Computer Communications 24 (2001) 486–495

computer communications

www.elsevier.com/locate/comcom

A new locality-based IP multicasting scheme for mobile hosts

Chien-Chao Tseng*, Kuang-Hwei Chi, Ting-Lu Huang

Department of Computer Science and Information Engineering, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 30010, Taiwan, ROC

Received 1 February 2000; accepted 13 July 2000

Abstract

Internet protocol multicasting is a mechanism widely used in the Internet to disseminate information. Multicast delivery paths may need to be frequently restructured along with host migrations, at the expense of nontrivial overhead or multicast latency. This paper presents a scheme with short multicast latency, while retaining acceptable overhead in adapting the delivery paths to mobile host locations. We exploit the locality in host movements and keep active the networks that mobile hosts have most recently visited. As long as hosts migrate within these active networks, multicast delivery paths can remain unchanged upon host movements. Performance results show that, compared with the best known proposal, our scheme is promising. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Mobile network; Multicast; TCP/IP; Mobile Internet protocol; Locality

1. Introduction

Internet protocol (IP) multicasting provides a useful facility that disseminates information from a single site to a set of destinations. Typical applications include stock trading, teleconferencing, coordinating updates to replicated file systems [6] etc. Given the prevalence of portable computing devices, and the current trend towards ubiquitous computing, mobile users should be capable of accessing information on the Internet anywhere and anytime.

A mobile environment allows hosts to roam around freely while retaining networking connectivity over wireless media. Host mobility, however, challenges multicasting in this environment: since a multicast route locates all of the participants in a group, the established route may need to frequently adapt to mobile users' locations, at the expense of nontrivial overhead or communication delay. In addition, because multicast delivery paths are likely to become obsolete upon host migrations, mobile hosts could experience significant packet loss. Lost datagrams are retransmitted if required, causing a waste of network bandwidth and a burden to upper layer protocols.

This paper presents an efficient approach to delivering multicast messages in mobile internetworks, while retaining acceptable multicast route adjustment costs. We exploit the locality in host movements and keep the networks that mobile hosts visited most recently *active* on a multicast route. As long as hosts migrate within these active networks, the delivery paths need not change in the event of host mobility. As a host moves to a network without group members, multicasts for the host may be disrupted transiently because new routes are under construction. To avoid this, a temporary unicast tunnel is set up for the host between the point of attachment to the previous network and that to the new network. The previous network will redirect a copy of each multicast datagram in flight to the new location, until a designated timer expires.

The rest of this paper is organized as follows. Our system model and previous research are presented in the Section 2. Section 3 describes our scheme, and Section 4 presents an analytical model to evaluate our proposal and a counterpart scheme. Numerical results are shown in Section 5. Section 6 concludes this paper.

2. Preliminaries

2.1. System model

We adopted the mobile IP networking architecture specified in [20, pp. 129–166]. (Fig. 1) A mobile node (MN) is a host or a router that may move around while retaining access to the Internet over a wireless medium. Each MN has a permanent IP address, namely *home address*, on its home network. On a local network, there is a router called mobility agent, or agent for short, that acts as a point of attachment to the system for mobile nodes. An agent is

---

* Corresponding author.

E-mail addresses: cctseng@csie.nctu.edu.tw (C.-C. Tseng), chikh@csie.nctu.edu.tw (K.-H. Chi), tlhuang@csie.nctu.edu.tw (T.-L. Huang).

PII: S0140-3664(00)00297-8

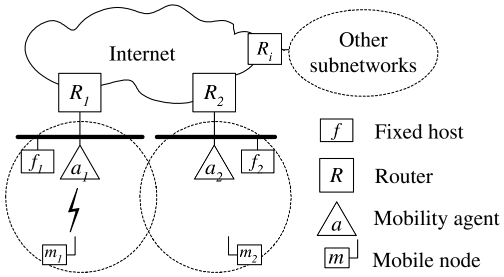


Fig. 1. Mobile internetworking architecture.

said to be home to a mobile node if the agent has a network prefix matching that of the MN's home address, or a foreign agent otherwise. Upon detecting movement into a new network, an MN registers at the local agent, with the information about its home agent and its previous agent. The new agent then informs the two entities of the MN's migration. For brevity a (wireless) subnetwork hereinafter refers to a local network, an administered area of a mobility agent.

To deliver datagrams to a group, multicast-capable routers, namely *multicast routers*, are introduced for group management and Internet-wide delivery service. Multicast routers learn which groups have members on each of their attached subnetworks, using the Internet Group Management Protocol (IGMP) [12]. A multicast router keeps a list of multicast group memberships for each attached subnetwork. Routers periodically transmit query messages in their respective administered areas. In response, local hosts send a report message for each group to which they belong, within a specified period. Notice that when a host first joins a group, it transmits a report for the group rather than waiting for a router query. Conversely, when a host wishes to leave a group, if it was the last host that replied to an IGMP query with a membership report for that group, it initiates a *leave-group* message. This causes the local multicast router to send a group-specific query onto the subnetwork. If no reports are received within 10 s, the router removes the group and will not forward remotely originated multicasts for that group onto the subnetwork.

In conjunction with IGMP, multicast routers construct delivery paths and forward datagrams towards group members across internetworks. Several routing protocols are in common use or in some stage of development [2]. For the rationale and deployment of such protocols, we refer the reader to [17]. Effectively a route in the form of a spanning tree is established to locate all participant hosts of a group. However, some dense mode protocols, like the Distance Vector Multicast Routing Protocol (DVMRP) [21] or Multicast Open Shortest Path First (MOSPF) [18], build a delivery tree for *each* source subnetwork per group. Joining a group means that a user must be *grafted* into the multicast tree(s). The existing routing protocols are capable of seamlessly providing both join (graft) and leave (prune) functions [3].

2.2. Issues and related work

In a mobile environment multicast delivery paths tend to be transient in nature and may need to adapt accordingly when participants move. It is nontrivial to restructure the paths along with host movements all the way because a large number of multicast routers could be involved for update, especially when dense mode protocols are employed.

Most of the previous schemes avoid adjusting multicast routes to group members' locations, by hiding host mobility from multicast tree constructions. In the approach by Acharya [1], datagrams are delivered using campus-wide broadcast. Therefore MNs are able to receive multicasts without modifying multicast routes upon each move. As a framework designed for an obsolete mobile IP [13], this scheme is less popular.

One of the two options proposed in the current mobile IP standard is that an MN on a foreign subnetwork uses a *co-located* care-of address to conduct multicast communication [20, pp. 119–120]. In this scenario the system could suffer from frequent multicast route reconstructions. Alternatively, the other proposed option builds multicast routes as if each MN were always situated at its home subnetwork. Each MN maintains a bi-directional tunnel with its home agent via which multicasts are sent and received. As a consequence, multicast datagrams will be delivered as unicast packets to each mobile host that is away from home, even though multiple mobile hosts served by a home agent happen to visit a common subnetwork. Duplicate multicast messages will arrive at that foreign subnetwork, causing unnecessary network load.

Similar to the second multicasting option in Mobile IP, Chikarmane et al. [9] presented a scheme that maintains delivery paths on the basis of MNs' home agents, while with an important distinction as follows. When a home agent serves the mobile hosts of a given group at several foreign subnetworks, it tunnels only one copy of the received multicast datagrams to each such foreign subnetwork. Link-level multicast is used by the local agents at these subnetworks to complete the last-mile delivery. When the MNs of different home agents are attached to a particular foreign agent, multiple tunnels will terminate at that foreign agent. To avoid duplicate multicasts being directed to that subnetwork in this case, the foreign agent designates one of these MN home agents to forward multicast traffic. Other non-designated home agents suppress traffic re-direction. As an optimization of Mobile IP multicasting, Chikarmane's scheme is very general: it uses home agents to accommodate mobile host memberships and supports administratively scoped (private) multicasts on home networks. The performance evaluation of this scheme is approached by simulations in [24].

A flaw in the aforementioned research is the maintenance of multicast routes regardless of the MNs' whereabouts. Message exchanges among MNs that are away from home

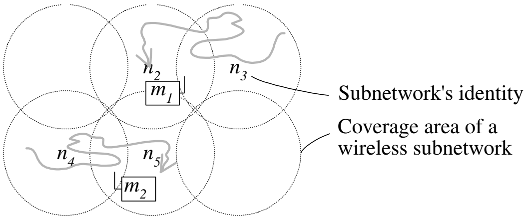


Fig. 2. A locality resulting from host migrations.

can traverse a long delivery path. This may increase the communication delay, waste network bandwidth, and place a burden on the network entities along the delivery paths. As a remedy, we propose a cost-effective approach allowing for the dynamic adjustment of multicast trees to mobile hosts' locations. Remarkably our scheme is independent of the multicast routing protocol in use, to be described below.

3. The proposed approach

We exploit the locality property in mobile host migrations. As evidenced in previous research such as [11,14], most of the MNs usually possess locality in their mobility behaviors — a subnetwork visited by an MN tends to be revisited in the near future.

3.1. Overview

In this text, a *locality* is defined as the set of subnetworks that were visited by the mobile hosts of a group during the most recent Δt time units. Fig. 2 illustrates an example of the locality $\{n_2, n_3, n_4, n_5\}$ for a group of members m_1 and m_2 at some time instant. We say that a subnetwork is *active* on a multicast route if it is included in the route, or *inactive* otherwise. In our approach, each subnetwork in a locality is kept active for Δt time units after the corresponding group of MNs has moved off. If a host of the group moves into such a subnetwork before Δt expires, this subnetwork is revived. Otherwise the subnetwork will be dropped from the locality. Hence a locality specifies an area where the hosts of a multicast group are able to migrate without altering delivery paths. This benefit is effective unless the locality changes, upon which the multicast delivery paths are adjusted accordingly.

3.2. Realization

A simple way to capture the notion of localities is to let mobility agents learn local MN group memberships and perform IGMP reports for Δt time units on behalf of those MNs that have moved away. This will effectively hold the subnetworks in a locality from being pruned from multicast delivery paths for a period of time.

Each mobility agent keeps track of local MN group memberships, using a triple $\langle g, \Delta t, mn\_ids \rangle$ per group g, where *mn_ids* denotes a set of local MNs in group g (Fig. 3a). Initially, when an MN, say m_i, requests to join group[1] g toward the local multicast router, this request is also overheard by the local mobility agent. Given that group g is new to the agent, a corresponding triple is created locally. Then m_i is added to the *mn_ids* field. Instantly the agent sets its network interface to recognize the multicast group address, namely joining group g simultaneously along with m_i.

Whenever MN m_i moves and changes its attached agent, say from a_j to a_k, the associated membership information with m_i will be handed from a_j to a_k locally. Fig. 3b illustrates the handoff procedure, where a_h is m_i's home agent and the dashed line represents the exchange of Registration Request and Registration Reply messages defined by the mobile IP. To begin, a_k sends a_j a Membership Binding message, indicating the identities of groups to which a_k currently belongs. In response, a_j returns a Binding Reply message, containing the group identities for m_i. Then a_k immediately joins those groups present in the Binding Reply message, but absent from the Membership Binding message, in the same way as a normal host joins a group. Meanwhile a_j redirects multicast datagrams in flight to a_k (thereby to m_i) for the new groups that a_k joins, until a designated timer expires.

This handoff procedure is inspired from [20, pp. 129–166]. We extend the use of *forwarding pointers* to forward multicast traffic in flight to the new agent of m_i. Such a technique, called *tunneling* in the literature, forwards the traffic to each applicable MNs using unicast. In this manner, m_i is able to receive timely multicasts in the course of rebuilding multicast delivery paths.

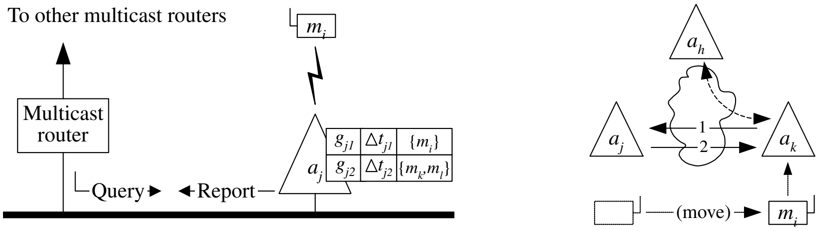
Whenever all MNs of a group have moved off a subnetwork, the local mobility agent determines the value of Δt in the group's triple and will issue IGMP reports for the group for Δt time units (a method for choosing Δt is proposed in Section 4.1). If any host of the group arrives before Δt has expired, this subnetwork is revived. The local agent is suppressed from originating IGMP reports, since the host will do that. On the other hand, if none of the group returns before Δt time units have elapsed, the local mobility agent sends a *leave-group* message to the local multicast router, to depart from the group. As a result, the subnetworks that the MNs of a group visited most recently are kept active as the group locality.

3.3. Multicast routing

The foregoing treatment is independent of the multicast routing protocols in use. To avoid ambiguity, each fixed or mobile host should originate multicast packets using its own home address as the IP source address so that recipients can distinguish the multicast senders. For this, we need to support multicast delivery from visiting mobile nodes for

---

[1] This can be an IGMPv3's Report message indicating its interest in receiving multicast traffic from a specific source [7].



(a) Group management on a subnetwork (b) Message flow in handoff procedure

Fig. 3. Maintenance of locality information.

those routing protocols, such as DVMRP or MOSPF, that forward multicasts according to the datagram source address. We could use the IP-in-IP encapsulation technique [19] as follows. As shown in Fig. 4, the local mobility agent intercepts and encapsulates the multicast datagrams whose IP source addresses are foreign to the subnetwork, and re-sends these datagrams using its own IP address. Here the encapsulation of a datagram means that an outer IP header is inserted in front of the original datagram. 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, packets will be routed along the established paths toward the downstream subnetworks and routers, hop by hop, to group members.

On receiving a multicast datagram forwarded to a subnetwork, the local mobility agent examines whether or not the datagram has been encapsulated, i.e. whether an integer 4 has been assigned to the Protocol Number field in the datagram's IP header. If so, the outer IP header is removed and the resultant datagram is forwarded to the intended MNs on the subnetwork, over wireless media.

Here mobility agents are assumed to be multicast-aware so that multicast datagrams originating from a visiting MN will not be dropped but processed for outgoing delivery. Besides, fixed hosts are assumed to know IP-in-IP so as to determine whether they should decapsulate the received multicast datagrams beforehand.

3.4. Remarks and discussion

Our scheme, in addition to saving the overhead of rebuilding multicast routes, reduces packet losses experienced by migrant hosts. Within active subnetworks, the hosts are able to receive multicasts without disruption from each of these subnetworks locally. Even as a host moves to a previously inactive subnetwork, packet forwarding for the host is activated. In contrast, those schemes using home agents to join delivery trees might route multicast datagrams in flight to incorrect destinations in the event that mobile nodes have changed locations and their respective serving home agents are not yet aware since the registration messages are underway. As a remedy, a packet forwarding technique could also be employed in those approaches. Despite this, those schemes will still encounter packet loss, if network-layer handoff is not carefully implemented [15,23]. Packet loss reduction will maintain communication efficiency, free higher layer protocols from re-transmitting lost datagrams, and avoid a significant waste of network bandwidth in case of applications with streaming data.

The locality property in host movements has been studied considerably, such as in [11,14]. Most of the previous research used this property to track the location or to reduce the registration costs of mobile nodes. In this study we deploy this property orthogonally to maintain multicast routes. The group management protocol software performed on mobility agents is the entity to be modified.

Our scheme can be employed to reduce the excessive cost caused by frequent host movements [8], especially with inter-regional mobility. A region refers to a cluster of subnetworks and routers encompassed by an enterprise or a campus. As an example, assume that a group locality includes subnetworks n_2 and n_3, in Fig. 2 that belong to two different regions. Although the MNs of the group make inter-region moves between n_2 and n_3, multicast routes, once established, need not change as [8] does.

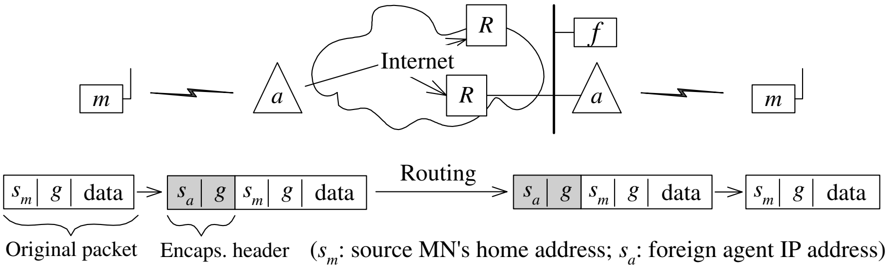


Fig. 4. A paradigm of multicast routing.

In our architecture a subnetwork without group members remains active for up to extra Δt time units. Multicast traffic directed to such subnetworks during this period is wasted. Indeed, yet another form of wasting network bandwidth is shown in previous schemes like in [9,10,20], in the sense that they may use longer paths to deliver multicast datagrams. We will investigate the network bandwidth demanded by these schemes in the next section.

4. Performance analysis

The choice of Δt is important to our scheme; the longer Δt is, the fewer multicast routes are reconstructed, yet the more multicast traffic is likely to be wasted for delivery to subnetworks without recipients. Given a multicast arrival rate at a subnetwork, we suggest a proper value for Δt such that the cost of rebuilding multicast routes and wasted network traffic is minimized. Then our scheme, using the suggested Δt, is compared with a counterpart approach in terms of message cost and communication delay. Here we choose Chikarmane's framework [9,10] as the base for comparison, since it outperforms other peer schemes in these dimensions. Note that all of the performance measures in this section describe the time averaged (long term) behavior of a single multicast group. A group containing fixed hosts benefits our scheme in that the multicast delivery paths are less volatile.

4.1. Selection of Δt

Consider subsequently a subnetwork in the system. Since the MNs could enter or leave the subnetwork independently at will, the arrivals and departures of MNs could be treated as Poisson processes [16, p. 61]. Sojourn time for MNs in a subnetwork can have a general distribution with mean $1/\mu$, where μ is called the mobility rate in some literature. Let λ be the MNs' aggregate arrival rate at a subnetwork. Provided that a subnetwork accommodates an average of n MNs, then by Little's rule, we have $\lambda = n\mu$. In practice, the rates at which local hosts, fixed or mobile, join and leave a group can be incorporated into λ and μ, respectively.

We say that a subnetwork is *busy* for the group under discussion if it accommodates at least one member of the group, or *idle* otherwise. As an idle period starts, this period will terminate immediately upon the arrival of an MN member. Since the time until the next MN's arrival is exponentially distributed, it follows that

$$\text{Prob[idle period duration} \le y] = 1 - \mathrm{e}^{-\lambda y}, \qquad y \ge 0$$

and the probability density function is $p(y) = \lambda \mathrm{e}^{-\lambda y}$.

Now let us investigate the extra traffic induced by a subnetwork in our scheme. As mentioned before, multicast routes are reconstructed unless an idle subnetwork turns busy within Δt time units. We represent the reconstruction cost as the number of packets, γ, to be generated, for removing the idle subnetwork from and re-incorporating the subnetwork later into multicast routes. Letting θ denote the multicast packet arrival rate at a subnetwork, we can express the ensemble *overhead traffic* as a function

$$G(\Delta t) = \int_0^{\Delta t} \theta y p(y)\,\mathrm{d}y + \int_{\Delta t}^{\infty} \gamma p(y)\,\mathrm{d}y = \frac{\theta}{\lambda} + \left(\lambda - \frac{\theta}{\lambda} - \theta\Delta t\right)\mathrm{e}^{-\lambda\Delta t} \qquad (1)$$

The first integral indicates the number of multicast datagrams received by an idle subnetwork when the subnetwork's idle period, y, is shorter than the chosen value of Δt. The second integral above states that, in case the idle period y exceeds Δt, γ packets are produced to adjust the multicast delivery paths.

Observe that γ may vary vastly in different multicast routing protocols, and is related to the topology of the Internet multicast backbone, the locations of group members, whether newly arrived MNs send multicasts, etc. In particular, a multicast-sender move may cause a significant amount of traffic for dense mode protocols, for example, incurring DVMRP broadcast-and-prune operations. Namely, the first few multicast packets from the source subnetwork will be delivered to all other DVMRP subnetworks, and those routers that do not lead to the group member are then pruned off the delivery tree. Aside from counting messages, protocol processing overhead on multicast routers can also be weighted as a part of γ.

Our goal is to find Δt such that $G(\Delta t)$ is minimized. For this, differentiating $G(\Delta t)$, we have

$$G'(\Delta t) = (\theta\Delta t - \gamma)\lambda\,\mathrm{e}^{-\lambda\Delta t}$$

and

$$G''(\Delta t) = \left(\gamma + \frac{\theta}{\lambda} - \theta\Delta t\right)\lambda^2\,\mathrm{e}^{-\lambda\Delta t}$$

When $G'(\Delta t) = 0$, it follows that $\Delta t = \gamma/\theta$. Since $G''(\gamma/\theta) > 0$, $G(\gamma/\theta)$ is minimal. In order to decide whether $G(\gamma/\theta)$ is also minimum, we need to examine $G(0)$, since $G(\Delta t)$ is indifferentiable when Δt is 0, where the minimum may take place. If $G(\gamma/\theta) < G(0) = \gamma$, then we set Δt to γ/θ. Otherwise, the overhead traffic could be prohibitively heavy and thus Δt is suggested to be *zero*. Lastly, substituting the value of Δt as above into Eq. (1), we measure the overhead traffic in our scheme. In other words

$$\Delta t = \begin{cases} \dfrac{\gamma}{\theta} & \text{if } \dfrac{\theta}{\lambda}(1 - \mathrm{e}^{(-\lambda(\gamma/\theta))}) < \gamma \\ 0 & \text{otherwise} \end{cases}$$

and $G(\Delta t) = \min\{\gamma, (\theta/\lambda)(1 - \mathrm{e}^{(-\lambda(\gamma))/\theta})\}$.

4.2. Message cost

This subsection addresses the number of multicast messages resulting from our approach and from Chikarmane's scheme [9,10], respectively. As before, we

Table 1
Network parameters

Notation	Definition	Value
n	Expected number of group members on a subnetwork involved in multicasting	0.1, 0.5, 1.0, 2.0
γ	Message count of adjusting multicast routes	500, 1000, 2000, 4000
θ	Average number of packets directed to a subnetwork per second	10
d	Packet delivery time through the Internet	0.3 s [11]
v	Timeout value for tunneling messages in our scheme	10 s
ϵ	Probability that an MN dwells at its home subnetwork	0.3 [24], 0.7 [11]

restrict attention to a subnetwork. Traffic induced by other subnetworks can be obtained similarly using the method introduced here. A subnetwork changes its idle or busy state alternatively, like a renewal process. We will measure the message count of interest in the two subject schemes during a renewal period, E[busy time] + E[idle time].

To see the expectation of renewal periods, we model a subnetwork using the $M/G/\infty$ queue in which host arrivals are Poisson distributed with parameter λ and mean service time is $1/\mu$. Then, as noted in [16, p. 234], we have

$$\text{Prob}[i \text{ members on a subnetwork}] = \frac{(\lambda/\mu)^i}{i!}\,\mathrm{e}^{-\lambda(1/\mu)}$$

Therefore the probability of finding none on a subnetwork is

$$\text{Prob}[\text{a subnetwork is idle}] = \mathrm{e}^{-\lambda(1/\mu)}$$

Moreover, because the distribution of a renewal period is non-lattice, by Theorem 3.4.4 of [22], it follows that

$$\text{Prob}[\text{a subnetwork is idle}] = \frac{E[\text{idle time}]}{E[\text{busy time}] + E[\text{idle time}]}$$

Since E[idle time] is equal to $1/\lambda$, these last two equations lead to

$$E[\text{busy time}] + E[\text{idle time}] = \frac{\mathrm{e}^{\lambda(1/\mu)}}{\lambda}$$

and

$$E[\text{busy time}] = \frac{\mathrm{e}^{\lambda(1/\mu)} - 1}{\lambda}$$

Now with a multicast arrival rate of θ, we have a message count in our scheme of $\theta \times E$[busy time] plus our overhead traffic, i.e.

$$\frac{\theta}{\lambda}\,\mathrm{e}^{\lambda(1/\mu)} + \left(\gamma - \frac{\theta}{\lambda} - \theta\Delta t\right)\mathrm{e}^{-\lambda\Delta t}$$

For Chikarmane's scheme, message counts are examined in two separate cases. First, when a subnetwork is home to *no* MN in the group under discussion, the subnetwork will receive multicasts during its busy period, hence $(\theta/\lambda)\times(\mathrm{e}^{\lambda(1/\mu)} - 1)$ messages. Second, if a subnetwork is home to *some* MN in the group, then the subnetwork will receive multicasts during the group lifetime, thus $(\theta/\lambda)\mathrm{e}^{\lambda(1/\mu)}$ packets during a renewal period.

4.3. Communication delay

We define communication delay as the averaged message delivery time between two MNs in a group, across internetworks. Assume that the delivery time between any pair of hosts has the same distribution with mean d.

Chikarmane's scheme needs $d + (1 - \epsilon)d + (1 - \epsilon)d$ of communication delay, where ϵ denotes the probability of an MN being home. The first term, d, accounts for the case where these two mobile nodes are home. The second and the third terms, respectively, indicate the extra message delivery time from a host that is away from home to its home agent, and from a recipient home agent to a corresponding node served.

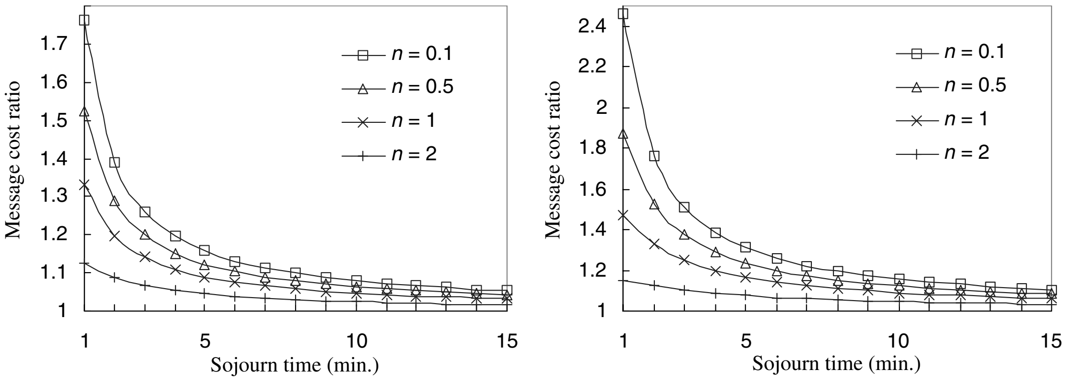
In our approach, mobile nodes are normally straightaway reachable from multicast routes, so a datagram could be delivered to the intended group members within d time. However, an exception arises as a destination MN moves to a previously inactive subnetwork. In the very beginning when the subnetwork turns active, multicast packets for the MN need to be delivered indirectly via its previous mobility agent. Such indirect delivery of messages continues until a timer of v seconds expires, thus with the probability v/E[active time]. E[active time], the expectation of a subnetwork's active time, is derived below. To summarize, the communication delay of our scheme is $d + (v/E[\text{active time}])d$.

A subnetwork in our architecture changes its active or inactive state alternatively. Specifically in terms of the state transitions, a subnetwork behaves as if it were the $M/G/\infty$ queue where MNs' arrival rate is still λ and service time averages $(1/\mu) + \Delta t$. Again, from [16, p. 234] and Theorem 3.4.4 of [22], we have the equality

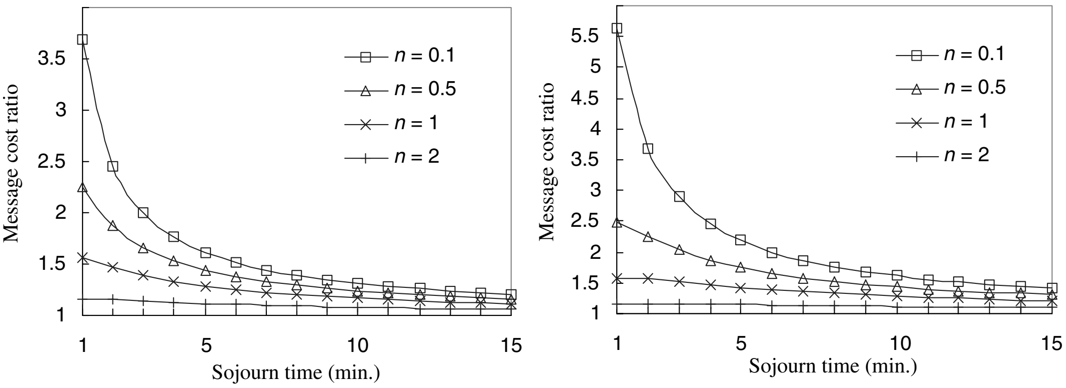
$$\text{Prob}[\text{a subnetwork is inactive}] = \mathrm{e}^{-\lambda((1/\mu)+\Delta t)}$$

$$= \frac{E[\text{inactive time}]}{E[\text{active time}] + E[\text{inactive time}]} \qquad (2)$$

Observe that when a subnetwork starts inactive, this period will terminate immediately upon the arrival of the next mobile node. Since the time until the next MN has *memoryless* distribution, we have $E[\text{inactive time}] = 1/\lambda$. This



(a) $\gamma = 500$ (b) $\gamma = 1000$



(c) $\gamma = 2000$ (d) $\gamma = 4000$

Fig. 5. Comparison in message count (the subnetwork is home to no MN in a given group).

results in

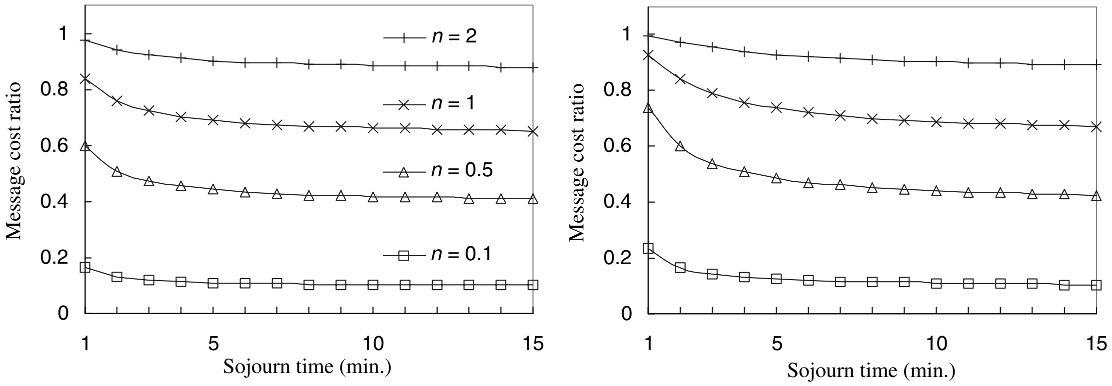
$$E[\text{active time}] = \frac{1}{\lambda}(e^{\lambda((1/\mu)+\Delta t)} - 1).$$

5. Numerical results

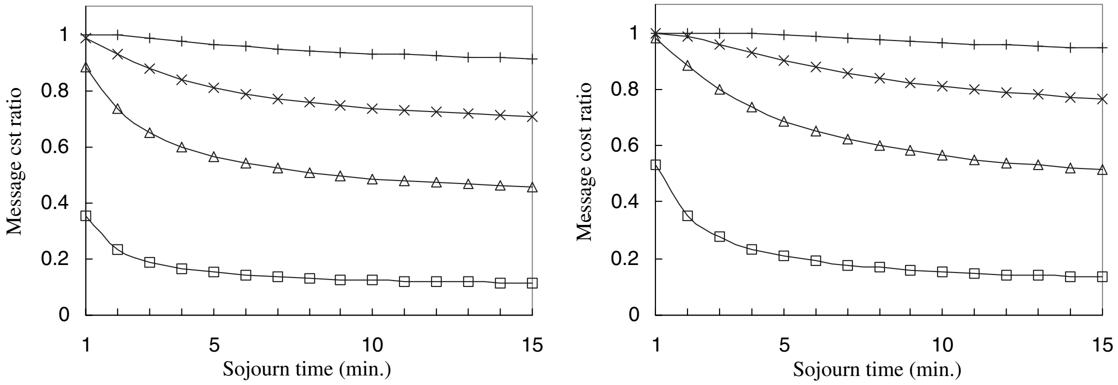
This section demonstrates the evaluation results of the proposed approach in comparison with Chikarmane's scheme under a parameter setting summarized in Table 1. To see the relative performance, metrics of interest are depicted as a normalized form, where the numerator represents the corresponding measure for our approach and the denominator the counterpart measure for Chikarmane's scheme. The horizontal axis of all subsequent figures represents MNs' sojourn time, denoted as $1/\mu$ in this text, within a subnetwork.

To our best knowledge, there are no proper reference materials for determining parameters n, λ, θ, and v in our model. For a better understanding of their effects on system performance, on principle a wider range or a worse-case (with respect to our scheme) of these values is chosen. For example, we consider different ns to correspond to the cases where group members are sparsely or densely distributed in a mobile environment. For γ, a light value refers to a setting where maintaining multicast routes upon host movements is not costly. In contrast, an extremely large value, 4000, corresponds to a case where DVMRP is deployed in the system and a multicast sender move incurs broadcast-and-prune operations. This γ value could be already large enough, assuming hundreds of DVMRP networks on today's multicast backbone [5]. Next, to decide θ we continue tracking an MBone session of multicast address 224.2.172.238 as the representative, the largest observed group so far in the *Mlisten*-collected statistics [4]. From the statistics, the number of packets received by each member during the period it joins the session is known. This could give us an overestimate of θ. Lastly consider v in our scheme. Since the packet delivery time through the Internet is 0.3 s [11], setting v to 10 s will suffice for a round-trip delay to ensure that graft messages are reliably processed.

Fig. 5 compares message costs for the two subject schemes, provided that the subnetwork under discussion is home to no MN in a multicast group. It can be seen that Chikarmane's scheme outperforms ours in this case, while the outperformance reduces as the MN sojourn time increases. This is because, with longer sojourn time, MNs tend to behave like stationary hosts and therefore we have



(a) $\gamma = 500$ (b) $\gamma = 1000$



(c) $\gamma = 2000$ (d) $\gamma = 4000$

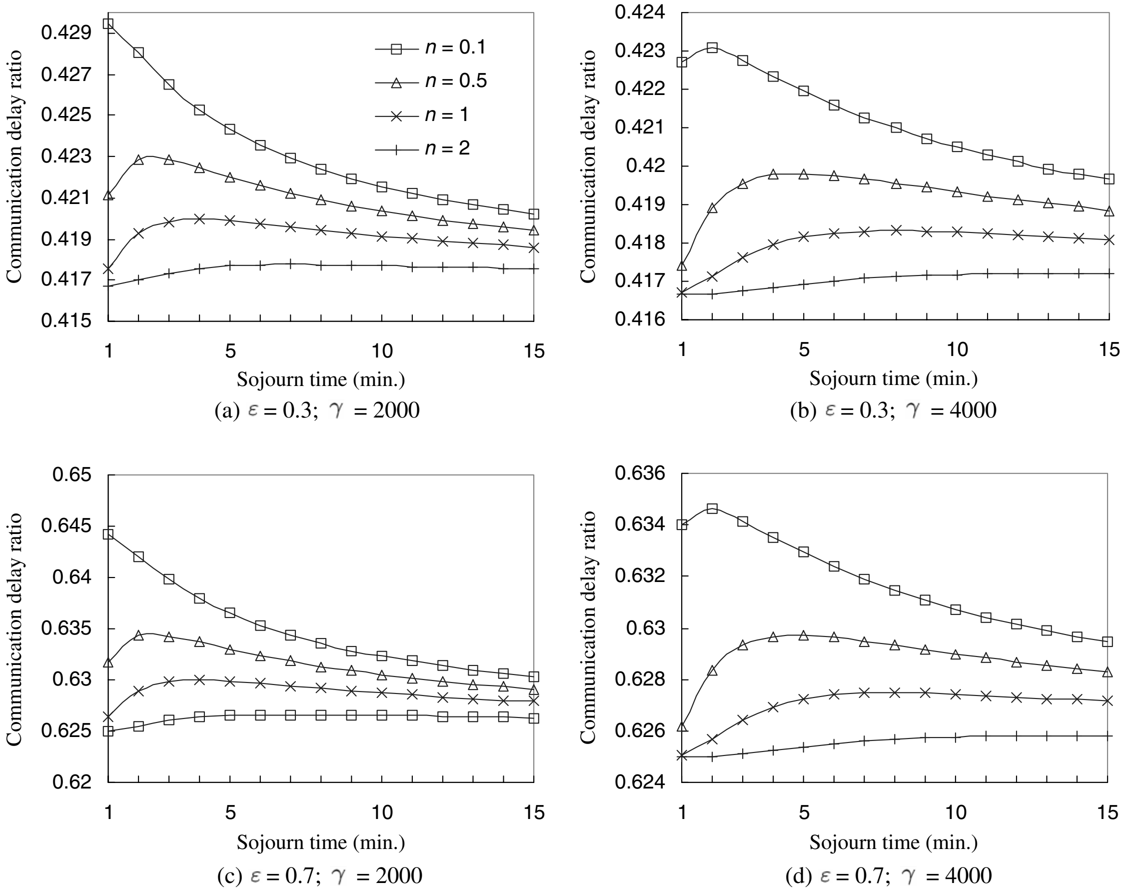
Fig. 6. Comparison in message cost (in case the subnetwork is home to some mobile node in the group).

less overhead traffic for adjusting multicast delivery paths. In particular, when n is 0.1, our message cost appears unacceptably high. This corresponds to the subnetwork rarely accommodating group members but is still included in the group's locality. We believe that such a scenario is uncommon, as reasoned in the following. A locality can be viewed as a group of mobile hosts' favorite area, where they are likely to reside. This implies that a large portion of the subnetworks in the locality could usually see the group member(s). In other words, a comparatively larger number of subnetworks, on average, could accommodate more MNs than n being 0.1. Thus the high message cost induced by a single subnetwork with rare group membership could only impose a light load on the overall system.

Fig. 6 demonstrates that, when a subnetwork is home to an MN of a given group, our approach will cost fewer messages than the counterpart scheme does. Recall that the measure refers to counting the messages delivered to or from the system by a subnetwork during a renewal period, a busy period plus the succeeding idle period. In Chikarmane's model, the subnetwork will receive multicast messages throughout the renewal period, regardless of whether the mobile nodes of the group are locally present. We treat those messages directed to the idle subnetwork as additional traffic. Such traffic resulting from Chikarmane's scheme is shown to exceed the overhead traffic for our scheme due to delayed changes in multicast routes.

As another remark on Fig. 6, the message cost ratios for n being 0.1, 0.5, and 1 are smaller than that for n being 2, as the MN sojourn time increases. One can expect that a subnetwork with less population remains idle most of the time during a renewal period. And, as Little's rule $\lambda = n\mu$ suggests, the subnetwork's idle period $(1/\lambda)$ increases along with the MN sojourn time $(1/\mu)$, while inversely with n. In consequence, Chikarmane's approach results in more and more multicast traffic directed to an idle subnetwork with fewer group members. From this figure, such additional messages in the case of a lower n are shown to outnumber those in the case of n being 2.

Due to space limitations, in the following we provide figures only under high γ traffic, a condition disadvantageous to us. From Fig. 7 it can be seen that we demand shorter multicast latency than the counterpart scheme does by at least 36%. Further, the smaller ϵ is, the greater performance we gain in this dimension. A higher ϵ means that the mobile nodes are often located at their home networks. This circumstance is advantageous to Chikarmane's approach, in that multicast messages can be delivered to and from MNs without redirections, saving communication delay.



(a) $\varepsilon = 0.3$; $\gamma = 2000$

(b) $\varepsilon = 0.3$; $\gamma = 4000$

(c) $\varepsilon = 0.7$; $\gamma = 2000$

(d) $\varepsilon = 0.7$; $\gamma = 4000$

Fig. 7. Communication delay versus mobile nodes' sojourn time.

6. Conclusions

This paper presented a scheme, backward compatible with the TCP/IP protocol suites, for delivering multicast messages in a mobile internetworking environment. To lower the overhead of adjusting multicast delivery paths to mobile host locations, we exploited the locality property in host migrations. We kept active those subnetworks that mobile hosts visited most recently. In this manner, as long as hosts roam within these active networks, the established delivery paths need not change along with host movements.

The length of Δt is important to the performance of our scheme. In Section 4.1, we suggested a value for Δt such that the overhead for rebuilding multicast routes and network traffic waste is minimized. Further, an analytical model was developed to evaluate our multicasting scheme using the suggested Δt. In comparison with the best known proposal, numerical results show that our scheme demands remarkably shorter communication delay, while at the expense of small traffic overhead. This indicates that our scheme is promising.

This study was inspired from an observation that host movements exhibit temporal locality. Occasionally host mobility is random. For cost and performance considerations, we would avoid keeping active those subnetworks that are located on MNs' random moving paths. This can be achieved as follows. When a subnetwork starts busy, it determines whether it *should* become active on multicast delivery paths. We averaged the length of a subnetwork's idle periods in the past and then applied the technique presented in Section 4.2 to derive the expected message cost. If our scheme outnumbers Chikarmane's approach, say by 30%, then the subnetwork is not incorporated into multicast routes. In this way only those subnetworks at which MNs dwell long enough are kept active in the system. Messages intended for a mobile member in an inactive subnetwork will require tunneling from somewhere in the set of active subnetworks. The detailed protocol as well as the related issues require thorough future investigation.

Acknowledgements

This study is supported by the National Science Council, ROC, under grants NSC88-2213-E009-081 and NSC89-2213-E009-028. We thank Dr. W. Fenner, from AT&T Research in Menlo Park, and T. Maufer, from 3Com, and anonymous referees for their valuable help and comments.

References

[1] A. Acharya, A. Bakre, B.R. Badrinath, IP multicast extensions for mobile internetworking, IEEE INFOCOM'96, 1996, pp. 67–75.

[2] K.C. Almeroth, The evolution of multicast: from the MBone to Inter-domain multicast to Internet2 deployment, IEEE Network 14 (1) (2000) 10–20.

[3] K.C. Almeroth, A long-term analysis of growth and usage patterns in the multicast backbone (MBone), Proc. IEEE 2 (2000) 824–833 (INFOCOM 2000, 19th Annual Joint Conf. IEEE Comput. Commun. Soc.).

[4] K.C. Almeroth, M.H. Ammar, Mlisten: MBone collection tool, http://www.cc.gatech.edu/computing/Telecomm/mbone/.

[5] K.C. Almeroth et al., The Mantra Project, http://imj.ucsb.edu/mantra. For DVMRP information, see http://imj.ucsb.edu/mantra/geographic/geo-java/geo-plot-dvmrp-networks.html.

[6] K.P. Birman (Ed.), Building Secure and Reliable Network Applications Manning, 1996.

[7] B. Cain, A. Thyagarajan, S.E. Deering, Internet group management protocol, version 3, Internet Draft ⟨draft-ietf-idmr-igmp-v3-00.txt⟩, 1997.

[8] K.H. Chi, C.C. Tseng, T.L. Huang, A framework for mobile multicast using dynamic route reconstructions, The Comput. J. 42 (6) (1999) 522–533.

[9] V. Chikarmane, R. Bunt, C. Williamson, Mobile IP-based multicast as a service for mobile hosts, Proceedings of the 2nd IEEE International Workshop on Services in Distributed and Network Environments, 1995, pp. 11–18.

[10] V. Chikarmane, et al., Multicast support for mobile hosts using mobile IP: design issues and proposed architecture, ACM/Baltzer Mobile Network. Appl. 3 (4) (1999) 365–379.

[11] G. Cho, L.F. Marshall, An efficient location and routing scheme for mobile computing environment, IEEE J. Select. Areas Commun. 13 (5) (1995) 868–879.

[12] W. Fenner, Internet group management protocol, version 2, RFC 2236, Network Working Group, 1997.

[13] J. Ioannidis, D. Duchamp, G.Q. Maguire, Jr., IP-based protocol for mobile internetworking, SIGCOMM'91, 1991, pp. 235–245.

[14] R. Jain, et al., A caching strategy to reduce network impacts of PCS, IEEE J. Select. Areas Commun. 12 (8) (1994) 1434–1444.

[15] M. Khalil, et al., Buffer management for mobile IP, Internet Draft ⟨draft-mkhalil-mobileip-buffer-00.txt⟩, 1999.

[16] L. Kleinrock, Queueing Systems, vol. I, Wiley, New York, 1975.

[17] T.A. Maufer (Ed.), Deploying IP Multicast in the Enterprise, Prentice-Hall, Upper Saddle River, NJ, 1998.

[18] J. Moy, Multicast extensions to OSPF, RFC 1584, Network Working Group, 1994.

[19] C. Perkins, Encapsulation within IP, RFC 2003, Network Working Group, 1996.

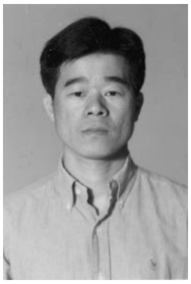
[20] C. Perkins (Ed.), Mobile IP Design: Principles and Practices, Addison-Wesley Longman, Reading, MA, 1998.

[21] T. Pusateri, Distance vector multicast routing protocol, Internet Draft, ⟨draft-ietf-imdr-dvmrp-v3-08⟩, 1999.

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

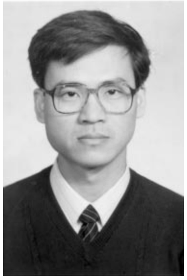
[23] C.H. Tsai, K.H. Chi, C.C. Tseng, Integration of network and data-link layer protocols for mobile internetworking, Proceedings of the 5th Workshop on Mobile Computing, Taichung, Taiwan, 2000, pp. 83–88.

[24] C.L. Williamson, et al., Performance evaluation of the Mom mobile multicast protocol, ACM/Baltzer Mobile Network Appl. 3 (4) (1998) 189–201.



Chien-Chao Tseng *is currently a professor in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Hsin-Chu, Taiwan. He received his BS degree in Industrial Engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981; MS and PhD degrees in Computer Science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. His research interests include Mobile Computing, and Wireless Internet.*

Kuang-Hwei Chi *received his BS degree in Computer Science and Information Engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1991 and his MS degree in Computer Science and Information Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1993. He is currently a PhD candidate in the Department of Computer Science and Information Engineering from National Chiao-Tung University. He is a member of the ACM.*



Ting-Lu Huang *studied at Tung-hi University (BS, 1976), University of Texas at Arlington (MS, 1981), and Northwestern University (PhD, 1989). He is currently an associate professor in the Department of Computer Science and Information Engineering at National Chiao-Tung University. He has been working on mutual exclusion algorithms, causality in distributed computing, testing and debugging of concurrent software, and verification.*