

REDUCING NEIGHBOUR DETECTION LATENCY IN OLSR

Yangcheng Huang

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

Saleem Bhatti

School of Computer Science
University of St Andrews
St Andrews, Fife KY16 9SX, UK

Søren-Aksel Sørensen

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

ABSTRACT

This paper presents a fast neighbour detection scheme for proactive a MANET routing protocol. Instead of using periodic HELLO messages, the proposed scheme adopts explicit handshake mechanism to reduce the latency in neighbour detection. In particular, two route handshake options are presented, namely the Broadcast based handshake (*BHS*) algorithm and Unicast based handshake (*UHS*) algorithm. Our simulation results show that the proposed scheme improves routing performance, especially in networks with moderate or high mobility. In low-density networks, the unicast option improves the routing throughput significantly without introducing extra control overhead.

I. INTRODUCTION

A mobile ad hoc network comprises a set of nodes connected in by wireless links in a temporary manner. The nodes are free to move, and may do so randomly and frequently, organising themselves arbitrarily. In addition, wireless networked nodes, such as sensors, may fail due to power exhaustion, or may need to be rebooted or re-programmed because of system failure and/or incorrect behaviour. New nodes may need to join the network when powered-on. Link connectivity may change rapidly and unpredictably, which leads to packet drops and performance degradation. This requires MANET protocols, especially routing protocols, to detect neighbour changes with low delay.

Traditionally, MANET routing protocols such as OLSR [1] and AODV [2] detect neighbour changes through exchanging periodic HELLO messages. Although the HELLO based neighbour detection is very simple in implementation and robust in presence of channel loss, there have been concerns about its performance in the dynamic environments like MANETs.

First, detection latency. The HELLO based mechanism has a relatively large delay in neighbour detection. For example, it takes around 3 seconds on average for OLSR nodes to detect established connections [1]. Such latency might lead to unnecessary packet drops due to route unavailability, especially in low-density networks with scarce network connectivity.

Second, resource waste. Periodic HELLO messages are broadcast even if no link changes occur, which wastes bandwidth and battery life. A smaller HELLO interval increases channel contention and might lead to congestion.

In this study, we propose a fast neighbour detection scheme. Instead of relying on periodic HELLO message, the proposed scheme uses explicit route handshake mechanisms in neighbour detection, which reduces the latency in connection es-

tablishment and improves path availability. In particular, we present two handshake options based on OLSR, namely Unicast based handshake (*UHS*) and Broadcast based handshake (*BHS*) options. We validate our scheme using simulation for a modified version of OLSR, showing that our proposed scheme improves routing performance.

The rest of the paper is organised as follows. Section 2 lists the existing neighbour detection mechanism. The proposed route handshake scheme is described in section 3, followed by an analytical study on its performance in Section 4. Section 5 introduces our simulation configurations. Section 6 presents our observations based on NS2 simulations. Conclusions and future work are summarised in Section 7.

II. EXISTING NEIGHBOUR DETECTION MECHANISMS

In this section, we briefly describe the existing neighbour detection approaches in MANET routing protocols.

A. HELLO Based Neighbour Detection

In the HELLO based neighbour detection approach, each node maintains a list of its neighbouring nodes, and broadcasts periodic HELLO messages to its neighbours to acknowledge its existence. Receiving a HELLO message from an unknown node indicates the arrival of a new neighbour, which further leads to link establishment. If, within a certain period of time, a neighbour state entry is not refreshed, the node time-outs the neighbour entry in its neighbour repository and assumes the connection to the neighbour is broken.

Each node sends HELLO messages periodically, so the protocol can tolerate some message loss without requiring reliable transmission. Such losses could occur frequently in radio networks, for example due to collisions or other transmission problems.

B. Mobility Adaptive Extension

Benzaid et al [3] propose an adaptive scheme to address the issues for high mobility nodes. In this scheme, the HELLO intervals are tuned in proportion to the rate of node mobility. That is, a node increases its HELLO refresh frequency when it moves fast, while decreasing refresh frequency when there is no mobility.

C. Link Layer Notification Extension

RFC 3626 [4] suggests that link layer notification could facilitate neighbour detection. If the link layer information is available for the routing layer, which describes connectivity to neighbouring nodes, this information can be used to maintain neighbour status, as a supplement to a HELLO based neighbour detection scheme. For example, Voorhaen and Blondia

[5] use data packet delivery failure events as link layer feedback to help detect link breakage. Such schemes introduce no extra control overhead, but require support from the MAC layer implementation.

III. PROPOSED FAST NEIGHBOUR DETECTION SCHEME

In this section, we present our neighbour detection scheme based on explicit neighbour handshake for routing.

A. Outline

MANET routing protocols like OLSR only use symmetric links in route calculation. The established (physical) connections would not be available for data transfer until identified as *symmetric* links by the routing protocols. Therefore, the delay in neighbour detection might lead to routing performance degradation.

The neighbour detection latency of HELLO based routing protocols is caused by the periodic nature of HELLO messages. After receiving the first HELLO message from a neighbouring node, the OLSR node does not respond until it broadcasts the next HELLO message. Essentially, the neighbour handshake process is done *implicitly* through exchanging periodic HELLO messages.

In our scheme, we use *explicit* handshake messages to facilitate connectivity detection. More specifically, in addition to periodic HELLO messages, the node sends explicit handshake messages to its neighbours. The basic process is described as follows.

1. Each node broadcasts periodic HELLO messages to its neighbours.
2. When node A receives its first HELLO message from its unknown neighbour B , it creates a new entry for directed link ($B \rightarrow A$), and responds with an ACK message *immediately* to node B , with the status of the new link ($B \rightarrow A$).
3. When node B receives such an ACK message, it infers the existence of bi-directional link ($B \leftrightarrow A$); then node B sends *immediately* an ACK message to node A , confirming the symmetric link status between them.
4. If, for any reason, the ACK message from A is lost or dropped, the following periodic HELLO messages would cover such a loss and finish the connection detection, as currently in a proactive neighbour detection scheme.
5. Similarly, if the ACK message from B to A is dropped, the following periodic HELLO message covers the loss and acknowledges A with the symmetric link status, as currently in a proactive neighbour detection scheme.

B. Handshake Options

In this section, we propose two handshake options based on above scheme, namely Unicast based handshake (*UHS*) and Broadcast based handshake (*BHS*).

Table 1: Notation Used in Analysis

r	HELLO interval
λ	Link change rate
δ_t	Time gap between successive messages
L_p	Detection latency of HELLO based method
l_p	Expected detection latency of HELLO based method
L_h	Detection latency of handshake based method
l_h	Expected detection latency of handshake based method

- **Unicast based handshake.** The handshake packets are transmitted as unicast messages between the neighbouring nodes. For example, when node A receives its first HELLO message from B , it only sends ACK messages to node B . Other neighbouring nodes of A would may receive the handshake packets but the packet is not addressed to them.
- **Broadcast based handshake.** The handshake packets are transmitted using broadcast messages. In order to reduce control overhead, the expiration time of the HELLO timer is delayed by r seconds after each handshake (a HELLO interval of r). For example, when node A receives its first HELLO message from B , it broadcasts ACK messages to each of its neighbours. Other nodes may also observe the packet.

IV. ANALYSIS

In this section, we present an analytical study on the performance of the proposed algorithm. In particular, we compare the link detection latency under HELLO based neighbour detection mechanism with that under the proposed fast neighbour detection scheme. In the following discussions, we assume that:

1. The arrival of a link establishment event is an independent, identically distributed Poisson process with arrival rate λ .
2. The delay in packet transmission and processing (i.e. t_p) is small enough (compared with link detection latency) to be ignored.

The assumptions are reasonable if the node degree is small and the nodes are moving randomly so that the process of link establishment or link breakage is totally random.

A. HELLO Based Neighbour Detection

Let L_p be the link detection latency of the proactive neighbour detection mechanism; let X_p be the time when the *first* symmetric link is established after t_0 ; let r be the HELLO interval. Then the link discovery latency can be approximated by:

$$L_p = t_0 + \delta_t - X_p + r = r + \delta_t - (X_p - t_0) \quad (1)$$

According to the assumptions:

$$X_p - t_0 \sim Exponential(\lambda) \quad (2)$$

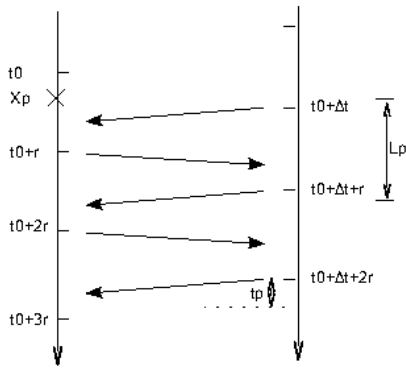


Figure 1: HELLO based Neighbour Detection Process

Therefore, the expected link discovery latency can be approximated by:

$$l_p = E[L_p] = r + \delta_t - E[X_p - t_0] = r + \delta_t - \frac{1}{\lambda} \quad (3)$$

For example, if the average new link arrival rate is one link per second (i.e. $\lambda = 1$) and the HELLO interval r is by default 2s, the expected link discovery latency is around 3s.

B. Proposed Fast Neighbour Detection

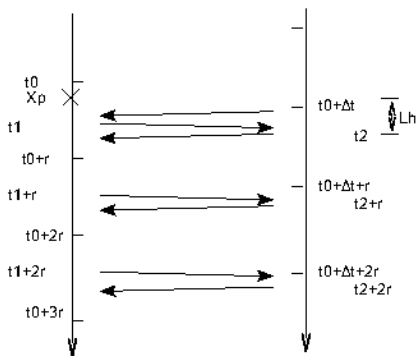


Figure 2: Fast Neighbour Detection Process

Let L_h be the link detection latency of the proposed handshake based neighbour detection mechanism; let X_h be the time when the *first* symmetric link is established after t_0 ; let r be the HELLO interval. Then, under our proposed scheme, the link discovery latency can be approximated by:

$$L_h = t_0 + \delta_t - X_h = \delta_t - (X_h - t_0) \quad (4)$$

According to the assumptions:

$$X_h - t_0 \sim \text{Exponential}(\lambda) \quad (5)$$

Therefore, the expected link discovery latency therefore can be approximated by:

$$l_h = E[L_h] = \delta_t - E[X_h - t_0] = \delta_t - \frac{1}{\lambda} \quad (6)$$

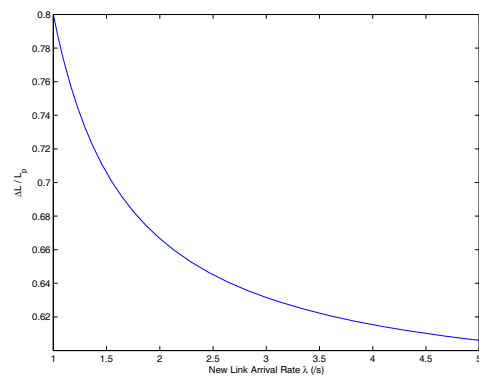
C. Factorial Analysis

From the above discussions we can see that the handshake scheme has a smaller link discovery latency. Smaller link discovery latency provides better route availability, which leads to better routing performance. We present now an analysis on the impacts of factors such as node density and node transmission range on the performance of the proposed handshake scheme.

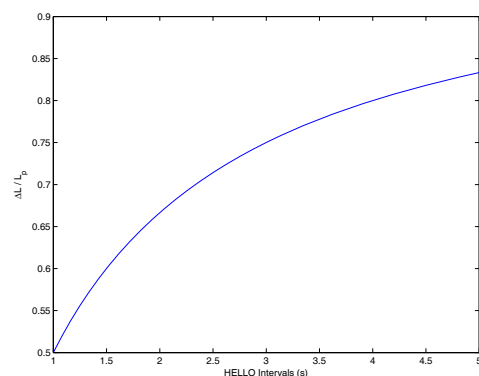
Consider the improvements of the proposed scheme on neighbour detection latency:

$$\frac{\Delta l}{l_p} = \frac{l_p - l_h}{l_p} = \frac{r}{r + \delta_t - \frac{1}{\lambda}} \quad (7)$$

Equation (7) presents a quantitative relationship between the improvements on latency and the variables of interest.


 Figure 3: $\frac{\Delta l}{l_p}$ vs. λ ($r=2$, $\delta_t=1.5$)

- **Rate of new link arrivals λ .** Studies on link dynamics [6] show that, the rate of new link arrivals λ increases with node velocity v , node density, ρ , and node transmission range R . We can see in Figure 3 that decreasing link arrival rate gives improvements for link detection latency. From this we can infer that, in the low-density networks with relatively smaller transmission radius, the proposed handshake scheme is expected to outperform the proactive neighbour detection scheme.


 Figure 4: $\frac{\Delta l}{l_p}$ vs. r ($\lambda=2$, $\delta_t=1.5$)

- **Refresh Intervals r .** From Equation (7) and Figure 4, we see that increasing the refresh interval, r , improves link detection latency. Therefore, the proposed handshake scheme is expected to have a better performance in a network with large refresh interval.

V. SIMULATIONS

A. Set-up

We implement the proposed options in the OLSR implementation which runs in version 2.9 of NS2 and uses the ad-hoc networking extensions provided by CMU, with a radio range of 250m radius and the use of MAC/802.11 as the media access control.

We use a network consisting of n nodes: $n = 20$ to simulate a low-density network, $n = 50$ to simulate a high-density network. The nodes are randomly placed in an area of 1000m by 1000m. All simulations run for 100s.

We use the Random Trip Mobility Model, “a generic mobility model that generalizes random waypoint and random walk to realistic scenarios” [7] and performs perfect initialisation. Unlike other random mobility models, Random Trip reaches a steady-state distribution without a long transient phase and there is no need to discard initial sets of observations.

The mean node speed, v , ranges between 1m/s to 30m/s. For example, when the mean node speed is 20m/s the individual node speeds are uniformly distributed between 0m/s and 40m/s. The average node pause time is set to 5s.

A random distributed CBR (Constant Bit Rate) traffic model is used which allows every node in the network to be a potential traffic source and destination. The CBR packet size is fixed at 512 bytes. There are at least $n/2$ data flows that cover almost every node.

For each sample point presented, 100 random mobility scenarios are generated. The simulation results are thereafter statistically presented with the mean of the metrics and the errors. This reduces the chances that the observations are dominated by a certain scenario which favours one protocol over another.

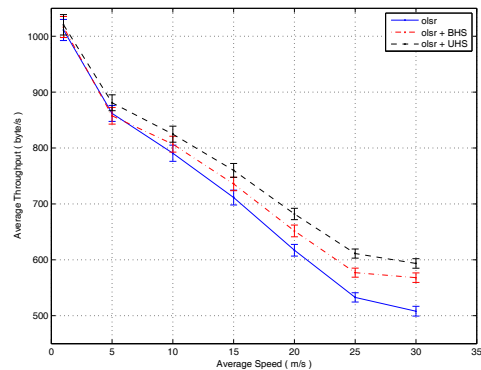
B. Metrics

In each simulation, we measure each CBR flow’s throughput and control traffic overhead and then calculate the mean performance of each metric as the result of the simulation.

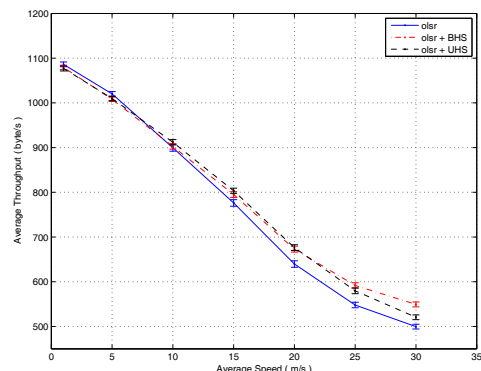
Throughput is considered as the most straightforward metric for the MANET routing protocols[8]. It is computed as the amount of data transferred (in bytes) divided by the simulated data transfer time (the time interval from sending the first CBR packet to receiving the last CBR packet).

The control overhead consists of HELLO messages and topology control (TC) messages. Considering the broadcast nature of the control message delivery, the packets are counted by summing the size of all the control packets *received* by each node during the whole simulation period.

Normalised routing overhead (NRO) is defined as the ratio $\frac{P_c}{P_d}$, of the number of control packets propagated by every node in the network, P_c , to the number of data packets received by the destination nodes, P_d .



(a) Low Density



(b) High Density

Figure 5: Throughputs ($h=2$ $t=5$ $R=250$)

VI. PERFORMANCE ANALYSIS

In this section, we present our observations on the routing performance under various factors, such as node velocity, node density and topology advertisement redundancy options.

A. Throughput

From Figure 5 we can see that, the proposed handshake options outperform the HELLO based option in high-mobility networks. For example, as shown in Figure 5(a), the unicast option leads to 18% increase in throughput, while the broadcast option has 12%. With no mobility or low mobility, the occurrence of link arrival is relatively rare; therefore the proposed handshake scheme has no significant impact on the routing performance.

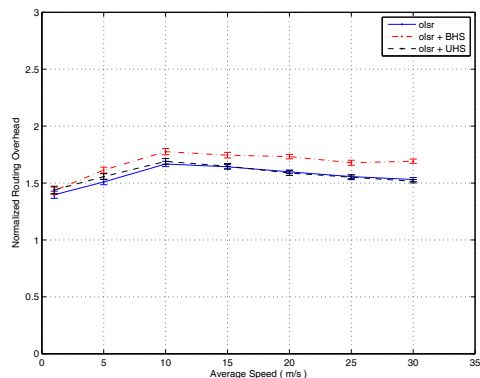
By comparing Figure 5(a) with Figure 5(b) we can see that in low-density networks, the proposed handshake options significantly improve routing throughput. In high density networks, however, the handshake options give little benefit. This matches our analytical results very well.

In addition, the UHS option outperforms the BHS option in low-density networks. Figure 5(b) shows that in high-density networks, the performance of the unicast option is not quite satisfactory. This is caused by the increased (address resolution protocol) ARP processing overhead and can be solved by improving the ARP processing capacity.

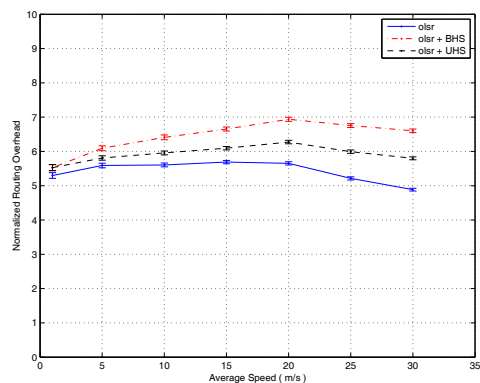
B. Control Overhead

From Figure 6(a) and Figure 6(b), the UHS option almost introduces no extra control overhead in low-density networks, while 18% extra costs in high-density networks. The BHS option introduces up to 36% extra control overhead.

Therefore, with the proposed handshake based neighbour detection algorithms, the performance improvement in data packet delivery is at the cost of extra control traffic overhead. Moreover, the control overhead introduced might be significant, especially in dense networks with high node mobility.



(a) Low Density



(b) High Density

Figure 6: Control Overhead ($h=2$ $t=5$ $rr=250$)

VII. CONCLUSIONS AND FUTURE WORK

This paper presents a fast neighbour detection scheme for MANET routing protocols. Instead of using periodic HELLO messages, the proposed scheme adopts an explicit handshake mechanism to reduce the latency in neighbour detection. The analytical study on neighbour detection latency shows that the proposed scheme reduces the link detection latency. Our simulation results support the analytical results, by showing that the proposed scheme improves routing performance (especially in networks with moderate or high mobility). In low-density networks, the UHS option improves the routing throughput significantly without introducing extra control overhead.

One drawback of the proposed algorithm is extra traffic over-

head introduced in dense networks with high node mobility. The increased control overhead *may* cause channel congestion and reduce network performance overall. Therefore, the proposed algorithm is more suitable for mobile networks with high bandwidth and a key requirement for timely packet delivery and throughput, rather than networks with very tight resource constraints.

In order to mitigate the side effects of the proposed algorithms, we have been working on an adaptive solution to adjust the delay in sending handshake messages dynamically based on network conditions such as bandwidth availability. This is to balance the trade-off between throughput and control traffic overhead.

In addition, the proposed algorithm is not limited to OLSR. In order to demonstrate its generic aspect, we are currently implementing and evaluating the algorithm in several other MANET routing protocols, including DSDV and AODV.

The original data, the source code and the scripts used in this study are all available from <http://www.cs.ucl.ac.uk/staff/y.huang/fh.tar.bz2>.

REFERENCES

- [1] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, a. Qayyum, and L. Viennot, "Optimized link state routing protocol," in *IEEE INMIC Pakistan*, 2001.
- [2] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (aodv) routing," IETF, Request for Comments 3561, July 2003.
- [3] M. Benzaid, P. Minet, and K. Agha, "Integrating fast mobility in the olsr routing protocol," in *Fourth IEEE Conference in Mobile and Wireless Communications Networks*, 2002.
- [4] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (olsr)," RFC 3626, October 2003.
- [5] M. Voorhaen and C. Blondia, "Analyzing the impact of neighbor sensing on the performance of the olsr protocol," in *Proc. 4th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt06)*, 2006.
- [6] P. Samar and S. B. Wicker, "Link dynamics and protocol design in a multihop mobile environment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1156–1172, 2006.
- [7] J. Y. L. Boudec and M. Vojnovic, "Perfect simulation and station-arity of a class of mobility models," in *Proc. IEEE Infocom Conference*, Miami, USA, 2005.
- [8] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," IETF, Request for Comments 2501, January 1999.