Mobile Networks and Applications 9, 379–391, 2004

A Two-Tier Heterogeneous Mobile Ad Hoc Network Architecture and Its Load-Balance Routing Problem [*]

CHI-FU HUANG, HUNG-WEI LEE and YU-CHEE TSENG [**]
Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu 30050, Taiwan

Abstract. The *mobile ad hoc network (MANET)* has attracted a lot of interest recently. However, most of the existing works have assumed a stand-alone MANET. In this paper, we propose a two-tier, heterogeneous MANET architecture which can support Internet access. The low tier of the network consists of a set of mobile hosts each equipped with a IEEE 802.11 wireless LAN card. In order to connect to the Internet and handle the network partitioning problem, we propose that the high tier is comprised of a subset of the mobile hosts, called *gateways*, which can access to cellular/infrastructure networks. The high tier is heterogeneous in the sense that the network interfaces in the gateway hosts could be IEEE 802.11 cards, PHS handsets, or GPRS handsets characterized by different bandwidths and latencies. Observing that the gateways could become the bottlenecks of the two-tier network, we propose a set of solutions, namely *boundary-moving*, *host-partitioning*, and *probabilistic* solutions, to solve the load-balance routing issue. Implementation issues/concerns of these schemes are discussed. Simulation results are presented to compare these load-balance routing schemes.

Keywords: ad hoc network, load balance, mobile computing, routing, wireless network

1. Introduction

Wireless communications and mobile computing have attracted a lot of attention recently. Wireless communication devices, such as IEEE 802.11 WLAN cards, Bluetooth, and PHS/GPRS phone card, are becoming popular or even standard equipments in portable computing devices. People can carry these devices while traveling around to enjoy the tremendous services on the Internet and live an easier life. Mobility has added a new dimension to the area of computing and communications.

An emerging wireless network architecture is the *mobile ad hoc network (MANET)*, which can be flexibly and conveniently deployed in almost any environment without the need of infrastructure base stations. A MANET is one consisting of a set of mobile hosts, which can roam around at their own will. Since no base stations are supported in such an environment, hosts may have to communicate with each other in a *multi-hop* manner. Applications of MANETs occur in situations like battlefields, disaster areas, and outdoor assemblies. A working group called "manet" has been formed by the Internet Engineering Task Force (IETF) to study the related issues and stimulate research in MANET. Intensive research has been devoted to MANET recently [7,11,17,20].

In the development of MANET, we have observed that most of the existing works have assumed a *stand-alone* MANET. While this is acceptable, we feel that it would be more attractive if one can simultaneously enjoy both the flexibility provided by MANET and the tremendous services/data provided in the Internet. The latter would not be possible without connecting the MANET to the Internet. While so doing, we should still maintain a high level of mobility for MANET. References [5] and [22] propose two different architectures, which integrate the concepts of MANET and cellular networks. In [5], an architecture called *Multihop Cellular Network (MCN)* is proposed to extend the service range of a base station by incorporating the flexibility of MANET. MCN can reduce the required number of base stations and improve the network throughput. In [22], an *Integrated Cellular and ad hoc Relaying Systems (iCAR)* is proposed to dynamically relay traffic from one cell to another by using ad hoc technologies. The iCAR system can balance traffic loads between cells. It can also increase system capacity, reduce power consumption, and extend system coverage. The integration of MANET and Mobile IP is addressed in [21], where it is proposed to extend the capability of access points (such as IEEE 802.11 access points) to support multiple MANETs. Each access point serves as a foreign agent (FA) in Mobile IP and has its own serving range, which can be dynamically changed. Two MANETs can partially overlap with each other in their serving ranges. Mobile hosts can access the Internet through FAs by proper registrations through Mobile IP messages. From Mobile IP's prospective, FAs' service ranges can be extended to multiple hops away from access points. From MANET's prospective, mobile hosts can immediately enjoy tremendous resources already existing in the Internet through Mobile IP.

Due to the advancement of cellular networks, the architecture discussed in [21] can be further extended to a more flexible mobile network architecture. In this paper, we propose a two-tier, heterogeneous MANET architecture which has Internet access capability. The low tier of the network consists of a set of mobile hosts each equipped with a wireless

---

[*] This work is co-sponsored by the MOE Program for Promoting Academic Excellence of Universities under grant numbers A-91-H-FA07-1-4 and 89-E-FA04-1-4, and by the Chung-Shan Institute of Science and Technology under contract number BC93B12P.

[**] Corresponding author.

802.11 wireless LAN card. In order to connect to the Internet and handle the network partitioning problem, we propose that the high tier is comprised of a subset of the mobile hosts, called *gateways*, which can access cellular/infrastructure networks. The high tier network is heterogeneous in the sense that the network interfaces in the gateway hosts could be IEEE 802.11 cards, PHS handsets, or GPRS handsets characterized by different bandwidths and latencies. It is clear nowadays that cellular networks have much wider coverage and longer transmission distances, but much lower communication bandwidths, compared to IEEE 802.11-based networks. Adopting cellular networking interfaces on the high tier would not reduce the mobility of MANET, but would improve MANET's connectivity to the Internet.

The two-tier network architecture can potentially be applied to many Internet Service Provision (ISP) business scenarios. For example, in a train, we can install in each trunk a router, which has a GPRS/PHS interface to connect to the cellular network and an IEEE 802.11 interface to provide services to passengers. Passengers can connect to the Internet via their own IEEE 802.11 interfaces to the trunk's IEEE 802.11 interface, and then to the GPRS/PHS interface and the Internet. In this case, the router in each trunk can be regarded as a *roaming router*. In addition, the multiple roaming routers in the train will allow passengers to freely roam around inside the train. This conforms to the proposed two-tier architecture. Similar scenarios may happen in cars, buses, airplanes, etc.

A key issue in the proposed two-tier architecture is how to utilize the bandwidths provided by the high-tier gateways efficiently. First, the high-tier links provided by cellular networks typically have much narrower bandwidths compared to low-tier links. Second, since a high-tier interface will have to serve multiple low-tier hosts, the traffic concentration effect can easily saturate the former. Load-balance routing is required for hosts to share the gateways. In addition, factors such as host mobility, which may change the members of a gateway dynamically, and heterogeneity of gateways have to be considered. All these motivate us to study this challenging, but attractive, issue. In this paper, we will propose three sets of solutions, each characterized by different computational and communication overheads, and implementation difficulty. The simulation results are also reported.

The rest of this paper is organized as follows. In section 2, we describe the system model with a formal problem statement. Then, the three sets of solutions, namely *boundary-moving*, *host-partitioning*, and *probabilistic* solutions, are proposed in sections 3, 4, and 5, respectively. Section 6 reports our simulation results. Section 7 concludes this paper.

2. System model

In this section, we first define our two-tier heterogeneous MANET architecture, followed by the load-balance routing problem under such an environment. We consider a set S of mobile hosts, each equipped with a broadband, short-distance, wireless LAN card, such as IEEE 802.11. The network formed by the mobile hosts constitutes the low tier of the two-tier MANET, and these hosts working in ad hoc mode may communicate with each other in a multi-hop manner.

In order to connect to the Internet, a subset S' of S, called *gateway hosts*, are each equipped with an extra network interface connecting to the cellular/infrastructure network. The network formed by the gateway hosts constitutes the high tier of the two-tier MANET. The proposed two-tier architecture is illustrated in figure 1. These high-tier network interfaces are heterogeneous in the sense that they could be a mixture of PHS handsets, GPRS handsets, or again IEEE 802.11 interfaces, which are characterized by different bandwidths and latencies. For example, a GPRS channel can in theory support a data rate up to 115.2 Kbps, while a PHS handset can support a data rate of 128 Kbps with 1 channel, and 384 Kbps with 3 channels. A gateway host with an extra IEEE 802.11 card is a special case. It is considered as a router with two IEEE 802.11 interfaces. The low-tier interface should be set to the ad hoc mode to support routing in the MANET, while the high-tier interface should be set to the infrastructure mode to connect to an IEEE 802.11 access point. An IEEE 802.11b interface can support rates of 11/5.5/2/1 Mbps, while an IEEE 802.11a interface can support rates of 54/48/36/24/18/12/9/6 Mbps.

Although intra-MANET communications are possible, we assume that the two-tier architecture is mainly used to support Internet access. A mobile host without a high-tier link can access the Internet through one or more low-tier links leading to a gateway host. Therefore, the Internet service ranges of gateway hosts are extended through ad hoc links. The mobility management problem can be handled by Mobile IP [10,12,23]. However, since Foreign Agents (FAs) in Mobile IP can only serve hosts that are one-hop away, it has to be extended to support mobile hosts that are several hops away from FAs. This issue have been addressed in [21], where the required modifications on the agent-advertisement, agent-solicitation, and registration messages are discussed.

In addition to providing the Internet access capability, the high tier also has the advantage of solving the network partitioning problem. Due to mobility, a MANET can get easily partitioned into multiple segments. Normally, this issue is unsolvable in traditional ad hoc routing, and is concluded as a route-unreachable problem. Through the connections provided by the Internet infrastructure and the high-tier interfaces, which have much longer communication distances, two hosts resident in separated MANET segments can remain connected.

2.1. Problem statement

A key issue in the proposed two-tier MANET architecture is how to utilize the bandwidths provided by the high-tier gateways efficiently. To connect to the Internet, each low-tier host should choose a high-tier gateway as its serving gateway. Load-balance routing is required for hosts to share bandwidths of high-tier gateways. There are several concerns to be addressed. First, the high-tier links formed by PHS/GPRS handsets have much narrower bandwidths compared to the low-tier links. Second, even for a gateway host that can con-

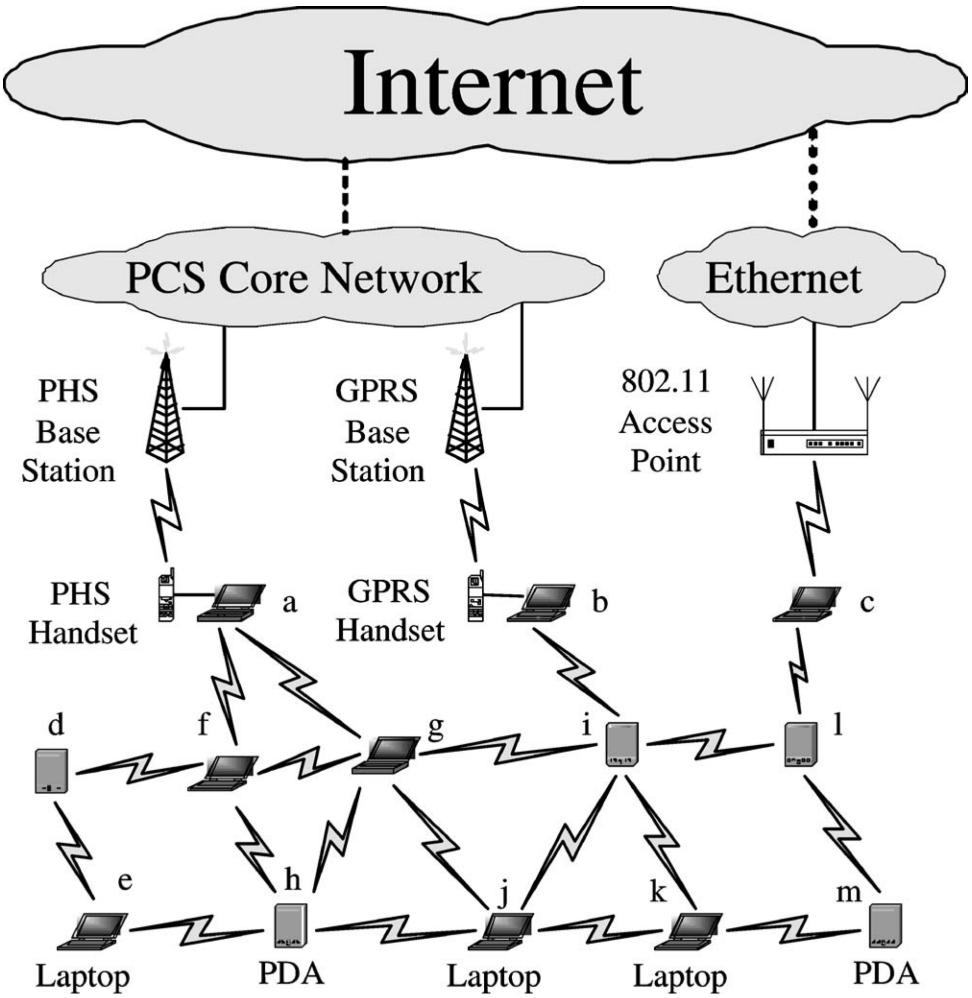


Figure 1. An example of the two-tier heterogeneous mobile ad hoc network architecture.

nect to an IEEE 802.11 access point, the traffic concentration effect, which is due to multiple non-gateway hosts connecting to this gateway, can easily saturate the gateway. Third, considering host mobility, the members served by a gateway may change dynamically, and so does the aggregated traffic load from these members. Fourth, as noted earlier, gateway hosts are heterogeneous, which further increases the difficulty of this problem.

To solve the load-balance routing problem, we model the two-tier network by a node-weighted graph as follows. Each host $x \in S$ is translated into a node denoted by (x, T_x), where T_x is the traffic index of x. The traffic index represents the traffic load between x and the Internet. The traffic indices of hosts may differ from time to time. Note that we mainly consider the two-tier architecture as an Internet access environment, and thus do not take intra-MANET communications into account. For any two hosts which have a link on the low tier, we introduce an edge between them. For each gateway host $y \in S'$ with a high-tier interface of capacity C_y, we introduce another node (y, C_y) to the graph with an edge connecting to its corresponding node (y, T_y) on the low tier. That is, each gateway host will introduce two nodes in the graph.

An example is shown in figure 2, which models the network described in figure 1. Each gateway host introduces a triangle node (high-tier) and a circle node (low-tier), while each non-gateway host introduces a circle node. Gateway hosts are sinks for Internet traffics in the network. The traffics of other hosts need to be drained out to proper sinks with load-balance concerns in mind. One possible solution is shown in figure 2, where hosts are separated, based on shortest-path routing, into three groups to be served by gateways a, b, and c, respectively. Such arrangement is obviously imbalanced in most cases. Moreover, gateway c, which has the largest bandwidth, is responsible for the least number of hosts. It is possible that gateway a has run out of all its bandwidth, while c still has extra unused bandwidth. In subsequent sections, we will propose several schemes to solve this problem.

Without loss of generality, we assume that the network is connected. Otherwise, either some hosts are not connected to any gateway, or the problem can be divided into multiple sub-problems that can be solved independently.

To evaluate how balanced a routing scheme can achieve, we define the concept of *load-balance index (LBI)* as follows. Let $x \in S'$ be any gateway host. The *load index (LI)* of a

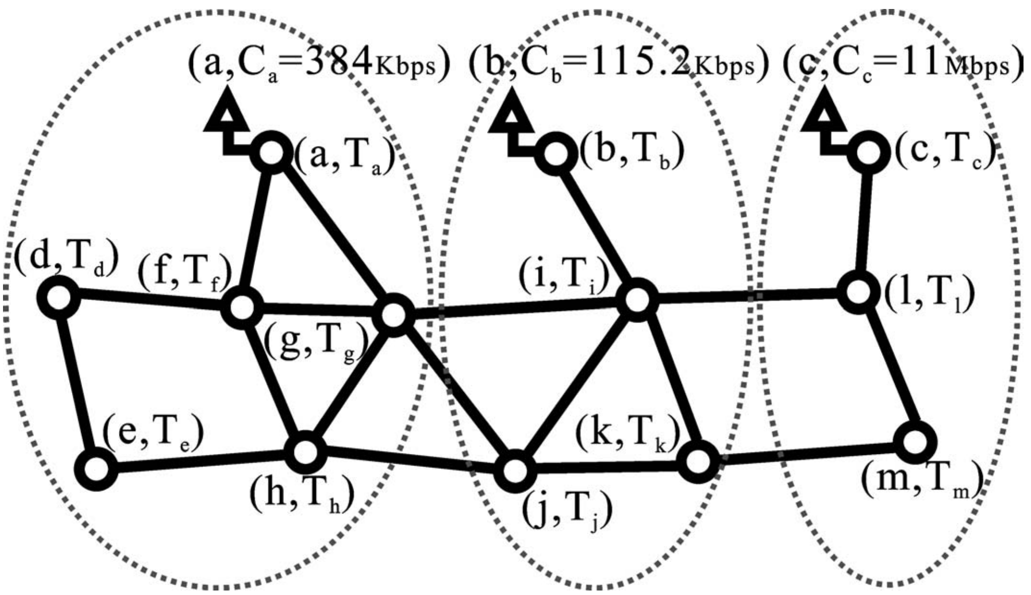


Figure 2. Modeling the two-tier network in figure 1 by a graph, where high-tier/low-tier interfaces are represented by triangles/circles.

gateway x is defined to be

$$L_x = \frac{\sum_{i \in S} \rho_{i_x} \cdot T_i}{C_x},$$

where ρ_{i_x} is the fraction of i's traffic that is sent to gateway x. That is, L_x is the ratio of the traffic load to the bandwidth of the high-tier interface for gateway x. The *load-balance index* of the network is defined to be

$$LBI = \frac{\max\{L_x\} - \min\{L_x\}}{\max\{L_x\}}.$$

We use *LBI* to judge how balanced the routing is. Note that $0 \leqslant LBI \leqslant 1$ and a lower *LBI* implies a better balanced situation.

Our goal is to reduce the *LBI* of the whole network. In the following sections, we will propose three types of solutions, each characterized by different computational and communications overheads, and implementation difficulties. We comment that when a gateway runs out of its capacity, i.e., its $LI \geqslant 1$, two cases may happen. If certain bandwidth-guaranteed routing mechanism is supported, then some fair scheduling or flow control algorithms (such as [6,9,18]) have to be run at the gateway to share its bandwidth to its members. If such a routing mechanism can not be supported (which is the case in most Internet environments), then the higher-layer transportation protocol, such as TCP, will do the flow control to slow down packet transmission. This is similar to reducing the traffic index T_x of a node (x, T_x) when a node x experiences TCP flow control. This may result in dynamic changes of hosts' traffic indices. In this case, the perceived *LI* at the gateway will be close to 1 (but never exceed 1) and there is no guarantee how much share each connection can get. For the latter case, our goal is to exploit load balance at the gateway level by fully utilizing the bandwidths provided by gateways.

3. Boundary-moving solutions

In this type of solutions, some boundaries will be established between gateways. These boundaries define the service range of each gateway host for low-tier hosts to route their traffics. Two solutions are proposed. The first one is derived based on routing distances, while the second one is derived based on gateways' loads.

3.1. Shortest-path (SP) routing

We divide hosts by a nearest-neighbor rule as follows. For any two distinct gateways $p, q \in S'$, the *dominance of p over q* is defined to be

$$dom(p, q) = \{x \mid x \in S, \delta(x, p) \leqslant \delta(x, q)\},$$

where $\delta()$ is the distance function between two hosts. The service region of a gateway $p \in S'$, denoted by $reg(p)$, is the subset of hosts lying in every dominance of p over all other gateways:

$$reg(p) = \bigcap_{q \in S' - \{p\}} dom(p, q).$$

The above definition is equivalent to the case where each host chooses the closest gateway, in terms of hop count, as its serving gateway. When there is a tie, the host is in the service regions of all its closest gateways. Figure 3 shows an example, where the thin dashed lines are the boundaries separating the service regions of three gateways.

Next, we discuss the implementation issues of this approach. Each gateway should advertise, through broadcast, its existence to its direct neighbors periodically. Initially, each low-tier host has no serving gateway. On receipt of an advertisement, it sets the advertising gateway as its serving gateway. After having a serving gateway, a host has to

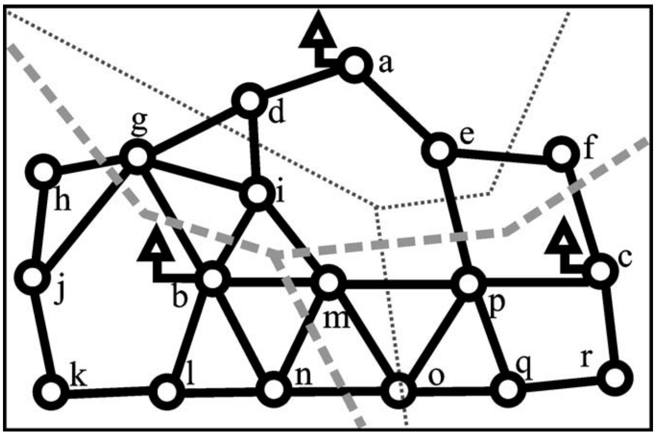


Figure 3. The service regions of gateways determined by the SP routing (thin dashed lines) and MLI routing (thick dashed lines).

help rebroadcast the advertisements for the gateway by increasing the hop-count information in the advertisements by one. A host changes its serving gateway only if it receives an advertisement from a different gateway with a smaller hop count than its current one. To calculate hop counts, the advertisement messages should contain a hop-count field. Since the network is targeted at providing Internet access, it is reasonable to assume that each host adopts the TCP/IP protocols. To facilitate the IP packet forwarding, each host should set its neighbor which leads to its serving gateway with the minimum hop count as its *default gateway*. Any packet with an unknown destination will be forwarded to the default gateway.

It is apparent that this scheme can not achieve the goal of load-balance routing in many cases. First, the traffic demands from hosts are not necessarily equal. Second, even if the demands are equal, hosts may not be evenly distributed to all gateways. This can be better explained by the concept of "Voronoi diagram" in computer geometry [1]. Given a plane and a set P of arbitrary points on the plane, the Voronoi diagram partitions the plane into a number of segments such that each segment contains exactly one point, say p, in P, and the segment contains all points that are closer to p than to any other point in P. Intuitively, the Voronoi diagram contains a number of boundaries that are perpendicular bisectors of neighboring points in P that divide the plane. An example is shown in figure 4. Imagining points in P as gateways, we see that gateways in the outer part of the plane are likely to be responsible for larger areas compared to those in the inner part. This explains why load balance is difficult to achieve by the SP routing.

3.2. Minimum load-index (MLI) routing

In this scheme, the boundaries between gateways will be adjusted dynamically by taking the load indices of gateways and the traffic loads of hosts into account. This scheme is fully distributed and is run by each host independently. Each gateway g will periodically broadcast advertisement messages containing its current load index (L_g). Each low-tier host x should keep a record of the load information of its current serving gateway. When x hears an advertisement from g, the following rules are executed:

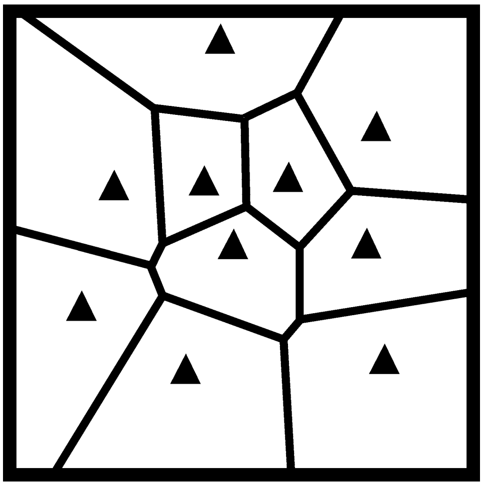


Figure 4. An example of the Voronoi diagram in a plane of 10 points.

1. If x currently has no serving gateway, it chooses g as its serving gateway by recording g's current load information and setting the host leading to g with the shortest distance as its default gateway. Then x rebroadcasts the advertisement.
2. If g is current x's serving gateway, x records g's load index and rebroadcasts the advertisement.
3. If g is different from x's current serving gateway, say g', then x checks if x has accepted g' as its serving gateway for over a time threshold τ and $(L_{g'} - T_x/C_{g'})/(L_g + T_x/C_g) \geqslant \Delta_l$, where Δ_l is a predefined gateway-switching load threshold. If the check passes, x will accept g as its new serving gateway with a gateway-switching probability P_{MLI}.

The above steps are similar to a diffusion procedure. A gateway with a lower traffic load will extend its service range, while one with a higher load will shrink its service range. Note that in the above rule 3, we do not require x to rebroadcast the advertisement, no matter x accepts g as its new gateway or not. We do this on purpose so as to achieve a slow diffusion. Otherwise, a gateway with a very low load may be quickly saturated by too many hosts and this may lead to a fluctuating situation. As a result, the service ranges of gateways will be extended at most one hop in each advertisement. Δ_l is designed with the similar purpose. To slow down the diffusion process, we further guard a host from switching to new gateway with a probability P_{MLI} even if the check passes. The MLI routing can lead to a more load-balanced status compared to the SP routing. In figure 3, assuming that each host has the same load, we show a possible result of the MLI routing (in thick dashed lines). The number of hosts served by each gateway is more balanced than that of the SP routing.

The MLI routing does not necessarily always lead to a perfectly load-balanced status. Due to its diffusion nature, the service boundary of a gateway cannot move beyond other

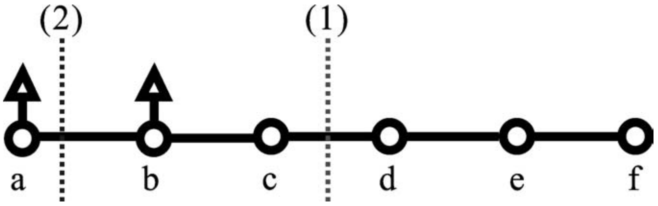


Figure 5. An example to illustrate the limitation of the MLI routing.

gateways. Figure 5 shows an extreme example. The network topology is a linear line with two gateways at the left-hand side. Assuming that hosts all have the same traffic loads, boundary (1) would be an optimal solution. However, boundary (2) is what the MLI routing can achieve because of its diffusion nature. Note that moving toward the optimal boundary is theoretically possible, but it is practically difficult, as reasoned below. First, to achieve so, a host needs to be served by one gateway while forwarding advertisements for other gateways. Unfortunately, this violates the diffusion nature. In the above scenario, host b will not forward a's advertisements. Second, most Internet routing follows the *destination-oriented* style. Any IP packet with an unknown destination will be forwarded to the default gateway. In the above scenario, depending on what b's default gateway is, all IP packets from b, c, d, e, and f with unknown destinations will be forwarded to the same gateway, leading to a very unbalanced situation. To solve this problem, a host has to specify the whole path from itself to its serving gateway (this is known as *source routing*). The design is more complicated.

4. Host-partitioning solutions

In this type of solutions, hosts will be divided into groups, each to be served by a gateway. Compared to the previous boundary-moving solutions, there are no clear boundaries being established between different groups. A host can be assigned to any gateway for the load-balance purpose. We will discuss two schemes below. The first one is a centralized approach. The second one leaves the decisions to all distributed gateways.

4.1. Centralized assignment (CA)

We assume that there is a centralized server, which is responsible for gathering the traffic load information of all hosts and assigning hosts to gateways. Unfortunately, even under such a simple model, the problem is strongly related to the renowned *number-partitioning problem*, which is known to be NP-complete [2,3,8]. Given a set of n non-negative integer numbers $a_1, a_2, \ldots, a_n$, the problem is to divide this set into two subsets such that the two sums of numbers in both subsets are as equal as possible. The problem is found to have many practical applications in the areas of multiprocessor scheduling and VLSI circuit size and delay minimization. The proposed assignment problem can be simplified and reduced to the number-partitioning problem by regarding the n integers as the loads of hosts and restricting that there are only two gateways each with the same capacity of $C = \sum_{i=1}^{n} a_i/2$. In this case, the dividing result must contain one subset with sum $\geqslant C$ and one with sum $\leqslant C$. The optimal division would be an optimal solution to the proposed assignment problem. So finding an optimal solution would be computationally expensive.

A heuristic is proposed in the following based on a greedy approach.

1. Sort all low-tier hosts into a list in a descending order of their traffic indices.
2. Sequentially assign each host x in the list to the gateway g with the minimum L'_g, where $L'_g = L_g + T_x/C_g$, until the whole list is examined.

The above procedure can be executed periodically in the central server. This scheme is costly because there will be a lot of message exchanges in the network. Besides, without considering the hop-count factor, the central server may assigns a host to a gateway which is far away from it.

4.2. Distributed assignment (DA)

For scalability and reliability reasons, distributed solutions are generally more favorable in a larger environment. In this subsection, we propose a distributed host-partitioning scheme.

The basic idea is to configure a *logical network* to connect all gateways together. Links between gateways, or referred to as *logical links*, are constructed by low-tier links. The logical network is for gateways to exchange load and capacity information and to trade low-tier hosts so as to achieve load-balanced routing. The logical network can be of any topology, but must be connected (we leave the design open here).

Based on the traffic demands of the low-tier hosts and the information gathered from other gateways, each gateway decides which of its low-tier hosts can be delegated to another gateway in a distributed manner to achieve a more balanced status. To avoid network fluctuation and reduce overheads, gateways only exchange information in defined periods and can delegate at most one host to another gateway in each time. So the *LBI* may be improved gradually. More precisely, each gateway will estimate the potential reduction in *LBI* if a member of it is delegated. The host leads to the largest reduction will be delegated to another gateway. To avoid a host being swapped repeatedly among gateways, we also enforce that a host can be delegated only if it has resided in that gateway for over a predefined time interval.

Below, we formally describe the scheme for one gateway g.

1. Gateway g should periodically convey its load and capacity information to its neighboring gateways on the logical network in every *information-exchange interval*. (Note that neighbors of a gateway are defined depending on the topology of the logical network connecting gateways. Since we do not enforce the logical network to be fully connected, a gateway may only have a partial view on other gateways' load indices.)

2. Gateway g determines the candidate gateways to which it may delegate hosts. A gateway g' is a candidate gateway if $L_g/L_{g'} \geqslant \Delta_l$, where Δ_l is a predefined load threshold. The rationale of this rule is similar to the design in MLI routing – to avoid network fluctuation.
3. For each candidate gateway g' and each host i currently served by g that has a resident time over a predefined threshold τ, gateway g tries to compute the expected *LBI* if host i is delegated to gateway g'.
4. For each of the possibilities in step 3, let (g', i) be the pair that leads to the largest positive reduction in *LBI*. Then gateway g notifies host i this decision with a *Delegate_to*(g') message.
5. Wait for one *information-exchange interval* and go back to step 2.

The above steps are to be taken by gateways. For low-tier hosts, they should inform their traffic demands to their current serving gateways. For high-tier gateways, they have to delegate hosts to other gateways. This requires interactions between hosts and gateways. We suggest that this be supported by extending the registration/reply mechanism of Mobile IP. Hosts can update their traffic demands via the extended registration message, and gateways can delegate hosts to other gateways via the extended reply message. Furthermore, in order to protect the low-tier hosts from becoming “orphans”, an “urgent” bit can be extended to the registration message to enforce a gateway to accept a host.

4.3. Implementation concerns: Source routing and IP-in-IP encapsulation

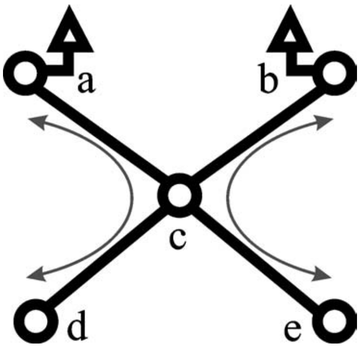


Figure 6. A routing scenario which is unsupported in destination-oriented routing.

The boundary-moving solutions are compatible with the destination-oriented routing (or next-hop routing) in typical IP networks, which has a default gateway in the routing table of each host. The host-partitioning solutions are capable of achieving a more load-balanced status potentially, but unfortunately, it violates this routing principle. For example, in figure 6, if we intend to deliver d’s and e’s traffics to gateways a and b, respectively, host c must carefully examine the source address of each IP packet which carries an unknown destination address and properly forward the packet to the serving gateway of the source host.

There are two solutions to this problem. The first one is to use IP-in-IP encapsulation as that in Mobile IP [10]. In front of each original IP packet, the source further inserts an outer IP header containing the IP address of its serving gateway. The new packet looks like a normal IP packet and thus can be directed to the desired gateway. The serving gateway then decapsulates the packet and routes the original packet toward the Internet. This approach requires all hosts including gateways and non-gateway hosts to be able to process IP-in-IP encapsulation. Inside the low-tier network, routing may be supported by AODV [14,15] or DSDV [13].

The second solution is to use *source routing* [16]. In general, source routing means the insertion of complete routing information into the packet header by the host that originates the packet. Here, the routing information is the whole path to be traveled by the packet within the low-tier network, followed by the real destination in the Internet. This host-partitioning approach can be integrated with the well-known *dynamic source routing* [4] for ad hoc networks. IPv6 also supports such routing through the extension of *routing headers* [19].

5. Probabilistic solutions

In the above boundary-moving or host-partitioning solutions, each host can only have one serving gateway. Considering the NP-complete property of the number-partitioning problem, it is understandable that these solutions cannot achieve a perfect load-balance status in many cases. The probabilistic routing schemes proposed in this section can eliminate the limitation because a host is allowed to split its traffic to multiple gateways.

5.1. Fully probabilistic (FP) routing

In the FP routing, a host will choose all possible gateways in the network as its serving gateways. The traffic of each host will be delivered to all gateways proportional to their capacities. To support FP routing, the capacity information of gateways should be advertised throughout the network. All hosts should collect and help distribute these information. For each host, a fraction of $C_x / \sum_{y \in S'} C_y$ of its traffic will be sent to gateway x. Under an ideal situation, all gateways will share the traffic load proportional to their capacities, so the resulting routing is perfectly balanced.

Nevertheless, the FP routing may incur a lot of routing and control overheads. First, the capacity information of gateways should be advertised throughout the network. Second, all hosts should split their traffics to all gateways proportional to their capacities. To reduce the overheads, in the next section, we propose another scheme that can balance performance and overheads.

5.2. Partially probabilistic (PP) routing

The PP routing only requires some hosts to conduct probabilistic routing. It follows the concept of boundary-moving solutions, but allows hosts at the service boundaries between gateways to conduct probabilistic routing. Also, to be more practical, we limit the number of serving gateways to which a host can send traffic probabilistically. Below, we develop our PP routing based on the earlier SP routing.

More specifically, the PP routing runs the SP routing at the background. This gives clear boundaries between gateways. The SP routing ensures that each host will pick the nearest gateway as its serving gateway, which we call the *primary gateway* of the host.

However, the PP routing will send data packets following a different rule from the SP routing. Gateways will broadcast advertisements carrying their load information around the network. When a host x, whose current primary gateway is g, hears an advertisement from g, it will help rebroadcast the advertisement. When a host x, whose current primary gateway is g, hears an advertisement from another gateway g', it is allowed to rebroadcast the advertisement only if $L_g/L_{g'} \geqslant \Delta_l$, where Δ_l is a predefined load threshold. As a result, the advertisements from gateways with lower traffic loads are more likely to be distributed around the network.

A host will route its data packets to one or multiple gateways depending on how many gateways it can receive advertisements from. For a host which can hear advertisements from only one gateway, all its data packets will be delivered to that gateway. For a host which can hear advertisements from multiple gateways, it will route its packets probabilistically. For each packet, it will be routed to the host's primary gateway with a predefined traffic-redistributing probability P_{PP}. (The purpose is to lower down the probability for a sudden burst of traffics being sent to a gateway with a particularly low load.) With a probability of $1 - P_{\text{PP}}$, the packet will be routed probabilistically. Let y_1 and y_2 be the two gateways from which the host can receive advertisements and which have the lowest loads. Based on their capacities, the host will send the packet to gateway y_1 with a probability $C_{y_1}/(C_{y_1} + C_{y_2})$, and to gateway y_2 with a probability $C_{y_2}/(C_{y_1} + C_{y_2})$. To summarize, the packet will be sent to the primary gateway with a probability P_{PP}, to gateway y_1 with a probability $(1 - P_{\text{PP}}) \cdot \frac{C_{y_1}}{C_{y_1}+C_{y_2}}$, and to gateway y_2 with a probability $(1 - P_{\text{PP}}) \cdot \frac{C_{y_2}}{C_{y_1}+C_{y_2}}$.

5.3. Implementation concerns: Binding with multiple FAs

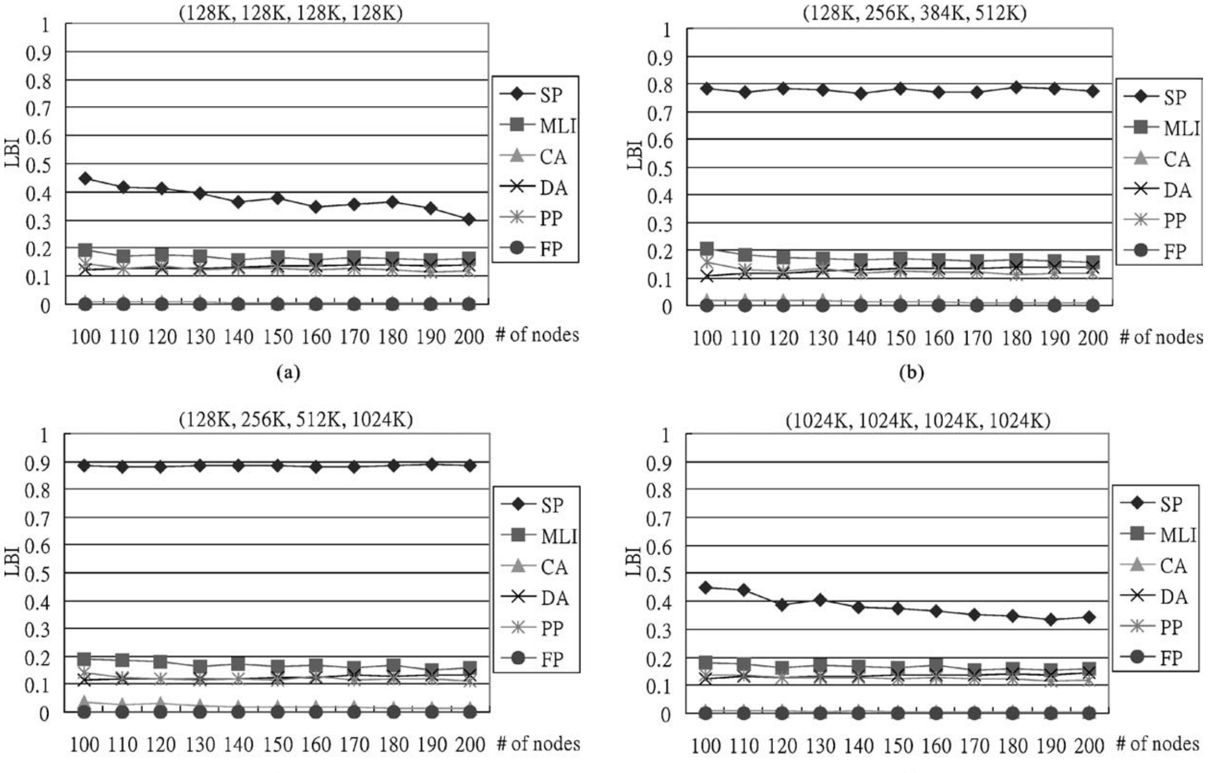
The probabilistic routing can potentially arrive at a more load-balanced status compared to the earlier solutions. Outgoing packets (those to be sent to the Internet) can be routed either in an IP-in-IP encapsulation or source routing manner as described in section 4.3. However, care must be taken for incoming packets (those coming from the Internet). Clearly, whichever gateway a packet arrives at, it should be delivered from that gateway to the desired host. This is another factor that could result in an unbalanced status.

One possible solution to this problem is to use the *multiple binding* option supported in the Mobile IP. This option allows a host to concurrently register with multiple mobile agents as their foreign agents. By this option, the home agent of the mobile host can tunnel packets to multiple foreign agents for the mobile host. To achieve load-balanced routing, the traffic distribution determined by the FP/PP schemes should be conveyed to the home agent. This requires modification of the Mobile IP protocol. With such support, the incoming traffic can also be distributed to gateways in a balanced manner.

6. Simulation results

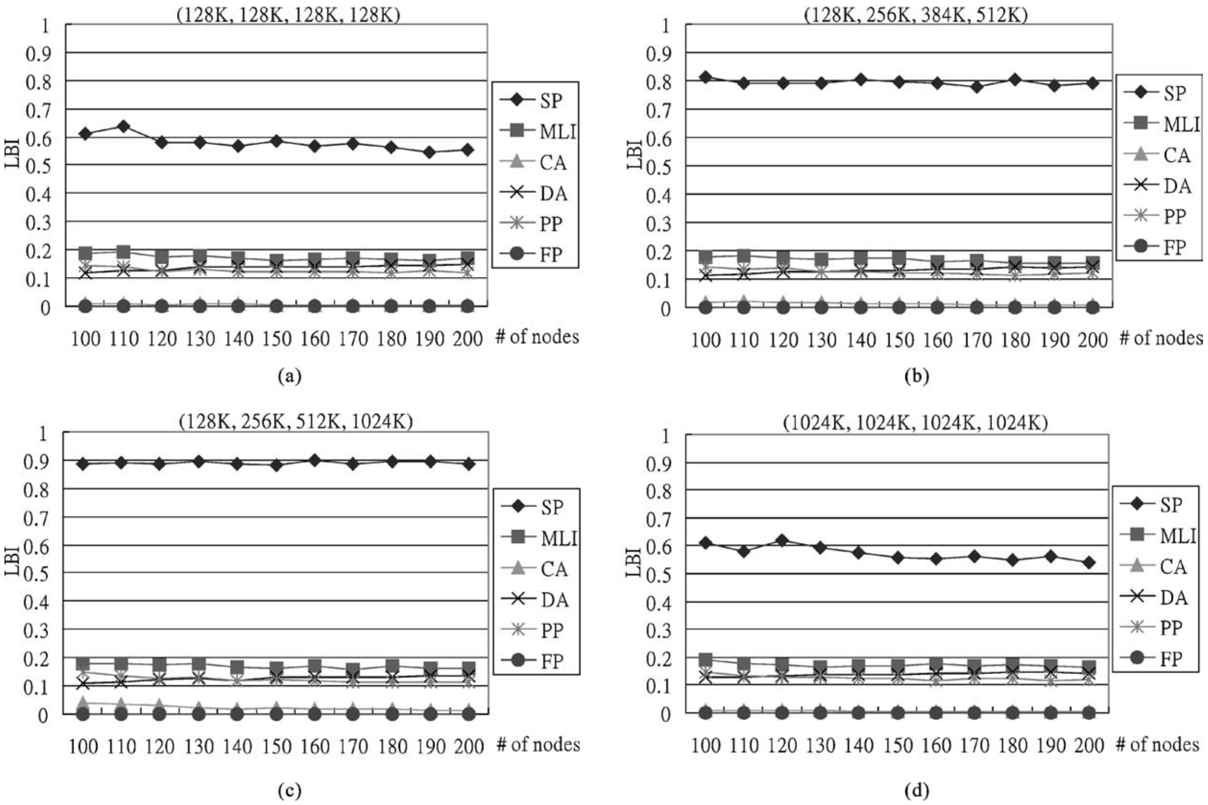
We have developed a simulator to compare the performance of the proposed routing schemes. There are two kinds of simulation environments. The first one is a 500×500 square area, on which 100~200 hosts are randomly generated. The second one is a 1,000 × 1,000 square area, on which 400~800 hosts are randomly generated. Among these hosts, we choose four hosts as gateways in the first environment, and sixteen hosts as gateways in the second environment. Each host has a low-tier transmission distance of 80 units and a transmission rate of 11 Mbps. Each host has a traffic load uniformly distributed between 1 and 20 kbps. To test homogeneity and heterogeneity of gateways, we set four combinations of gateway capacities: (128 K, 128 K, 128 K, 128 K), (128 K, 256 K, 384 K, 512 K), (128 K, 256 K, 512 K, 1024 K), and (1024 K, 1024 K, 1024 K, 1024 K). For simplicity, all gateways and hosts have no mobility (and for this reason routing overheads can be ignored). Gateways can be deployed in three ways. Below, we present our simulation results according to these deployments of gateways. The performance index is *LBI* of the whole network. The parameters for the MLI scheme are $\Delta_l = 1.3$ and $P_{\text{MLI}} = 0.7$. The parameter for the DA scheme is $\Delta_l = 1.2$. The parameters for the PP scheme are $\Delta_l = 1.2$ and $P_{\text{PP}} = 0.8$.

(A) Regular deployment of gateways. In this experiment, we consider the first environment and divide the network area evenly into four 250 × 250 regions. The four gateways are each placed at the center of one region. The simulation result is in figure 7. The x-axis is the number of hosts, which reflects the level of load from low-tier hosts. The ideal FP scheme always performs very well. Second to the FP scheme is the CA scheme. The CA scheme needs to collect global load information and makes decisions in a central server, so it is reasonable that CA can achieve a quite load-balanced status. Following the CA scheme are the PP and DA schemes, which do not try to collect global load information. The boundary-moving solutions (MLI and SP schemes) do not perform as well as the previous schemes, but the MLI scheme is much better than the SP scheme. As there are more and more hosts, the performance of the SP scheme can be slightly improved in the cases of figures 7(a) and (d), where gateway capacities are homogeneous. However, when the capacities of gateways are heterogeneous, as figures 7(b) and (c) show, such an im-



(a) (b) (c) (d)

Figure 7. *LBI* vs. number of hosts under a regular deployment of gateways in a 500 × 500 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K), (b) (128 K, 256 K, 384 K, 512 K), (c) (128 K, 256 K, 512 K, 1024 K), and (d) (1024 K, 1024 K, 1024 K, 1024 K).



(a) (b) (c) (d)

Figure 8. *LBI* vs. number of hosts under a semi-regular deployment of gateways in a 500 × 500 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K), (b) (128 K, 256 K, 384 K, 512 K), (c) (128 K, 256 K, 512 K, 1024 K), and (d) (1024 K, 1024 K, 1024 K, 1024 K).

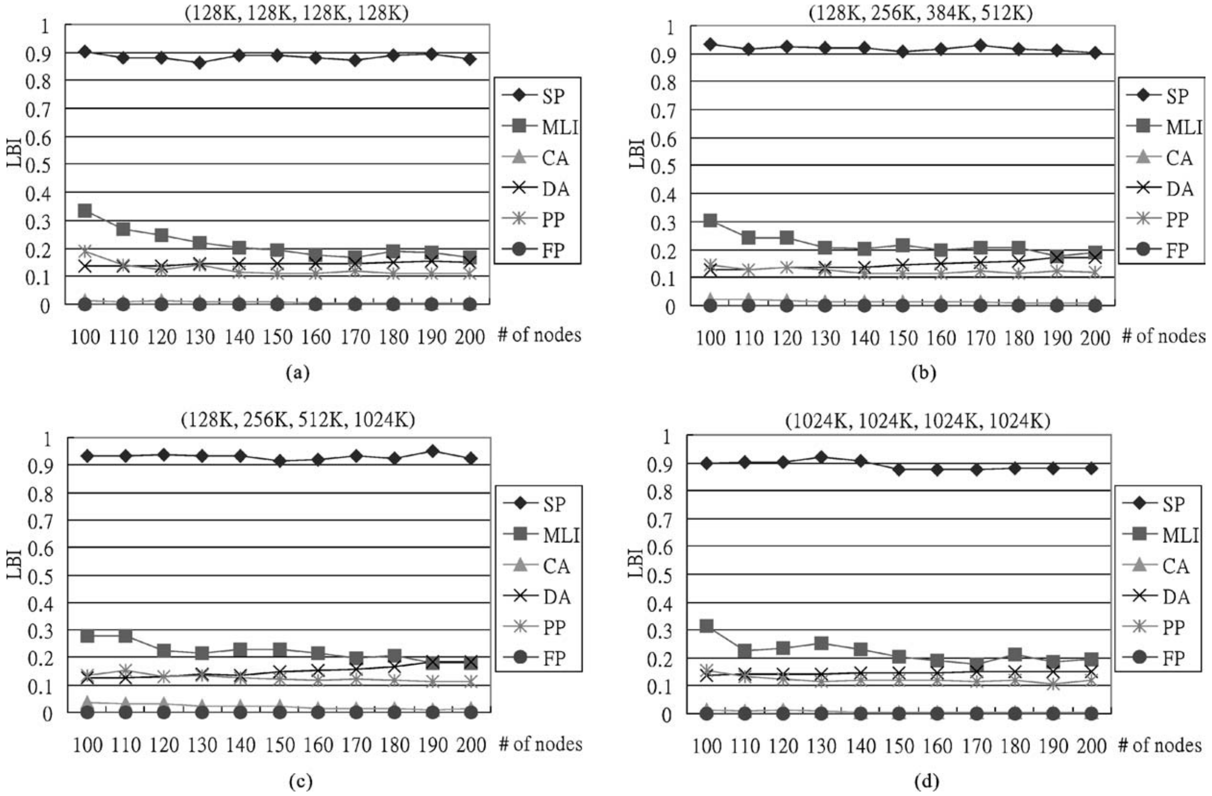


Figure 9. *LBI* vs. number of hosts under a concentrated deployment of gateways in a 500 × 500 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K), (b) (128 K, 256 K, 384 K, 512 K), (c) (128 K, 256 K, 512 K, 1024 K), and (d) (1024 K, 1024 K, 1024 K, 1024 K).

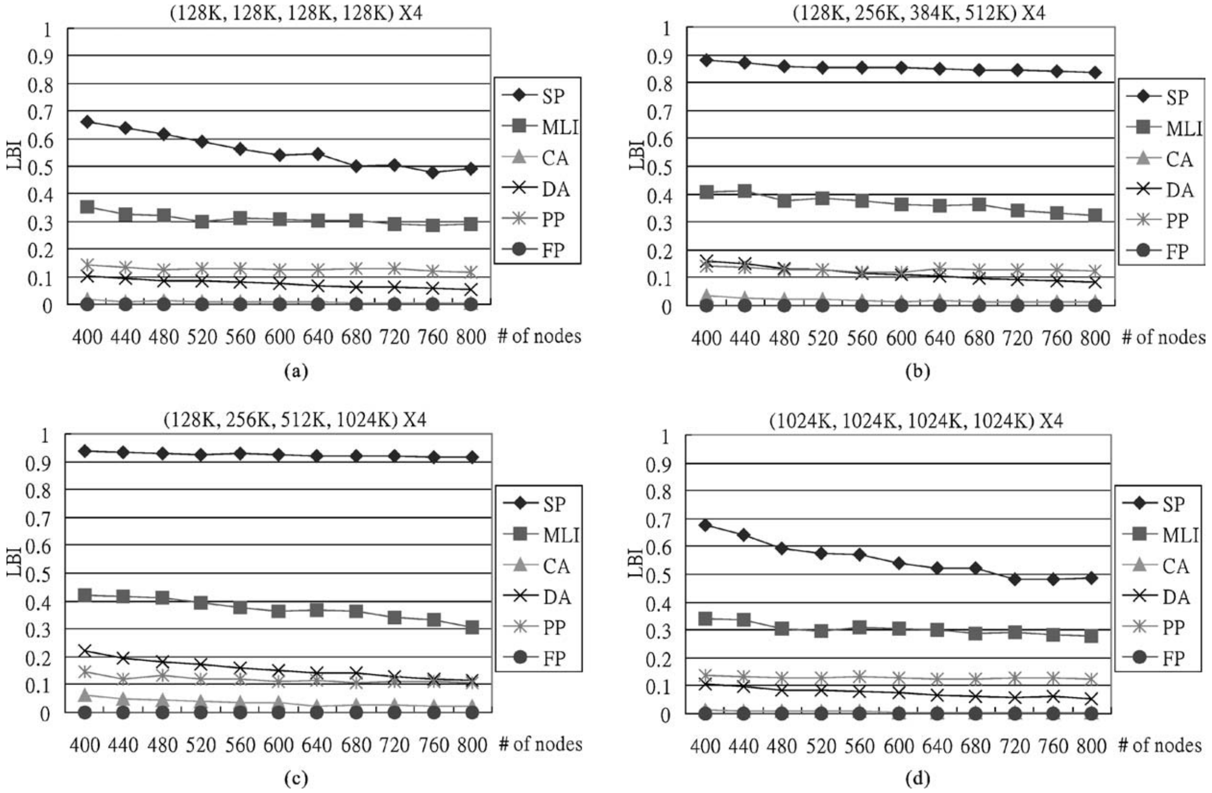


Figure 10. *LBI* vs. number of hosts under a regular deployment of gateways in a 1,000 × 1,000 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K) × 4, (b) (128 K, 256 K, 384 K, 512 K) × 4, (c) (128 K, 256 K, 512 K, 1024 K) × 4, and (d) (1024 K, 1024 K, 1024 K, 1024 K) × 4.

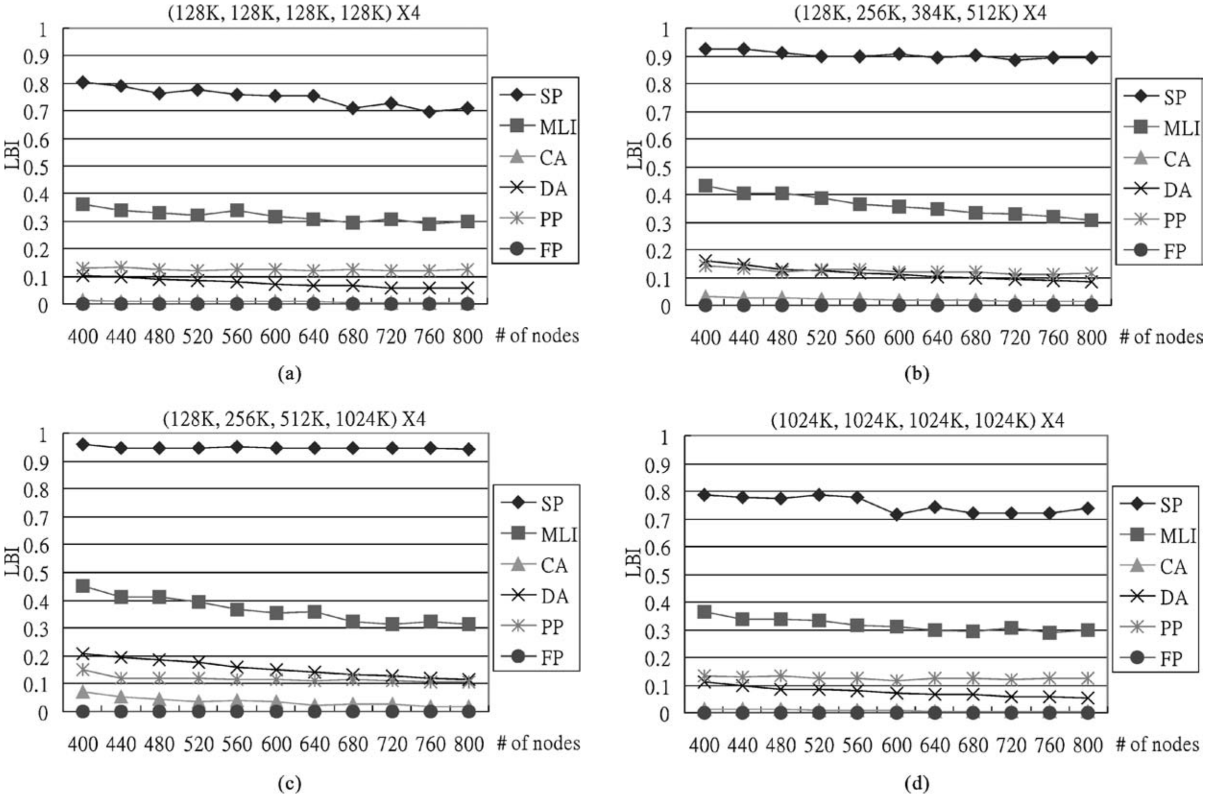


Figure 11. *LBI* vs. number of hosts under a semi-regular deployment of gateways in a 1,000 × 1,000 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K) × 4, (b) (128 K, 256 K, 384 K, 512 K) × 4, (c) (128 K, 256 K, 512 K, 1024 K) × 4, and (d) (1024 K, 1024 K, 1024 K, 1024 K) × 4.

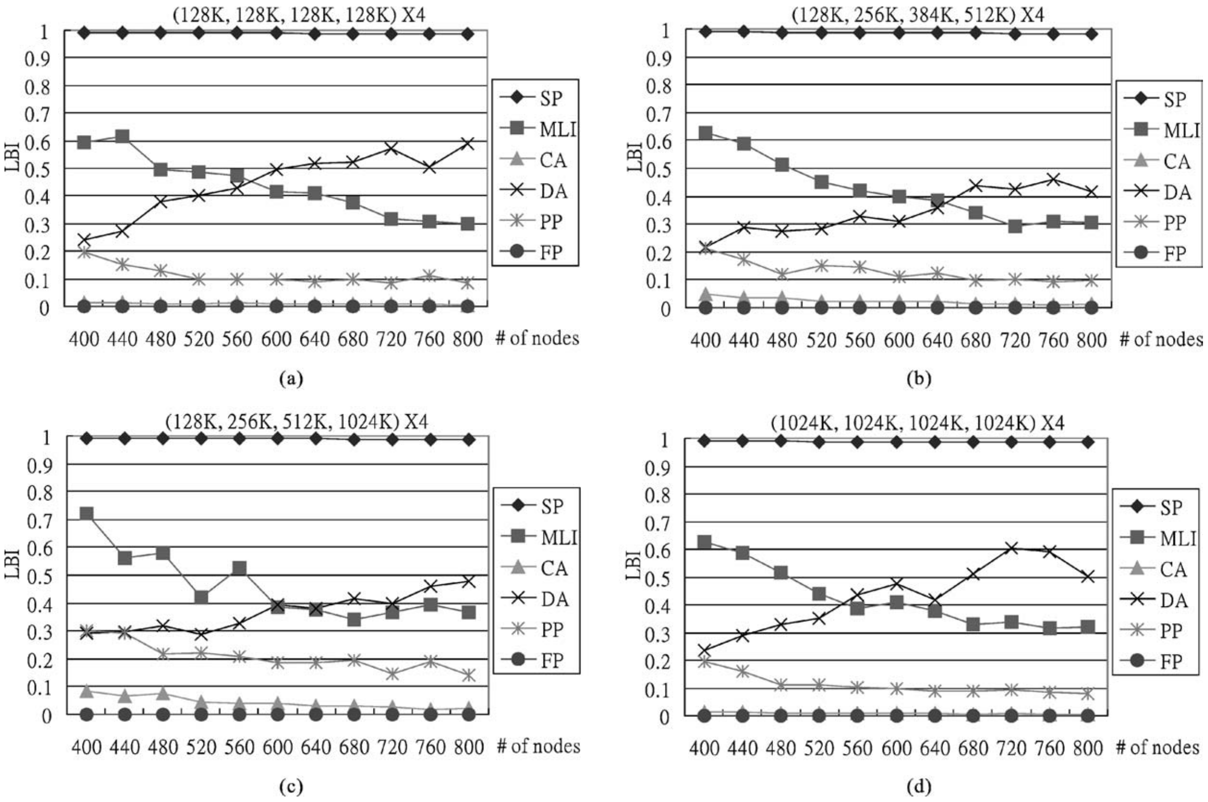


Figure 12. *LBI* vs. number of hosts under a concentrated deployment of gateways in a 1,000 × 1,000 network, where the capacities of gateways are: (a) (128 K, 128 K, 128 K, 128 K) × 4, (b) (128 K, 256 K, 384 K, 512 K) × 4, (c) (128 K, 256 K, 512 K, 1024 K) × 4, and (d) (1024 K, 1024 K, 1024 K, 1024 K) × 4.

provement cannot be seen. So the straightforward SP routing scheme cannot handle the heterogeneity of gateways well.

(B) Semi-regular deployment of gateways. In this experiment, we still consider the same network environment as in part (A). However, in each 250 × 250 region, one gateway is randomly placed inside the region so as to simulate some heterogeneity in gateway deployment. That is, the deployment of gateways is not so uniform compared to the earlier case. The result is in figure 8. In this case, load balance is more difficult to achieve compared to the previous case for the boundary-moving solutions (MLI and SP schemes). So we see some degradation for these schemes. The other schemes are less affected by the randomness of gateways.

(C) Concentrated deployment of gateways. In this experiment, we still consider the same network environment as in parts (A) and (B). However, to simulate more heterogeneity, the four gateways are placed in the upper-left 250×250 square region in a random manner. So this will create some traffic concentration effect toward the upper-left region. The result is illustrated in figure 9. In this case, finding proper boundaries betwen the serving regions of gateways is more difficult for the boundary-moving solutions. So we see further performance degradation for such solutions. The performance of the DA scheme is also affected by such heterogeneity in gateway deployment, especially when there are more hosts and when there is more heterogeneity in gateways' capacities.

(D) Larger network environment. Different from the previous experiments, this experiment considers the second network environment (of size 1,000 × 1,000 with 400∼800 random hosts). We divide the network area into sixteen 250×250 square regions. Again, gateways are deployed in three ways: regular (each region with a gateway at the center), semi-regular (each region with a gateway at a random location), and concentrated (all gateways at the upper-left corner region). The parameters for the MLI scheme are $\Delta_l = 1.4$ and $P_{\mathrm{MLI}} = 0.7$. The parameter for the DA scheme is $\Delta_l = 1.2$. The parameters for the PP scheme are $\Delta_l = 1.1$ and $P_{\mathrm{PP}} = 0.9$. The simulation results are shown in figures 10, 11, and 12, respectively. In such an environment, load balance is more difficult to achieve compared to the previous environment for the boundary-moving solutions (MLI and SP schemes). So we see larger degradation for these schemes. We can also observe some performance degradation for the DA scheme, especially when there are more hosts and when the deployment of gateways is more irregular (such as the concentrated deployment in figure 12). The other schemes are less affected these factors.

7. Conclusions

In the literature, most works consider mobile ad hoc networks as stand-alone networks. In this work, we have extended the definition of ad hoc networks by considering equipping some hosts in an ad hoc network with extra network interfaces that can access cellular-based networks, such as PHS or GPRS, which have much longer communication distances compared to typical IEEE 802.11/HyperLAN interfaces. We believe that this can greatly improve the usefulness of ad hoc networks in both conquering the network partitioning problem and improving its Internet access capability. The proposed architecture can potentially support train/car/bus Internet by considering the high-tier gateways as roaming routers.

We have proposed several solutions for load-balance routing in this two-tier, heterogeneous network architecture. The boundary-moving solutions are easy to implement and compatible with current IP routing, but in many cases cannot lead to a load-balanced status. It is particularly weak in handling the heterogeneity of gateways and irregularity in gateway deployment. The host-partitioning solutions are able to handle these situations very well by paying some more routing overheads. To be compatible with IP routing, either IP-in-IP encapsulation or source routing must be used to support routing inside the ad hoc network. The probabilistic solutions further allow a host to bind with multiple gateways, through Mobile IP's multiple binding option, and send/receive traffic to/from the Internet proportional to the capacities of gateways. Thus, the FP scheme can achieve a perfect load-balanced status in most cases. The PP scheme does not perform that well, but it also fairly good with less overheads. In this work, we have assumed that the traffic in each connection is symmetric in the uplink and downlink directions. In reality, this is usually not the case. For high-tier interfaces, they may have different capacities in uplink and downlink directions too. This deserves further investigation. We are currently conducting implementations. The related result will be reported in future papers.

References

[1] F. Aurenhammer, Voronoi diagrams: A survey of a fundamental geometric data structure, ACM Computing Surveys 23(3) (September 1991) 345–405.

[2] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, England, 1994).

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).

[4] D.B. Johnson and D.A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Vol. 353, Chapter 5 (Kluwer Academic Publishers, 1996) pp. 153–181.

[5] Y.-D. Lin and Y.-C. Hsu, Multihop cellular: A new architecture for wireless communications, in: *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)* pp. 1273–1282.

[6] S. Lu, V. Bharghavan and R. Srikant, Fair scheduling in wireless packet networks, IEEE/ACM Transactions on Networking 7(4) (August 1999) 373–389.

[7] J.P. Macker and M.S. Corson, Mobile ad hoc networking and the IETF, ACM Mobile Computing and Communications Reviews 4(4) (October 2000) 12–13.

[8] R.E. Miller and J.W. Thatcher (eds.), *Complexity of Computer Computations* (Plenum Press, New York, 1972).

[9] A.K. Parekh and R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: The single node case, IEEE/ACM Transactions on Networking 1(3) (June 1993) 344–357.

[10] C.E. Perkins, *Mobile IP Design Principles and Practices* (Addison-Wesley, Boston, MA, 1997).

[11] C.E. Perkins, *Ad Hoc Networking* (Addison-Wesley, Boston, MA, 2001).

[12] C.E. Perkins, Mobile IP, IEEE Communication Magazine 40(5) (May 2002) 66–82.

[13] C.E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: *Proc. of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications* (1994) pp. 234–244.

[14] C.E. Perkins and E.M. Royer, Ad hoc on-demand distance vector routing, in: *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications* (February 1999).

[15] C.E. Perkins, E.M. Royer, S.R. Das and M.K. Marina, Performance comparison of two on-demand routing protocols for ad hoc networks, IEEE Personal Communications 8(1) (February 2001).

[16] J.B. Postel (ed.), Internet Protocol, Internet Request for Comments RFC 791 (September 1981).

[17] E.M. Royer and C.-K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications 6(2) (April 1999) 46–55.

[18] D. Stiliadis and A. Varma, Efficient fair queueing algorithms for packet-switched networks, IEEE/ACM Transactions on Networking 6(2) (April 1998) 175–185.

[19] S.A. Thomas, *IPng and the TCP/IP Protocols*, Chapter 4 (Wiley, New York, 1996) pp. 113–117.

[20] Y.-C. Tseng and T.-Y. Hsieh, Fully power-aware and location-aware protocols for wireless multi-hop ad hoc networks, in: *Proc. of the Int'l Conf. on Computer Communication and Networks (ICCCN)* (2002).

[21] Y.-C. Tseng, C.-C. Shen and W.-T. Chen, Mobile IP and ad hoc networks: An integration and implementation experience, IEEE Computer 36(5) (May 2003) 48–55.

[22] H. Wu, C. Qiao, S. De and O. Tonguz, Integrated cellular and ad hoc relaying systems: iCAR, IEEE Journal on Selected Areas in Communications 19(10) (October 2001) 2105–2115.

[23] I.-W. Wu, W.-S. Chen, H.-E. Liao and F.-F. Young, A seamless handoff approach of mobile IP protocol for mobile wireless data networks, IEEE Transactions on Consumer Electronics 48(2) (May 2002) 335–344.

Chi-Fu Huang received his B.S. and M.S. degrees in computer science from the Feng-Chia University and the National Central University in 1999 and 2001, respectively. He is currently a Ph.D. student at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. His research interests include wireless communication and mobile computing.
E-mail: cfhuang@csie.nctu.edu.tw

Hung-Wei Lee received his B.S. and M.S. degrees in computer science and information engineering from the National Chiao-Tung University in 2001 and 2003, respectively. His research interests include wireless communication and mobile computing. He is going to work in the Science-based Industrial Park, Hsin-Chu, Taiwan, as an engineer.
E-mail: leehw@csie.nctu.edu.tw

Yu-Chee Tseng received his B.S. and M.S. degrees in computer science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He obtained his Ph.D. in computer and information science from the Ohio State University in January of 1994. From 1994 to 1996, he was an Associate Professor at the Department of Computer Science, Chung-Hua University. He joined the Department of Computer Science and Information Engineering, National Central University in 1996, and has become a Full Professor since 1999. Since August 2000, he has become a Full Professor at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan. Dr. Tseng has served as a Program Chair in the *Wireless Networks and Mobile Computing Workshop*, 2000 and 2001, as an Associate Editor for *The Computer Journal*, as a Guest Editor for *ACM Wireless Networks* special issue on "Advances in Mobile and Wireless Systems", as a Guest Editor for *IEEE Transactions on Computers* special on "Wireless Internet", as a Guest Editor for *Journal of Internet Technology* special issue on "Wireless Internet: Applications and Systems", as a Guest Editor for *Wireless Communications and Mobile Computing* special issue on "Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing", as an Editor for *Journal of Information Science and Engineering*, and as a Guest Editor for *Telecommunication Systems* special issue on "Wireless Sensor Networks". He received the Outstanding Research Award, 2001–2002, from the National Science Council, ROC. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng is a Senior Member of the IEEE.
E-mail: yctseng@csie.nctu.edu.tw