A Framework for Mobile Multicast Using Dynamic Route Reconstructions

KUANG-HWEI CHI, CHIEN-CHAO TSENG AND TING-LU HUANG

Department of Computer Science and Information Engineering, National Chiao Tung University 1001, Ta-Hsueh Road, Hsinchu, Taiwan 30010
Email: cctseng@csie.nctu.edu.tw

A mobile computing environment allows hosts to roam while retaining access to the Internet. Multicasting is one of the most important facilities for constructing reliable distributed systems and cooperative applications. Host mobility, however, challenges multicasting in this environment: the established multicast delivery paths may frequently restructure along with host migrations, incurring expensive overheads. This paper presents a framework for network-layer multicasting while keeping a low overhead in adapting multicast routes to mobile host locations. This is achieved by partitioning the mobile environment into non-overlapping regions, so that changes in the multicast routes due to host intra-region movements are hidden from other regions. An analytical model was developed for performance evaluation. It shows that, compared with the best known proposal, our scheme reduces the average multicast latency by more than 66%, while causing less than 7% overhead.

Received August 1, 1998; revised April 21, 1999

1. INTRODUCTION

Multicasting is a technique allowing a single message to be passed to a set of destinations. This is useful in information dissemination. In practice, a multicast route is realized as a tree rooted at the sender with a receiver at each leaf. Paths in the tree diverge and the message delivery is parallelized to the destinations along the branches of the tree. Multicasting is one of the most important facilities for constructing reliable distributed systems and cooperative applications, such as coordinating updates to replicated file systems [1].

The explosive growth of wireless communications leads to the integration of wireless networks with the Internet. Such an integrated environment, called a mobile computing environment, allows hosts to roam around freely while retaining accesses to the Internet over a wireless medium. In mobile environments, a typical multicasting application is teleconferencing, where a meeting is held via mobile computers. Users in this multi-party conversation have the freedom of mobility, for example, riding in the back seat of a taxi, and work as if they were in offices. Other applications specific to mobile multicasting can also be found in [2, 3, 4].

An ideal mobile networking environment provides hosts with *anytime* and *anywhere* Internet services. Host mobility, however, causes a severe problem to IP multicasting in this environment: since a multicast route locates all of the participants in a group, the established route may frequently restructure along with the changes of mobile host locations. This incurs costly overheads because incorrect routes require modifications throughout the Internet.

All previous work that supported multicasting in mobile environments maintained multicast routes statically, irrespective of the mobile host locations [5, 6, 7, 8, 9]. As a result, multicasts are delivered in an inefficient way: multicasts to or from mobile hosts that are away from their home must be routed indirectly by way of their respective home networks. This may cause unacceptably long latencies, waste network bandwidth and overload the networks along the delivery path.

This paper presents a framework for routing multicast datagrams to recipients directly and efficiently in mobile IP internetworks. We dynamically adapt multicast routes to mobile host locations, while maintaining low cost. This is achieved by partitioning the mobile environment into a number of non-overlapping regions, so that most of the location changes due to host movements are hidden from other regions. As a result, when a host makes intra-region movements, the changes of multicast routes only affect that region. Global changes in multicast routes, due to host movements, might be thus considerably reduced.

When a mobile host moves to an *inactive* network where no group members reside, multicast routes are reconstructed. The host will experience a delay in reconstructing new routes and cannot proceed group communications instantly. To solve this, we set up a temporary channel for the mobile host between the point of attachment to the previous network and that to the new network. The previous network will re-direct the multicast packets in flight to the host's new location, until a designated timer for the channel expires.

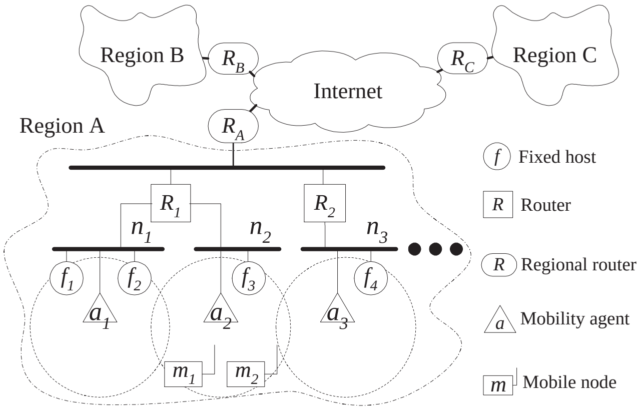


FIGURE 1. Reference mobile computing architecture.

The rest of this paper is structured as follows. A brief background on the system model and the problem of our concern are presented in the next section. Section 3 summarizes previous work and outlines its insufficiencies. Section 4 describes our framework and Section 5 discusses its qualitative performance and applicability. Performance evaluation is conducted in Section 6. Lastly, in Section 7 we draw conclusions.

2. BACKGROUND

This section provides a brief background on the system architecture and the problem to be studied. To interoperate with the existing networking systems, we use several protocols in the TCP/IP suites as the building blocks.

2.1. System model

A mobile computing environment could be augmented from static internetworks that span across wide areas of multiple regions, as shown in Figure 1. A region in this text refers to a cluster of routers and subnets encompassed by an enterprise or a campus network. Host mobility is supported by the standard mobile IP protocol [9], to be summarized below.

A mobile node (MN) is a host or a router that may move around, while retaining connections to a static network over a wireless medium. An MN is assigned a permanent IP address, namely a *home address*, on its home network. On a local network, there is a router called the mobility agent (MA) that serves as a point of attachment to the static network for MNs. For an MN, an MA refers to either a home agent (HA) or a foreign agent (FA), depending on whether the network is the MN's home.

When an MN moves to a foreign network, it obtains a *care-of address* from the FA and registers the new address with its home agent. The care-of address indicates the MN's current location and is generally the FA's IP address. Datagrams sent by the MN use the FA as a default router and are delivered to their destinations by standard routing mechanisms. In contrast, datagrams meant for the MN are forwarded by the normal routing mechanisms to the MN's home network, where they are intercepted by the MN's HA. The HA then encapsulates the datagrams within new IP datagrams directed to the MN's current care-of address. On receipt of these datagrams, the FA decapsulates and delivers them to the destination MN. The method of encapsulating datagrams to work around normal IP routing is called *tunneling.*

On a foreign network, an MN may acquire a co-located care-of address locally. In this case, the MN itself performs datagram encapsulation and decapsulation. This can cause heavy power consumption on the MN and therefore is not preferred for consideration in this article. We hereafter suppose that an MN only uses FA's care-of address on a foreign network. For brevity, a subnetwork (or subnet for short) henceforth refers to a local network.

2.2. Network-layer multicast support

A multicast IP address is capable of identifying a dynamic set of hosts, which may join or leave a group at any time. A multicast-capable Internet router is referred to as a *multicast router*. To deliver packets to a group, routers need mechanisms to track group membership and deliver datagrams towards members.

The standard mechanism for membership tracking is the Internet Group Management Protocol (IGMP, version 2) [10]. IGMP learns which groups have members on which subnetworks. A router only records the presence of a group on each attached subnet, rather than the knowledge about which local hosts belong to the group. Periodically IGMP queries the local network to determine if any hosts are still group members. When the last host to respond to a query with a report wishes to depart from the group, it sends a leave-group message onto the subnet. The local router is thus triggered to send a group-specific query. If no reports are received for the group before the designated timer (the elapsed time is called the *leave latency*) has expired, the router removes the group and will not forward remotely-originated multicasts for that group onto the local network.

IGMP provides the final step in a multicast packet delivery service since it is only concerned with the forwarding of multicast traffic from the local router to group members on directly attached subnetwoks. IGMP is not concerned with the delivery of multicast packets across an internetwork. In conjunction with IGMP, a multicast routing mechanism is responsible for the construction of multicast delivery trees and performs multicast packet forwarding, to support an Internet-wide delivery service.

The multicast routing mechanism is not yet standardized. We deploy the Distance Vector Multicast Routing Protocol (DVMRP) [11], the main one used nowadays. Alternatives can be found in [12, 13, 14, 15]. In DVMRP, each router periodically exchanges with its neighbors routing information that encodes the respective *distance* to each subnet. The routing information is propagated throughout the multicast backbone so that every router keeps track of the shortest paths to all of the subnets. Effectively a tree, namely a multicast route, locating all the members of a particular group is set up per subnetwork with multicast senders. Such a source-rooted tree is refreshed whenever a multicast datagram gets delivered and will be removed after a timeout period (5 minutes in practice) since the last packet emanated from the source subnet. DVMRP uses the 'broadcast and prune' approach, in which multicast packets are initially delivered to all the multicast routers; those routers that are not on the path to a group member are pruned off the delivery tree for the group. If a new host appears later, the pruned router grafts paths back to the tree.

In this paper a subnetwork is said to be *inactive* if it is pruned from a multicast tree for a group; otherwise it is said to be *active*.

2.3. The problem

Host mobility challenges multicasting in a mobile environment, since the traditional routing protocol implicitly assumes static hosts when setting up a multicast route. In a mobile environment, the network must not only manage multicast group membership and establish the delivery paths, but also contend with the fact that the established routes are themselves transient in nature.

An MN can fail to receive multicasts when it moves. This is because multicast routing relies on the pre-setup of delivery paths on routers, while routers do not track MNs' movements. Thus, when an MN moves, routers have no knowledge of the movement and the established paths become obsolete. Packet delivery among the group members can be thereby disrupted.

3. PREVIOUS WORK

The above problem may be solved by reconstructing the delivery tree along with MNs' migrations. However, this can incur significant overheads because incorrect routes, with respect to each multicast source, require entry modifications on routers throughout the Internet. The overhead becomes overwhelming when multiple MNs move simultaneously and frequently. Instead, most of the previous schemes maintained delivery trees independent of MNs' movements and multicast trees thus appear static [5, 6, 7, 8, 9]. A summary of previous work is as follows.

Acharya's approach [5]: Datagrams are delivered using region-wide broadcast, so every subnetwork receives every multicast packet, irrespective of whether recipient MNs are present in the region. Because this scheme was intended for Columbia Mobile IP [16], its applicability is limited. This paper henceforth does not consider this scheme.

Mobile-IP home subscription [9]: Multicast trees are constructed as if the MNs are always located at their home networks. Multicast to or from MNs that are away from home must be re-routed indirectly via their respective home agents. This scheme can result in quadrangular routing, as depicted in Figure 2. Suppose that MNs m_1 and m_2 are attached to foreign agents FA_1 and FA_2, respectively. Multicast packets from m_1 are first tunneled from FA_1 to its home agent, HA_1, and are thereafter propagated over the established tree. As the packets arrive, m_2's home agent HA_2 tunnels them to FA_2, and thereby to the destination m_2. This scheme is degraded by the *tunnel convergences*: multiple tunnels (from different HAs) can terminate at a given FA [6]. Thus one copy of every multicast packet would be forwarded to the FA; duplicate multicast packets are delivered to local MNs, wasting network bandwidth.

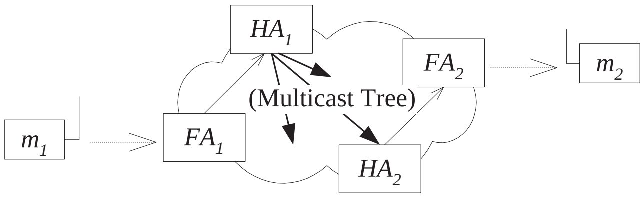


FIGURE 2. Schematic diagram showing quadrangular routing.

Mobile-IP remote subscription [9]: As an MN moves to a foreign subnet, it acquires a new co-located care-of address locally and henceforth acts as a stationary host on that subnet. However, this scheme suffers from the penalty of frequent multicast route reconstructions and is thus impractical.

Chikarmane's approach [6, 7, 8]: This is an optimized Mobile-IP home-subscription approach that eliminates tunnel convergences. Nevertheless, duplicate packets still occur when an MN member visits a foreign network whose own MN members are also away from home. Moreover, since multicast trees are still constructed according to the MNs' home addresses, the quadrangular routing problem remains.

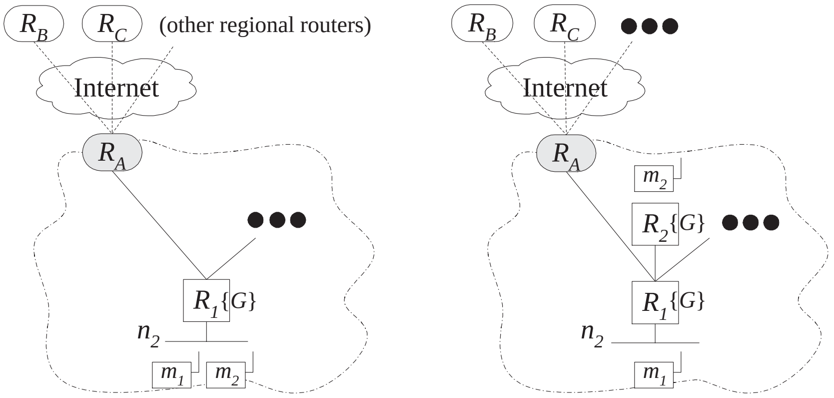
It can be evidenced that a major common flaw of the previous schemes is packet quadrangular routing. This might cause unacceptably long communication latencies, wasted network bandwidth and burdens on the networks along the delivery path. These disadvantages are attributed to multicast trees being maintained regardless of the MNs' locations. To improve group communications efficiency, this paper presents a framework for routing multicast packets directly to or from the networks where the participant MNs are situated. We dynamically adapt multicast routes to mobile host locations, while maintaining a low overhead, as described in the following section.

4. THE DYNAMIC ROUTE SCHEME

It is hereafter assumed that there is a multicast router (possibly co-located at the mobility agent) attached to the subnet where an MN dwells.

4.1. Overview

The overhead of reconstructing multicast routes is reduced by partitioning the mobile environment into regions, so that



(a) m_2 dwells at subnet n_2 (b) after m_2 moves to subnet n_3

FIGURE 3. The multicast tree rooted at subnet n_2.

changes in the multicast routes due to MNs' intra-region movements are isolated to the same region. A mobile environment, as illustrated in Figure 1, is intrinsically a hierarchical structure which is composed of regions. A region contains some number of subnets and routers in a geographical area, e.g. a campus, as a whole. In essence, each region is treated as if it were a single subnetwork.

Within a region, each router manages its group and exchanges periodic routing information with each of its neighbors as usual. Such routing information is propagated throughout the region, thereby to the regional router. Then the regional router initiates the routing information of its own downstream, on behalf of local routers. Hence the far-end routers outside a region are only aware of the regional router and its attached whole network. MNs' intra-region movements do not change the membership aggregated within the region and, thus, the delivery paths off the region remain. In other words, the changes in multicast routes due to the MNs' intra-region movements are limited to that region, rather than the entire Internet.

For example, considering region A in Figure 1, suppose MNs m_1 and m_2 both belong to group G. First, MN m_2 uses a delivery tree (shared with m_1) to distribute multicasts from subnet n_2, as depicted in Figure 3a. To construct such a tree, the routing information originating from R_1 is propagated throughout Region A, while it is suppressed at the regional router R_A from further outgoing propagation. Instead, on behalf of R_1, R_A sends R_A-initiated routing information to the neighboring routers off the region. When m_2 later moves to subnet n_3 without group members, the local router R_2 detects membership for group G (by some means proposed in Section 4.4) and grafts paths onto the multicast trees rooted at other subnetworks, say n_2, as shown in Figure 3b. In this example, the graft message issued by R_1 is propagated inside, but not outside, region A. As a result, the changes in multicast trees due to host intra-region movements are localized within a region and hidden from the routers off the region.

4.2. Protocol notation

A router maintains a routing table that contains an entry for every reachable multicast source on the Internet. Here a source refers to either the whole network of an external region or a single subnet internal to the same region. Therefore, multicast senders with the same network identifier are treated as the same source and thus share the same delivery tree. The network identifier, hereafter denoted as *NETID*, can be derived from the prefix of an IP address.

A routing table consists of entries, each of which has the following fields:

source	address of the network where a multicast sender dwells,
group	multicast group address,
distance	shortest-path distance to the source,
parent	identifier of the shortest-path link toward the source,
pruned	set of links in the absence of the group members,
timer	time left for this entry to become stale.

Note that *distance* is typically measured in the number of hops and *parent* indicates the one-hop-back subnet address from the router toward the source. A route entry in the routing table is indexed by a (*source*, *group*) pair; $RT[source, group]$ indicates the corresponding entry in the table. Otherwise it is **nil**, provided there exists no information about that (*source*, *group*) pair.

Periodically each router exchanges routing information with its neighbors, carrying a source and distance known to the sending router. By comparing the contents of the information with its own routing table, the receiving router learns of other sources and of the best, precedent, hop routers to reach those sources. Each router keeps a list, for each incident link, of those groups that are present on that link. This information is used to update the receiving router's table and is, in turn, reported in this router's periodic routing information. By such neighbor-to-neighbor exchanges, all routers eventually end up with routing tables identifying all reachable sources in the system.

We concentrate on the routing protocol performed on a regional router, since this is the sole entity to be modified. The protocol is tailored to control the outgoing routing information so that the route changes due to the MNs' intra-region movements are hidden from other regions. Multicast routing protocols performed elsewhere are unchanged.

An abstraction of the routing protocol is as follows. The routing process has a set of global and local constants, a set of local variables, and a set of actions. Actions are separated from each other with the symbol [], using the following syntax:

begin *action* [] *action* [] ... [] *action* **end**

Each action is of the form *guard* → *command* [18]. A guard is either a Boolean expression involving the local variables of its process, or a receive statement of the form **rcv** msg **from** q, where msg is a message and q is an identifier of the router's incident links and is assumed to be an integer type. A command is con-structured from sequencing (;), conditional (**if fi**), and iterative (**for rof**) constructs that group together **skip**, assignment and send statements of the form **send** msg **to** q. Similar notations for defining network protocols are given in [17, 18].

An action is said to be *enabled* if its guard is either a Boolean expression that evaluates to true, or a receive statement of the form **rcv** msg **from** q and has a message msg arrived from link q. An execution step of a protocol consists of choosing any enabled action and executing the action's command. Protocol execution is fair; that is, each action that remains continuously enabled is eventually executed.

4.3. Protocol description

The routing protocol executed on a regional router, modelled by a routing process P, can be written in the following Figure 4. We merely show the most essential actions in the process: handling the routing information and forwarding the multicast data messages using its routing table. The routing information is of the format (*routing*, *source*, *group*, *distance*), while the data message is of the format (*data*, *source*, *group*).

In the action at label 0, process P, on receiving routing information (*routing*, *source*, *group*, *distance*) from one of its neighbors j, determines which of its attached subnets is its parent link for that *source*, with respect to that *group*. If j offers a shorter path toward the *source*, P updates the associated route entry and resets the parent to j. Then, at label 0A, P determines whether the routing information is originated internally to the region that P serves. If it is, P sends new routing information (*routing*, *NETID*(P), *group*, 0) to each external link. Otherwise, P relays the routing information (*routing*, *source*, *group*, *distance* $+ 1$) to its neighbors other than j. Effectively, the regional router suppresses the routing information originating from inside the region and initiates its own towards external subnets. In this manner, external routers are only aware of the region as a whole. However, the regional router does not suppress any incoming routing information.

Only minor modifications are needed on a regional router. The only modification to the normal routing protocol is the command at label 0A. Originally this command is simply

send (*routing*, *source*, *group*, *distance* $+ 1$) **to** i.

In the second action (at label 1), process P, on receipt of a multicast message (*data*, *source*, *group*) from j, examines if the message comes from the shortest-path link toward the *source*. If it is, P forwards the message to its neighbors except j and those links without group members. Otherwise, the message is discarded.

Other important actions involved in the protocol are summarized as follows. The routing protocol uses a 'broadcast and prune' approach, in which multicast datagrams are initially delivered to all the routers on the Internet. Those routers that are not on the path to a group member are subsequently pruned off the delivery tree. Information about the absence of group members propagates back up the tree towards the source, along all branches that do not lead to group members. This information is recorded in RT[*source*, *group*].*pruned*. Subsequent packets from the same source to the same group are blocked from traveling down the unnecessary branches.

When a new group member appears on a pruned link j, the pruned branch is quickly grafted back onto a multicast tree. This is achieved by sending a *graft* message to the immediate router that is attached to j. So j is removed from RT[*source*, *group*].*pruned* accordingly. The graft message is propagated as far as necessary to rejoin the originating router into the specified multicast tree.

4.4. Group management

We use the standard group management protocol, IGMP, to manage multicast groups. Each mobility agent maintains a membership table containing an entry for every local MN of some group(s). The table entry is a tuple ⟨*MN-address*,*group-list*⟩, where *MN-address* is an MN's home address and *group-list* represents the identifiers of the groups the MN participates in currently. MNs are unaware of such a membership table.

Initially when an MN, say m_i, sends a request to join a group G, this request will be overheard by all the hosts and routers attached to the local network. The local mobility agent checks if there is an entry for m_i in its membership table. If not, an entry is created, whose *MN-address* field is initialized to m_i. In all cases, this new group identifier G is added into the *group-list* field. Instantly the local mobility agent sets its network interface to recognize the multicast group address.

When m_i detects that it has moved to a new subnet, a handoff procedure commences. Suppose m_i, whose HA is a_h, changes MAs from a_j to a_k. The associated membership table entry for m_i will be transferred to the new agent a_k.

```
type
   neighbors = set of integer;                          (* neighboring routers *)
   route = record
               source, group, distance, parent, timer: integer;
               pruned: set of integer init $\emptyset$
         end
process P;
var
   $i, j$ : neighbors;                                  (* $i, j$ range over all neighbors *)
   new : route;
   $RT$ : set of route init $\emptyset$;                (* routing table *)
begin
   true $\rightarrow$
0:    rcv (routing, source, group, distance) from $j$ $\rightarrow$
         if $RT[source, group] = \mathbf{nil}$ $\rightarrow$        (* add new route *)
            new.source, new.group, new.parent := source, group, $j$;
            $RT := RT \cup new$;
            send (routing, source, group, $\infty$) to $j$
         [] $RT[source, group] \neq \mathbf{nil}$ $\rightarrow$     (* update an existing route *)
            if $distance \geq RT[source, group].distance$ $\rightarrow$ skip
            [] $distance < RT[source, group].distance$ $\rightarrow$
               $RT[source, group].parent, RT[source, group].distance := j, distance$;
               send (routing, source, group, $\infty$) to $j$
            fi
         fi;
         $RT[source, group].timer := T_L$;              (* refresh group-membership timer *)
0A:      for $i \in neighbors - \{j\}$ do
            if $NETID(source) = NETID(P) \wedge NETID(i) \neq NETID(P)$ $\rightarrow$
               send (routing, $NETID(P)$, group, 0) to $i$
            [] $NETID(source) \neq NETID(P) \vee NETID(i) = NETID(P)$ $\rightarrow$
               send (routing, source, group, $distance + 1$) to $i$
            fi
         rof
      []
1:    rcv (data, source, group) from $j$ $\rightarrow$
         if $RT[source, group].parent \neq j$ $\rightarrow$ skip
         [] $RT[source, group].parent = j$ $\rightarrow$
            for $i \in neighbors - \{j\} - RT[source, group].pruned$ do
               send (data, source, group) to $i$
            rof
         fi
      []
      .
      ⋮
end
```

FIGURE 4. Protocol description.

The handoff procedure is illustrated in Figure 5, with the following sequence of steps.

Step 1. When m_i detects it has moved to a new subnetwork, it sends a registration request message to the new agent, a_k. The message contains encryption keys and m_i's previous agent information.

Step 2. The new agent will forward the request to m_i's HA, a_h, in case a_k is foreign to MN m_i. On receiving a registration reply from a_h, a_k relays the message to m_i.

Step 3. The new agent a_k sends to m_i's previous agent a list of identifiers of groups, $glist_k$, to which a_k currently belongs.

Step 4. Previous agent a_j returns a binding reply, $glist_j$, a list of identifiers of groups in which m_i participates.

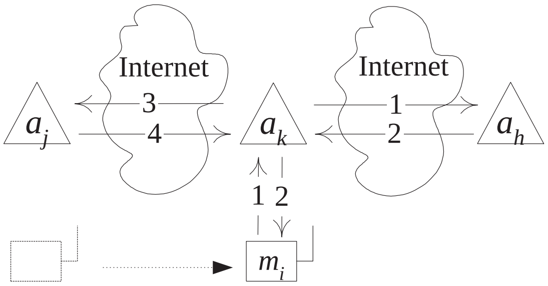


FIGURE 5. Message flow in the handoff procedure.

> At this moment, for MN m_i, a_j establishes a tunnel between a_j and a_k for each group in $glist_k$ but not in $glist_j$.

The new agent a_k, for the MN m_i, immediately joins these groups in $glist_k$ but not in $glist_j$, in the same way as a normal host joins a group.

The set-up of a tunnel between a_j and a_k is advantageous when the MN m_i moves to an inactive subnet. In this manner m_i is able to resume receiving multicasts shortly after its handoff is completed, even though, in the course of multicast, the route changes. The previous agent a_j redirects the multicast datagrams in flight to m_i's new location. Such a tunnel remains until a timer expires.

The proposed handoff procedure is extended from the mobile IP with route optimizations [9, pages 129–167]. We mainly use the 'forwarding pointer' concepts to establish a tunnel that supports the MNs to proceed with group communications while multicast delivery paths are restructuring.

4.5. Multicast routing

Multicast packets are delivered directly to and from where MNs are currently located. A source MN sends multicast packets using its home address as the IP source address. As shown in Figure 6, a local mobility agent encapsulates multicast datagrams whose IP source addresses are foreign to the network and re-sends these datagrams using its own IP address. An outer IP header is inserted before these original datagrams. The outer IP header's Source Address and Destination Address specify the agent IP address, and the destination multicast group address, respectively. In this way, multicasts will be routed over the established paths to group members.

On a leaf subnetwork, a router delivers datagrams received from upstream to the local agent by static routing. That is, by always setting the physical frame's destination address to the agent's MAC address, the agent is the sole recipient in the physical layer. On receiving multicast datagrams, the agent determines by the protocol field in the datagram IP header whether these datagrams are encapsulated, i.e. protocol numbers are four. If they are, the agent must decapsulate them first and deliver the datagrams to the intended group members on the local network. The datagrams are assigned the multicast addresses in physical frames, so that local members can receive them.

4.6. Remarks

Partitioning the multicast backbone into regions was studied previously [19, 20]. However, these works considered a stationary networking system rather than a mobile environment. In [19], Deering proposed an approach to hierarchical routing based on hierarchy encoded in the (unicast) source address of the multicast packets. In [20], Thyagarajan and Deering used region identifiers that are not encoded in the addresses and use encapsulation for the inter-region forwarding of datagrams. This method is amenable to incremental deployment and reduces the amount of topological information that routers must store and exchange. However, regional routers can be overloaded in performing a decapsulation and encapsulation to each multicast packet. When a region contains a large number of members, the regional router is vulnerable to heavy traffic loads.

The concept of regionalizing the network also appears in hierarchical mobile IP by Perkins [9, pp. 187–199]. In the hierarchical mobile IP, an MN's registration can be transacted with a regional agent without requiring approval by or rebinding at the HA to smooth the registration procedure. The localized registration of the mobile IP is not in the context of multicasting and the issues specific to multicast group communications remain. However, our scheme is orthogonal to the hierarchical mobile IP and could be regarded as an augmentation to that proposal which deals with multicast packet routing.

5. QUALITATIVE DISCUSSION

We investigated our framework qualitatively in terms of three dimensions as follows.

Scalability. Our scheme scales well since we partition the mobile environment into regions, each of which is treated as if it were a single subnet. A router maintains an entry for each subnet within the same region and an entry for each external region. The routing table size can be thus kept small, even though the MN population is large.

Inter-operability. Our scheme is inter-operable with the existing systems, in that the TCP/IP protocol suites are used as the base building blocks in our framework. The modified entities only require minor changes. No sophisticated algorithms are involved.

Traffic overhead. Due to the leave latency, an idle subnet defers from being pruned from multicast trees. Multicast route reconstructions are avoided if a new local host joins the group or an MN member moves into the subnet before the associated group membership timer expires. However, this is likely to lead to waste network bandwidth, because multicast packets could be delivered to subnets without recipients.

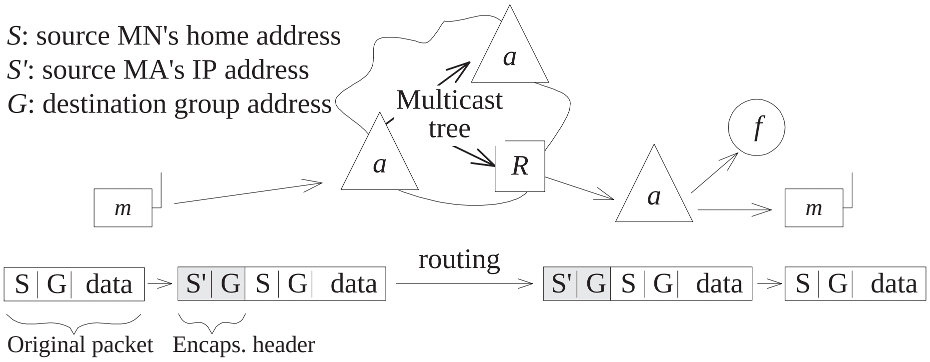


FIGURE 6. Multicast routing with encapsulations and decapsulations of datagrams.

6. PERFORMANCE EVALUATION

A multicast group containing fixed hosts benefits our scheme in that the delivery trees are less volatile. For simplicity, we restrict ourselves to the case where a group consists only of MNs and none joins or departs from the group over time, i.e. a router's membership change is solely affected by the MNs' movements.

6.1. Assumptions and metrics

The assumptions used in our performance study are as follows.

(A1) Each MN is a multicast sender and issues at least one multicast datagram after it moves to a subnetwork.

(A2) Wherever an MN migrates, it is able to connect to a MA that provides multicast services. This implies group communications are not disrupted regardless of the MNs' locations.

(A3) MNs are uniformly distributed over a mobile environment and their movements are arbitrary in direction.

(A4) MNs are indistinguishable and have the same mobility rate.

From assumptions A3 and A4, we infer that all subnets are statistically identical. (Conventionally this is called a homogeneous system.) Therefore it is sufficient to consider a single subnet instead of the entire mobile environment.

Two performance indices are investigated: reconstruction overhead and multicast latency. The reconstruction overhead refers to the extra cost to dynamically modify multicast trees upon host movements. The extra cost is compared with the total cost of periodically maintaining multicast trees in Chikarmane's scheme. The multicast latency refers to the expected delivery time required for an MN to receive a multicast from a source MN. Both the performance metrics are measured in terms of a single multicast within a single group. We compare our multicasting scheme with Chikarmane's approach since that approach is an optimized work of the standard Mobile IP.

6.2. Reconstruction overhead

As mentioned previously, we adjust multicast routes to mobile host locations. Therefore our scheme causes extra reconstruction overhead due to host mobility. The reconstruction overhead is measured in terms of a time interval defined below. As a multicast recipient, a subnetwork moves back and forth between active and inactive states. As shown in Figure 7, a subnet's life cycle is defined to be an active period and the following inactive period. A subnet accommodating at least one member is said to be *busy*; otherwise it is said to be *idle*. When a subnet turns idle, it does not depart from the multicast trees immediately. On the contrary, it will remain active until L time has elapsed. L is the IGMP leave latency (10 seconds in practice.) If an MN moves into an idle subnet before the leave latency has terminated, it revives the subnet and keeps the subnetwork active.

Let Y be the random variable for a subnet's life-cycle duration. From the Appendix, we can derive the expected value $E[Y]$ as follows.

$$E[Y] = \frac{\mathrm{e}^{N_m(1+\eta L)}}{N_m \eta}$$

where N_m denotes the average number of MNs present on a subnet and η is the mobility rate for each participant mobile host. ($1/\eta$ is the average time for an MN to dwell in a subnetwork.)

The above addresses the case in which a subnetwork is a recipient. Considering a subnet as a source, the multicast tree rooted from the subnet is refreshed whenever a packet is transmitted from the source. The source-rooted tree will expire 5 minutes after the last datagram has been delivered thereon. Similar to the above arguments, the source subnetwork also has another life cycle for building the delivery tree whose duration is likely to be longer than Y. For simplicity we suppose that these two time periods are equal in length.

The reconstruction overhead could be represented as a measure proportional to the number of router-updates to maintain multicast routes. Let r and R, respectively, be the number of subnetworks within a region and over the entire multicast backbone, on which each subnet is assumed

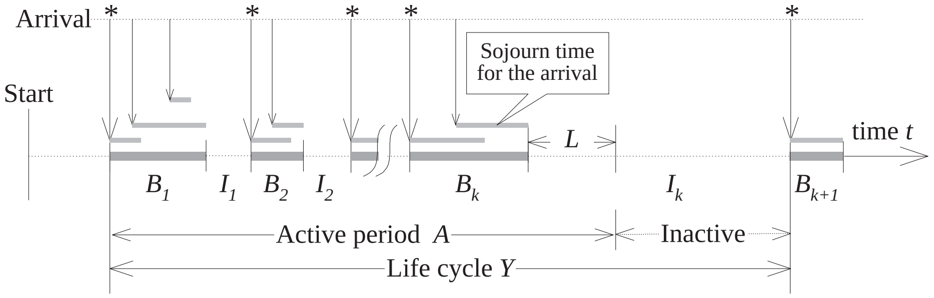


FIGURE 7. A subnet behaves as an alternating renewal process. An arrival marked with '$*$' initiates a new busy period.

to be connected via a respective router. Suppose further each region contains the same number of subnetworks. A region is inactive if all the r subnets within are inactive, at probability π_0^r, where π_0 denotes the probability that a subnetwork is found to be idle. When an inactive subnet turns active, it grafts paths to the multicast trees rooted at other sources, in which multicast routers are updated. If the subnet is within an inactive region, the subnet will activate the region and all the R routers may be updated. Otherwise only the routers within the region are affected. Thus in equilibrium, a subnet activation causes $(R\pi_0{}^r + r(1-\pi_0^r))$ router-updates. When the subnet later starts inactive, it is pruned from multicast trees. Such pruning could also go to $(R\pi_0{}^r + r(1-\pi_0^r))$ routers. In addition, the subnet, as a source, updates $2(R\pi_0{}^r + r(1-\pi_0^r))$ routers to build a multicast tree in a life cycle. To summarize, in our approach a subnet incurs $4((R-r)\pi_0^r + r)$ extra router-updates per $E[Y]$ time.

On the other hand, in DVMRP each router periodically exchanges routing information with its neighbors every minute. The routing information is propagated globally. In other words, a subnet both in our scheme and in Chikarmane's approach regularly causes $(RE[Y])$ router-updates during time interval $E[Y]$, to maintain routing tables. Therefore, we have

$$\text{Reconstruction overhead} = \frac{4((R-r)\pi_0^r + r)}{RE[Y]}. \qquad (1)$$

6.3. Multicast latency

Multicast latency refers to the expected delivery time required for MNs to receive a multicast datagram from a source MN. Assume that the packet delivery time between any two hosts through the Internet has the same mean X_1.

6.3.1. Chikarmane's approach

Given a source and a destination MN, the multicast latency can be addressed in three cases as follows.

Case 1. If both MNs are at their home networks, datagrams originating from a source MN are delivered toward the destination MNs through the multicast tree (Figure 8a). The multicast latency in this case is X_1.

Case 2. If one of the two MNs is visiting a foreign network, the HA of the MN that is away from home is thus responsible for re-routing the multicast datagrams. The delivery path is viewed to consist of two segments as depicted in Figure 8b, yielding $2X_1$ of multicast latency.

Case 3. If neither of the two MNs are home, both HAs are responsible for re-routing multicast datagrams for the MNs. The delivery path is composed of three segments, as shown in Figure 8c, whose multicast latency is thus $3X_1$.

Because MNs move randomly, the probability of finding an MN in its home network is $1/R$. The overall latency of Chikarmane's scheme, T_C, results from weighting the above three cases:

$$T_C = \left(\frac{1}{R}\right)^2 X_1 + 2\frac{1}{R}\left(1-\frac{1}{R}\right)2X_1 + \left(1-\frac{1}{R}\right)^2 3X_1$$
$$= \left(3-\frac{2}{R}\right)X_1. \qquad (2)$$

6.3.2. Our approach

In our architecture, MNs are, primarily, immediately reachable from multicast trees, so multicast packets are delivered directly to destination MNs in X_1 time. However, when an MN moves to an inactive subnet, multicast datagrams to the MN are delivered via a temporary tunnel, passing through its previous MA indirectly, until a designated timer expires. Suppose that the tunnel's timeout value is U. Multicast delivery using tunneling takes place with the probability $(U/(E[Y]))$. $E[A]$ is the mean length of an active period which can be found in the Appendix:

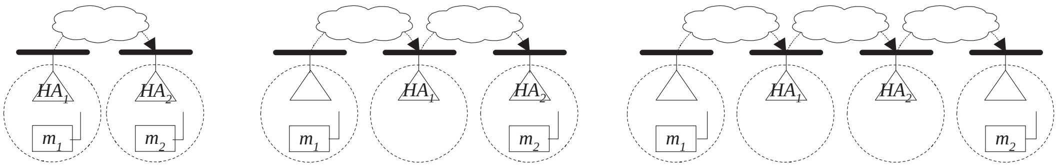
$$E[A] = L + \frac{e^{N_m(1+\eta L)} - 1}{N_m\eta}.$$

Note that such tunneling only occurs between adjacent subnetworks and its latency differs from X_1. Let X_2 be the mean latency of the tunneling. The multicast latency of our dynamic route approach, T_D, is

$$T_D = X_1 + \frac{U}{E[A]}X_2. \qquad (3)$$

6.4. Numerical results

The parameters used in our analytical model are as follows: $R = 3000$; $r = 25$; $X_1 = 0.3$ (s); $X_2 = 0.1$ (s);



(a) Case 1 (b) Case 2 (c) Case 3

FIGURE 8. Packet delivery across the Internet.

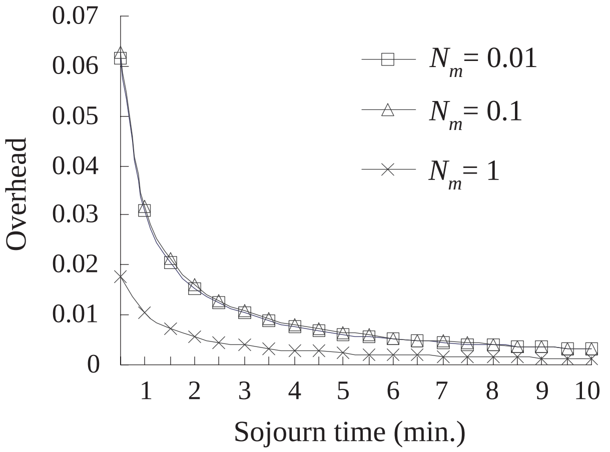


FIGURE 9. Route reconstruction overhead.

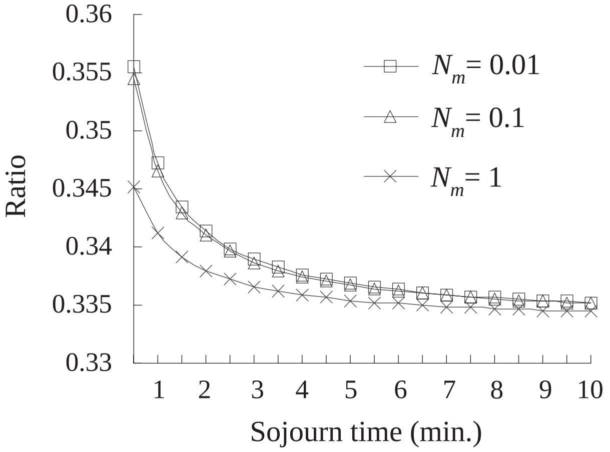


FIGURE 10. Comparisons of multicast latencies.

$U = 10$ (s). The numerical results are shown in the context of three N_m values, namely 0.01, 0.1, 1, respectively. The horizontal axis of the subsequent figures, denoted as *sojourn time*, represents the average time for an MN to dwell in a subnetwork. A short sojourn time means that MNs are highly mobile.

Figure 9 demonstrates our route reconstruction overhead, in comparison with the necessary cost to maintain multicast trees in Chikarmane's approach. It can be seen that the overhead is insignificant, below 7%. In particular, as $N_m = 1$, the overhead almost approaches to zero. This phenomenon might be justified as follows. Since $N_m = 1$, each subnetwork is expected to accommodate an MN. In this case, a subnet is very likely to remain active for a long time and thus the multicast routes are reconstructed rarely.

The multicast latencies of the two subject schemes are depicted in Figure 10. The vertical axis is T_D/T_C, i.e. the ratio of the multicast latency required in our approach to that in Chikarmane's scheme. It can be evidenced that our approach outperforms the counterpart. The ratio T_D/T_C is around 0.34. When the MNs' sojourn time becomes sufficiently long, MNs behave as if stationary hosts and the multicast latency is nearly constant. Note that T_C, as formulated in Equation 2, is dominated by the probability that an MN is located at its home subnetwork, rather than the sojourn time.

7. CONCLUSION

Dynamically reconstructing multicast routes according to mobile host locations incurs costly overhead because incorrect routes require modifications throughout the multicast backbone. To avoid this overhead, all previous work maintained multicast routes statically irrespective of the MNs' locations. However, these schemes lead to inefficient datagram delivery, in that multicasts to or from MNs that are away from their home must be routed indirectly by way of their respective HAs. This may cause unacceptably long communication latencies, wasted network bandwidth and overload the networks along the delivery path.

This paper has presented a framework for routing multicast datagrams to recipients efficiently, using network-level protocols in the TCP/IP suites as basic building blocks. We allow multicast routes to be dynamically adapted to MNs' locations, while maintaining low cost. This is achieved by partitioning the mobile environment into regions, so that most of the location changes due to MNs' movements are hidden from other regions. As a result, when an MN moves intra-regionally, the multicast route change only affects that region, rather than the whole Internet. This scheme is considerably beneficial when MNs are roaming within a region.

To accomplish efficient routing, in our scheme, local MAs encapsulate and re-send multicast datagrams originated from foreign MNs. Datagrams are thus forwarded normally by the multicast routing protocol. A destination MA decapsulates such datagrams and then delivers them to the local MN receivers.

When an MN moves to an inactive subnet, the MN may experience a delay during which group communications cannot proceed. To solve this, we set up a temporary tunnel between the MN's previous MA and the current agent. The previous agent re-directs the multicast datagrams in flight to the new locations. Such tunneling remains until a timer expires.

We have developed an analytical model to quantify our framework in terms of two performance indices: route reconstruction overhead and multicast latency. Numerical results show that our scheme, compared with the best known proposal, reduces the average multicast latency by more than 66%, while causing less than 7% overhead. The preliminary evaluation results show that our scheme is promising. We are proceeding to implement this framework in real systems.

ACKNOWLEDGEMENTS

The authors sincerely appreciate Dr William Fenner, Tom Maufer and anonymous referees for their valuable comments and assistance. This work has been supported by the National Science Council, ROC, under grants NSC88-2213-E009-001 and NSC88-2213-E009-081.

REFERENCES

[1] Birman, K. P., Schiper, A. and Stephenson, P. (1991) Lightweight causal and atomic group multicast. *ACM Trans. Comp. Syst.*, **9**, 272–314.

[2] Acampora, A. S. and Naghshineh, M. (1994) An architecture and methodology for mobile-executed handoff in cellular ATM networks. *IEEE J. Selected Areas Commun.*, **12**, 1365–1375.

[3] Mysore, J. and Bharghavan, V. (1997) A new multicasting-based architecture for Internet host mobility. In *Proc. 3rd Ann. ACM/IEEE Conf. Mobile Comp. and Networks*, New York, 26–30 September, pp. 161–172.

[4] Seshan, S., Balakrishnan, H. and Katz, R. H. (1996) Handoffs in cellular wireless networks: the Daedalus implementation and experience. *Kluwer Int. J. Wireless Comm. Syst.*, **4**, 141–162.

[5] Acharya, A., Bakre, A. and Badrinath, B. R. (1996) IP multicast extensions for mobile internetworking. In *IEEE INFOCOM'96*, Piscataway, NJ, pp. 67–75.

[6] Chikarmane, V., Bunt, R. B. and Williamson, C. L. (1995) Mobile IP-based multicast as a service for mobile hosts. In *Proc. 2nd IEEE Int. Workshop Services in Distributed and Network Environments*, pp. 11–18. IEEE Computer Society Press, Los Alamitos, CA.

[7] Chikarmane, V., Williamson, C. L., Bunt, R. B. and Mackrell, W. L. (1999) Multicast support for mobile hosts using mobile IP: design issues and proposed architecture. *ACM/Baltzer Mobile Networking Applications*, **3**, 365–379.

[8] Harrison, T. G., Williamson, C. L., Mackrell, W. L. and Bunt, R. B. (1997) Mobile multicast (MoM) protocol: multicast support for mobile hosts. In *Proc. 3rd Annual ACM/IEEE Conf. Mobile Comp. and Networks*, New York, 26–30 September, pp. 151–160.

[9] Perkins, C. (1998) *Mobile IP Design Principles and Practices.* Addison-Wesley Longman, Reading, MA.

[10] Fenner, W. (1997) Internet group management protocol, Version 2, *RFC 2236*, Network Working Group (Updates *RFC 1112*).

[11] Pusateri, T. (1999) Distance vector multicast routing protocol, Internet Draft, <draft-ietf-imdr-dvmrp-v3-08>, Network Working Group (Updates *RFC 1075*).

[12] Ballardie, A. *et al.* (1993) Core based trees (CBT)—an architecture for scalable inter-domain multicast routing. *Comp. Commun. Rev.*, **23**, 85–95.

[13] Deering, S. E., Estrin, D., Farinacci, D., Jacobson, V., Liu, C.-G. and Wei, L. (1994) An architecture for wide-area multicast routing. *Comp. Commun. Rev.*, **24**, 126–135.

[14] Moy, J. (1994) Multicast extensions to OPSF, *RFC 1584*, Network Working Group.

[15] Xylomenos, G. and Polyzos, G. C. (1997) IP multicast for mobile hosts. *IEEE Commun. Mag.*, **35**, 54–58.

[16] Ioannidis, J., Duchamp, D. and Maguire, Jr G. Q. (1991) IP-based protocol for mobile internetworking. In *SIGCOMM'91, Conf. Commun. Architectures and Protocols*, Zurich, Switzerland, 3–6 September, pp. 235–245.

[17] Cobb, J. A. and Gouda, M. G. (1997) The request reply family of group routing protocols. *IEEE Trans. Comp.*, **46**, 659–672.

[18] Dijkstra, E. W. (1975) Guarded commands, nondeterminancy and formal derivation of programs. *Commun. ACM*, **18**, 453–457.

[19] Deering, S. E. (1991) Multicast routing in a datagram internetwork. Ph.D. Thesis, Stanford University.

[20] Thyagarajan, A. S. and Deering, S. E. (1995) Hierarchical distance-vector multicast routing for the MBone. *Comp. Commun. Rev.*, **25**, 60–66.

[21] Kleinrock, L. (1975) *Queueing Systems, Vol. I: Theory.* Wiley, New York.

[22] Ross, S. M. (1996) *Stochastic Processes*, 2nd edn. Wiley, New York.

APPENDIX

A.1. Distribution of participant population

Since MNs could enter or leave a subnetwork independently at will, the arrivals and departures of MNs, with respect to a subnetwork, could be treated as Poisson processes [21, p. 61]. Therefore, we model the MN population distribution in a subnet using an $M/M/\infty$ queue. Such a population is Poisson distributed with the equilibrium probability π_k, the probability that there are k MNs present on a subnet, as follows [21, p. 234]:

$$\pi_k = \frac{N_m^k}{k!}\, \mathrm{e}^{-N_m},$$

where N_m is the expected number of MNs located on a subnetwork. In particular, the probability of an idle subnet is

$$\pi_0 = \mathrm{e}^{-N_m}. \tag{4}$$

An MN dwells in a subnet for a period of time and moves away (into another subnetwork.) The sojourn time for each MN is exponentially distributed with parameter η. That is,

η is the MN's mobility rate; $1/\eta$ is the mean sojourn time. Let λ be the MNs' aggregate arrival rate at a subnet. If, on average, a subnet accommodates N_m members, then by Little's rule, we have $\lambda = N_m\eta$.

A.2. Life cycle of a recipient subnetwork

Let random variables B and I denote the durations of a busy period and an idle period, respectively. A subnet behaves like an alternating renewal process, as shown in Figure 7, where a renewal corresponds to the start of a busy period. When a busy period terminates, a new idle period begins. This idle period will terminate immediately upon the arrival of an MN member. The time until the next MN's arrival is exponentially distributed. Let F_I denote the distribution function of that subnet's idle period. Then

$$F_I(t) = P[I \le t] = 1 - \mathrm{e}^{-\lambda t}, \quad t \ge 0,$$

where λ is the MNs' arrival rate for that subnet. The average idle time for a subnet, $E[I]$, is thus $1/\lambda$. Since F_I is non-lattice, by Theorem 3.4.4 in [22], we have the relation

$$\pi_0 = \frac{E[I]}{E[B] + E[I]}.$$

Consequently, the average renewal period is

$$E[B] + E[I] = \frac{E[I]}{\pi_0} = \frac{\mathrm{e}^{N_m}}{\lambda}. \tag{5}$$

Since $E[I] = 1/\lambda$, it follows that $E[B] = (\mathrm{e}^{N_m} - 1)/\lambda$.

A subnet's life cycle is defined to be an active period and the following inactive period. The life cycle containing k pairs of busy and idle periods is geometrically distributed with probability mass function $(P[I \le L])^{k-1}P[I > L]$. $P[I > L]$ is the probability that *no* MNs arrived during the first L time in an idle period and is equal to $\mathrm{e}^{-\lambda L}$. Thus we have

$$P[k \text{ renewals}] = (1 - \mathrm{e}^{-\lambda L})^{k-1}\,\mathrm{e}^{-\lambda L}. \tag{6}$$

Additionally with π_0 and $P[I > L]$, we derive P_{ia}, the probability that a subnet is found to be inactive.

$$P_{ia} = \pi_0 P[I > L] = \mathrm{e}^{-N_m(1+\eta L)}. \tag{7}$$

As shown in Figure 7, a subnetwork could be viewed to experience MNs' arrivals generated by two streams. One generates Poisson arrivals marked with '$*$' at rate $\pi_0\lambda$, while the other generates Poisson arrivals without any marks at rate $(1-\pi_0)\lambda$. An arrival marked with '$*$' initiates a renewal. A life cycle Y containing k renewals is known to be Erlang distributed with probability density function

$$f(y \mid k \text{ renewals}) = \frac{(\pi_0\lambda)^k y^{k-1}}{(k-1)!}\,\mathrm{e}^{-(\pi_0\lambda)y}.$$

Therefore

$$\begin{aligned} E[Y \mid k \text{ renewals}] &= \int_0^\infty y f(y \mid k \text{ renewals})\,\mathrm{d}y \\ &= \frac{k}{(\pi_0\lambda)}. \end{aligned} \tag{8}$$

Note that $1/(\pi_0\lambda)$ is the mean inter-arrival time for '$*$' marks and such an inter-arrival time consists of a busy period and an idle period. Thus $1/(\pi_0\lambda) = E[B] + E[I]$, which agrees with Equation (5).

From Equations (6) and (8), it follows that

$$\begin{aligned} E[Y] &= \sum_{k=1}^{\infty} E[Y \mid k \text{ renewals}]P[k \text{ renewals}] \\ &= \frac{\mathrm{e}^{N_m+\lambda L}}{\lambda}. \end{aligned} \tag{9}$$

Furthermore, the mean length of an active period A is

$$\begin{aligned} E[A] &= E[B] + L \\ &\quad + \sum_{k=1}^{\infty} P[k \text{ renewals}](k-1)(E[B]+E[I]) \\ &= L + \frac{\mathrm{e}^{N_m(1+\eta L)} - 1}{N_m\eta}. \end{aligned} \tag{10}$$