all information is available for the MAC layer. Here we explain the measurement of channel utilization first and then describe the LWR scheme.

**A. Channel Utilization Measurement**

The channel status of ad hoc network can be defined as IDLE, TRANSMITTING, RECEIVING, and COLLISION. MAC layer monitors changes of status, and whenever a change happens, duration of previous channel status is accumulated to idle_time, trans_time, recv_time and coll_time respectively. Based on accumulated values (idle_time, trans_time, recv_time, coll_time) channel utilization Ui of node Pi is calculated periodically on every interval I (in sec).

Ui can be classified in 4 levels:
- FULL_IDLE: Ui is very low.
- HALF_IDLE: Ui is a little low.
- FULL_FULL: Ui is a little high.
- FULL_HEAVY: Channel is fully utilized.

We determine each threshold based on previous research on the performance of standard 802.11 [8,9]. We consider channel is fully busy in the following cases:
- The difference (\(\rho_{\text{max}} - \text{Ui}\)) ≤ \(\delta\). \(\rho_{\text{max}}\) is the maximum performance of 802.11 experienced in the scenario at hand. It generally depends on the collision probability and the number of neighbors. \(\delta\) is a small safety margin.
- \(P_{\text{coll}} > C_{\text{max}}\). The probability of collision (\(P_{\text{coll}}\)) is larger than COLLISION_THRESHOLD (\(C_{\text{max}}\)).
- Under high load situation, some nodes tend to occupy the shared channel longer than the fair (round robin) fraction. This scheme tries to keep channel utilization equal among all active neighbors.

The list of active neighbors is managed by timeout and is updated at each packet arrival.

Appendix I. shows the algorithm in detail.

**B. Load aWare Routing (LWR)**

Reactive routing uses two phases to find a route: Route request and Route reply phase. In this paper, we focus on the route request only, since the load aware method in route reply phase requires extra control data.

Whenever an intermediate node Pi receives a route request packet, it forwards or drops the packet based on TTL and current load level. It drops the packet only when TTL expires or the load level Li is higher than threshold FULL_HEAVY. A node drops route request flooding packets when the load level Li reaches the near maximum load level. The load level Li is associated with channel utilization Ui, current queue size Qi, number of active neighbors Ni, and the value of backoff timer Bi. The larger values are, the larger Li is.

Those values have following meaning.
- The channel utilization Ui: If Ui is near saturation point \(\rho_{\text{max}}\), then small increase in offered load would results in a large increase in data packet drop probability. Therefore, in such situation a new route request packet should be dropped.
- The queue size Qi: Queue size is also an indicator of load level. High Qi means that the queue is fully loaded.
- The number of neighbors Ni: [8,9] shows that the number of neighbors is critical in determining \(\rho_{\text{max}}\).
- The backoff timer Bi: Bi represents the number of failures of a data packet transmission. Hence Bi is clearly related to the probability of collision [16].

The intermediate node Pi drops route request packet if Ui is FULL_HEAVY or all 4 values are large enough. The
latter condition helps keeping stable performance and prevents saturation.
Appendix II. illustrates in more detail the collection of load level measurements.

IV. SIMULATION AND EXPERIMENTS
A. Simulation Model
We evaluate the LWR scheme by comparing the performance with AODV [12,13]. We use the Global Mobile Simulation (GloMoSim) library [10], which is a scalable simulation environment for wireless network. GloMoSim uses Parsec [14]. We implement LWR based on AODV and 802.11 DCF [15].

We use a CBR (Constant Bit Rate) application, AODV, 802.11 DCF MAC protocol and TWORAY radio model with 2Mbits/sec-channel bandwidth. TWORAY radio model is similar to the model used in the ns-2 wireless extension [11]. Under no interference, the range at maximum transmission power covers 250m. We place nodes randomly, and simulate 600 seconds for all cases.

First we evaluate LWR compared to AODV by using various mobility parameters. We use 100 nodes over 2000 x 600m² terrain and 40 random CBR connections. Nodes are moving following random waypoint model with speed 2m/s, 4m/s, 8m/s, and 16m/s without pause time.

Second we compare LWR and AODV over various offered loads. We use 100 nodes in 2000 x 600 m² and 200 nodes in 2500 x 900 m². We increase the number of CBR connections from 10% to 100% the number nodes. Here we use random waypoint model with minimum speed 2 m/s, maximum speed 20 m/s and 100 seconds pause time.

For both comparisons, we consider two rates -512 bytes/pkt and 4pkts/sec- at each CBR connection. All values averaged over several simulation experiments using different random seeds.

B. Simulation Results

B.1. Results based on change of mobility parameter

Figures 2 to 4 show that LWR generally works better than ordinary AODV since LWR reduces the control overhead. The performance gain by LWR increases as the mobility increases.

B.2. Results based on offered load (Num of nodes =100)

Throughout the experiments (Figure 5-7), we can easily see that LWR survives even under heavy traffic loads, while AODV suffers from huge control overhead. Figure 6 shows the average hop count. After saturation point
B.3. Results based on offered load (Num of nodes =200)

To show the scalability of LWR, we simulate large network. Figure 8 shows that LWR is fairly stable even in a dense and large network. The decline in performance by LWR as a function increasing offered load is gradual, while AODV performance drop dramatically.

A possible undesirable consequence of dropping route request packets is loss in connectivity. To gauge this effect, we measure the number of successfully connected pairs with 200 nodes. Recall that LWR drops flooding packet only when the load level is too high. Figure 9 shows the gain in connectivity achieved by LWR.

V. CONCLUSIONS

This paper has presented a novel load aware routing LWR. The paper has made several contributions. First, LWR guarantees stable performance even when the network is dense and heavily loaded. Second, LWR balances the load using detour paths and avoiding congested areas. Finally, LWR reduces interference caused by flooding packet enormously by dropping flooding packets.

Simulation results show that LWR works effectively.

<table>
<thead>
<tr>
<th>Table 1. Threshold Values for Channel Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
</tr>
<tr>
<td>ALPHA</td>
</tr>
<tr>
<td>IDLE_THRESHOLD</td>
</tr>
<tr>
<td>BUSY_THRESHOLD</td>
</tr>
<tr>
<td>TRANS_THRESHOLD</td>
</tr>
<tr>
<td>COLL_BUSY_THRESHOLD</td>
</tr>
</tbody>
</table>

APPENDIX

1. Measurement of channel utilization

Variables used in our experiment are summarized in Table 1. Values in Table 1 are decided based on [8,9]. ALPHA value is used to mitigate sudden effect. The performance of one transmission over 802.11 can be calculated easily. And $\rho$ is over 0.8 under our environment. If more than 2 neighbors share the channel, then we drop the packet after channel utilization reaches 0.7 and keep the fair share to every neighbor. Under collision, we drop maximum throughput threshold to 0.6 since collision generates more idle time due to backoff.

TABLE 2. Pseudo code to calculate fraction of each state

```python
Table 2. Pseudo code to calculate fraction of each state
```
MIN and MAX values. Based on MIN and MAX, level of each is determined as following.

1: \{MIN, (MAX-MIN)*1/4\}
2: \{(MAX-MIN)*1/4, (MAX-MIN)*2/4\}
3: \{(MAX-MIN)*2/4, (MAX-MIN)*3/4\}
4: \{(MAX-MIN)*3/4, MAX\}

The load level Li of node Pi is determined by using one linear function of four values (Bi, Ni, Qi and Ui).

\[ Li = \text{CHANNEL\_UTIL\_PORTION} \times U_i + \text{BACKOFF\_PORTION} \times B_i + \text{NEIGHBOR\_PORTION} + \text{QUEUE\_PORTION} \times Q_i/4; \]

- if Li \geq \text{FULL\_THRESHOLD}, Li = \text{FULL\_BUSY}
- if Li = \text{HALF\_FULL}, Li = \text{FAIRY\_BUSY}
- if Li \geq \text{HALF\_IDLE}, Li = \text{LITTLE\_BUSY}
- Li = \text{FULL\_IDLE}

REFERENCES