Crossover Node Discovery for IEEE 802.11s Wireless Mesh Networks

Li-Hsing Yen*, Jiun-Jang Su†, Kuei-Li Huang†‡, Chien-Chao Tseng†§ and Kuan-Ming Liao†
*Dept. of Computer Science & Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan 811, R.O.C.
†Dept. of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan 300, R.O.C.
‡Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan
§Responsible for all communications. E-mail: cctseng@cs.nctu.edu.tw

***Abstract*—Crossover nodes have been utilized to achieve smooth handoffs for micro mobility management schemes. IEEE 802.11s supports proxy-based frame delivery services for mobile hosts roaming within a wireless mesh network (WMN). Discovering crossover node for mobile hosts in 802.11s WMNs, however, faces several challenges due to intrinsic properties of mesh networks. This paper identifies these challenges and proposes a scheme suitable for crossover node discovery under 802.11s WMNs. This scheme is characterized by source-oriented, MAP-centric, and per-source. It can be performed off-line with the derived results cached for on-line retrievals during handoffs. Simulation results show that the proposed scheme can reduce packet losses due to handoffs, and confirm the necessity of the source-oriented principle.**

I. Introduction

Wireless mesh networks (WMNs) [1], [2] have been conceived as an extension of Wireless Local Area Networks (LANs) that provide broadband data access services to mobile users. In such an infrastructure, a mobile node (MN) should make an association with a base station or access point (AP) that is within its radio communications range. Afterward the MN is allowed to send packets to the infrastructure via the associated AP. When an MN roams across APs, it should change the associated AP to continue its ongoing sessions. This process is known as a *handoff* [3].

A crucial step for some fast or seamless handoff schemes is to find *crossover nodes* (CrNs) for an imminent or underway handoff, a task termed crossover node discovery (CND). A crossover node is a fork point of two paths that both start from a correspondent node (CN) which sustains an ongoing session with the MN undergoing a handoff. These two paths ends at the old AP (oAP) and the new AP (nAP) associated with the MN, respectively. Crossover nodes can be utilized to achieve *smooth handoffs* (i.e., handoffs with unnoticed interruption) in the context of micro mobility management schemes such as Cellular IP [4], Hawaii [5], and MIP Regional Registration [6].

CrNs have the potential for facilitating handoffs of MNs within an IEEE 802.11s WMN [7]. In this environment, CrNs are essentially mesh points, and MNs roam across mesh access points (MAPs). Frame delivery in an 802.11s WMN is proxy-based, where an MAP serves as a routing proxy for all MNs associated with it, and all frame delivery paths among MAPs form a mesh structure. The proxy-based frame delivery supporting MN-to-MN traffic does not require each MAP to keep MN-to-MAP proxy associations of *all* MNs. This proxy-based design saves considerable storage cost in MAP and increases routing efficiency of WMN by replacing MN-to-MN routes with MAP-to-MAP routes and *partial* MN-MAP associations.

We found that existing CND approaches [4], [5], [6] rely on two basic properties of micro mobility schemes. First, each candidate CrN has location information for the handoff MN. Second, all routing paths among routers in a micro mobility domain form a tree hierarchy. These two properties are of micro mobility schemes and do not hold in WMNs, making a direct adaptation of existing CND approaches to IEEE 802.11s WMNs impossible. In this paper, we first identify potential problems that may arise in such an adaption attempt. We then suggest principles and propose a feasible approach to CND under 802.11s WMNs. This approach is characterized by source-oriented, MAP-centric, and per-source. It can be performed off-line with the derived results cached for retrievals during handoffs. Simulation results show that the proposed scheme can reduce packet losses due to handoffs.

The rest of this paper is organized as follows. The next section describes applications and principles of CND and briefs the frame-forwarding mechanism used in WMNs. We then point out potential problems that may arise during the application of existing CND schemes to WMNs and propose our approach (Section III). Section IV presents our simulation results and the last section concludes this paper.

II. Background

A. Crossover Node Discovery

There are several ways by which CrNs can be utilized to improve layer-3 handoff performance in a micro-mobility domain. After an MN changes its point of attachment to the Internet from an old access router (oAR) to a new access router (nAR), packets that are sent by a CN and destined for the MN are still routed toward the MN until the CN or any intermediate router is notified of the MN's new point of attachment. These packets are considered lost if no special treatment is made. As a fork point between the old and new delivery paths from the CN to the MN, the CrN could intercept all these in-transit packets and redirect them to nAR [8] before the CN is finally notified of the MN's new location. This *packet-redirection* could reduce packet losses due to handoff. Another technique named *bi-casting* [4], [5] can also minimize possible pause of data stream during handoffs. This technique designates a router to intercept any data stream sent from a CN toward an MN during handoff and duplicate it, sending one copy to the oAR and the other to the nAR. Crossover node is a natural choice to perform bi-casting from the perspective of the CN.

Depending on which entity initiates the procedure, existing CND procedures can be classified into *MN-oriented* and *source-oriented*. The former type starts a search for CrN from MN while the latter does so from a source node (the term source node refers to a CN.) In MN-oriented CND (MO-CND) (e.g., [4], [5], [9], [6]), the first router that finds itself on the fork point where the path from the MN (through the nAR) to oAR and that from the MN (through the nAR) to CN diverge identifies itself a CrN. In case of source-oriented CND (SO-CND), a router on the old delivery path identifies itself a CrN if, based on its routing table, the next-hop node to the nAR of the roaming MN differs from that to the oAR of the MN.

Regardless of specific protocol designs, there are common elements associated with conventional CND approaches. First, as a CrN performs frame-redirection or bi-casting specifically for a roaming MN, each candidate CrN must have location information specific to the MN. The location information in the form of *binding cache* maps an MN to its current serving AR. Such information is also needed for a router to identify itself a CrN in both MO-CND and SO-CND. Furthermore, MO-CND implicitly assumes that routers within a micro-mobility domain form a tree structure such that all delivery paths between any two routers are uniquely defined. If this assumption does not hold, the CrN reported by MO-CND can be different from that identified by SO-CND.

In short, the validity of existing CND approaches relies on two properties of micro mobility schemes. First, each candidate CrN has location information for the MN performing handoff. Second, all routing paths among routers in a micro mobility domain form a tree structure.

B. Frame Delivery in IEEE 802.11s WMNs

In IEEE 802.11s WMNs, MAPs are primary units that perform frame-forwarding tasks. Each MAP maintains two tables: a forwarding table and a proxy table. The forwarding table maintains MAP-to-MAP (rather than MN-to-MN) routing information. MAPs invoke Hybrid Wireless Mesh Protocol (HWMP) to create or update forwarding table entries. HWMP by default uses an on-demand route discovery protocol called Radio Metric AODV (RM-AODV). It also has the option to proactively maintain a tree-based route for every MAP in a WMN. The root of the tree is usually a mesh portal point (MPP), a gateway MAP that interconnects the WMN and an outer network. The proxy table records all association relations between MAPs and MNs known by the MAP. This includes the associations made by all MNs with the MAP itself. If the MAP ever receives or forwards a frame for an MN associating with another MAP, the MAP may have an entry in its proxy table that memorizes the MN's association. In other words, MAPs may cache MN-MAP proxy relations for peers which they recently communicate with. However, there is no guarantee that the proxy MAP of an MN is known by any other MAP in the WMN.

Delivery of frames between two MNs in a WMN is based on a two-tier architecture. When a source MN (sMN) has some frame to send to a destination MN (dMN), sMN first sends the frame to its *proxy MAP*, i.e., the MAP it associates with. Denote this MAP by source MAP or sMAP. When sMAP receives the frame, it consults its proxy table for the proxy MAP of the destination MN (dMAP). If sMAP can find dMAP in its proxy table, it then looks up its forwarding table for the delivery of the frame to dMAP. Otherwise, sMAP may either initiate a route discovery procedure to find a route to dMN or forward the frame to the MPP by tree-based routing. As MAPs are allowed to independently discover their routes to other MAPs, all frame forwarding paths in a WMN generally form a mesh rather than a tree topology.

In short, frame-delivery service in IEEE 802.11s WMNs is characterized by three points. First, frame delivery is proxy-based and MN-to-MN traffic is supported by a two-layer hierarchy. Second, an MAP may not keep location information for a particular MN. Third, routing paths among MAPs generally form a mesh structure.

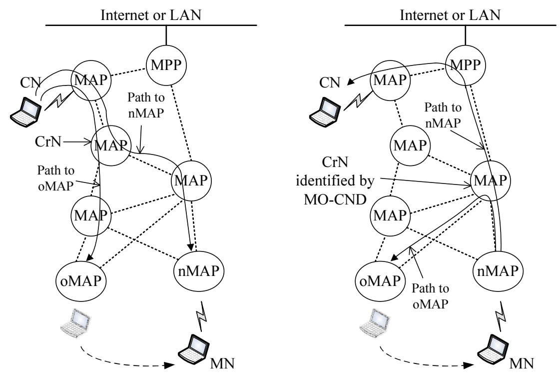
III. Problem and Solution

A. Problems of CrN Discovery in IEEE 802.11s WMNs

Existing CND schemes are based on layer-3 infrastructure. Our aim is to develop a CND scheme for WMNs so as to reduce frame losses due to the switching of wireless link between MAPs. To this end, we first disclose the problems that may occur when adapting existing CND schemes to WMNs. We assume that a WMN forms a single layer-2 mobility domain. However, no particular mobility management scheme is presumed to maintain up-to-date frame-forwarding and proxy information in the WMN. Hereafter we use oMAP and nMAP to denote the MAPs before and after an MN's handoff, respectively.

Both MO-CND and SO-CND rely on adequate routing information to operate. However, an MAP may not cache MN-MAP association information for all MNs involved in CND detection. Consequently, the MAP may not be able to identify itself a CrN for some MN due to the lack of the MN's association information. MAPs may initiate a route discovery procedure as a remedy, or an independent mobility management scheme may assist candidate CrNs in obtaining such information. Either way incurs costly overhead and extra delay. Extra delay in particular may make CrN too late to be useful.

Even that all MAPs have adequate routing information, non-tree frame-forwarding topology in WMNs still causes MO-CND to find *ineffective* CrNs. The reason is that MAPs are allowed to independently discover their routes to other MAPs, so the path from the MN to the CN may differ from that form the CN to the MN. Recall that a CrN reported by MO-CND will be the first router on the fork point where the path from



(a)

(b)

Fig. 1. Problems specific to MO-CND. (a) CrN by definition (b) CrN reported by MO-CND, which is incorrect.

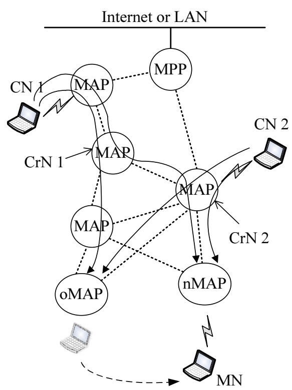


Fig. 2. Needs for multiple CrNs. CrN 1 for the communication pair (MN, CN 1) differs from CrN 2 that is for the pair (MN, CN 2).

the MN (through nMAP) to old MAP and that from the MN (through nMAP) to the CN diverge. It may differ from the CrN by definition. For example, the CrN by definition shown in Fig. 1a differs from that reported by MO-CND as shown in Fig. 1b. Here the CrN reported by MO-CND cannot perform bi-casting or frame redirection since frames from the CN to oMAP do not pass through it.

Furthermore, some layer-3 CND schemes for micro-mobility management such as [10] implicitly assume that all CNs are outside the current mobility domain of MN and thus find only one CrN for all CNs. However, the possibility of intra-WMN traffic implies that CND schemes need to find one CrN for each CN-MN pair. Fig. 2 illustrates an example where one CrN per CN-MN pair is needed. Here the CrN for CN1-MN differs from the CrN for CN2-MN.

Finding one CrN for each CN-MN pair may incur redundant protocol overhead under 802.11s two-tier frame-forwarding infrastructure. Consider two or more CNs that associate with the same MAP and communicate with the same MN. Performing SO-CND individually for these CNs will find a common CrN, so much efforts can be saved if SO-CND is performed only once for these CNs. Redundant protocol overhead may also

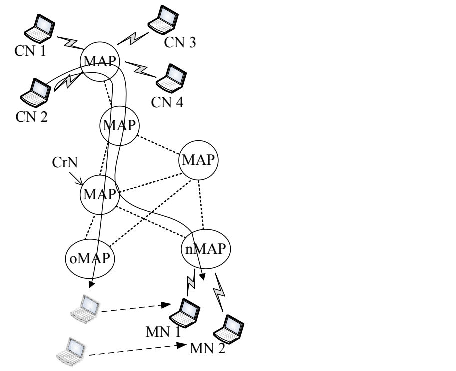


Fig. 3. Suppose that MN 1 is communicating with CN 1, CN 2, and MN 2 is communicating with CN 2, CN 3. There are two types of redundant protocol overhead. (1) The CrN for (CN 1, MN1) also serves as the CrN for (CN 2, MN 1), but CN 1 and CN 2 may perform SO-CND individually to find the same CrN. (2) The CrN for (CN 2, MN1) also serves as the CrN for (CN 2, MN 2), but MN 1 and MN 2 may perform MO-CND individually to find the same CrN.

occur to MO-CND. Consider two or more MNs that migrate from the same oMAP to the same nMAP. If there is a non-zero set of CNs common to these MNs, performing MO-CND individually by each MN will find the same CrN for these CNs. Fig. 3 illustrates such an example. In fact, a CrN is uniquely defined for a particular set of oMAP, nMAP, and the MAP which a CN associates with. If each CN or MN independently makes its own effort to find a CrN, extra protocol overhead may be incurred. It greatly reduces protocol overhead if only one CND is performed for all MN-CN pairs that share a common CrN.

B. Proposed Scheme

Discussions in the previous subsection reveal that a direct porting of existing layer-3 CND schemes to WMNs is not appropriate. We therefore propose an efficient CND for WMNs called MAPS that is characterized by the following properties:

- *MAP-centric*: By MAP-centric we mean that a proxy MAP invokes CND on behalf of all the hosts associating with it. That is, one CrN is defined among the oMAP, the nMAP, and each MAP that is a proxy of some CN (called source MAP or sMAP) rather than among the oMAP, the MN, and each CN. Consequently, an MAP is a CrN if it is on the fork point where the path from a sMAP to the oMAP and that from the same sMAP to the nMAP diverge. This property eliminates the problem caused by inadequate routing information as MAPs need not obtain routing information for all MNs. Furthermore, the first type of redundant protocol overhead in Fig. 3 can be alleviated since only one SO-CND is performed for all CNs that associate with the same MAP. The second type of redundant protocol overhead in Fig. 3 can also be avoided when only one MO-CND is performed for all MNs that share the same oMAP and nMAP.
- *Source-oriented*, which means that CrNs are defined from the perspective of source nodes (CNs) rather than MNs.

This property avoids the ineffective CrN problem specific to MO-CND as shown in Fig. 1.

- *Per-source*, which means that, for any handoff, one CrN is defined for each CN. This is to avoid the correctness problem caused by intra-WMN traffic as illustrated in Fig. 2. When combined with the MAP-centric requirement, this property implies that for a handoff between two particular MAPs, one CrN is found for each sMAP. The key point is: although one CrN should be *defined* for each CN, only one CrN is *needed* for all CNs that share the same proxy MAP.

For MAP-centric CND, one CrN is uniquely defined for each combination of three distinct MAPs (sMAP, oMAP, nMAP). If frame-forwarding paths are static or quasi-static, crossover nodes can be identified off-line and the result can be cached for on-line retrievals to minimize redundant CND sessions. More specifically, every MAP keeps a *CrN table* that memorizes one CrN for each possible handoff case. When a handoff occurs, the nMAP looks for corresponding CrNs in its table without carrying out on-line CND procedure. When backbone traffic condition changes significantly, all MAPs should update their CrN tables accordingly. The procedure to set up and update CrN tables is as follows.

1) Each node (MAP) m finds all frame-forwarding paths from it to every other node. These paths collectively form a spanning tree that is rooted at m.
2) Node m finds CrNs for each pair of other MAPs with m being the source MAP by calling *Partition&Set* algorithm (shown in Algorithm 1).
3) After *Partition&Set* returns, m multicasts the result along the spanning tree to all its descendants.
4) Every descendant, upon receiving the set of CrNs, copies those for which it is an nMAP to its CrN Table.

With a given sMAP, *Partition&Set* identifies one CrN for each possible pair of nMAP and oMAP. The algorithm traverses all nodes in the spanning tree rooted at the source MAP. Every visited node x in the tree is identified as a CrN for every possible pair of two MAPs p and q, where p and q being decedents of x belong to two different subtrees of x.

Algorithm 1 Partition&Set(s, x)

Let k_x be the number of child nodes x has
if $k_x = 0$ **then** $\triangleright$ x is a leaf node
  return
end if
Let D_x be the set of all of x's descendants and $c_x^{(i)}$ be the ith child node of x
Partition D_x into k_x disjoint sets $D_{x,1}, D_{x,2}, \cdots, D_{x,k_x}$ such that $D_{x,i}$ consists of all nodes in the subtree rooted at $c_x^{(i)}$
for each $D_{x,i}, D_{x,j}$, where $1 \leq i, j \leq k_x$ and $i \neq j$ **do**
  for each $p \in D_{x,i}$ and $q \in D_{x,j}$ **do**
    $X(s, p, q) \leftarrow x$
  end for
end for
for each $c_x^{(i)}$, where $1 \leq i \leq k_x$ **do**
  Partition&Set(s, $c_x^{(i)}$)
end for

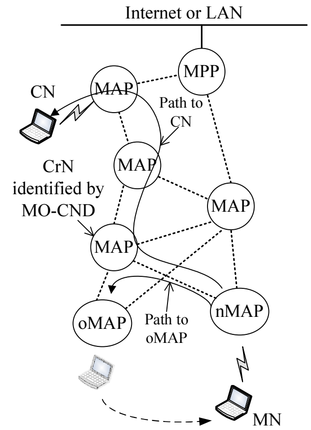


Fig. 4. An example where CrN reported by MO-CND is a better choice than the CrN by definition

CrN knowledge in Algorithm 1 is coded as a function $X(s, o, n)$, which returns the CrN with s being the sMAP, o being the oMAP, and n being the nMAP. All frame-forwarding paths from a sMAP s to every other node collectively form a directional spanning tree T_s rooted at s. By calling *Partition&Set*(s, s), MAP s can find all CrNs for which it serves as an sMAP. Let x be a non-leaf node in T_s (including s), the procedure determines the CrN to be x for any two nodes that belong to different descendant partitions of x. To determine the CrN for any two nodes that belong to the same descendant partition, procedure *Partition&Set* is recursively applied to each partition.

IV. Simulation Results

We investigated the performance of our design through simulations. The reason to adopt the MAP-centric and per-source principles in MAPS is easy to see. However, the benefit of the source-oriented principle is not so obvious. Fig. 1b does show a synthesized example where MO-CND finds an ineffective CrN, but this is not always the case. Fig. 4 shows another example where MO-CND finds a CrN that is effective yet different from that identified by SO-CND (Fig. 1a). The former is considered even better than the latter since the former is closer to nMAP and therefore can be notified of its role earlier than the latter. This specific example is still synthesized. To study the general behaviors of the source-oriented principle, we considered an alternative CND called MOCND in the simulations that also follows the MAP-centric and per-source principles but is MN-oriented.

A. Simulation Setup

We used ns2 2.33 [11] to study the performance of MAPS, MOCND, and NOOP (handoff without assistance from crossover nodes). Sixteen MAPs numbered from 0 to 15 were organized into a 4×4 grid in the simulation. Each MAP in the grid was set a transmission range that is able to communicate with and only with its neighboring MAPs in the vertical and horizontal directions. MAP 0 in the grid was assumed the only MPP in the WMN. An MN was assumed with MAP 0 initially and performed 30,000 handoffs per simulation run. In each handoff, the MN moved from the serving MAP to a surrounding MAP with equal probability, resulting in at most eight target MAP candidates for each handoff. We also deployed a CN with which the MN held a session throughout a whole simulation run. The CN resided in one MAP called the source MAP and remained stationary in each simulation run. The source MAP differed in different simulation runs, however. The CN generated a data packet toward the MN every 20 ms.

For routing, links between pairs of neighboring MAPs were assigned weights that represent traffic loads on these links, and routing paths between all possible MAP pairs were statically established by Dijkstra's shortest-path algorithm. Consequently, routing paths originating from an MAP to all possible destination MAPs collectively form a tree rooted at the source MAP. We considered both symmetric and asymmetric routing paths. For symmetric routing paths, identical weights are assigned to both directions of links between all pairs of neighboring MAPs. For asymmetric routing paths, the weight assigned to link (u, v) is different from that assigned to link (u, v) for any two neighboring MAPs u and v. More specifically, letting $d(x)$ be the minimal number of hops from MAP x to the MPP (i.e., MAP 0), link (u, v) was assigned weight $4.0 - d(u) \cdot 0.5$ if $d(u) > d(v)$ and weight 1.0 otherwise. As a result, the routing paths established between any two MAPs may be asymmetric, meaning that the routing path of a request message may differ from that of the corresponding response.

For MAPS and MOCND, packet-redirection was implemented and tested. The detailed simulation procedure is as follows. After a link-layer handoff is completed, the MN sends a location update message (through nMAP) toward the CN in order to notify the sMAP (as the routing proxy of the CN) of the MN's new point of attachment (nMAP). In MAPS, the nMAP identifies the CrN toward which the location update message is first routed. The CrN then forwards the location update message to the source MAP. The CrN starts packet redirection as soon as it receives the location update message. In MOCND, the location update message is directly routed to the sMAP. The first MAP on the delivery path that identifies itself as a CrN starts packet redirection right after it forwards the received location update message.

B. Delay

We studied the time needed for the recovery of packet stream after handoff. The first metric measured is location update delay, which counts the time spent on the delivery of location update messages. Fig. 5a shows the result for asymmetric routing paths, where we can see that the location update delays are identical in all the three methods. The same result is also observed with symmetric routing paths (not shown here). In fact, the delays are directly proportional to the hop counts of the routing paths on which location update messages propagate (Fig. 5b), which closely relate to the location of the source MAP.

Next, we investigate CrN notification delay, which measures the length of the time period from the completion of a handoff to the time at which the CrN becomes aware of its role. Figs. 6a and 7a show the results. There is no result associated with NOOP as it finds and utilizes no CrN. Since CrN is generally closer (in terms of hop counts) to the MN than the CN is, CrN notification delays are generally shorter than location update delays. For both symmetric and asymmetric routing paths, MOCND has shorter CrN notification delays than MAPS. This result reflects the fact that CrNs identified by MOCND in general are closer to the MN than CrNs identified by MAPS are. Refer to Figs. 6b and 7b for the corresponding hop count values. However, some CrNs identified by MOCND may be ineffective, which shall be discussed in the next subsection.

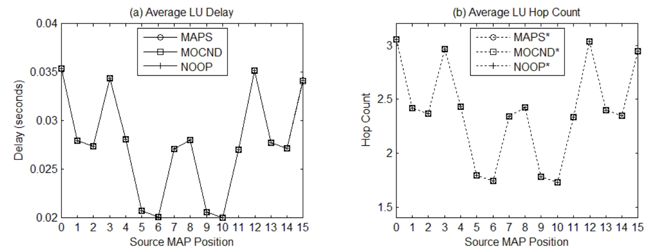


Fig. 5. (a) Location update delay (b) Corresponding hop count value

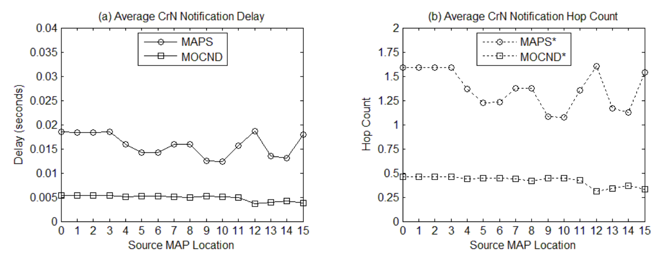


Fig. 6. (a) CrN notification delay (symmetric routing paths) (b) Corresponding hop count value

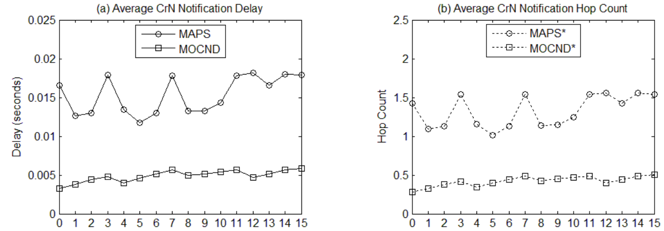
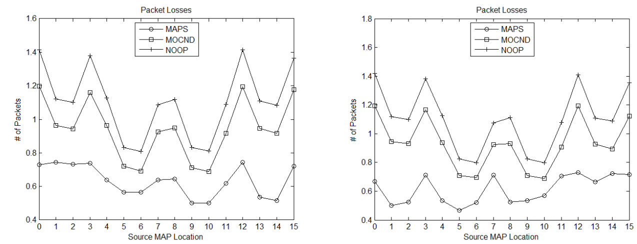


Fig. 7. (a) CrN notification delay (asymmetric routing paths) (b) Corresponding hop count value



(a) (b)

Fig. 8. Average number of lost packets per handoff (a) symmetric routing paths (b) asymmetric routing paths

C. Packet Losses

To focus on packet loss associated with CND, we set link-switch time in the simulations to zero to ensure that no packets got lost during that time. After handoffs and before the CrN or the sMAP is notified of the location of the new MAP, packets destined for the MN are still propagated toward the old MAP. Only after an effective CrN or the sMAP (in case of NOOP or when the CrN identified by MOCND is ineffective) receives the location update message can packets destined for the MN be redirected to the new MAP. We therefore count the number of packets received by the oMAP after a handoff.

The result of packet losses for symmetric routing paths is shown in Fig. 8a. As expected, packet loss in MAPS is directly proportional to the CrN notification delay, while that in NOOP is directly proportional to the location update delay. Packet loss in MOCND is higher than that in MAPS, which appears odd firstly since MOCND does have an average CrN notification delay lower than that of MAPS (Fig. 6). The reason is that a portion of CrNs identified by MOCND are ineffective. In that case, MOCND leads to the same amount of packet losses as NOOP does. In fact, the number of lost packets is determined by the CrN notification delay (in case of effective CrN) or location update delay (in case of no CrN or ineffective CrN). More specifically, letting L be the number of lost packets and p_e be the probability of finding an effective CrN, we have

$$E[L] = (p_e T_c + (1 - p_e) T_l)\, \lambda,$$

where T_c is the CrN notification delay, T_l is the location update delay, and λ is the packet sending rate. We know that $T_l \geq T_c$, $p_e = 1$ in MAPS while $p_e < 1$ in MOCND, and the values of T_l and λ are identical for all the three methods. Although MOCND on average has smaller T_c values than MAPS, we found that when the CrN identified by MOCND is closer to nMAP than that found by MAPS, the CrN identified by MOCND is ineffective. On the other hand, whenever the CrN identified by MOCND is effective, it is not better than the CrN identified by MAPS in terms of T_c due to the symmetry of routing paths. Consequently, CrNs identified by MOCND cause equal or higher number of packet losses than those identified by MAPS.

The result of packet losses for asymmetric routing paths is shown in Fig. 8b. It exhibits the same trend as that shown in Fig. 8a. We found that 40% to 50% CrNs identified by MOCND are ineffective. Although scenarios such as those shown in Fig. 4 did occur in this simulation setting, the proportion of such occurrences was not high enough to dominate the final result. This explains the inferior performance of MOCND compared with MAPS.

V. Conclusions

We have identified four main issues that may arise when adapting existing CND schemes to IEEE 802.11s WMNs. First, MAPs in WMNs may not have adequate MN location information for CND. Second, MN-oriented CND may find ineffective CrNs with non-tree frame-forwarding paths in WMNs. Third, the possibility of intra-WMN traffic calls for multiple CrNs, which may not be supported by existing CND schemes. Four, it causes redundant efforts to find a CrN for each CN-MN pair under 802.11s two-tier hierarchy. To deal with these issues, we have pointed out that CND in WMNs should be source-oriented, MAP-centric, and per-source. We also develop an efficient CND scheme for WMNs which can be conducted off-line with the derived result cached for retrievals during handoffs. Simulation results show that the proposed scheme reduces more packet losses due to handoffs than a counterpart that is MAP-centric and per-source but not source-oriented.

Acknowledgement

This work was supported in part by National Science Council, Taiwan, under Grants NSC 100-2221-E-009-072-MY3 and NSC 101-2219-E-009-028.

References

[1] P. H. Pathak and R. Dutta, "A survey of network design problems and joint design approaches in wireless mesh networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 396 –428, 2011.

[2] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh networks design: A survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 299–310, 2012.

[3] K. Mase, "Layer 3 wireless mesh networks: mobility management issues," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 156 –163, july 2011.

[4] A. G. Valko, "Cellular IP: a new approach to internet host mobility," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 1, pp. 50–65, Jan. 1999.

[5] R. Ramjee, K. Varadhan, L. Salgarelli, S. R. Thuel, S.-Y. Wang, and T. L. Porta, "HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, pp. 394–410, Jun. 2002.

[6] E. Fogelstroem *et al.*, "Mobile IPv4 regional registration," IETF, Tech. Rep. RFC 4857, Jun. 2007.

[7] IEEE, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE, New York, 2008.

[8] J.-Y. Hu and C.-C. Yang, "Enhancement of Cellular IP routing by redirection at crossover base stations," in *IEEE VTC-2005 Fall*, Sep. 2005, pp. 25–28.

[9] R. V. Ferré and J. P. Aspas, "Improving fast handovers for mobile IPv6: optimal crossove discovery using geopaging routing tables," in *VTC 2005-Spring*, May 2005.

[10] D. H. Cuong, J. K. Choi, D. Guha, J. S. Park, H. J. Kim, and J. H. Hahm, "Proactive crossover router discovery with FMIPv6," in *Proc. 4th ACM Int' Workshop on Mobility Management and Wireless Access*, Malaga, Spain, Oct. 2006, pp. 104–107.

[11] "The network simulator - ns2." [Online]. Available: http://www.isi.edu/nsnam/ns/