A Multipath Approach for Fast Re-routing in OLSR

Ashiq KHAN　Toshihiro SUZUKI　Motonari KOBAYASHI　and　Masanori MORITA

Network Laboratories, NTT DoCoMo, Inc.　3-5 Hikarino-oka, Yokosuka, Kanagawa, 239-8536 Japan
E-mail: {ashiq-khan, toshi, kobayashi, morita}@netlab.nttdocomo.co.jp

Abstract Mobile Ad-hoc Networks (MANET) are self-organized networks where independent mobile nodes (MN) organize themselves in an autonomous manner to form a communication network. Maintaining connectivity is the main challenge as links, due to mobility and interference, break quite frequently. After the link break, end-to-end routes have to be re-constructed. As the MANET routing protocols are mainly single path protocols, the route re-construction process takes a long time and the success in finding a route heavily depends on the protocols' performance and the network topology. In this paper, we investigate a multipath construction scheme on the proactive Optimized Link State Routing Protocols (OLSR) in order to reduce the route reconstruction latency. We make end-to-end link-disjoint multipath in an on-demand basis. Once the main OLSR path breaks, session is instantly switched to the backup path. This reduces the route re-construction latency to a great extent. Besides, constructing the multipaths only after a demand from particular applications and around the base OLSR single path greatly reduce the control overhead and frees up network resources.

Keyword MANET, OLSR, Multipath, Route Change Latency

OLSRにおけるマルチパス方式によるルート再構築の高速化の検討

カーン アシック　鈴木 俊博　小林 基成　森田 正範

（株）NTT ドコモ ネットワーク研究所 〒239-8536 神奈川県横須賀市光の丘 3-5
E-mail: {ashiq-khan, toshi, kobayashi, morita}@netlab.nttdocomo.co.jp

あらまし モバイルアドホックネットワーク(MANET)は，既存ネットワークの拡大に有力な候補である．MANETでは，接続性が主な課題であり，ルーティングプロトコルについて数多くの研究がされてきた．しかし，既存ルーティングプロトコルは主にシングルパス方式に基づいたものであり，ルート切断後のルート再構築に時間がかかる．ルート再構築遅延は，特に通信中にルートが切断される場合に，大量データ損失の原因になる．本論文は，プロアクティブ型ルーティングプロトコルのOLSRに基づき，マルチパス方式により予備パスを用意し、ルート切断後瞬時にルート切り替え方式を提案しその優位性を示す．

キーワード MANET，OLSR，マルチパス，ルート再構築

1. Introduction

Mobile Ad-hoc Network (MANET) is a potential technology to increase the capacity and coverage of any conventional network. Due to advancement in mobile wireless technologies like cellular and WLANs, mobile networking is becoming a standard. A wide range of services are being provided in mobile wireless environment. However, mobile wireless infrastructure is also expensive to build and difficult to plan. Coverage is still a problem and in many of the congested areas, wireless resources allocated to the operators proved insufficient to cover all the users during the peak hours. MANET can play an important role in the places where access points (AP) or base stations (BS) are either absent or unavailable for services. By employing its self-organizing and multi-hop characteristics, communication within a locality or connecting to a far away AP/BS can be realized in a cost effective manner.

Topology management and routing are the main challenges towards the realization of fully autonomous carrier-grade MANET. Mobility and usage of unlicensed spectrum which is prone to interference and noises change MANET topology frequently. Routing protocols have to be fast and robust enough to cope with the frequent topology changes in MANET.

Numerous researches have been carried out on routing in MANET. As a result, we have mainly two kinds of routing protocols for data transmission in MANET, namely, Reactive or On-demand Routing Protocols and Proactive Routing Protocols. AODV [1], DSR [2] are on-demand protocols. OLSR [3], TBRPF [4] are proactive protocols. On-demand routing protocols do not hold any routing table and search for routes only when it is necessary, e.g., when a node has some data to send. On the other hand, proactive routing protocols periodically exchange routing and topology information and keep an up-to-date routing table in each node.

The above mentioned protocols, both on-demand and proactive, are single path routing protocols. Basically, the on-demand ones, when searching for a route, search for one route to the destinations. The proactive ones keep a single route record for every reachable destination in the network. This single path approach inherently slow in response to frequent topology and the resulting route changes in MANET. A new route search has to be performed after a route break is detected which usually takes long time. During this re-routing process, high packet loss is observed which cannot be ignored when considering real time and streaming type applications.

In this paper, we consider multipath approach to cope with the long re-routing latency problem. We investigate the potential of multipath techniques on the Optimized Link State Routing Protocol (OLSR) [3]. Our main objective is path redundancy where primary routes and backup routes are maintained in the routing table and instant route switching is implemented as soon as the primary path fails. Besides, we also consider the control signal overhead in such proactive protocols and propose our multipath scheme as an on-demand extension on the basic proactive protocol.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of the related works. Section 3 describes the basics of OLSR. Section 4 discusses the Route Change Latency (RCL) in MANET. Section 5 introduces our multipath scheme. Section 6 gives an evaluation of our scheme and Section 7 concludes this paper.

2. Related Works

Multipath approaches have been investigated intensively in the context of On-demand Routing Protocols. Examples of such techniques are AOMDV [5], AODV-BR [6] and MP-DSR [7]. However, Multipath approach on Proactive Routing Protocols is a little addressed field. This is due to the fact that Proactive Protocols, presumably hold the current and complete or near-complete topology of the network. Hence, it is possible to find out an alternative path quite easily once a route break is detected. We argue that, the validity of Proactive Protocols information heavily depends on the pre-defined control message exchange interval and the completeness on the scale of the network. Longer routing message exchange interval makes it quite reactive in detecting topology changes. And the larger the networks get, the incomplete the topology information becomes. Hence, it is not an easy task to build and maintain end-to-end multipaths on the Proactive Routing Protocols.

[8] proposes a multipath technique on OLSR to support QoS in MANET. When the OLSR single path cannot provide the bandwidth requirement of an application, multipath is built to distribute loads over multiple paths. The proposal in [8] considers interference among the paths and makes paths with least interference among them. However, [8] does not explain the complete and detailed mechanism on how multiple paths are built and maintained.

[9] proposes a Multipath_OLSR to avoid congestion in MANET. They build multipath based on the multiple topology entries. In order to avoid common nodes among the paths, [9] proposes source routing. At the same time, the whole route information is maintained in the routing table. While sending packets, the whole route is copied in the IP header and packets are source routed. This violates the distributed routing principle of OLSR. Besides, source routing increases the IP header size to multiple times and consumes network bandwidth. They occupy more spaces in the queue and prone to create congestion themselves.

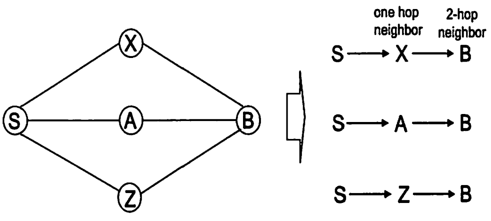
3. The Optimize Link State Routing Protocol (OLSR)

Optimized Link State Routing Protocol (OLSR) [3] is a proactive routing protocol designed for intra-MANET communication. It reduces the control signal overhead observed in conventional link state protocols by limiting flooding to specially selected nodes. We explain the relevant of OLSR here.

Control Messages: OLSR uses two kinds of control messages- HELLO and Topology Control (TC) message. HELLO is used for link sensing to the neighbors. An OLSR node broadcasts HELLO messages every two second by default. The HELLO message includes the ID of the transmitting node and the neighbor set of the node.

2-hop Neighbor Set: OLSR holds a 2-hop neighbor set among its information repositories. It includes all the

2-hop neighbors of a node and the intermediate one hop neighbors to each of them.



Network topology till 2-hops

2-hop neighbor set at S

Fig 1: 2-hop Neighbor set

Fig 1 shows the graphic and the data structure of OLSR 2-hop Neighbor Set. Node S includes X, A and Z as the one-hop neighbors through which 2-hop neighbor B could be reached.

MPR Selection: A node running OLSR selects the minimum number of one-hop neighbors through which all its 2-hop neighbors could be reached. In Fig 2, the black nodes are enough to cover all the 2-hop neighbors of node S. The set of these one-hop neighbors are called Multi-Point Relay (MPR) set. A node selecting another node as its MPR is called the MPR Selector (MS) of the MPR.

In OLSR, only MPRs generate and forward Topology Control (TC) message. A TC message includes the address of the originator and links to its MS at minimum. This restricted generation and forwarding of routing control messages greatly reduces the flooding overhead found in other conventional link state routing protocols.

Routing Table Calculation: A node running OLSR calculates its routing table in a recursive way based on its one-hop neighbor set and the topology set. OLSR keeps one record for each reachable destination in the network. Partial of broken routes are not recorded. OLSR performs hop by hop routing. Therefore, only the next hop and the hop number are recorded to reach a destination.

At first, all the one-hop neighbors are inserted in the routing table. In this case, the next hop to these one-hop neighbors are the one-hop neighbors themselves. After that, entries in the topology set which originator addresses equal the destination entry in the routing table are inserted where the destination is the advertised *MS* address. The next hops of these new destinations are the same as the next hop of the routing entries which destinations equaled the originator address of the topology entry. The hop number is incremented by one.

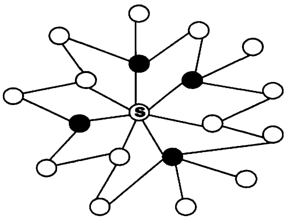


Fig 2: Selection of MPR

Routing Table Update: In OLSR, the routing table is recalculated every time there is a change in any of the Neighbor set, 2-hop Neighbor set or the Topology set.

Once a link to a MPR is broken, after selecting a new MPR, a node may instantly send a HELLO to inform the new MPR. Besides, instant TC may also be diffused in the network in order to inform the network about a link break or addition in the network.

4. Route Change Latency (RCL)

Route Change Latency (RCL) in mobile wireless environment can be divided into two main phase as shown below

$$RCL = d_{link\_break\_detection} + d_{new\_route} \quad ---(1)$$

$d_{link\_break\_detection}$ is the time necessary to detect a link break. In OLSR [3], the standard is 3 X HELLO transmission interval that is 6 seconds. As an option, the corresponding MPR, detecting a link break can broadcast a TC message about this link break. Although it takes few seconds to detect a link break from the Network Layer, it is possible to reduce $d_{link\_break\_detection}$ greatly by employing Link or Physical Layer triggering.

After the link break detection, the corresponding node selects a new MPR and informs it by a HELLO message. The node can instantly generate such HELLO message instead of waiting for the periodic one. On receiving this HELLO message, the new MPR can send an instant TC message informing the topology around it [10].

Besides, as soon as a link break is detected at a node, the node recalculates its routing table. Please note that, the corresponding node, may find a new route by using the information in the topology set it already has in this process of routing table recalculation. But we consider the worst case where a route to the destination of the on-going session may not be available from the topology information. The node has to wait until a route could be built by the TCs received later. Therefore, d_{new_route}

becomes to,

$$d_{new\_route} = d_{mpr\_selection} + d_{TC\_arrival} ----(2)$$

The value of (2) ranges from few milliseconds to few seconds depending on the topology information.

5. Our Scheme: Multipath on OLSR

In this section, we present a simple multipath construction method on OLSR. We intend not to change any of the OLSR specifications and external data structure. The multipath constructed here is partially link-disjoint multipath, shortest and having an equal length and share only one link in the worst case.

5.1 Node-disjoint Multipath within 2 hops

As described in Section 3, OLSR maintains an up-to-date 2-hop neighbor table. The 2-hop neighbor table holds the information of connectivity within the 2-hop neighborhood of a node (Fig 1). When there are multiple one-hop neighbors covering the same 2-hop neighbor, it is possible to construct node-disjoint multipath till the 2-hop neighbor.

Explaining it using Fig 4, the dotted line which represents symmetric links but not reflected in the routing table will be utilized to build multipath till the 2-hop neighbors.

5.2 End-to-end link-disjoint Multipath

In order to build end-to-end multipaths, we connect the 2-hop node-disjoint multipaths described above from the source to the destination. This will result into end-to-end link-disjoint multipaths where the nodes at every 2-hop distance are shared by the paths. We explain the details of our scheme in the following subsections.

5.3 On-demand Multipath Construction

In order to restrict control signal overhead, we construct our multipath on an on-demand basis. Constructing and maintaining disjoint multipaths will increase the control overhead already serious enough in case of proactive protocols. We, by principle, construct multipath only when it is necessary, for real time applications, where shorter route re-construction latency is desired. Otherwise, OLSR behaves in its original single path mode.

src	dst	next hop	msg_type	num_mpath	Intermediate node addr

Fig 3: *Multipath_REQ/REP* message

When an application requires multipath to a particular destination, the routing module, based on the already existing OLSR routing table, unicasts a *Multipath_REQ* message to the destination. As shown in Fig 3, the *Mutlipath_REQ* message includes source (src), destination (dst), message type (msg_type) and number of multipath (num_mpath) to be constructed in its header.

Each node on the OLSR path between the source and the destination, appends its own address at the tail of the message and forwards it to the next hop. Forwarding of this message is performed by OLSR at each node.

When the *Multipath_REQ* message arrives at the destination, it has all the intermediate nodes between source and destination in the upstream order. The destination changes msg_type to *Multipath_REP*, reverses the src and dst and unicasts the *Multipath_REP* back to the source.

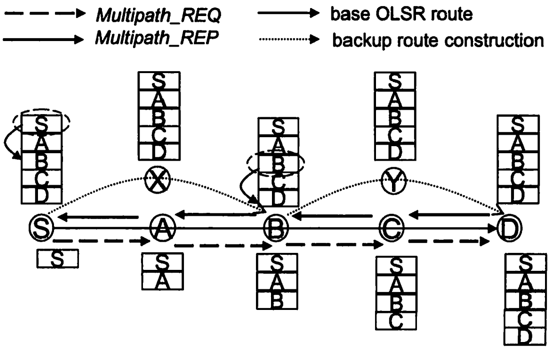


Fig 4: Constructing end-to-end link-disjoint multipath

The *Multipath_REP* message will traverse the same path as the *Multipath_REQ* message. Each node now, on receiving the *Multipath_REQ* message, checks its position on the route by looking at the addresses at the tail of the message. Nodes at a position of 2N (N is a natural number) hop distance from the destination (or source) are responsible to make a node-disjoint multipath to its 2-hop neighbor towards the destination (or source). As described in 5.1, the 2-hop neighbor table is used for this purpose.

As shown in Fig 4, node A, B and C are the intermediate nodes on the OLSR route between source S and destination D. After receiving the *Multipath_REQ* message, A, B and C append their addresses at the end of their received message and forward it to the next node en-route to the destination D. Destination D, upon receiving the message, change the message type to *Multipath_REP*, reverses the src and dst and unicasts it towards S.

C receives the message and checks its position on the route by checking the address order and finds itself only one-hop away from the destination D. Without doing anything, it forwards the *Multipath_REP* to B. B finds its position as 2-hop away from D. B checks its 2-hop neighbor table and sees if any one-hop neighbors are available other than C. In this case, B finds Y and includes it as the secondary next hop entry for the destination D in its routing table.

A forwards packet without doing anything and S finds itself at a distance of 4 hops (2X2) from the destination D. S also finds out B as its next 2 hop neighbor to the destination D. Following this, S consults its 2-hop neighbor table and finds X beside A which also covers B. The same as it was the case for B, S includes X as its secondary next hop for the destination D in its routing table.

As mentioned before, the src indicates its desired number of multipaths in the *Multipath_REQ* message. If any node fails to meet the number, it replaces the num_mpath value with the number of multipath it can make. If it cannot make any multipath, the num_mpath value in the *Multipath_REQ* is replaced by 1 which indicates the base OLSR single path. The source, upon receiving the *Multipath_REP*, checks the value of num_mpath field. It's completely upon the source to decide whether to adapt the present path with a different multipath value against what it had desired.

5.4 Route Change Policy

Our primary objective is to reduce Route Change latency. Therefore, as soon as a link break is detected, route is instantly switched to the secondary route. To make the link break detection phase faster, lower layer detection schemes could also be employed in addition to our scheme.

When the primary route fails and the secondary route is employed, the status is changed to primary. OLSR itself starts constructing a new route as its route has broken. This new OLSR route is inserted in the routing table as a secondary route if it is not identical to the route in use. If OLSR finds the same route that is in use now, the *Multipath_REQ/REP* message is used to construct multipath around the present route as explained in section 5.3. It is also possible to periodically transmit the *Mutipath_REQ* over a desired route to maintain the integrity throughout the lifetime of the ongoing application session. After the application session is over, all multipath information is removed from the routing table and the nodes follow the normal OLSR single path principle.

6. Evaluation

In this section, we present a qualitative evaluation in order to estimate the reduction in Route Change Latency (RCL), overhead in constructing and maintaining multipath and other features of our proposed scheme.

6.1 Reduction in Route Change Latency (RCL)

The outline of RCL has been presented by relation (1) and (2) in details in section 4. The link break detection $d_{link\_break\_detection}$ phase is outside the scope of this paper. Our objective is to reduce the time necessary to find a new route, d_{new_route}.

d_{new_route} can range from a few milliseconds in the best case to a few seconds in the worst case. As our scheme keeps a backup route throughout the lifetime of a session, instant re-routing becomes possible. As soon as a link break is detected, the backup route is employed to carry data from then onward. By such instant route switching, it becomes possible to reduce d_{new_route} to as low as zero.

While using such multipath techniques, the RCL completely depends on the length of the link break detection scheme. A fast link break detection [11], used in conjunction with our multipath scheme can reduce the total RCL to as low as a few milliseconds.

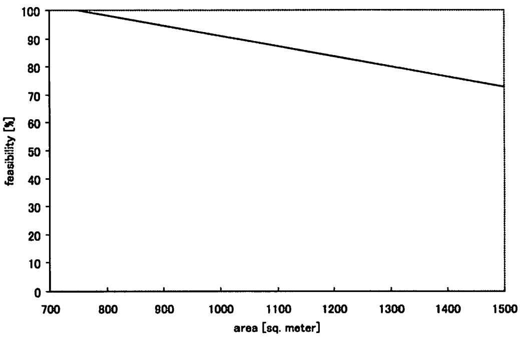


Fig 5: Percentage of the source-destination pairs covered in our scheme

6.2 Control Overhead

Unlike other multipath schemes on OLSR, our multipath is an on-demand extension. The multipath scheme explained in Section 5 is used only when such kind of extensive service is necessary for a particular application. Such applications could be VoIP, streaming video etc. as such applications are not tolerant to long RCL values.

By constructing multipath after the demand from a few applications, the control overhead is restricted to a great deal as the best effort (BE) applications are supported by the OLSR single paths. Thus, the number of Multipath REQ/REP messages are proportional to the number of applications and number of route breaks.

Besides, we employ the route determined by base OLSR as our primary route and construct the multipath around it. The *Multipath_REQ/REP* messages are unicast messages that travel only on the OLSR single path between the source–destination pair. In this way, we reduce the generation of multipath control messages to the extreme locality and the rest of the network is kept unaware.

6.3 Local Route Reconstruction

In our scheme, every 2 hop nodes are shared among the paths. This in turn means that, paths within two 2-hop neighbors are independent on the end-to-end paths. Therefore, when a link is broken, the nearest common node switches the route to its backup path and the rest of the route remains intact.

6.4 Feasibility

Feasibility is an important factor while constructing link-disjoint multipaths. By feasibility, we mean the probability of being able to make such paths in an end-to-end manner. In other words, in an arbitrary topology, how many of the source-destination pairs could be supported by our proposed scheme comparing to the total number of multipaths possible in the network.

Our investigation shows that the feasibility depends on the node density. When the node density is higher which increases the number of one-hop neighbors to reach any 2-hop neighbor, the feasibility performance of our scheme converges to the total number of multipath. When 25 nodes are randomly placed on a 750 square meter plane and using 802.11 radio specifications, we found that it is possible to support any source-destination pair by our multipath scheme. The result of our investigation is shown in Fig 5.

7. Summary and Future Works

In this paper, we have proposed a novel link-disjoint multipath scheme for robust routing in MANET. Our scheme is based on proactive routing protocols OLSR. The proposed scheme is used to reduce Route Change Latency (RCL) and have the potential to reduce RCL to as low as zero.

We propose our scheme in an on-demand manner where the multipaths are constructed and used only when an application requires such services. Besides, by constructing the multipaths around the OLSR single path, we restrict the multipath control along the OLSR single path. The on-demand multipath construction based on the already available OLSR single path significantly reduces the control overhead.

Our future works include detailed simulation of our proposal and comparison with the existing multipath schemes.

References

[1] C. E. Perkins, E. B. Royer, S. Das: Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561, July 2003.

[2] D. B. Johnson, D. A. Maltz, Y. C. Hu: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, draft-ietf-manet-dsr-10.txt, July 2004.

[3] T. Clausen, P. Jacquet: Optimized Link State Routing Protocol (OLSR), RFC 3626, October 2003.

[4] R. Ogier, F. Templin, M. Lewis: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC 3684, February 2004.

[5] M. Marina, S. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks," Proc. of the International Conference for Network Protocols (ICNP '01), Riverside, Novermber, 2001.

[6] S. Lee, M. Geral, "AODV-BR: Backup routing in ad hoc networks," Proc. of the IEEE WCNC 2000, pp. 1311-1316, Chicago, September, 2000.

[7] R. Leung, J. Liu, E. Poon, A. Chan and B. Li, "MP-DSR: A QoS-Aware Multi-Path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks," Proc. of the 26th Annual Conference on Local Computer Networks (LCN '01), pp. 139-172, 2001.

[8] H. Badis and K. A. Agha, "QOLSR Multi-path Routing for Mobile Ad Hoc Networks Based on Multiple Metrics: Bandwidth and Delay," Proc. of the 59th Vecular Technology Conference (VTC '04), IEEE, Vol. 4, pp. 2181 – 2184, 2004.

[9] M. Kun, Y. Jingdong, R. Zhi, "The Research and Simulation of Multipath_OLSR for Mobile Ad hoc Networks," Proc. of the IEEE Intl. Symposium on Communications and Information Technology (ISCIT), Vol. 1, pp. 540-543, October, 2005.

[10] C. Gomez, D. Garcia, J. Paradells, "Improving Perof a Real Ad-hoc Networks by Tuning OLSR Parameters," Proceedings of the IEEE Symposium on Computers and Communications (ISCC '05), pp. 16-21, June, 2005.

[11] M. Tauchi, T. Ideguchi, T. Okuda, "Ad-hoc Routing Protocol Avoiding Route Breaks on AODV," Proc. of the 38th Hawaii International Conference on System Sciences (HICSS '05), January, 2005.