Wireless Personal Communications **4:** 103–139, 1997.

Associativity-Based Routing for *Ad-Hoc* Mobile Networks*

CHAI-KEONG TOH**
University of Cambridge, Computer Laboratory, Cambridge CB2 3QG, United Kingdom

Abstract. This paper presents a new, simple and bandwidth-efficient distributed routing protocol to support mobile computing in a conference size *ad-hoc* mobile network environment. Unlike the conventional approaches such as link-state and distance-vector distributed routing algorithms, our protocol does not attempt to consistently maintain routing information in every node. In an *ad-hoc* mobile network where mobile hosts (MHs) are acting as routers and where routes are made inconsistent by MHs' movement, we employ an associativity-based routing scheme where a route is selected based on nodes having associativity states that imply periods of stability. In this manner, the routes selected are likely to be long-lived and hence there is no need to restart frequently, resulting in higher attainable throughput. Route requests are broadcast on a per need basis. The association property also allows the integration of *ad-hoc* routing into a BS-oriented Wireless LAN (WLAN) environment, providing the fault tolerance in times of base stations (BSs) failures. To discover shorter routes and to shorten the route recovery time when the association property is violated, the localised-query and quick-abort mechanisms are respectively incorporated into the protocol. To further increase cell capacity and lower transmission power requirements, a dynamic cell size adjustment scheme is introduced. The protocol is free from loops, deadlock and packet duplicates and has scalable memory requirements. Simulation results obtained reveal that shorter and better routes can be discovered during route re-constructions.

Key words: *ad-hoc* mobile networks, associativity-based routing, neighbour-aware mobile computing.

1. The Problem and Motivation

From the best-effort-delivery type of Wireless LANs (which are basically extensions of the existing Ethernet networks running Mobile IP network protocol) to the state-of-the-art quality-of-service based Wireless ATM networks (such as SWAN-Seamless Wireless ATM Network [16–19], the "wireless last hop" from the wireline network to the mobile host has been supported through the use of wireless base stations.

Unlike these infra-structured WLANs with base-stations (BSs) providing coverage for mobile hosts (MHs), *ad-hoc* mobile networks do not have any access to BSs. The problem here relates to how MHs can communicate with one other, over the wireless media, without any infra-structured network component support. The most obvious problem is to devise a scheme to compute routes which can adapt well to link changes. Conventional distributed routing schemes attempt to maintain consistent routing information by performing periodic link and topology updates. These, however, are undesirable for MHs in an *ad-hoc* mobile network since MHs' migrations cause frequent link changes, which result in enormous transmissions over the wireless media to propagate and update routes. This is very inefficient in an environment where radio bandwidth and battery power are scarce resources. Hence, there is a need for a new, efficient and robust routing scheme for MHs in an *ad-hoc* mobile network.

---

[*] Parts of this Letter appeared in IEEE International Phoenix Conference on Computers & Comunications (IPCCC'96) as "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing". This invention, also named as "Cambridge Ad-Hoc Routing Protocol", has been filed for a patent.

[**] C.-K. Toh is supported by King's College Cambridge and Cambridge Commonwealth Trust Scholarships. He can be reached at: Chai-Keong.Toh@cl.cam.ac.uk.

The organisation of this Letter is as follows. Section 2 discusses briefly on the existing *ad-hoc* mobile routing schemes while Section 3 describes the characteristics of *ad-hoc* mobile networks. Section 4 introduces a new concept of associativity-based routing (ABR), followed by a detailed description of the protocol in Section 5. The performance of the routing protocol is evaluated with a migration-based mobile network simulator, which is described in Section 6. Comparison of our proposed protocol with existing routing schemes is presented in Section 7. Finally, Section 8 reveals a dynamic cell size adjustment scheme (DCSAS) for ABR while Section 9 provides a brief discussion on *ad-hoc* Wireless ATM LANs and mobile applications.

2. *Ad-Hoc* Mobile Network Routing Schemes

2.1. Early Mobile Routing Schemes

The ARPANET Packet Radio Network (PRN) [21, 30] is the earliest deployment of a regional-wide wireless data network. It has existed since the 1970s and has proven to be feasible and practical. In a PRN, all components (repeaters, terminals and stations) can be mobile or certain components can remain fixed while others are moving. There are basically two approaches to routing and packet forwarding in PRNs and since these approaches are used here as the basis of reference for *ad-hoc* mobile routing, they are briefly described below.

- *Routing in* PRNs
  In “point-to-point” routing, the station computes all the routing information and the decision is either distributed to the repeaters involved in the route or to the source packet radio (PR). This scheme was found to be suitable for slow moving user terminals. However, in “broadcast routing”, each packet radiates away from the source PR with a wave-front-like propagation. Since no station needs to be present to compute routes, the destination address serves to identify the intended recipient. For fast moving user terminals, broadcast routing was found to be useful as it avoided the need to process rapidly changing routes.
- *Packet forwarding in* PRNs
  The connectionless approach to packet forwarding requires some background operation to maintain up-to-date network topology and link information in each node. This means that as topology changes, the background routing traffic can be substantial. This is commonly associated with broadcast routing, where each packet carries sufficient routing information for it to arrive at the destination. However, in the connection-oriented approach, an explicit route establishment phase is required before data traffic can be transported. This approach is commonly associated with point-to-point routing, where each node in a route has a lookup table for forwarding incoming packets to the respective out-going links. Hence, if the topology changes, a route re-establishment phase is needed. A detailed contrast between these approaches is found in [2] and [26].

2.2. Current Mobile Routing Schemes

Many *ad-hoc* mobile routing schemes have evolved. They include those of [3, 8, 9, 13, 20, 22, 25, 28]. Most of these schemes are based on either broadcast or point-to-point routing using either the connectionless or connection-oriented packet forwarding approach. We shall briefly describe five such schemes.

The “Layer Net” self-organising protocol proposed in [22] uses a connectionless packet forwarding approach. Broadcast routing is used for the initial network connectivity construction and the subsequent topology maintenance as a result of nodes’ movements and link changes. Because routes are not constructed based on demand and topology updates have to be performed in sympathy with link changes, the overall signalling traffic can be quite substantial.

Cluster-based routing [20], in essence, uses the broadcast routing and connectionless packet forwarding approach. It relies on existing routing schemes such as link-state or distance-vector routing to derive network topology and link information. On top of this, a clustering methodology is used to reduce the number of updates due to MHs’ migrations. Routes are constructed between all pairs of nodes and route maintenance is essentially cluster maintenance. Hence this method is inefficient.

In source-initiated distributed routing [8], however, a combination of point-to-point and broadcast routing using the connection-oriented packet forwarding approach is used. Here routes are initiated by the source and are constructed based on demand. Hence, this scheme forgoes the need to constantly propagate up-to-date routing information throughout the network. However, because alternate route information is used during route re-construction (RRC), problems of stale routes exist.

As for the dynamic source routing scheme proposed in [13], periodic route advertisements are avoided and route caches are used to store source routes that a mobile host has learnt over time. Routes are source-initiated and discovered via a route discovery protocol. With source routing, the sender explicitly lists the route in each packet’s header, so that the next-hop nodes are identified as the packet transits towards the destination. Because route cache information is used, accurate updates of these route caches are essential. Because the sender has to be notified each time a route is truncated, the route maintenance phase does not support fast route re-construction.

Finally, for destination sequence distance-vector (DSDV) routing [4], an enhancement to the existing distance-vector Bellman–Ford routing is made in order to support *ad-hoc* MHs. Because each MH has to periodically advertise its view of the network topology, this scheme is inefficient. Similar to cluster-based routing, this scheme uses the broadcast routing and connectionless packet forwarding approach. Hence, the problems associated with existing routing schemes motivate us to seek a better routing approach to support *ad-hoc* mobile computing.

3. Characteristics of *Ad-Hoc* Mobile Networks

3.1. *AD-HOC* COMMUNICATION SERVICE DISCOVERY

Conventional computer supported groupware technology forces users to be in the same place where the collaboration facilities are installed. Users are also compelled to reserve the collaboration facilities well in advance to prevent usage conflicts. In addition, partners in collaboration are restricted to those who can connect to the same network. *Ad-hoc* mobile networks, on the other hand, allow spontaneous LANs to be created anytime and anywhere. Such networks support nomadic collaborative computing [1], which occur anytime and anywhere, without the support of BSs. Hence, users are no longer bounded by the time and place to perform collaborative computing.

For *ad-hoc* mobile communications to be possible, a host discovery mechanism must exist. When an association is formed (through recognising a MH's beacon), the mobile application is informed of the new mobile user who can then participate in nomadic collaborative computing. This can appear as an icon which when clicked, reveals details of the new participant. The new associativity of one MH with another can also be propagated to the nodes' neighbours, so that all other neighbouring MHs within the mobile subnet can be aware of the existence of the new user. Alternatively, an MH can choose only to be aware of its immediate neighbours and can later perform an on-demand neighbour discovery broadcast when it desires to communicate with MHs outside its vicinity. In this manner, MHs can also discover what services are available from which MHs.

3.2. TYPES OF COMMUNICATION

An *ad-hoc* mobile network has no BSs to provide connectivity to backbone hosts or to other MHs. There is no need for handover and location management at all. In such a network, each MH acts as a router, forwarding packets from one MH to another. The communication connectivity is fairly "weak", as any migration by MHs participating in one or more routes will cause the route to become invalid. The MH should not spend most of its time updating and computing routes in sympathy with MHs' movements. Such schemes are highly impractical, inefficient and result in low data throughput. Hence, a novel routing scheme is needed to provide efficient and high throughput communication among *ad-hoc* MHs. Prior to that, we shall examine the possible types of *ad-hoc* mobile communications.

MHs in an *ad-hoc* mobile network can communicate with their immediate peers, i.e. peer-to-peer, which is a single radio hop away. However, if three or more MHs are within range of each other (but not necessarily a single hop away from each other), then "remote-to-remote" MHs communications exist. Remote-to-remote communications are associated with group migrations. The traffic characteristics for different types of *ad-hoc* communications are discussed below.

3.3. TRAFFIC PATTERNS

What are the traffic and migration patterns associated with MHs in an *ad-hoc* mobile network? Depending on the MHs' associativity and migration profiles and based on the inference that *ad-hoc* MHs wish to communicate when they are within range of each another, we identify the following three types of traffic patterns.

Figure 1(a) shows the network traffic for the case of peer-to-peer communications, where communications among MHs are usually between pairs that are a single radio hop away from each other. These MHs can be migrating in pairs or in groups.

The second scenario is remote-to-remote communications, where mobile users migrate in groups and communicate with destination hosts that are beyond a single radio hop away. An example of this is a group of mobile users gathering together in some foreign place for a discussion, a seminar or a mini-conference where there is no luxury of radio coverage by BSs. Here, remote-to-remote communications allow multi-party collaboration and spontaneous *ad-hoc* network formation to be possible. The resulting network traffic is shown in Figure 1(b).

The third scenario refers to MHs migrating without any correlation with one another and communications can exist between or among the MHs. This is the extreme scenario where the MHs are highly dynamic, resulting in very poor connectivity and low throughput as compared

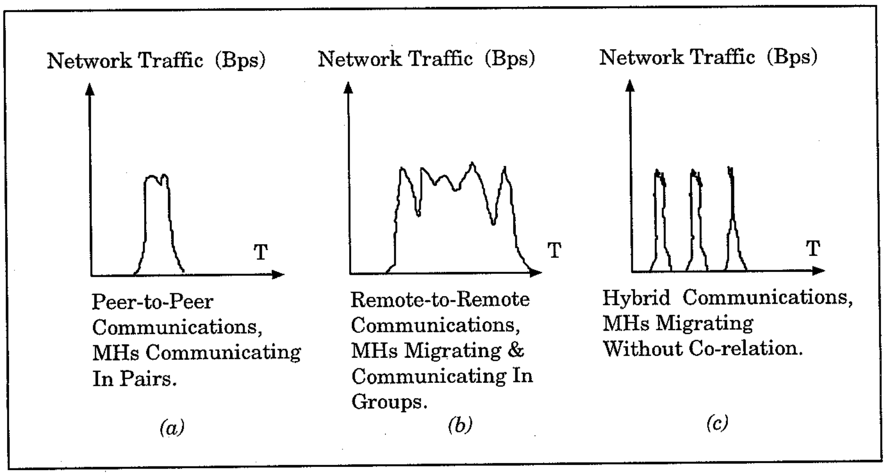


Figure 1. Traffic patterns in an *ad-hoc* mobile network.

to MHs migrating in groups. Hence, the network traffic may consist of a series of abrupt bursts, as shown in Figure 1(c).

3.4. INTEGRATION WITH BS-ORIENTED WLANS

It is desirable for a MH to be able to function in both *ad-hoc* or BS-oriented WLAN environments.[1] When a MH receives beacons generated by other MHs, it automatically invokes *ad-hoc* routing to support mobile-to-mobile communication. However, when it receives beacons generated by BSs, the MH knows that it has access to the wired network and hence conventional routing protocols supported by location management, registration, handovers, etc., can be invoked.

Mobile applications can also be made intelligent enough to decide which communication mode, i.e. *ad-hoc* or BS-oriented, best suits the service requirements. It shall be shown later that both *ad-hoc* and BS-oriented modes can be combined to provide fault tolerance against BSs' failures.

3.5. TYPES OF MH MOVEMENTS

3.5.1. *Movements by* SRC, DEST *and* INs

A route in an *ad-hoc* mobile network comprises of the source (SRC), destination (DEST) or/and a number of intermediate nodes (INs). Movements by any of these nodes will affect the validity of the route.

A SRC in a route has a downstream link and when it moves out of its downstream neighbour's radio coverage range, the existing route will immediately become invalid. Hence, all the downstream nodes may have to be informed so as to erase their invalid route entries.

Likewise when the DEST moves out of the radio coverage range of its upstream neighbour, the route becomes invalid. However, unlike the SRC, the upstream nodes will have to be informed so as to erase their invalid route entries.

---

[1] This, in fact, is one of the functional specifications laid down by the IEEE 802.11 committee.

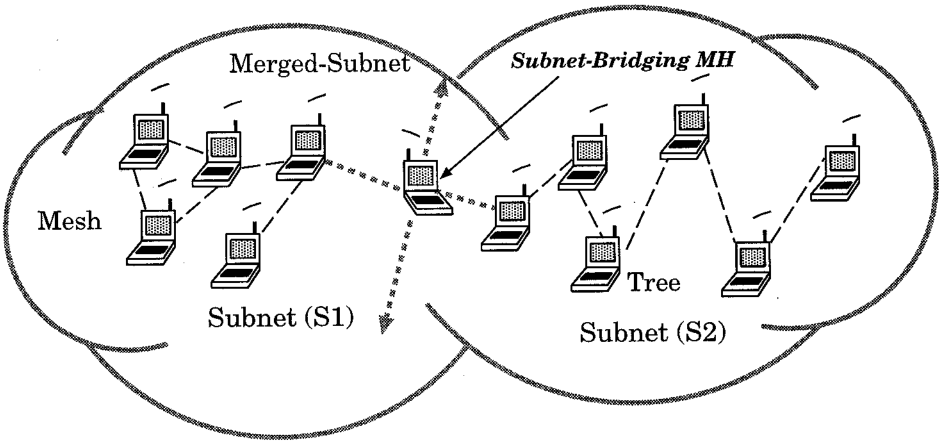


Figure 2. Ad-hoc mobile subnets merging and fragmenting.

Lastly, similar to the SRC and DEST nodes' movements, any movements by one of the INs supporting an existing route may cause the route to become invalid.

All these nodes' movements cause conventional distributed routing protocols to respond in sympathy with the link changes, in order to update all the remaining nodes within the network, so that consistent routing information can be maintained. However, this involves broadcasting over the wireless medium which results in wasteful bandwidth and an increase in the overall network control traffic. Consequently, a better route re-construction scheme is required.

3.5.2. *Movements by Subnet-Bridging* MH

In addition to the above-mentioned moves, moves by a MH which is performing subnet-bridging function between two virtual mobile subnets can fragment the virtual subnet into smaller subnets. Likewise, some MHs can cause subnets to be merged, resulting in bigger subnets. This is illustrated in Figure 2. When the mobile subnets merge to form bigger subnets, the routing algorithm can accept the new subnet by updating all the nodes' routing tables (RTs). This is however very inefficient. In our routing protocol, we forgo this inefficient scheme and only choose to update the affected MHs' associativity tables, which is already an inherent part of the MH's radio data-link layer functions. Likewise, this applies to partitioning subnets.

The property of a mobile subnet states that if both the SRC and DEST are elements of the subnet, a route or routes should exist unless the subnet is partitioned by some subnet-bridging MHs. Mobile subnets can be used to support nomadic collaborative computing, and the collaboration partners can grow in size when two collaboration groups join or when new mobile users join by coming into range.

3.6. CONCURRENT MHS' MOVEMENTS

In reality, concurrent moves by SRC, INs and DEST exist. Hence, consistency is needed to ensure that multiple route re-constructions or updates will ultimately converge and no deadlock or stale routes will exist.

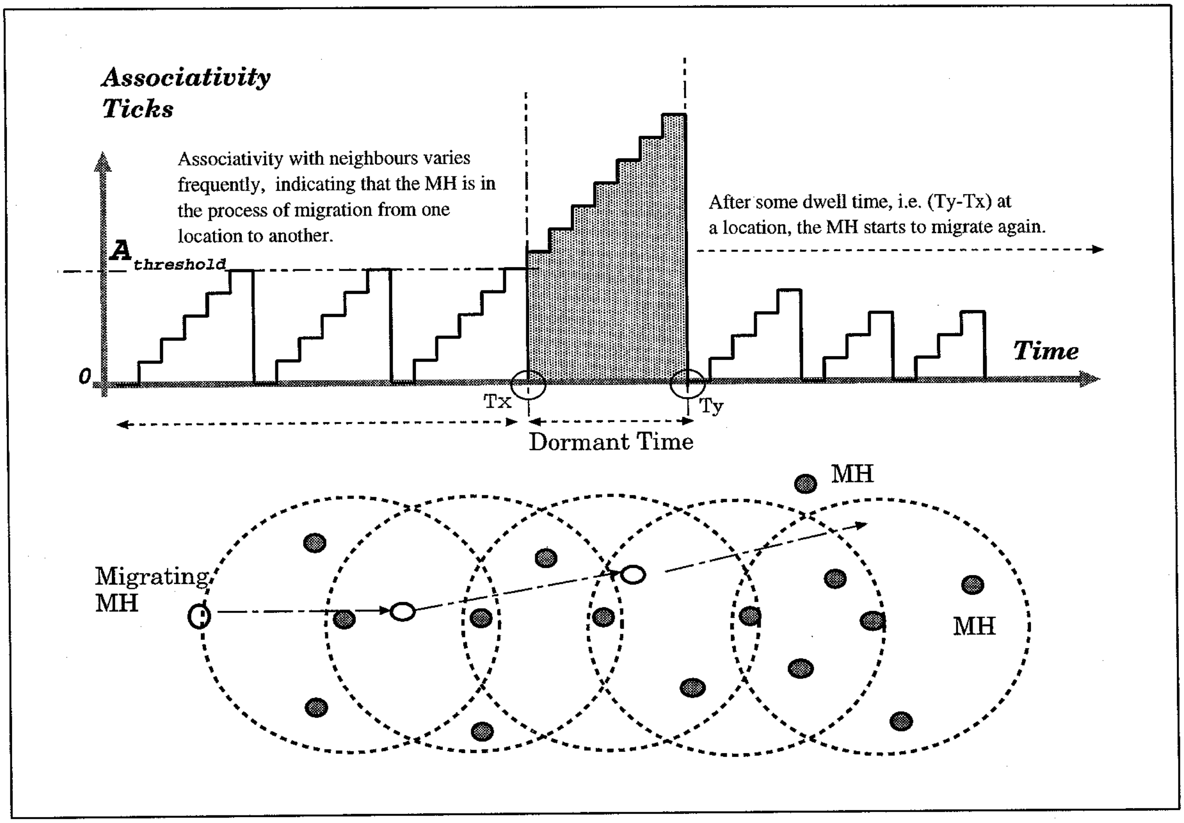


Figure 3. Time and spatial representation of associativity of a MH with its neighbours.

4. **Associativity-Based Routing** (ABR)

ABR is a compromise between broadcast and point-to-point routing. Similar to [8], ABR only maintain routes for sources that actually desire routes. However, ABR does not employ route re-construction based on alternate route information stored in INs (thereby avoiding stale routes). In addition, routing decisions are performed at the DEST and only the best route will be selected and used while all other possible routes remain passive. This, therefore, avoids packet duplicates. Furthermore, the selected route tends to be more long-lived due to the property of associativity, which is described below.

4.1. RULE AND PROPERTY OF ASSOCIATIVITY

4.1.1. *Rule of Associativity*

This rule states that a MH's association with its neighbour changes as it is migrating and its transiting period can be identified by the associativity "ticks". The migration is such that after this unstable period, there exists a period of stability, where the MH will spend some dormant time within a wireless cell before it starts to move again.[2] The threshold where the associativity transitions take place is defined by $A_{\text{threshold}}$, as shown in Figure 3.

---

[2] Refers to moves causing link changes with the MH's neighbours.

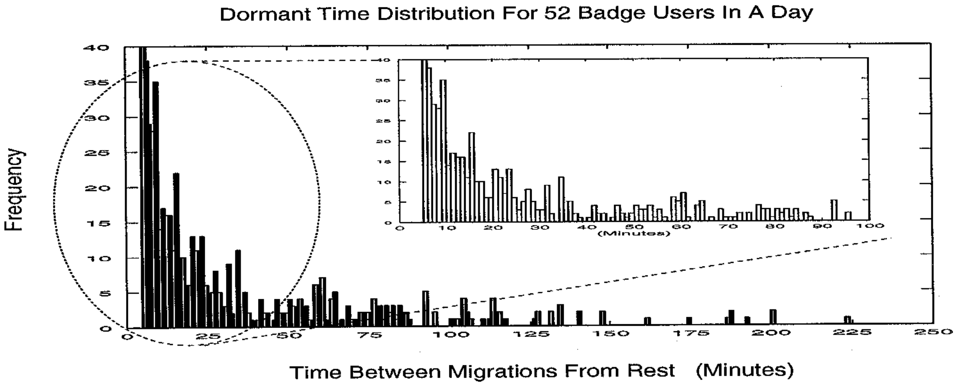


Figure 4. Dormant time distribution of 52 badge wearers in a day at the Computer Laboratory.

Table 1. Dormant time distribution of 52 badge wearers in a week at the computer laboratory.

Dormant time (minutes)					
Distributions	Day of the week				
	Mon	Tue	Wed	Thur	Fri
Maximum	299.15	277.00	281.68	223.06	297.64
Minimum	5.08	5.06	5.10	5.01	5.02
Mean	35.79	36.26	41.08	40.84	47.99
Standard deviation	46.63	50.88	50.55	55.40	62.81

Associativity ticks are updated by the MH's data-link layer protocol, which periodically transmits beacons identifying itself and constantly updates its associativity ticks in accordance with the MHs sighted in its neighbourhood. In a scenario where an *ad-hoc* WLAN has a wireless cell size of 10 m with a MH's minimum migration speed of 2 m/s and a beacon transmission interval of a second, the maximum possible associativity ticks of the migrating MH with its neighbours is five. Likewise, the neighbouring MHs will also record associativity ticks of no more than five. This value is $A_{\text{threshold}}$ and any associativity ticks greater than this threshold implies periods of association stability.

To further support our claim that a mobile user will have some dormant time at a location before it starts to move again, we gathered the mobility traces of 52 badge wearers from the Active Badge System[3] for five consecutive days (from 8 am till 6.30 pm) at the Cambridge University Computer Laboratory.[4] The traces provide specific information on the sighted location (domain and network identifiers), the wireless cell (badge sensor identifier), the MH (the badge identifier) and the time when the MH is sighted (time-stamps).

Figure 4 and Table 1 show the dormant time distributions obtained for a day and a week respectively. The average dormant time ranges from 35.79 to 47.99 mins. Hence, we believe that a practical mobile user will spend some dormant time at a location before he/she decides to move again.

---

[3] Details available from http: //www.cl.cam.ac.uk/abadge/documentation/abwayin.html.

[4] Details available from http: //www.cl.cam.ac.uk/site-maps/site-maps.html.

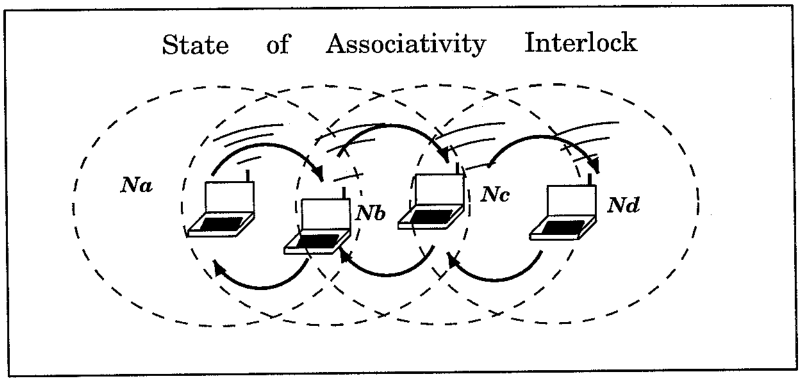


Figure 5. Interlocking phenomenon in *ad-hoc* mobile networks.

4.1.2. *Property of Associativity*

A MH is said to exhibit a high state of mobility when it has low associativity ticks with its neighbours. On the other hand, if high associativity ticks are observed, the MH is in the stable state and this is the ideal point to select the MH to perform *ad-hoc* routing. Consequently, if all the MHs in a route have high associativity ticks, an inter-locking phenomenon arises where "my" degree of associativity ticks will be high if "you" do not move out of reachability (i.e. symmetric mutual-dependent property) and are in stable state, as illustrated in Figure 5. The associativity ticks are reset when the neighbours or the MH itself moves out of proximity, not when the communication session is completed and the route made invalid.

4.1.3. *Applicability to* BS-*Oriented* WLANs

The properties of associativity can also be applied to BS-oriented WLANs. When a MH "sees" a BS, its associativity ticks with the BS will be high. But these associativity ticks will be reset on BS's failure (equivalent to an associated node moving away). Hence, under such circumstances, the MH can apply associativity-based *ad-hoc* routing to re-route its packets to its neighbouring MHs who may have access to other BSs. In this manner, robustness can be achieved during BSs' failures.

4.2. NEW ROUTING METRICS

Conventional routing qualities are characterised by:

- Fast adaptability to link changes (recovery time).
- Minimum hop path to destination.
- Propagation delay.
- Loop avoidance.
- Link capacity.

However, is fast adaptability to changes in network topology a plus? As mentioned earlier, some protocols go to the extreme of frequent broadcasts to attain fast route convergence. In an *ad-hoc* mobile network, fast adaptability at the expense of excessive radio bandwidth consumption is undesirable. The qualities of a good route should not only include the number of hops and the round trip propagation delay. We identify the following new routing metrics:

Table 2. The ABR route selection algorithm.

Let S_i be the set of possible routes from SCR → DEST, where $i = 1, 2, 3\ldots.$
Let RL^i_j be the relaying load in each node j of a route in S_i, where $j = 1, 2, 3\ldots.$
Let $\mathrm{RL}_{\max}$ be the maximum route relaying load allowed per MH.
Let $\mathrm{AT}_{\text{threshold}}$ be the min. associativity ticks required for association stability.
Let AT^i_j represent the associativity ticks in each node j of a route in S_i
Let H_i represent the aggregate degree of association stability of a route in S_i.
Let L_i represent the aggregate degree of association instability of a route in S_i.
Let $H_{i_{\text{ave}}}$ represent the average degree of association stability of a route in S_i.
Let $L_{i_{\text{ave}}}$ represent the average degree of association instability of a route in S_i.
Let Y_i represent the number of nodes of a route in S_i having acceptable route relaying load.
Let U_i represent the number of nodes of a route in S_i having unacceptable route relaying load.
Let $Y_{i_{\text{ave}}}$ represent the average acceptable route relaying load factor.
Let $U_{i_{\text{ave}}}$ represent the average unacceptable route relaying load factor.

```
Begin
    For each route $i$ in $S_i$
    Begin
    $a \leftarrow 0$
      For each node $j$ in route $S_i$
      Begin
          If $(\mathrm{AT}^i_j \geq \mathrm{AT}_{\text{threshold}})$ $H_i++$;
            else $L_i++$;
          If $(\mathrm{RL}^i_j \geq \mathrm{RL}_{\max})$ $U_i++$;
            else $Y_i++$;
          $a++$;
      End
      $H_{i_{\text{ave}}} = H_i/a$; $L_{i_{\text{ave}}} = L_i/a$;
      $U_{i_{\text{ave}}} = U_i/a$; $Y_{i_{\text{ave}}} = Y_i/a$;
    End

    Best Route Computation
    Let the set of acceptable routes with $U_{i_{\text{ave}}} = 0$ and $H_{i_{\text{ave}}} \neq 0$ be $P_l$, where $P_l \subseteq S_i$
    Begin
        * Find Route With Highest Degree of Association Stability*
          Compute a route $k$ from $P_l$ with $H_{k_{\text{ave}}} > H_{l_{\text{ave}}}, \forall\, l \neq k$;
          or if a set of routes $K_n$ exists such that $H_{K1_{\text{ave}}} = H_{K2_{\text{ave}}} \cdots = H_{Kp_{\text{ave}}}$
          where $n = \{1, 2, 3, \ldots, p\}$
          Begin
              * Compute Minimum Hop Route Without Violating Relaying Load*
                Compute a route $K_k$ from $K_n$ with Min$\{K_k\}$ < Min$\{K_m\}$, $\forall\, m \neq k$;
                or if a set of routes $K_o$ exists such that Min$\{K_1\}$ = Min$\{K_2\} \cdots$ = Min$\{K_q\}$,
                where $o = \{1, 2, 3, \ldots, q\}$
                Begin
                    * Multiple Same Associativity & Minimum Hop Routes Exists*
                      Arbitrarily select a minimum hop route $K_k$ from $K_o$
                End
          End
    End
End
```

- Longevity of a route.
- Relaying load of INs supporting existing routes.
- Knowledge of link capacities of the selected route.

The "longevity" of a route is important as the merits of a shorter but short-lived route will be denigrated due to frequent data flow interruptions and the need for route re-constructions. In addition, even relaying load distribution is important in an *ad-hoc* mobile network, as no one particular MH should be unfairly burdened to support many routes and perform many packet-relaying functions. This also alleviates the possibility of network congestion. Finally, since the associativity of a MH with its neighbours also reflects the number of contenders within a wireless cell, the approximate aggregated throughput for the selected route can be made known to the mobile user prior to transmission. This, therefore, allows the user to either proceed with or abort the transmission.

4.3. ROUTE SELECTION RULES

Given a set of possible routes from the source node (SRC) to the destination node (DEST), if a route consists of MHs having high associativity ticks (therefore indicating spatial and connection stability), then that route will be chosen by the DEST, despite other shorter hop routes. However, if the overall degree of association stability of two or more routes are the same, then the route with the minimum hops will be chosen. If multiple routes have the same minimum-hop count, then one of the route is arbitrarily selected. The ABR route selection algorithm, which is executed at the DEST, is formally stated in Table 2.

5. ABR **Protocol Description**

The associativity-based routing protocol consists of three phases, namely

- Route discovery phase.
- Route re-construction (RRC) phase.
- Route deletion phase.

Initially when a source node desires a route, the route discovery phase is invoked. When the link of an established route changes due to SRC, DEST, INs or subnet-bridging MHs' migration, the RRC phase is invoked. When SRC no longer desires the route, the route deletion phase is initiated. These three phases will be discussed below. A summary of the characteristics of the routing protocol is presented in Table 11.

5.1. ROUTE DISCOVERY PHASE

The route discovery phase allows an approximation of the data throughput associated with the selected route to be computed. This is achieved through the knowledge of associativity ticks of neighbours in the route and the relaying load of nodes supporting the route. The route discovery phase consists of a broadcast query (BQ) and an await reply (REPLY) cycle, which is described below.

- BQ-REPLY *cycle*

Initially all nodes except those of DEST's neighbours, have no routes to the DEST. A node desiring a route to the DEST broadcasts a BQ message, which is propagated throughout the *ad-hoc* mobile network in search of MHs which have a route to the DEST. Here, a sequence

Table 3. BQ control packet.

TYPE	SRC ID	DEST ID	LIVE	IN IDs	METRICS	SEQ NO.	CRC

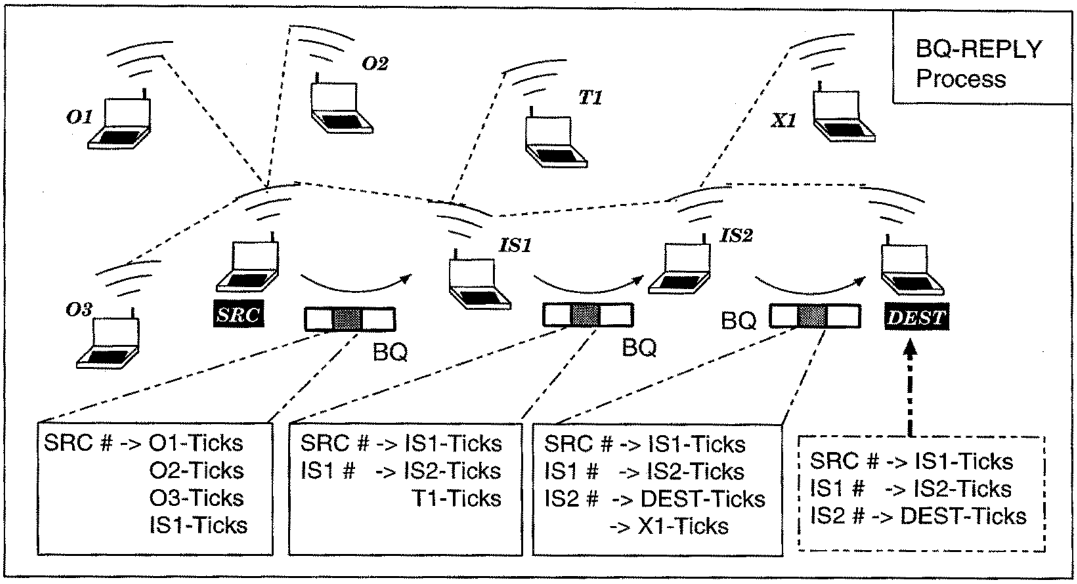


Figure 6. Updating associativity metric during BQ packet propagation.

number is used to uniquely identify each BQ packet and no BQ packet will be broadcast more than once.

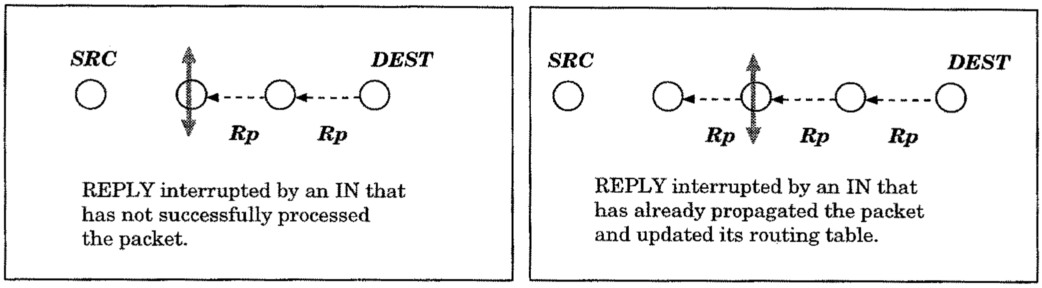
Once the BQ query has been broadcast by the SRC, all INs that receive the query will check if it has previously processed the packet. If affirmative, the query packet will be discarded, otherwise the node will check if it is the DEST. If it is not the DEST, the IN appends its MH address/identifier at the IN IDs field of the query packet and broadcasts it to its neighbours (if it has any). The associativity ticks with its neighbours will also be appended, along with its relaying load, link propagation delay and the hop count.

The next succeeding IN will then erase its upstream node's neighbours' associativity ticks entries and retain only those concerned with itself and its upstream node. In addition, because of the association ticks symmetry between nodes, the associativity ticks received from the upstream node can be checked for validity. In this manner, the query packet reaching the DEST will only contain the intermediate MHs' addresses (hence recording the path taken) and their associativity ticks (hence recording the stability state of the INs supporting the route) and the route relaying load, together with information on route propagation delays and hop count. The process is illustrated in Figure 6. The resulting BQ packet is variable in length and its format is shown in Table 3.

The DEST will, at an appropriate time after receiving the first BQ packet, know all the possible routes and their qualities. It can then select the best route (based on the selection criteria mentioned earlier) and send a REPLY packet back to the SRC, via the route selected. This causes the INs in the route to mark their routes to DEST as valid and this means that all other possible routes will be inactive and will not relay packets destined for the DEST, even if they hear the transmission. This, therefore, *avoids duplicated packets* from arriving at the DEST.

Table 4. REPLY control packet.

TYPE	SRC ID	DEST ID	INs IDs	SEQ ID	ROUTE QUALITIES	CRC



REPLY interrupted by an IN that has not successfully processed the packet.

REPLY interrupted by an IN that has already propagated the packet and updated its routing table.

Figure 7. REPLY interruption caused by unexpected INs' movements.

While the BQ query packet propagates to the DEST, each node relaying the BQ packet will know its hop count from the SRC. Likewise, when the REPLY packet propagates back to the SRC, the INs can also compute their distances to the DEST. The REPLY packet is variable in length and has the format shown in Table 4.

- *Case when* SRC *never receives* REPLY

There may be some rare instances when the SRC never receives DEST's REPLY because of some unexpected "not-yet-selected" INs' movement. In such circumstances, the SRC will eventually BQ_TIMEOUT and sends another BQ query. Since the downstream neighbour of the migrating IN realises the associativity change, it will send a RN[STEP=1][5] (Route Notification) packet in the downstream direction, deleting all the downstream nodes' invalid routing table entries. Another situation occurs when a selected IN moves while the REPLY propagation is still in progress. The upstream neighbour of the migrating node will perform a LQ[H][6] (Localised Query) process to discover a new partial route, while the downstream neighbour sends a RN[1] packet towards the DEST, thereby erasing all invalid downstream nodes' routing entries. Hence, while the RRC is in progress, the REPLY packet continues to propagate towards the SRC. Figure 7 illustrates these two scenarios.

5.2. ROUTE RE-CONSTRUCTION (RRC) PHASE

The route maintenance phase performs the following operations :

- Partial route discovery.
- Invalid route erasure.
- Valid route update.
- New route discovery (worst case).

These operations may be invoked by any of the four moves mentioned earlier. Before concurrent moves are analysed, it is essential to examine the consequence of individual node's movements. In the proposed routing protocol, the selected route is more likely to be long-lived due to the property of associativity. However, if unexpected moves do occur, the protocol will

---

[5] RN[1] will be explained in the next Section.

[6] LQ[H] will be explained in the next Section. Note that "H" refers to "LIVE" of BQ and LQ packets.

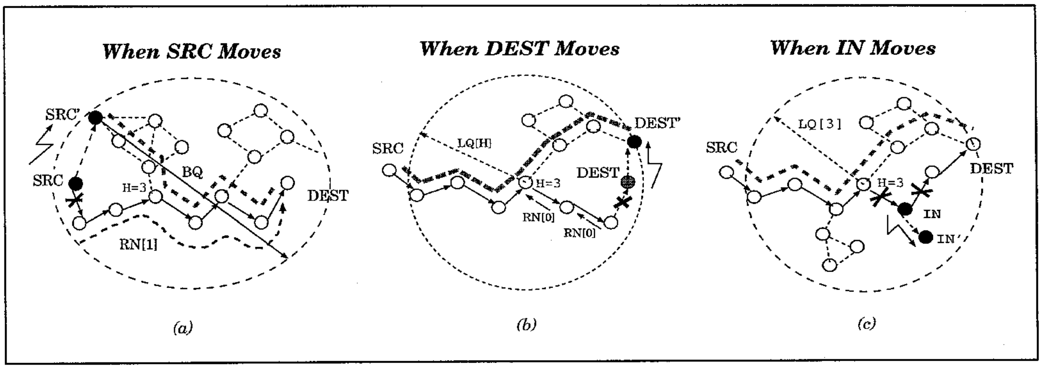


Figure 8. Route maintenance when SRC, DEST and IN moves.

Table 5. RN control packet.

TYPE	ORG ID	SRC ID	DEST ID	SEQ ID	STEP	DIR	CRC

attempt to quickly locate an alternate valid route without resort to broadcast query, unless necessary. The following narrations shall refer to Figures 8a, b and c respectively.

5.2.1. SRC *Node Movements*

Since the routing protocol is source-initiated, any moves by the SRC will invoke an RRC process equivalent to that of a route initialisation, i.e. via a BQ_REPLY process. It will be shown later that this avoids “multiple-RRCs” conflicts as a result of concurrent nodes’ movements.

5.2.2. DEST *node movements*

When the DEST moves, the DEST’s immediate upstream neighbour (i.e. the pivoting node) will erase its route. It then performs a LQ[H] process to ascertain if the DEST is still reachable. “H” here refers to the hop count from the upstream node to the DEST. If the DEST receives the LQs, it will select the best partial route and send a REPLY, otherwise the LQ_TIMEOUT period will be reached and the pivoting node will backtrack to the next upstream node.

During the backtrack, the new pivoting node will erase the route through that link and perform a LQ[H] process until the new pivoting node is greater than half hop$_{\text{src}-\text{dest}}$ away from the DEST or when a new partial route is found. If no partial route is found, the pivoting node will send a RN[1] packet back to the SRC to initiate a BQ process. While the RN packet is fixed in length, the LQ packet is not. The formats of the RN and LQ packets are shown in Tables 5 and 6.

The ORG ID is the pivoting node ID while the SRC and DEST IDs identify the route. STEP=0 in the RN control packet means that the backtracking process is to be performed one hop at a time (in the upstream direction) while a STEP=1 implies that the RN packet will be

Table 6. LQ control packet.

TYPE	SCR ID	DEST ID	INs IDs	METRICS	SEQ ID	LIVE	CRC

propagated straight back to the SRC to invoke a BQ process or to the DEST to erase invalid routes. The DIR flag serves to indicate the direction of RN[1] propagation.

5.2.3. *Intermediate Node* (IN) *Movements*

- *Upper arm* INs' *moves*

The "upper arm" of a route refers to the INs and the DEST that contribute to half the route length from SRC to DEST. When any IN moves, its immediate upstream node (i.e. the pivoting node) removes its outgoing node entry and its immediate downstream neighbour propagates a RN[1] packet towards the DEST, thereby deleting all the subsequent downstream nodes' invalid routing entries. A new partial route to the DEST needs to be found.

A LQ[H] process is then invoked by the pivoting node to locate alternate partial routes. The DEST may receive multiple LQs, hence it needs to select the best partial route and return a REPLY to the pivoting node. This causes all INs between DEST and the pivoting node to update their RTs. On receiving the REPLY, the pivoting node updates its RT entries and appends the next hop (outgoing) node ID into the data packet. This ensures that only one partial route is selected.

As before, the LQ[H] process is performed based on a suitable H value. If the pivoting node is X hops away from the DEST via the previous active route, then $H = X$ will be used in the hope that the DEST is still within X hops range (reachable via other paths) or shorter. This, therefore, attempts to rebuild partial paths of equal or shorter lengths (i.e. *route optimisation* during RRCs).

However, if no partial route exists, LQ_TIMEOUT will expire and a RN[0] packet will be sent by the pivoting node to the next upstream node, and the cycle repeats until the next pivoting node has a hop count greater than half $\text{hop}_{\text{src}-\text{dest}}$ or when a new partial route to the DEST is found.

- *Lower arm* INs' *moves*

The "lower arm" refers to the SRC and INs that contribute to half the route length from SRC to DEST. If any of these nodes moves, RN[1] packet will be propagated downstream towards the DEST, and the pivoting node will perform LQ[H] and await the DEST's REPLY. If no REPLY is received, a RN[0] packet is sent to the next upstream node and the new pivoting node then invokes the LQ[H] process again, but with a different value of H. The cycle proceeds until the new pivoting node is the SRC (where the BQ process will be initiated to discover a new route) or a partial route to the DEST is found.

5.2.4. *Subnet-Bridging* MH *Movements*

The migration of a subnet-bridging MH beyond the radio coverage of its neighbouring MHs will cause the mobile subnet to be partitioned. If an existing route does not span across the fragmented subnets, the route is not affected and only the subnet-bridging MH's upstream and downstream neighbours need to update their route and associativity entries. All other MHs remain ignorant and do not perform any route updates.

However, if existing routes span across subnets (i.e. the subnet-bridging MH is an IN of the route), then the route is invalidated as the DEST is no longer reachable, despite any LQ or BQ attempts. Under such circumstances, the LQ_RN cycle will eventually inform the SRC about the partitioning and the SRC can then invoke BQ query several times or it can inform the mobile user about the partitioning and prompt him to try later.

5.2.5. *Concurrent Nodes Movements*

Race conditions exist due to multiple invocations of RRC processes as a result of concurrent movements by SRC, DEST and INs. The following explains why the proposed routing protocol is immune to “multiple-RRCs” conflicts and how one RRC is valid ultimately.

- DEST-*moves* RRC *interrupted by upstream* INs’ *moves*
  When the DEST moves and while the RRC is in progress, any upstream INs moves will cause their respective downstream neighbours’ route to be deleted. The new pivoting node nearest to the SRC will perform the RRC and all other RRCs will be passive when they hear the newer LQ broadcast for the same route. Hence, only one RRC is valid.
- *Upper-arm* IN RRC *interrupted by lower arm* INs’ *moves*
  This is the same as the above-mentioned case. Note that the same argument can be applied to the case when a LQ process has to be aborted and a RN[1] packet has to be sent to the SRC to invoke a BQ but is hindered due to some upstream INs’ movements. The new pivoting node nearest to the SRC will swamp the earlier RRC processes by invoking a new LQ.
- *Lower-arm* IN RRC *interrupted by upper arm* INs’ *moves*
  While a lower arm IN RRC is taking place, any movements by any upper arm INs will not result in a LQ[H] or RN[1] process being initiated since the lower arm IN has earlier sent RN[1] packet downstream to erase invalid routes. If the RN[1] packet does not succeed in propagating towards the DEST, the LQ[H] process initiated by the lower arm IN will also serve to delete these invalid routes.
- *Lower/upper-arm* IN RRC *interrupted by* DEST’s *moves*
  This has no effect on the RRC, as the LQ[H] process uses a localised query approach to locate the DEST. Once the DEST is in its stable state and is reachable from the pivoting node, the RRC process will be successful.
- *Lower/upper-arm* IN RRC *interrupted by* SRC’s *moves*
  While lower or upper arm IN RRC is in progress, any moves by the SRC will result in a BQ, which will swamp out all on-going LQ_REPLY_RN processes related to that route. Hence, unfruitful and stale RRCs will not continue and a new route has to be discovered via the BQ process.
- SRC *and* DEST *nodes moving away from* INs
  When this occurs, RRCs as a result of DEST and SRC moves will be initiated. However, the BQ process initiated by the SRC will again swamp out all unnecessary on-going RRCs.
- DEST *migrating into* SRC’s *radio coverage range*
  When the DEST migrates, RRC is achieved via the LQ[H] process. However, when the DEST is within the SRC’s radio coverage range, packet duplicates will result at the DEST since the DEST now receives packets from the SRC directly and also from the original SRC-DEST route. Hence, to avoid packet duplicates and non-optimal routes, the SRC, on discovering that the DEST is within range and is in stable state, will send a RN[1] packet downstream to erase existing route and will re-establish a new single hop route with the DEST.

Table 7. RD Control packet.

TYPE	SRC ID	DEST ID	LIVE	SEQ NO.	CRC.

5.2.6. LQ-REPLY *Cycle Interruption*

During a LQ-propagation and REPLY-await process, if any of the upstream nodes[7] (i.e. lower arm INs) break up, a RN[1] packet will be propagated downstream, erasing all the downstream INs' routes entries. The existing pivoting node will ignore any subsequent REPLY to its LQ. The new pivoting node will resume with a new LQ-REPLY process.

5.2.7. *Route Erasure and Updates*

In ABR, no attempt is made to retain alternate routes, as maintaining them causes overhead. Only one route will be selected and only one route is valid for a particular route request. The avoidance of using alternate routing information means that problems associated with looping due to INs having stale routes are absent and there is no need for periodic network-wide broadcast and route updates. This is, however, not the case for the source-initiated distributed routing scheme proposed in [8].

Any alternate route will have to be discovered via a LQ or BQ process, which may give rise to better (shorter hop and long-lived) routes. The DEST, on receiving multiple LQs, will select the best route and reply to the SRC. During the LQ_REPLY_RN cycle, invalid INs routes are erased by RN[1] packets and INs forming the new partial route will have their route entries updated when they relayed the REPLY packet from the DEST to the pivoting node. If the LQ_REPLY_RN cycle fails, the subsequent new pivoting node will have its route entries erased by RN[0] packet during the backtrack process. If all the possible backtrack LQ_REPLY_RN cycles fail, all the upstream nodes will have their route entries erased via RN[0] and RN[1] packets and the SRC will then revert back to the BQ_REPLY cycle.

Finally, for the case of BQ query, any INs receiving a BQ and having invalid routes will result in these routes being erased, therefore ensuring that no invalid routes exist in the INs.

5.3. ROUTE DELETION PHASE

When a discovered route is no longer desired, a route delete (RD) broadcast will be initiated by the SRC so that all INs will update their routing table entries. A full broadcast is used compared to directed broadcast. This is so because the nodes in a route change during route re-constructions, hence using directed broadcast will be unsuitable unless the SRC is always informed about any changes to the route path. Similar to BQ, the RD control packet has a LIVE=∞ to achieve a full wave-like broadcast. Its format is shown in Table 7.

5.4. PACKET HEADER

Since a long packet header results in low channel utilisation efficiency, in our proposed protocol, each data packet header will only contain the neighbouring node routing information, not all the nodes in the route. Each IN will renew the next-hop information contained in the header before propagating the packet upstream or downstream. Hence, a hybrid routing scheme

---

[7] Downstream nodes' movements are not a concern in this case.

Table 8. Packet header.

Routing header field	Function
SRC ID	Packet forwarding
DEST ID	Route identification
Sequence no.	Duplicates prevention, uniqueness
Service type	Packet priority
Last IN	Passive acknowledgement
Next IN	Duplicates prevention, routing
Current IN	Acknowledgement, routing

Table 9. Routing table.

Destination	Source	Incoming IN	Outgoing IN	Distance
N_a	N_x	N_z	N_j	4
N_k	N_y	N_i	N_o	3
Total no. of active routes supported (relay load): 2				

which is a combination of broadcast and point-to-point routing is used. The purpose of the individual fields of the packet header is summarised in Table 8.

5.5. DATA FLOW ACKNOWLEDGEMENT

As in PRNs, we employ a *passive acknowledgement* scheme for packets in transition. When a node receives a packet and performs relaying via a radio transmission to its neighbours, its previous neighbour that has sent it the packet will have heard the transmission and hence this is indirectly used as an acknowledgement to the packet sent. On the other hand, *active acknowledgements* will only be sent by the DEST as it no longer has a neighbour to relay the packet to. Hence, this provides a data flow acknowledgement mechanism for packet forwarding in an *ad-hoc* mobile network, which is not present in any of the existing *ad-hoc* mobile routing schemes.

5.6. PACKET RETRANSMISSION

While the data flow acknowledgement scheme allows forwarded packets to be acknowledged, there are situations where the acknowledgements never reach the intended receiver. This can be a result of radio interference which causes a sudden loss of radio connectivity. Hence, if a MH has forwarded a packet and does not receive an acknowledgement within a certain time interval, it retransmits the packet for X times, after which the neighbouring MH will be considered as out-of-reach and RRC procedures will be invoked. If however, the radio link returns before the retransmission counter expires, the packet forwarding process continues, as illustrated in Figure 9.

5.7. ROUTING TABLE (RT)

The RT of a node supporting existing routes is shown in Table 9. The table reveals that every node supporting on-going routes will map incoming packets from a particular upstream

Table 10. Neighbouring table.

Neighbours	Associativity ticks (units)	Link delays (msecs)
N_a	5	75
N_b	15	50

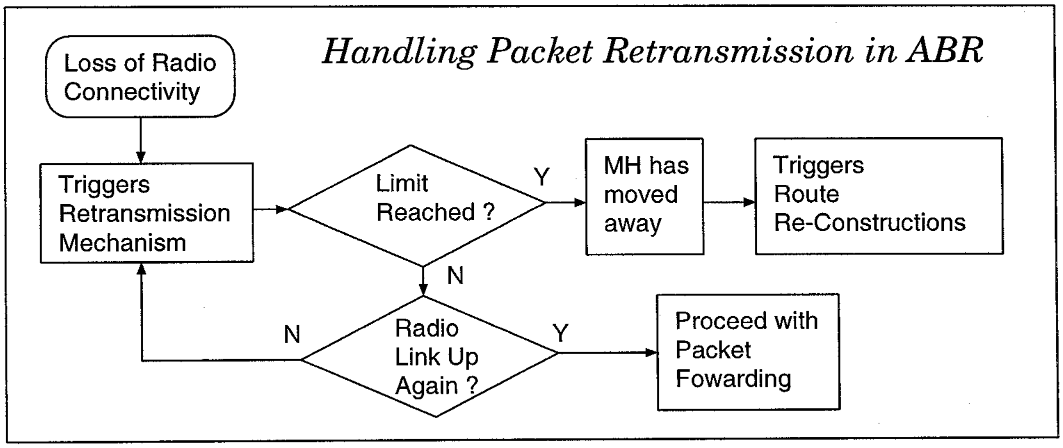


Figure 9. Handling packet retransmission in ABR.

node to the corresponding out-going downstream node. Every node will also keep track of its distance (hop count) to the DEST and a record of the total routes that it is currently supporting.

5.8. NEIGHBOURING TABLE (NT)

The NT is usually updated by the data-link layer protocol, which will generate, receive and interpret beacons from the neighbouring MHs or BSs and pass this information up to the higher protocol layers. Nomadic collaborative applications can then utilise the NT information to update their participants' present and absent lists. The structure of a NT is shown in Table 10.

5.9. CONTROL PACKETS "SEEN" TABLES

While the BQ query process is activated via a radio broadcast, the LQ query process is invoked via a localised broadcast. To avoid MHs from processing and relaying the same BQ or LQ packet twice, BQ and LQ "seen" tables are needed. If the received control packet type, route identifier and sequence number match an entry in the "seen" table list, then the packet is discarded. On the other hand, because the REPLY and RN control packets utilise "directed" broadcast (since intended recipients' addresses are contained in the control packet), "seen" tables for these packets are not necessary.

5.10. ABR PROTOCOL SUMMARY

Table 11 summarises the procedures of the routing protocol under different MHs' associativity states. The outstanding feature is that no RRCs is needed so long as the property of associativity

Table 11. Summary of route reconstructions under varying MHs' migrations.

Associativity valid	Associativity violated					
	INs & DEST moves		SRC moves	Subnet bridging MH moves		Concurrent moves
	Normal case	Worst case		Route within subnet	Route spans across subnets	
No route reconstructions are needed	LQ, REPLY cycle success	BQ, REPLY cycle success	BQ, REPLY cycle success	No route reconstructions are needed	Network is partitioned BQ REPLY cycle will retry before aborting	Ultimately only one route reconstruction cycle is valid

Table 12. Complexity comparison of routing protocols.

Routing protocol	TC	CC
ILS	$O(d)$	$O(2E)$
MS	$O(d^2)$	$O(N^2)$
DBF (link failures)	$O(N)$	$O(N^2)$
NP (QRY-RPY, initialisation)	$O(2d)$	$O(2N)$
NP (QRY-RPY, postfailure)	$O(2l)$	$O(2x)$
NP (FQ-RPY, postfailure)	$O(2l)$	$O(2x)$
ABR (BQ-RPY, initialisation)	$O(d+z)$	$O(N+y)$
ABR (LQ-RPY, postfailure)	$O(l+z)$	$O(x+y)$
Flooding (no control packets)	0	0

interlock remains valid. When this property is violated, the protocol will invoke a LQ or BQ process to quickly locate alternate routes. To further illustrate the simplicity and efficiency of the ABR protocol, we compare our ABR protocol with several existing routing algorithms in terms of time and communication complexities.[8] Table 12 (extracted with permission from [8]) reveals that ABR is simpler and more efficient than all other algorithms compared. The abbreviations used in Table 12 are listed below:

N = number of nodes in the network,
E = number of links in the network,
d = network diameter,
x = number of nodes affected by a topological change,
l = diameter of the affected network segment,
y = total number of nodes forming the directed path where the REPLY packet transits,
z = diameter of the directed path where the REPLY packet transits,
ILS = idealised link–state protocol,
MS = Merlin–Segall routing algorithm [23],
DBF = distributed Bellman–Ford algorithm,
NP = routing protocol proposed in [8],
ABR = associativity-based routing.

---

[8] Time complexity refers to the number of steps required to perform a protocol operation. Communication complexity refers to the number of messages exchanged in performing a protocol operation.

6. Migration-Based Simulation and Performance Evaluation

6.1. SIMULATION MODEL AND ENVIRONMENT

In order to evaluate the performance of the routing protocol during cases when the “rule of associativity” is violated (though this can be less frequent, as verified from the badge trace findings), the routing protocol is implemented within a migration-based *ad-hoc* mobile network simulator.

Unlike other *ad-hoc* mobile network models proposed in [7] and [20], our model is migration based, where MHs are laptops or personal digital assistants (PDAs) users in an in-building environment. The network simulator models MHs migrating in two dimensional space with a wireless cell size of 10 m in diameter. The simulator randomly allocates N MHs (where $N = 30$) on a 400 m^2 space. The resultant *ad-hoc* mobile network connectivity is determined from the overlapping of these wireless cells, i.e. if the shortest distance between MHs is less than the cell radius, these MHs are connected to one another. To evaluate the routing protocol for all possible source-initiated routes, i.e. $N(N-1)$, the resultant *ad-hoc* network so formed must not be partitioned.

For each of the possible routes, each IN belonging to the route is progressively moved away (i.e. moved out of connectivity range with its upstream or downstream neighbours), thereby invalidating the route. RRC procedures are then invoked to locate alternate routes. The average neighbouring factor (N.F.)[9] of a route defines the association of a route with surrounding MHs and is given by:

$$\text{N.F.} = \sum_{i=1}^{N} \frac{nf_i}{N} \,,$$

where

$$nf_i = \frac{\text{no. of neighbours associated with node } i}{\text{maximum possible no. of neighbours for a node}} \,.$$

In the simulator, the maximum possible number of active neighbours associated with a node is 10. To evaluate the protocol behaviour for routes having different N.F., RRCs are performed for each of the possible N.F. till its maximum. This is achieved by gradually moving “free-nodes” (nodes that are not part of and neighbours of the route) into the coverage range of the SRC, DEST and INs. To ensure that the N.F. is ≤ 1, the number of neighbours associated with a node on the route must be within the specified maximum. Hence, a node cannot be moved to the vicinity of the considered node if it causes neighbours overflow in any of the other neighbouring nodes’ cells. Figures 10 and 11 show an *ad-hoc* mobile network connectivity state, using the default nodes allocation seed, before and after nodes’ migrations. The route statistics associated with the initial *ad-hoc* mobile network is shown in Figure 12.

6.2. SIMULATION RESULTS AND OBSERVATIONS

To derive an overall picture of the protocol performance, RRCs are performed for all possible source-initiated routes. To ensure that the results obtained are valid for different initial network nodes’ location and connectivity states, different possible nodes’ allocation seeds for the

---

[9] This should not be confused with the degree of associativity (association stability) of MHs participating in the route.

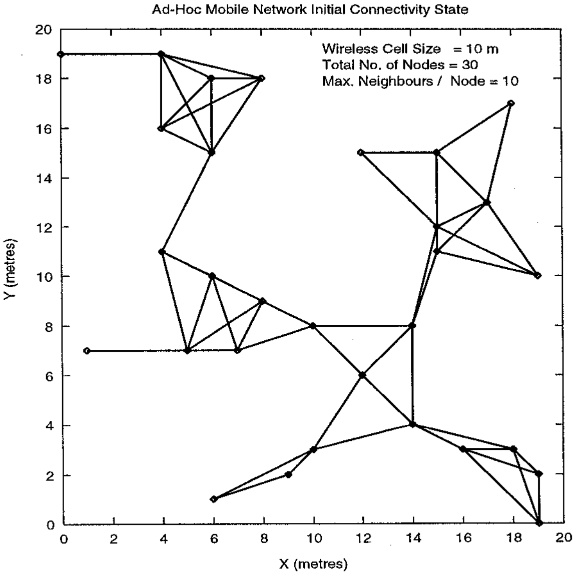


Figure 10. Ad-hoc mobile network topology before nodes' migrations.

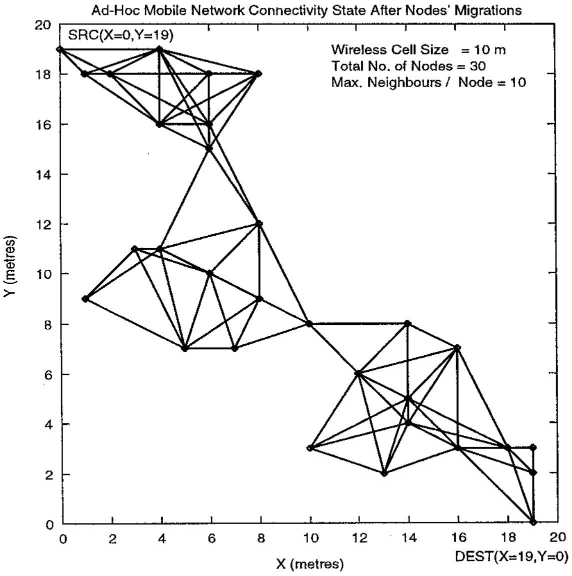


Figure 11. Ad-hoc mobile network topology after nodes' migrations.

random generator are used. For each of the re-constructed route, the following parameters are recorded

- Route length difference (w.r.t. INs' migrations and route N.F.).
- LQ difference (w.r.t. INs' migrations and route N.F.).
- Maximum no. of LQs performed (w.r.t. INs' migrations and route N.F.).

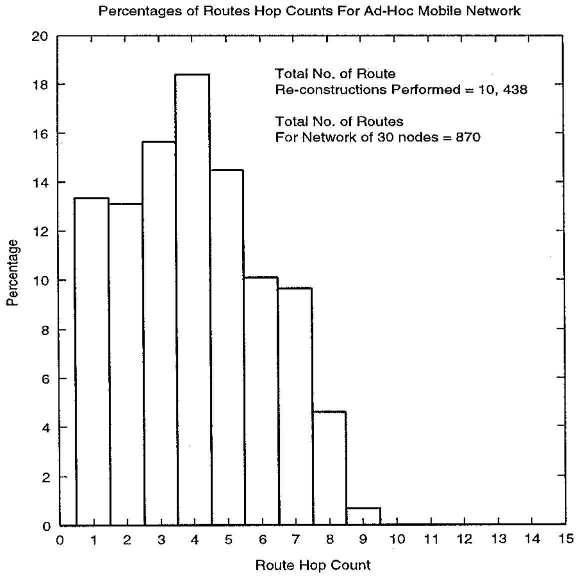


Figure 12. Routes statistics for *ad-hoc* mobile network using default nodes allocation seed.

6.2.1. *Results for a Single Ad-Hoc Mobile Network*

- LQ *difference*

As shown in Figure 13, the minimum possible LQ difference is one since the DEST node is not moved but its upstream nodes. This represents 42.96 percent of the total population since 57.04 percent has LQ differences of greater than one. This implies that the actual number of LQs performed before a successful RRC is less than its worst possible maximum. Note that the results of LQ difference are derived from routes having three or more hops only. Although it is not obvious from Figure 13 that LQ difference increases with increasing N.F., our study on the raw results for routes with similar hop count and IN moves but with different N.F., confirms this point. We found that the average N.F. threshold where LQ difference began to improve occurred at 0.67.

- *Maximum no. of* LQs *performed*

Figure 14 clearly reveals that the maximum number of LQs required before a successful RRC decreases with increasing N.F.. This is due to the exploitation of locality, i.e. increasing the route N.F. increases the probability of a LQ query success. To avoid being misled by the result (due to routes having hop counts of one or two), the network route types are computed (see Figure 12) and clearly the cumulative percentage of routes having $3 \rightarrow 7$ hops are greater than those having one or two hops. A zero maximum LQ is due to one and two hops routes and for cases when the migrated node is equal to half hop or (route hop count -1). From the raw results, the average N.F. threshold which resulted in reductions on the maximum number of LQs performed was found to be 0.66.

- *Path difference*

The difference between the original route length and the new route length (after re-construction) yields the path difference. In the simulator, “LIVE” is set to the hop count from the new pivoting node to the DEST. This implies that none of the discovered route will be longer than

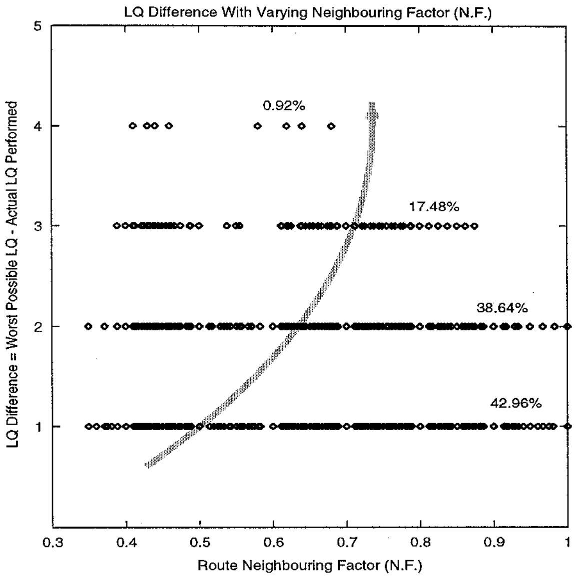


Figure 13. Results of LQ difference for all INs' moves and route reconstructions with varying N.F.

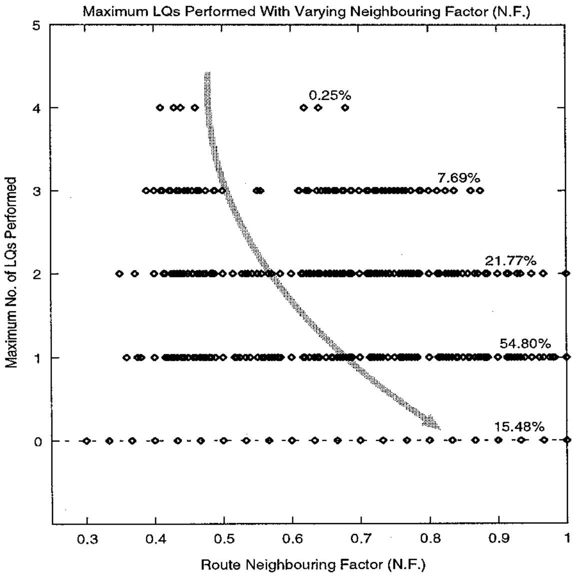


Figure 14. Maximum LQ performed for all INs' moves and route re-constructions with varying N.F.

the original route, as verified by results shown in Figure 15. Note that one and two hop routes do not contribute to the result. Out of a total of 10,438 RRCs, 52.31 percent have shorter routes while 47.69 percent have the same route length, implying that the LQ RRC process can discover shorter new routes. In addition, as the N.F. is increased, the path difference becomes significant. Again from the raw results, we computed the average N.F. threshold which caused improvements in the path difference and this value was found to be 0.74.

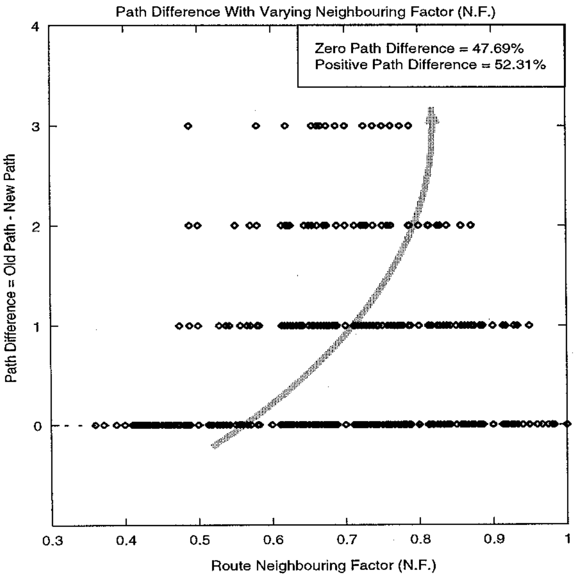


Figure 15. Results of path difference for all INs' moves and route re-constructions with varying N.F.

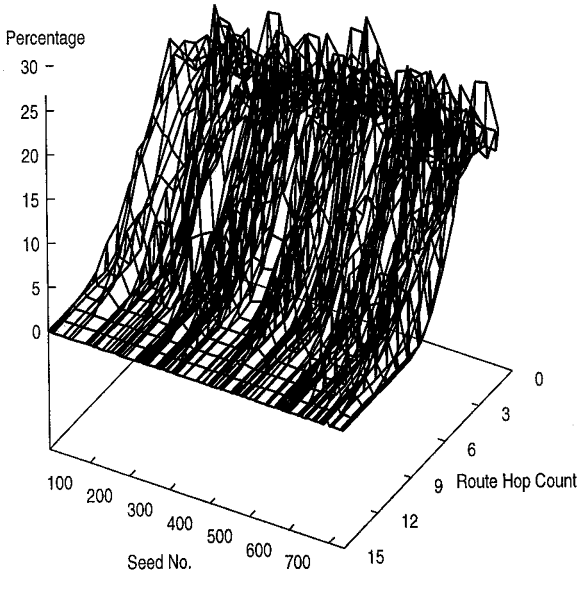


Figure 16. Three-dimensional plot of route characteristics for 113 *ad-hoc* mobile networks under tests.

6.2.2. *Overall Results for Different Ad-Hoc Mobile Networks*

To ensure that the deductions derived from the above simulations are valid for *ad-hoc* mobile networks having different initial nodes' location and connectivity states, a further 113 simulations (a total of 1,155,648 RRCs) are performed for all possible seeds in the 0 → 800 range. The following results are recorded:

- *Overall routes's characteristics*

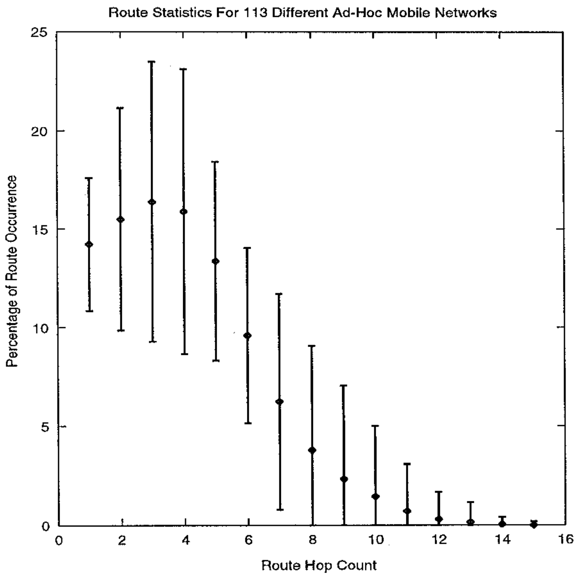


Figure 17. Two-dimensional plot of route characteristics showing mean values and 95% confidence intervals.

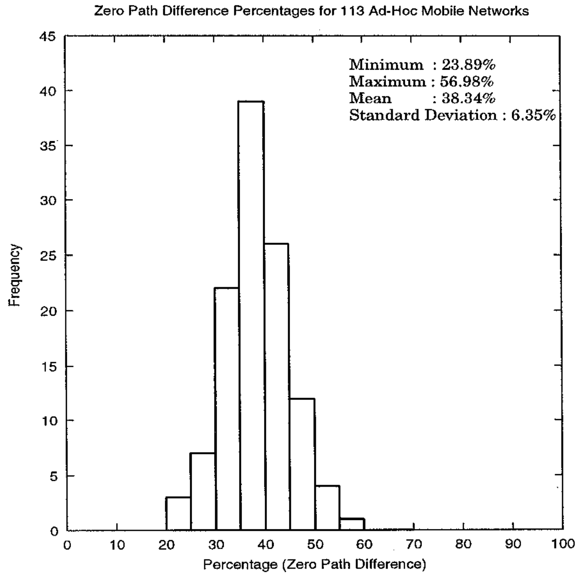


Figure 18. Zero path difference frequency distribution for 113 *ad-hoc* mobile networks.

The resulting route types for each of the possible *ad-hoc* mobile networks are plotted to give an overall view of the networks' route characteristics. Both Figures 16 and 17 show that for every *ad-hoc* mobile network under test, the route characteristics are different. In particular, minor variations exist over routes with larger hops while considerable variations exist for routes in the 1 $\rightarrow$ 10 hops range. This is due to the constraint imposed by the number of MHs confined in a 400 m^2 physical space. The 95 percent confidence intervals and mean percentage for different route type occurrences are shown in Figure 17. Note that routes with eight hops or more have positively-skewed frequency distributions while others have normal distributions.

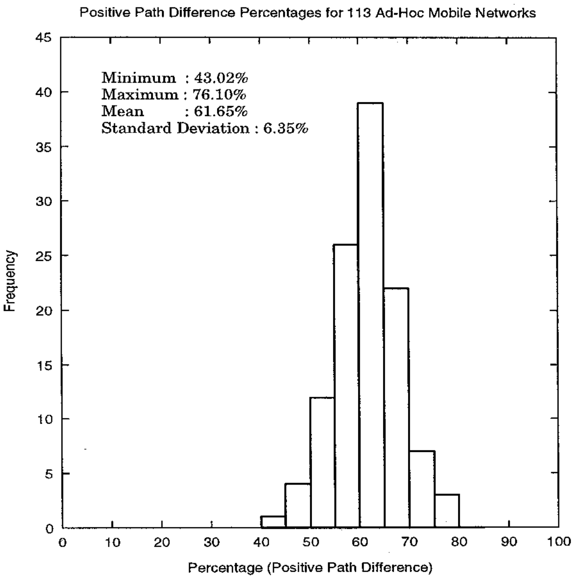


Figure 19. Positive path difference frequency distribution for 113 *ad-hoc* mobile networks.

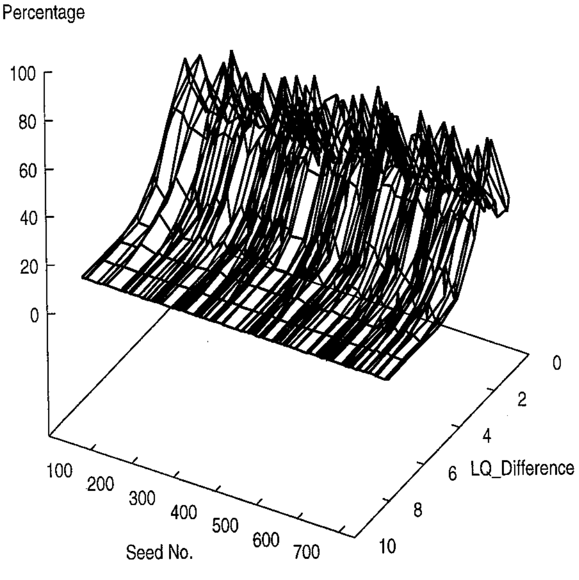


Figure 20. Three-dimensional plot of LQ differences for 113 *ad-hoc* mobile networks.

- *Overall path differences*

Figures 18 and 19 reveal that for each of the 113 different *ad-hoc* mobile networks (with every possible route undergoing INs' movements and hence invoking RRC processes over increasing route N.F.), the percentage of the resultant path being shorter than that of the original path has a mean of 61.65 percent, while those having the same path length is 38.34 percent. Hence, the protocol's nature of exploiting the advantages of locality via localised query has allowed the formation of shorter new paths.

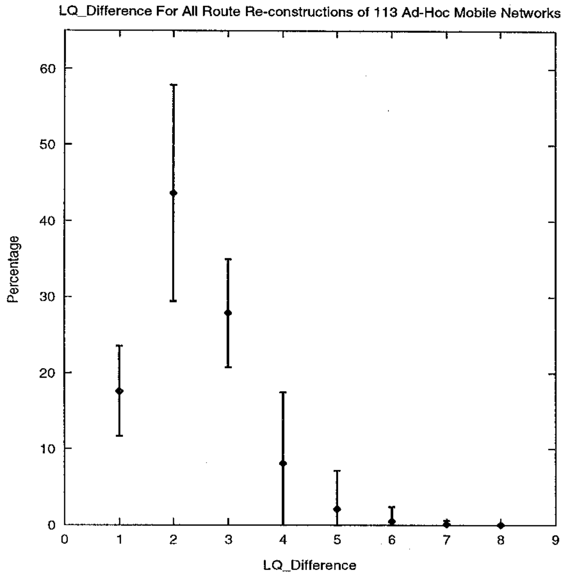


Figure 21. Two-dimensional plot of LQ differences showing mean and 95% confidence intervals.

Table 13. LQ successes and failures statistics (in %).

Process	Min	Mean	Max	Standard dev.
LQ success	20.84	37.95	61.13	8.49
LQ failure	38.87	62.05	79.16	8.49

- *Overall* LQ *differences*

As shown in Figures 16 and 17, since the maximum possible hop count of a route never exceeds 15, the maximum worst case LQ difference shall be 8. This is verified by the results shown in Figures 20 and 21. Two deductions can be derived from these graphs: firstly, the values of LQ differences vary with each set of mobile network under test and secondly, the aggregate percentage of LQ differences in the $2 \rightarrow 8$ range is greater than that of value 1. This infers that the number of LQ operations performed before a successful RRC is mostly less than its worst possible value. Notice again that LQ differences of value greater than four have positively skewed distributions instead of normal distributions.

- *Percentage of* LQ *successes and failures*

Table 13 shows the percentage of LQ successes (i.e., an alternate partial route is located) and LQ failures (i.e., when no partial routes can be located via LQ processes and a BQ process has to be invoked.)

Table 14. LQ failures contributed by routes having different hop count (in %).

LQ failures	Min	Mean	Max	Standard dev.
Routes < 5 hops	9.13	22.26	49.25	7.4
Routes ≥ 5 hops	11.77	39.79	63.80	10.44

Table 15. LQ failures contributed by routes having different N.F. (in %).

LQ failures	Min	Mean	Max	Standard dev.
Routes with N.F. < 0.7	15.99	36.50	56.36	8.94
Routes with N.F. ≥ 0.7	10.93	25.56	43.06	6.59

Table 16. Percentage of BQs invoked (in %).

Percentage of BQs invoked:	Min	Mean	Max	Standard dev.
(Total BQs invoked)/(total no. of moves)	39.70	56.72	78.33	8.30
Due to LQ failures only	23.04	32.07	39.44	3.19
Due to 2 hop routes	11.19	24.65	44.50	6.82

The results initially give the impression that the LQ RRC process performs badly, given the higher percentage of LQ failures. However, this is not the case since the high LQ failures are a result of the greater number of shorter hop routes, for which the protocol aborts the LQ process quickly and resorts to BQ in order to avoid long recovery time. This is verified by the results shown in Table 14. Out of the 62.05 percent of LQ failures, 22.26 percent are due to shorter hop routes. Hence, LQ failures as a result of shorter hop routes need not be a disadvantage, since little difference exists between LQ and BQ queries for shorter hop routes.

In addition, Table 15 reveals that the percentage of LQ failures as a result of routes having N.F. < 0.7 is greater than those having N.F. ≥ 0.7. A N.F. threshold of 0.7 is used since the N.F. ranges from $0.35 \rightarrow 1.0$. Because the chances of a LQ success only improve when the N.F. is high, this also partially explains why the percentage of LQ successes is lower than LQ failures.

- *Percentage of* BQ*s invoked*

Table 16 shows that the percentage of BQs invoked is > 50 percent (w.r.t. the total number of INs' moves). This again gives the illusion that the protocol performs badly. A breakdown of this percentage (i.e. 56.72 percent) reveals that up to 24.65 percent are due to 2-hop routes while 32.07 percent are due to routes that invoke BQs as a result of LQ failures. Hence, BQs invoked as a result of shorter hop routes should not be considered as a demerit, as explained earlier.

Finally, viewing from another perspective, Table 17 reveals the percentage of BQ invoked under high and low N.F.. For routes having N.F. ≥ 0.7, the percentage of BQ invoked is less than those with N.F. < 0.7, indicating that at high N.F., the probability of BQ being invoked is reduced.

Table 17. Percentage of BQs invoked under different route N.F. (in %).

BQs invoked when:	Min	Mean	Max	Standard Dev.
Route N.F. < 0.7	21.90	29.51	34.73	2.08
Route N.F. ≥ 0.7	10.43	27.21	48.10	8.94

7. Comparisons with Existing Protocols

7.1. CLUSTER-BASED ROUTING

Cluster-based routing for mobile networks is not new. Early work includes those proposed by [7]. The cluster-based routing scheme proposed by [20] employs a two-level routing strategy. It uses existing routing schemes (such as link-state or distance-vector Bellman–Ford routing) to derive network topology and link metric information in order to compute the shortest path to each destination. Each node consists of a cluster table (housing the mapping between the nodes and their clusters), a routing table (housing the destination node cluster ID, the next hop and the hop count), and an all-route table (housing all possible paths from a node to every possible destinations). Hence, the memory requirements are substantial, so are the control traffic required to maintain and update these tables.

Another deficiency associated with the protocol is that during the route construction phase, all routes are constructed between all pairs of nodes, i.e. routes are not constructed based on need. This means that maintaining these routes (even though they are not required) causes bandwidth, power and computation overheads. Because nodes are grouped into clusters, cluster management (such as cluster assignments, new cluster updates, cluster deletions, redundant clusters checks, etc.) results in further overheads.

In cluster-based routing, route maintenance is basically cluster maintenance – where the new cluster information has to be propagated throughout the network. The protocol proposed by [20] has not yet addressed issues on packet duplicates, retransmission and data acknowledgement problems associated with *ad-hoc* mobile networks.

7.2. SOURCE-INITIATED ROUTING

This routing scheme proposed by [8] for mobile PRNs employs the principle of source-initiated and demand-based routing. This scheme forgoes the conventional bandwidth-intensive routing approach which requires periodic broadcast to maintain up-to-date network topology and link metric information.

Although our protocol is also source-initiated and demand-based, there are many differences. Firstly, the protocol proposed by [8] allows any IN that claims to have route/s to the DEST to reply to a query. Secondly, as the reply propagates back to the SRC, one or more routes are "created" at each node participating in the route. This means that while other nodes can learn about the route, multiple route paths may result in packet duplicates. Thirdly, there is no route selection rules and hence routes are not optimised.

The route re-construction phase employs failure-query, await-reply, route-erase and backtrack methodologies to progressively locate alternate routes. Any neighbouring node that receives the failure query broadcast and have a route to the DEST can send back a reply. However, it is not known how the information for alternate routes are kept up-to-date. The backtrack methodology used does not employ a quick abort strategy – such as the abort after performing backtrack for a maximum of half $\text{hop}_{\text{src}-\text{dest}}$ times, as mentioned in our protocol. This means that in the worst case, the backtrack will continue until the next node is the SRC, after which the BQ process will be initiated. This, therefore, incurs a relatively long recovery time.

7.3. DYNAMIC SOURCE ROUTING

A similar approach of avoiding periodic router advertisements is adopted in [13]. Source routing is used instead of hop-by-hop packet routing and each packet carries explicit route information. Route caches are used to store discovered routes and routes that an MH may learn over time. However, the accuracy of the route cache's content is subjected to proper expiration and stale route erasure procedures.

Although routes are source-initiated and discovered through a discovery broadcast, only one route is considered instead of all other possible routes. In addition, no considerations have yet been given to other routing metrics (such as link capacities, delays, associativity states, etc.). Hence, the route discovered is not necessary the best route and may not be long-lived.

During the route maintenance phase, the operation of a route is monitored and the sender is informed of any routing errors. However, it is unclear what specific remedial actions can be taken when a route is truncated. There are no partial route discovery, quick abort and route reconstruction procedures.

8. **Dynamic Cell Size Adjustment Scheme** (DCSAS)

The above-mentioned Sections reveal that associativity is closely related to routing. High association of a node with other nodes enhances its communication capability and produces shorter hop routes. However, an increase in the number of active nodes in a wireless cell can cause greater contention for the available wireless bandwidth, resulting in lower throughput per MH. A suggestion by [14] for an environment which is congested with MHs is to dynamically adjust the transmission power of each MH such that both the cell size and the N.F. is reduced in order to achieve a reasonably high throughput while still maintaining acceptable routing performance. It has also been shown earlier by [10] that spatial channel reuse obtained by reducing MH transmission power to a level where only a few neighbours are within range, gives rise to an improved throughput. This is also further advocated in [28].

The throughput of a wireless network depends on the media access control (MAC) protocol and the spectrum bandwidth-allocation strategies. The MAC layer protocols proposed for WLANs include the following distributed schemes:

- Pure/continuous ALOHA.
- Slotted/discrete ALOHA.
- GRAP (group randomly addressed polling).
- CSMA (carrier sense multiple access)[10].

For a high speed wireless local area network (HSWLAN) employing the GRAP protocol [5], a double increase in the number of active MHs results in a 10 percent drop in the throughput and a triple increase in the delay. For the CSMA case, an increase in both the contention for wireless channels and the "hidden-terminal" phenomenon results in lower throughput and higher delay. Note that IEEE 802.11 proposes a CSMA/CA plus acknowledgement scheme [29] that avoids the hidden terminal problem.

Consequently, the dynamic cell size adjustment scheme (DCSAS) allows the wireless cell capacity associated with each *ad-hoc* MH to be increased but the formation of longer routes

---

[10] Both the IEEE 802.11 committee proposed DF-WMAC (distributed foundation – wireless media access control) and the ETSI HIPERLAN MAC [24], are based on CSMA. CSMA was originally designed for radio networks.

Table 18. Percentage reduction in numbers of contenders and wireless cell size.

Parameters	Min.	Mean	Max.	Standard dev.
Percentage reduction in number of contenders	25.00	36.95	60.00	26.96
Percentage reduction in wireless cell size	10.56	35.10	80.00	18.05

may result in longer delays. There is also an increased probability that the *ad-hoc* mobile network will be partitioned into multiple subnets. Power consumption, however, may not be greater as smaller cells require less transmission power.

8.1. Proposed DCSAS for ABR

Instead of striving to derive a compromise between throughput, delay and optimum routing, we applied the DCSAS in a different perspective.

The proposed DCSAS for ABR protocol activates when a MH finds itself in a congested environment (i.e. having many neighbours and heavily loaded with route relaying functions), contending for the limited available wireless bandwidth. Based on the MH's knowledge of which neighbouring MHs are active (i.e. supporting routes similar to itself) and which are not, and the distances of these neighbouring MHs from itself (computed from the received signal power levels), the MH can dynamically reduce its transmission range to exclude inactive neighbours but include all currently active neighbours.

In this manner, the DCSAS does not affect the operation of the ABR. Existing routes remain unaffected and no RRCs needs to be invoked due to the wireless cell size reduction. This gives rise to the following advantages:

- Reduction in transmission power of the MH.
- Increase in capacity over a given area (due to less beacon traffic and fewer contenders).

8.2. Simulation Results for DCSAS in ABR

To evaluate the percentage reduction in the number of contenders within the wireless cells and the size of resulting wireless cells after a DCSAS activation, the DCSAS mechanism is implemented within the simulator and simulation for all possible routes in the 30 nodes *ad-hoc* mobile network is performed. For each node in the currently considered route, the maximum distances between it and its neighbouring nodes and the upstream or downstream nodes (whichever is higher) are computed. The difference of these distances gives the amount of wireless cell size reduction possible without fragmenting the existing route. The percentage reduction in the wireless cell size and the number of contenders are then recorded. In addition, to ensure that the data is valid for nodes with different spatial locations, different node allocation seeds are used. A total of 500,000 samples of the results are gathered and the distributions are tabulated in Table 18.

Referring to the results obtained by [15] in Table 19 on the throughput of the IEEE 802.11 DFWMAC, it can be seen that decreasing the number of contenders increases the throughput significantly. Unlike [10] who strive for optimal transmission radii to maximize the expected

Table 19. Throughput performance for CSMA/CA.

Throughput of 802.11 DFWMAC			
Collision interval (msec)	No. of contenders		
	1	5	10
0.025	0.78	0.71	0.71
0.025	0.78	0.57	0.42
0.125	0.78	0.11	0.02

progress of packets in the desired direction for each of the different MAC layer protocols, our DCSAS do not. Nonetheless, our scheme continues to provide improvement in throughput in a manner that is independent of the underlying MAC layer protocols.

9. Discussion

9.1. *AD-HOC* WIRELESS ATM

9.1.1. AT&T BAHAMA

Many work on Wireless ATM (WATM) concerns building WLANs with BSs connected to high speed wired ATM networks. However, the lack of total wireless (radio or infra-red) coverage has lead to blind spots. This plus the need to retain compatibility with existing wired ATM networks have motivated the work on *ad-hoc* WATM. A WATM network with mobile INs is undesirable as it violates the notion of ATM connection-oriented service, where once a connection is set-up, data will be transported over the existing path in an orderly manner, and the path is expected to remain unchanged throughout the life of the connection.

The first *ad-hoc* WATM work was described in [11]. In BAHAMA (Broadband Ad-Hoc WATM LAN) [11], a prototype capable of supporting multi-Mbps access rates and multi-Gbps aggregate capacities is presented. However, the *ad-hoc* mobile network configuration is quite different. Other than MHs, specially designed PBSs (Portable Base Stations) (which are deployable) are used to provide microcell coverage to MHs. Hence, MHs are not routers but end hosts and the overall network configuration is similar to that of packet radio.

Because the concept of BS is employed, location management and handover operations are similar to those of BS-oriented WATM networks. However, such an *ad-hoc* WATM LAN will suffer when PBSs move about frequently, denigrating the benefits of ATM transport service. In addition, since conventional routing schemes such as Bellman–Ford or shortest path find (SPF) are used in BAHAMA, routing updates as a result of PBSs' migrations can be substantial. These shortcomings and the need to support instantaneous LAN formation and deformation have, therefore, motivated our work on an *ad-hoc* WLAN comprised solely of MHs.

9.1.2. UCLA WAMIS

Developed as part of the WAMIS (Wireless Adaptive Mobile Information Systems) project, a cluster-based routing scheme which takes into consideration the support of multi-media traffic in an *ad-hoc* mobile network environment is recently proposed in [28]. Compared to the cluster-based scheme in [20], this scheme supports communications over virtual circuits.

An underlying route maintenance protocol such as the distance-vector routing scheme is used to maintain routing information. Mobile hosts are elected as cluster heads if they have the highest degree of connectivity or the lowest ID. These cluster heads then perform channel access co-ordination to nodes within the cluster. A mixture of media access schemes are used: TDMA is used for mobile hosts residing within a cluster while CDMA is used for gateway nodes connecting two different clusters. The virtual circuit establishment process is performed on a hop-by-hop basis, with decision regarding next hop node and slot allocation performed by the cluster head. A modified version of this routing scheme which forgoes the cluster head concept is presented lately in [6].

Since the underlying routing protocol is Bellman–Ford, periodic broadcast of routing information throughout the network is inefficient in terms of both bandwidth and power consumption. Recognising this problem and the need to support quality of service (QoS), a QoS-based routing protocol [27] is currently being investigated by UCLA. Since each MH is subjected to migration, penalties occurred when the cluster head moves. Under such circumstances, the cluster head will loose control as the local channel access co-ordinator, affecting the channel throughput since a new cluster head has to be re-elected. In addition, changes in cluster head roles also occur when other MHs (perhaps with a lower ID or a higher degree of connectivity) migrate into the cluster or when the existing cluster nodes move away from the current cluster. Clustering also results in the problem associated with “distributed gateways”, as mentioned in [28].

Although our ABR scheme is not designed for *ad-hoc* Wireless ATM networks, we have not completely ruled out this possibility.

9.2. Mobile Applications

Unlike BS-oriented wireless networks where QoS are collaboratively supported by the wired and wireless links, *ad-hoc* mobile networks have to maintain end-to-end QoS based solely on the dynamically changing wireless links. Although our ABR protocol is not designed to support QoS guarantees, it does distinguish the qualities among possible routes. Consequently, in addition to supporting nomadic collaborative applications, ABR can also be used to support adaptive mobile multi-media applications [12, 16]. A less adaptive mobile application may prefer routes with a high degree of association stability while a highly adaptive mobile application may operate over routes with varying degrees of association stabilities.

10. Conclusion

In this Letter, the problems associated with routing for conference size *ad-hoc* mobile networks are presented. While conventional link-state and distance-vector Bellman–Ford routing protocols function well in a wired network, they incur extensive bandwidth, power and computation overheads for MHs in an *ad-hoc* mobile network. Hence, a simple and bandwidth-efficient distributed routing protocol based on a new concept of associativity is proposed. The association property allows the routing protocol to exploit the spatial and temporal relationships of *ad-hoc* MHs to construct long-lived routes, resulting in fewer route re-constructions and hence higher attainable throughput.

The proposed protocol uses a combination of point-to-point and broadcast routing approach. Routes are initiated by the source and are set-up based on demand. The protocol is free from loops since “seen” tables, single route selection and ultimate single route re-construction mechanisms are employed. Because a route is explicitly selected prior to usage (with all other possible alternate routes remaining passive), packet duplicates are avoided. Since there is no need to maintain alternate route information, problems associated with stale routes are absent. To ensure transmission integrity, a data flow acknowledgement and retransmission scheme is incorporated into the routing protocol.

The route re-construction process exploits the advantage of locality of neighbouring MHs to quickly construct alternate and even shorter routes, i.e. route optimisation. To fairly distribute the route relaying functions among *ad-hoc* MHs, the route relaying load is identified as a new routing metric, so is the longevity of a route. Neighbouring tables used to support associativity-based routing (ABR) can be simultaneously used to support “neighbour-aware” applications, such as nomadic collaborative computing. In addition, the integration of ABR routing with BS-oriented WLANs allows packets to be re-routed to another nearby BS via other MHs during a BS failure.

To evaluate the protocol performance during the cases when the associativity interlock property is violated, the routing protocol is implemented within a migration-based *ad-hoc* mobile simulator. From the results obtained out of 113 *ad-hoc* mobile networks simulations, we found that the re-constructed route can be shorter than its original route and the number of localised query operations are less than its worst possible value. In addition, as the route neighbouring factor is increased, the probability of a localised query success is also increased. We have also proposed a new dynamic cell size adjustment scheme for ABR, which provides further improvement in throughput and reduction in transmission power in a manner that is independent of the underlying MAC layer protocols.

While conventional networks have static topologies, *ad-hoc* mobile networks have changing topologies. While conventional packet-level network simulators can evaluate traffic performance as a result of node and link failures, they however cannot possibly model nodes in migration with respect to space. In addition, their operations are mainly synchronous and the ability to test for all possible route re-constructions are limited. This explains our approach in developing a migration-based *ad-hoc* mobile network simulator. Although the traffic performance is not evaluated in our simulator, we believe that the property of having long-lived routes has already enhanced the communication throughput considerably and the capability of the routing protocol to quickly locate an alternative shorter route enhances the response time to link changes.

Future work includes the implementation of the proposed routing protocol into existing WLANs (such as AT&T/NCR WaveLAN and Wi-LAN WiLAN) and the development of adaptive mobile applications so that *ad-hoc* mobile computing can be better supported.

Acknowledgements

I wish to thank Prof. D.R. McAuley (Computer Science Department, Glasgow University), Dr. David Greaves (Cambridge University Computer Laboratory), Benny Lee and Mike Webb for their interesting comments on this work. I am grateful to Prof. A. Ephremides (Electrical Engineering Department, University of Maryland) and Dr. Sanjiv Nanda (AT&T Bell Labs) for allowing me to use some of their published research results. Prof. Mario Gerla, Jack Tsai and Keith Scott from UCLA have also assisted in providing me with the relevant publications. I am also thankful to the researchers at Networked Computing Research Group at AT&T Bell Labs for their valuable feedback. Finally, I am indebted to King’s College Cambridge for funding my research work.

References

1. A. Sugikawa, Y. Morioka, K. Iwamura, Y. Tajika and M. Nakamura, “A computerized support system for nomadic collaboration”, in Proceedings of MoMuC2: Second International Workshop on Mobile Multi-Media Communications, April 1995.
2. D. Nielson, F. Tobagi and B. Leiner, “Issues in packet radio design”, Proceedings of the IEEE, Vol. 75, No. 1, pp. 6–20, 1987.
3. K.L. Scott and N.D. Bambos, “A fully distributed algorithm for the creation and maintenance of wireless networks of mobile users”, Technical Report, TR UCLA-ENG-95-112, Department of Electrical Engineering, University of California at Los Angeles, 1994.
4. C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers”, in Proceedings of ACM SIGCOMM’94, September 1994, pp. 234–244.
5. K.-C. Chen, “Medium access control of wireless LANs for mobile computing”, IEEE Network – Special Issue on Nomadic Personal Communications, Vol. 8, No. 5, pp. 50–63, 1994.
6. C. R. Lin and M. Gerla, “A distributed architecture for multi-media in dynamic wireless networks”, in Proceedings of IEEE GlobeCom’95, November 1995, pp. 1468–1472.
7. A. Ephremides, J.A. Flynn and D.J. Baker, “The design and simulation of a mobile radio network with distributed control,” IEEE Journal of Selected Area in Communications, Vol. 2, No. 1, pp. 226–237, 1984.
8. M. S. Corson and A. Ephremides, “A distributed routing algorithm for mobile radio networks”, ACM Wireless Networks Journal, Vol. 1, No. 1, pp. 61–81, 1995.
9. K. Scott and N. Bambos, “The self-organising wireless network protocol for communication among mobile users”, in Proceedings of IEEE GlobeCom’95, November 1995, pp. 355–359.
10. H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals”, IEEE Transactions on Communications, Vol. 32, No. 3, pp. 246–257, 1984.
11. K.Y. Eng, M.J. Karol, M. Veeraraghavan, E. Ayanoglu, C.B. Woodworth, P. Pancha and R.A. Valenzuela, “A wireless broadband ad-hoc ATM local area network”, ACM Journal on Wireless Networks (WINET), Vo. 1, No. 2, pp. 161–174, 1995.
12. K. Lee, “Adaptive network support for mobile multi-media”, in Proceedings of ACM First International Conference on Mobile Computing and Networking (MOBICOM’95), November 1995, pp. 62–72.
13. D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks”, in Mobile Computing, Thomasz Imielinski and Hank Korth (eds.), Kluwer Academic Publishers: Dordrecht. 1996.
14. D.R. McAuley, Private Communications. 1995.
15. A. DeSimone and S. Nanda, “Wireless data: Systems, standards, applications”, ACM Journal on Wireless Networks (WINET), Vol. 1, No. 2, pp. 241–252, 1995.
16. P. Agrawal, E. Hyden, P. Krzyzanowsji, M.B. Srivastava and J. Trotter, “Multimedia information processing in the SWAN mobile networked computing system”, in Proceedings of Multimedia Computing and Networking (MMCN’96), January 1996.
17. P. Agrawal, E. Hyden et al., “SWAN: A mobile multimedia wireless network”, IEEE Personal Communications Magazine, Vol. 3, No. 2, pp. 18–33, April 1996.
18. P. Agrawal et al., “A testbed for mobile networked computing”, in Proceedings of IEEE International Conference on Communications (ICC’95), June 1995, pp. 410–416.
19. P. Agrawal, P.P. Mishra and M.B. Srivastava, “Network architecture for mobile and wireless ATM”, in Proceedings of 16th International Conference on Distributed Computing Systems (ICDCS ’96), May 1996.
20. P. Krishna, M. Chatterjee, N.H. Vaidya and D.K. Pradhan, “A cluster-based approach for routing in ad-hoc networks”, in Proceedings of 2nd USENIX Symposium on Mobile and Location-Independent Computing, April 1995, pp. 1–10.
21. J. Burchfiel, R. Kunzelman, R. Kahn and S. Gronemeyer, “Advances in packet radio technology”, Proceedings of the IEEE, Vol. 66, No. 11, pp. 1468–1496, 1978.
22. A. Bhatnagar and T.G. Robertazzi, “Layer Net: A new self-organizing network protocol”, in Proceedings of MILCOM’90: IEEE Military Communications Conference, June 1990, pp. 845–849.
23. P. Merlin and A. Segall, “A failsafe distributed routing protocol”, IEEE Transactions on Communications, Vol. 27, No. 9, pp. 1280–1287, 1979.
24. T.A. Wilkinson, “HIPERLAN: An air interface designed for multi-media”, in Proceedings of MoMuC2: Second International Workshop on Mobile Multi-Media Communications, April 1995.
25. T. Izumoto, H. Shigeno, T. Yokoyama and Y. Matsushita, “A self-organizing wireless LAN system”, in Proceedings of IEEE 18th Conference on Local Computer Network, September 1993, pp. 467–473.
26. J. Jubin and J. Tornow, “The DARPA packet radio network protocols”, Proceedings of the IEEE, Vol. 75, No. 1, pp. 21–32, 1987.
27. J. T.-C. Tsai, Private Communications. 1996.

28. M. Gerla and J. T.-C. Tsai, “Multicluster, mobile, multimedia radio network”, ACM Journal on Wireless Networks (WINET), Vol. 1, No. 3, pp. 255–265, 1995.
29. V. Hayes, “IEEE P802.11”, in Digest of AT&T WCND WaveLAN Partners Conference, October 1995.
30. N. Shacham and J. Westcott, “Future directions in packet radio architectures and protocols”, Proceedings of the IEEE, Vol. 75, No. 1, pp. 83–99, 1987.

Chai-Keong Toh is a Honorary Cambridge Commonwealth Trust Scholar, a King’s College Cambridge External Research Scholar, a Fellow of the Cambridge Commonwealth Society and the Cambridge Philosophical Society. Born in 1965, he received his Diploma in Electronics & Communication Engineering with a Certificate of Merit award from the Singapore Polytechnic in 1986 and his B.Eng. (First Class Honours) in Electronics Engineering from the University of Manchester Institute of Science and Technology in 1991. C.-K. Toh is currently completing his Ph.D. requirements at the Computer Laboratory, University of Cambridge, England. His research interests include the design of wireless ATM network architectures, mobile routing, multicasting and handover protocol design, mobile connection and location management, the internetworking of wireless networks and operating systems support for mobile computing. Prior to Cambridge, he has held several technical positions in the industries, such as Network Specialist, R&D Engineer and Member of Technical Staff. He initiated the Cambridge University Mobile Special Interest Group in 1994 and since then has published several papers on mobile networking. He has served as a technical reviewer for several IEEE and ACM conferences and journals. He is a Technical Editor for IEEE Journal on Communications Survey and serves as a member of the Editorial Board for Personal Technologies Journal. He is a member of IEEE, IEE, ACM and USENIX.