A Hierarchical QoS Multicast Routing Protocol for Mobile Ad-Hoc Networks

Li Layuan Li Chunlin

Department of Computer Science, Wuhan University of Technology, Wuhan 430063.P.R.China
E-mail: jwtu@public.wh.hb.cn

Abstract: The provision of Quality-of-service (QoS) guarantees is of utmost importance for the development of the multicast services. These multicast services have been used by various continuous media applications such as the multicast backbone (Mbone) of the Internet has been used to transport real time audio/video for news, video conferencing and distance learning. This paper presents a hierarchical QoS multicast routing protocol (HQMRP) for mobile ad-hoc networks. It can provide QoS-sensitive routes in a scalable and flexible way, in the network environment with mobility. In the proposed HQMRP scheme, each local node just only needs to maintain local multicast routing information and/or summary information of other clusters (or domains), but does not requires any global ad hoc network states to be maintained. The HQMRP also allows that an ad-hoc group member can join/leave the multicast group dynamically, and supports multiple QoS constraints. The paper presents formal description and main procedures for realizing routing decision process of the HQMRP, and the proof of correctness and complexity analysis of the protocol. In also presents a theoretical analysis of the mobility in the mobile ad-hoc network environment. The performance measures of HQMRP are evaluated using simulation. The studies show that HQMRP can provide an available approach to QoS multicast routing for mobile ad-hoc networks.

Keywords: Ad-hoc networks, QoS routing, multicast, hierarchical routing, mobile wireless networks

I. INTRODUCTION

Mobile ad-hoc networks (MANET) are self-organized by a collection of mobile nodes, interconnected by multi-hop wireless paths in a strictly peer-to-peer fashion. Each node may serve as a packet-level router for its peers in the same network. Such networks have recently drawn significant research attention since they offer unique benefits and versatility with respect to bandwidth spatial re-use, intrinsic fault tolerance, and low-cost rapid deployment. Furthermore, near-term commercial availability of Bluetooth-ready wireless interfaces may lead to the actual usage of such networks in reality. However, the topology of ad-hoc networks may be highly dynamic due to unpredictable node mobility, which makes QoS provisioning to applications running in such networks inherently hard. The limited bandwidth of wireless channels between nodes further exacerbates the situation, as message exchange overheads of any QoS-provisioning algorithms must be kept at the minimum level. This requires that the algorithms need to be fully distributed to all nodes, rather than centralized to a small subset of nodes.

Conventional routing protocols [1-8] used for wired networks are based on distance vector or link state algorithms. However, these routing protocols haven’t been designed to cater to the dynamic property of the mobile networks. Although, it was possible to model each mobile node as a router to suit the needs of the conventional protocols, these protocols still placed a very heavy computational burden on the nodes. Moreover, the convergence characteristics of these routing protocols were not good enough to suit the needs of a MANET. Thus conventional routing protocols cannot be used for ad hoc networks as these networks are bandwidth and energy constrained [7-13].

Conventional multicast protocols, e.g., CBT and PIM [1-4], were designed for best-effort data traffic. They construct multicast tress primarily based on connectivity. Such tress may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms [1-6] provide heuristic solutions to the NP-complete constrained Steiner tree problem, which is to find the delay-constrained least-cost multicast trees. These algorithms however are not practical in the Internet environment because they have excessive computation overhead, require knowledge about the global network state, and do not handle dynamic group membership.[14-16,19] Jia’s distributed algorithm [3] does not compute any path or assume the unicast routing table can provide it. However, this algorithm requires excessive message processing overhead. The spanning join protocol by Carberg et al. [4-6] handles dynamic membership and does not require any global network state. However, it has excessive communication and message processing overhead because it relies on full flooding to find a feasible tree branch to connect a new member. QoS MIC, proposed by Faloutsos et al. [1] alleviates but does not eliminate the flooding behavior. In addition, an extra control element, called Manager router, is introduced to handle the join requests of new members.

Some routing protocols for mobile ad hoc networks, such as AODV, DSR, AND TORA[7-11], are designed without explicitly considering quality-of-service of the routes they generate. QoS routing in ad hoc networks has been studied only recently. QoS routing requires not only to find a route from a source to a destination, but the route must satisfy the end-to-end QoS requirement, often given in terms of bandwidth or delay. Quality of service is more difficult to guarantee in ad hoc networks than in other type of networks, because the wireless bandwidth is shared among adjacent nodes and the network topology changes as the nodes move. This requires extensive collaboration between the nodes, both to establish the route and to secure the resources necessary to provide the QoS.

This paper presents a hierarchical QoS multicast routing protocol for mobile ad-hoc networks (HQMRP). It not only ensures fast convergence but also provides multiple guarantees for satisfying multiple QoS Constraints. HQMRP also allows that an ad-hoc group member can join/leave the multicast group dynamically.

The rest of this paper is organized as follows. Section 2 describes the hierarchical MATET, its QoS multicast routing problem and model. Section 3 presents the QMRPD. Section 4 gives the correctness proof and complexity analysis. Some simulation results are provided in Section 5. The paper concludes with Section 6.

II. HIERARCHICAL MANET AND MODEL

In general, the clustering problem of MANIET depends on the network topology, geographical location of nodes (or routers), connectivity, as well as the relativity between nodes. In the viewpoint of hierarchical networks, each node of MANET can be considered as 0th-level. A region that consists of such several nodes can be called first-level cluster (or domain). Several first-level clusters are combined to form second-level clusters. Similarly, third-level clusters, forth-level clusters and *K*th-level clusters can be defined. Each first-level cluster contains at least one node and does not overlap with any other first-level clusters. Second-level clusters contain only first-level clusters and they do not overlap. All nodes that are within the same first-level cluster are called local nodes. The node that has links to nodes in other clusters in called bridge node (or domain border router). The local nodes are also called 0th-level bridge nodes. The nodes that connect two first-level clusters are called first-level bridge nodes; the nodes that connect two second-level clusters are called second level bridge nodes, and so on. A network that is formed with first-level bridge nodes within a given second-level cluster is called a first-level bridge network. An example hierarchical MANET is shown in Fig.1. In Fig.1, white circle denotes node or router, black circle denotes bridge node or domain border router and ellipse denotes domain.

As far as multicast routing is concerned, a network is usually represented as a weighted digraph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of communication links connecting the nodes. $|V|$ and $|E|$ denote the number of nodes and links in the MANET, respectively. Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes[13]. Associated with each link are parameters that describe the current status of the link.

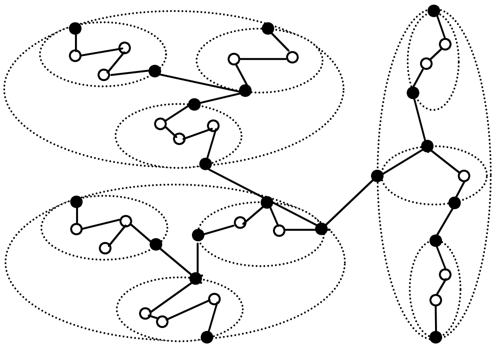


Fig.1 An example of MANET model

Let $s \in V$ be source node of a multicast tree, and $M \subseteq \{V\text{-}\{s\}\}$ be a set of end nodes of the multicast tree. Let R be the positive weight and R^+ be the nonnegative weight. For any Link $e \in E$, we can define the some QoS metrics: delay function *delay* (e): $E \rightarrow R$, cost function *cost* (e): $E \rightarrow R$, bandwidth function *bandwidth* (e); $E \rightarrow R$, and delay jitter function *delay-jitter* (e): $E \rightarrow R^+$. Similarly, for any node $n \in V$, one can also define some metrics: delay function *delay* (n): $V \rightarrow R$, cost function *cost* (n): $V \rightarrow R$, delay jitter function *delay-jitter* (n): $V \rightarrow R^+$ and packet loss function *packet-loss* (n): $V \rightarrow R^+$. We also use $T(s,M)$ to denote a multicast tree in which the following relations hold:

1) $delay\,(p\,(s,t)) = \sum_{e \in P(s,t)} delay\,(e) + \sum_{n \in P(s,t)} delay\,(n)$.

2) $cost\,(T(s,M)) = \sum_{e \in T(s,M)} cost\,(e) + \sum_{n \in T(s,M)} cost\,(n)$.

3) $bandwidth\,(p(s,t)) = \min\{bandwidth\,(e),\ e \in P(s,t)\}$.

4) $dealy\text{-}jitter\,(p\,(s,t)) = \sum_{e \in P(s,t)} delay - jitter\,(e) + \sum_{n \in P(s,t)} delay - jitter\,(n)$.

5) $packet\text{-}loss\,(p\,(s,t)) = 1 - \prod_{n \in P(s,t)} (1 - packet\text{-}loss(n))$

where $p\,(s,t)$ denotes the path from source s to end node t of $T\,(s, M)$.

The QoS-based multicast routing problem is to find the $T\,(s, M)$ which satisfies some QoS constraints:

Delay constraint: $delay\,(p\,(s,t)) \leqslant D_t$ (1)
Bandwidth constraint: $bandwidth\,(p\,(s,t)) \geqslant B$ (2)
Delay jitter constraint: $delay\text{-}jitter\,(p\,(s,t)) \leqslant J$ (3)
Packet loss constraint: $packet\text{-}loss\,(p\,(s,t)) \leqslant L$ (4)

Meanwhile, the *cost* ($T\,(s, M)$ should be minimum. Where D is delay constraint, B is bandwidth constraint, J is delay jitter constraint and L is packet loss constraint. In the above QoS constraints, the bandwidth is concave metric, the delay and delay jitter are additive metrics, and the packet loss is multiplicative metric. In these metrics, the multiplicative metric can be converted to the additive metric. For simplicity, we assume that all nodes have enough resource, i.e., they can satisfy the above QoS constraints. Therefore, we only consider the links’¯ or edge’¡¯ QoS constraints, because th links and the nodes have equifinality to the routing issue in question. The characteristics of edge can be described by a fourtuple (*D,J,B,C*), where *D,J,B* and *C* denote delay, delay jitter, bandwidth and cost, respectively. For simplicity, we also mainly consider the former two QoS constraints of the above QoS constraints (Equation1-4)

III. HQMRP

3.1 The protocol description

In order handle the dynamic and mobility of MANET, The HQMRP assumes that each local node measures periodically the delay along its outgoing links and forwards the information with the highest priority to all other nodes in the cluster. Other nodes will recompute their intralcuster routing tables after receiving the update message. Similarly, each bridge node also checks periodically the delay along its outgoing links and forwards the information to all other bridge nodes in the first-level (second-level or third-level) bridge network. Other bridge nodes will recompute their intercluster routing tables after receiving the intercluster updating message. The routing databases contain the main topological information that needs to be updated only when topology changes, a link (node) fails, or a node joins/leaves multicast tree.

The sender of a multicast may move while transmitting, or receiver may move while receiving the multicast message. In HQMRP, the remote subscription method for handling mobility is used. In the method, each mobile node subscribes again when it enters a new domain. This makes the local multicast node for that the new domain join the multicast tree. This re-subscription frequency really depends on the rate of hand-off that the mobile nodes faces.

HQMRP uses a receiver-initiated selection flooding (*SF*) algorithm in which the links that violate the bandwidth constraint will firstly be deleted, and the flooding message should keep clear of the violated links. In HQMRP, each node just only needs to maintain local multicast routing information and/or summary information of other clusters (or domains),but does not requires any global states of MANET to be maintained.

When a new member wishes to join a *T* (s,*M*), it will send a JOINreq message (*msg*) to its parent bridge node. The format of this message is JOINreq（*GA*、*NA*、*QM*）. Here, *GA* is the multicast group address, *NA* is the new member’¯s address, and *QM* is QoS metric. When the new node initiates the JOINreq *msg,* the first entry in the array is set to be the node address. The maximum number of entries in path is equal to the maximum number of levels in the hierarchy. If the bridge node receiving a JOINreq message is not aware of the multicast tree, it appends its own address to the array of addresses in path and forwards the JOINreq message to its parent bridge node. If the requested multicast tree does not exist in the network, then the JOINreq message will arrive at the bridge node of the top level domain, which is not aware of the multicast tree. In this case, the top bridge node sends multicast tree generating（*MT generate*）message towards the node.

The formal description of processing process of JOINreq *msg* of HQMRP can be described in the paragraph (1) of Fig.2.

When the new member receives a *MT generate msg*, it will generate the tree and forward the *MT update msg* to it parent bridge node. *MT generate msg* updates the multicast tree information of the bridge node and is sent towards the higher level bridge node. The formal description of the processing processes of *MT generate msg* and *MT update msg* is shown in the paragraph (2) of Fig.2. The variable *MT* is a Boolean variable. This variable is initialized to FALSE (F) and if the node is on-tree node then it will be set to TRUE (T), which can keep track of whether the node is on a specific multicast tree or not. Each bridge node should store address of all on-tree nodes within a domain and bridge addresses of the lower level domains that contains on-tree nodes. When a *MT update msg* arrives at a bridge node, the address of the node that sent the message is stored by the bridge node.

When a new node wants to join a multicast group, it sends a JOINreq *msg* to its parent bridge node. If the message arrives at a bridge node that is aware of the multicast tree, then the bridge node forwards the message to all the on-tree nodes or bridge node of the sub-domains having on-tree node. Otherwise, the bridge node forwards the message to its parent bridge node. Then JOINreq message arrives at an on-tree node, the node initiates a *SF* message. This message is flooded towards the new node by sending it to some neighbors, which in turn forward the message to their some neighbors. To reduce message overhead during reverse selection flooding, the messages are forwarded only in those directions that satisfy certain forwarding conditions. The forwarding conditions are selected to eliminate those messages that will not participate in establishing a feasible path between new node and the multicast tree.

In HQMRP protocol, each node has a data structure defined by *PF*, *PQ* and *PR*. Here *PF* is a Boolean variable that shows whether the *SF msg* has been sent by the current node or not. It is set to T when a flooding *msg* from multicast tree is forwarded to the new node else it is set to F. *PQ* denotes QoS metric (*QM*) of the most recent message that has been sent by the node and is initialized to zero. *PR* is the address of the neighboring node that sent the *SF msg* which has already been sent by the current node.

Even though a node has sufficient resources to satisfy the QoS requirements when it forwards the flooding message, it may not have the required resources to reserve while processing the join message. This problem can be avoided by reserving resources while forwarding the flooding message and releasing the resource if it is not used before a certain specified time. This approach will unnecessarily reserve more resources than required for a certain period of time.

The paragraph(3) of Fig.2 gives the formal description of processing process of join message. Here, *TN* is the array that stores the addresses of all the on-tree neighboring nodes. *R* is a variable that contains the address of the node that forwarded the join message. When a node receives a join message it adds *R* to the array *TN*. Similarly, when a node forwards a join message, it adds *PR* to the array *TN*.

In order to overcome loop routing, HQMRP allows only one *SF msg* for a particular pair (multicast, node) to pass through each node. This method can be implemented by setting the variable *PF* to T after forwarding a *SF msg*, which can be used to delete any future *msg* for the same pair.

When a *SF msg* with larger path delay arrives at a node before *SF msg* with smaller path delay, the message with larger path delay gets forwarded the node. If the forwarded message fails to satisfy the QoS requirement later, it will not arrive at the new node and the feasible path may be detected. In the unicast routing, this problem was solved by delaying the *SF* message at each node by *nd*, where *nd* represents the node delay. The introduction of delay at each node guarantees that the messages with smaller path delay arrive at nodes before messages with larger path delay. However, this method may increase the time delay for joining a multicast session group. If the differences between the *QM* of the current message and that of a previously forwarded message is more than *δ*, then the message is forwarded, where *QM* is the QoS metric of the path followed by the *SF msg* up to the current node, and *δÄ*= *Φ* *QR* and 0< *Φµ*<1, here *QR* is QoS requirement. The formal description of processing process of *SF* message can be described in the paragraph (4) of Fig.2. The *QT* denotes QoS type, in the paragraph (4) of Fig.2.

The above related process of HQMRP can be formally described as follows.

```
(1)  if (a new member, which is on the tree, wishes to
     join a T (s, M))
        then it sends JOINreq (GA, NA, QM)
                  to its parent bridge node
       else case1 (on-tree bridge nodes or nodes exist in
            the domain)
            sending JOINreq to all on-tree bridge nodes
            case2 (the new member is the top level
            bridge node)
            deleting JOINreq (GA、NA、QM、)
            else sending JOINreq to its parent bridge
                 node
     fi
 (2)  if (the current node is the new member)
            then MT = T
            ND = current node
 sending MT update msg to parent bridge node
     else forwarding the MT generate msg to the node
   fi
  if (the current node is top level bridge node)∨(the
     current node is a on-tree bridge node )
          then deleting the MT update msg
  else sending the MT update msg to the parent bridge
          node
     fi
  (3)  if (MT=T) then
          deleting JOINreq (GA、NA、QM)
          else MT=T
          reserving resources on the link to PR
          adding PR to TN
          R=current node address
            sending JOINreq to PR
     if (current node is not top level bridge node)
          forwarding MT update msg
     fi             to its parent bridge node
     fi
(4)  if (QT=bandwidth)∧(PF=F) then
          send msg=T
               nd=o
          else if (QT=delay)∧(PQ-QM)>δ
               sending msg = T else
               sending msg = F
     fi
     if (sending msg=T)
     for (all neighboring nodes)
     if (sending is open)
       PR=R
       PQ=QM
       PF=T
       R=current node address
     If (QT=delay) then
       QM=QM +nd
     fi
       Send SF msg to neighbor
     fi
     fi
     fi
```

3.2 Main routing decision procedures

Routing decision process is a key procedure of the proposed protocol. It is based on the discrete dynamic programming principle and allows the optimal routing to be found by computing the multi-segment map.

The mathematical description of the multi-stage routing decision based on the discrete dynamic programming principle can be given as follows.

In hierarchical MANET, the cost function of link delay can be expressed as

$$D_N=\sum_{i=0}^{N-1}F[x(i),u(i)],\, i=0,1,2,\cdots,N-1 \tag{5}$$

where $x(i)$ denotes the state of routing decision process, *i.e.*, the position of each node (v) u (i) denotes the routing decision in the routing selection process and $F[x(i),u(i)]$ denotes the delay cost of each segment path.

Initial state in the routing decision process is defined by

$$x(0)=x_0 \tag{6}$$

The dynamic equation of routing process is defined by

$$x(i+1)=g[x(i),u(i)] \tag{7}$$

Let D_N^* denote the minimum cost of optimal routing decision and let u^* be the optimal routing, then the optimal routing problem will be transformed to solve for $u^*(0)$, $u^*(1)$, $u^*(2)$, ⋯, $u^*(N-1)$, making the cost D_N along path connecting V_0 to V_t minimum.

From formula（5）and using formula（6）and（7）one by one, we get

$$D_N = F[x(0),u(0)] + F[x(1),u(1)] + \cdots + F[x(N-1),u(N-1)]$$

$$= F[x(0),u(0)] + F\{g[x(0),u(0)],u(1)\} + \cdots$$

The above equation can also be expressed as

$$D_N = [x(0), u(0), u(1), \cdots, u(N-1)] \tag{8}$$

If optimal routing sequence has been selected, then the minimum value of D_N only depends on the initial state and can be denoted by $D^*[x(0)]$. In general, the minimum value of D_N for initial state x can be expressed as $D_N^*[X]$, then we get

$$D_N^*[x(0)] = \min_{u(0),u(1),\cdots,u(N-1)} \{D_N[x(0), u(0), u(1), \cdots, u(N-1)]\}$$

$$= \min_{u(0)} \{D[x(0), u(0)] + D_{N-1}^*[x(1)]\} \tag{9}$$

where $x(1)=g[x(0),u(0)]$

The main procedure for realizing routing decision can be described as follows.

```
    Procedure routing decision
        (var, type and parameter declaration)⋯;
        begin for j:=1 to clu[i1[. bridgenum do
          r1:=r1+1;i1:=i1+1 end;
          i1:=1;while i1＜vd1 do begin
          for j1:=1 to clu[i1].bridgenum do
          r2:=r2+1;i1:=i1+1 end;
          cld:=r1;for i1:=1 to cld[vd1].bridgenum do begin
p:=p+1; r2:=r2+1; r1:=cld;
          for j:=1 to clu［vs1］do
  if   bdelay [p]＞bdelay[ j ]+node [vs2].ir_rout [r2,j] delay
then begin
bdelay[p]:=hdlay[j]+node[vs2],ir_rout[r2,j].delay;
d[p]:=j end
end;
p:=p+1
  for i1:=1 to clu[vd1].bridgenum do with clu[vd1] do      s
begin q2:=q2+1;
    if   bdelay [p] bdelay [q2] + node [vd2] .in_ rout [bridge
[ii]].delay
    then begin
    bdelay [p] := bdelay [q2]+node [vd2].in_ rout [bridge
[ii]] . delay; d[p]:=ii end
end;
writeln...;
begin
init;
routing update;
for i:=1 to clunum do with clu[i] do
    for.j:=1 to inclunum do
    begin writeln;...
        r3:=0;for i1:=1 to clunum do
        for jj:=1 to clu［ii］.bridgenum do
        begin
          r3:= r3+1;...
          for kk:= 1 to bridgenum do
          begin r4:0;for i4:=1 to clunum do
            for j4:=1 to clu［i4］.bridgenum do
            r4:= r4+1;
             if r4:=r4+1;
             then begin...
      write(node［j］.ir_rout［r3, kk］.delay...)
          end
        end
      end;...
    end
  end;⋯
```

3.3 Dynamics of nodes and numerical results

The following assumptions are predetermined (1) the new arrivals of multicast session request follow the Poisson distribution; (2) the domain residence time for any mobile node follows exponential distribution; (3) the time of multicast sessions follows exponential distribution; (4) the number of servers is assumed to be infinite since each user is independent of other users and thus they can be served independently. Let the average domain residence time for any mobile node be $1/u_R$ and the average multicast session duration time be $1/uT$, Let Q be the total number of nodes in a domain. For the assumed model, the probability that are K users in the system is given as follows:

$$P_K = \begin{cases} \dfrac{(\frac{\lambda}{\mu})^K \binom{Q}{K}}{(1+\frac{\lambda}{\mu})^Q}, & 0 \le k \le Q \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Where $\lambda = \lambda_N + \lambda_H$ and $\mu = \mu_R + \mu_T$, λ_N is the mean rate of new node arrival and λ_H is the mean rate of hand off call arrival. A new route is grafted in a multicast tree, when either a new call or a hand off call arrives to a domain which is not already a member of the multicast delivery tree. So the probability that any domain A will be added to the multicast tree at t is given as follows:

$P_g(t)$=*Prob* [a new call or a hand off call arrives in domain A at time t | Domain A does not have any user].

Let π_n^A be the probability of finding n users in domain A. Therefore, from Equation (10) we have

$$\pi_n^A = \frac{(\frac{\lambda}{\mu})^n \binom{Q}{n}}{(1+\frac{\lambda}{\mu})^Q} \tag{11}$$

Prob [a new call or a band off call arrives in domain A at time t]=$\lambda\ddot{E}^{\lambda\ddot{E}}$and *Prob* [domain A does not have user]=π_0^A .

We get

$$P_g(t) = \lambda e^{-\lambda t} \frac{(\frac{\lambda}{\mu})^0 \binom{Q}{0}}{(1+\frac{\lambda}{\mu})^Q} = \frac{\lambda e^{-\lambda t}}{(1+\frac{\lambda}{\mu})^Q} \tag{12}$$

Then the expected number of new domains grafted to the multicast tree in time T can be given by the following equation:

$$\int_0^T NP_g(t)\,dt = \int_0^T N\frac{\lambda e^{-\lambda t}}{(1+\frac{\lambda}{\mu})^Q}dt = \frac{N}{(1+\frac{\lambda}{\mu})^Q}[1-e^{-\lambda T}]$$

（13）

The computation of the probability of a domain getting detached from the multicast tree depends on the hand off policy. There are two main hand off policies namely Hard Hand off and Soft Hand Off. In the former, the connection in the old domain from which a mobile user moves is broken before any connection is made in the new domain. While in the later, the connection in the new domain is made even before the connection in the old domain is broken. The occupied channel in the old domain is released after some time, called the freeze time. But for all practical purposes this freeze time can be neglected.

A domain gets detached from the multicast tree when it looses all the multicast recipients from its coverage area. Let the probability of a domain getting detached from the multicast tree is denoted as $P_d(t)$. Thus we have

$P_d(t)$=*Prob*[a call terminates in domain *A* before time *t*| domain *A* has only one user and the call does not get hand off before time *t*] + *Prob*[*a* call is hand off in domain *A* before time *t*| domain *A* have only one user and the call does not terminate before time *t*].

Prob [a call terminates in domain *A* before time *t*| domain *A* have only one user and the call does not get hand off before time *t*] is given by,

$$\frac{\frac{Q\lambda}{\mu}\mu T e^{-\mu Tt} e^{-\mu Rt}}{(1+\frac{\lambda}{\mu})^Q} \qquad (14)$$

and *Prob* [a call is hand off in domain *A* before time *t* | domain *A* has only one user and the call does not terminate before time *t*] is given by

$$\frac{\frac{Q\lambda}{\mu}\mu \mathrm{Re}^{-\mu Rt} e^{-\mu Tt}}{(1+\frac{\lambda}{\mu})^Q},$$

Where $\mu = \mu_R+\mu_T$. Then we have

$$P_d(T) = \frac{Q\lambda e^{-\mu t}}{(1+\frac{\lambda}{\mu})^Q} \qquad (15)$$

Thus, the expected number of new domains deleted form the multicast tree in time T is given as follows:

$$\int_0^T NP_g(t)\,dt = \int_0^T NQ\frac{\lambda e^{-\mu}}{(1+\frac{\lambda}{\mu})^Q}dt = \frac{NQ\lambda}{\mu(1+\frac{\lambda}{\mu})^Q}[1-e^{-\mu T}]$$

（16）

When $T\rightarrow\infty$, the expected number of nodes deleted from the multicast tree will become

$$\frac{NQ\lambda}{\mu(1+\frac{\lambda}{\mu})^Q}.$$

When the group size is known, it is possible to get an estimate for the total length of the multicast distribution tree. The total length of the multicast tree (M_e), the average length of unicast routing path (V_e) and the multicast group size (G_S) can be related by the relation $M_e = VeG_S^K$, where k is typically found to be 4/5. So if we know G_S, which is basically the expected multicast group size, and get an estimate for V_e. Let N is the number of domains then it is easy to get an idea of the expected size of the multicast tree form Chang-Sirbu scaling law as follows:

$$M_e = V_e\left(N\left[1-\frac{1}{(1+\frac{\lambda}{\mu})^Q}\right]\right)^{4/5} \qquad (17)$$

This is an approximate estimate only since the length of the multicast tree is not static during to user mobility and dynamic membership causing addition and deletion of nodes to (or from) the multicast tree. The expected number of nodes grafted to the multicast tree gives an estimate of the number of times that the protocol might have to go for reconstruction of the multicast tree.

The numerical results can be presented using the following values[20] for the concerned parameters: Let arrival rate for new calls (λ_N) be 2.75e^{-02} calls/sec., a arrival rate for hand off calls (λ_H) be 5.91e^{-02} calls/sec., service rate for calls or the call termination rate (μ_T) be 1.55e^{-02} calls/sec., rate that a call gets hand off (μ_R) be 3.33^{-02} calls/sec., and the number of domains be 60.

Fig.2 shows the dynamics of the number of nodes left from *T* (*s*, *M*) with time. In Fig.2, *T* denotes time and *NL* denotes expected number of nodes left from *T*(*s*,*M*).

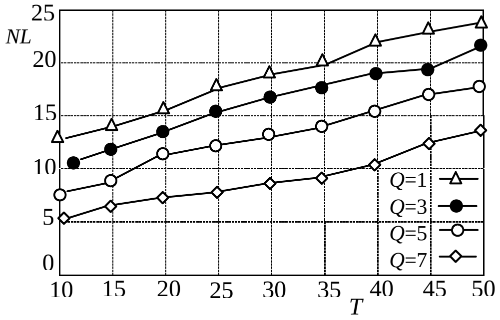


Fig.2 The left nodes’ number vs. time

Fig.3 shows that the dynamics of the expected number of nodes joined to *T*(*S*,*M*) with time. In Fig.3, *NJ* denotes the expected number of nodes joined to *T*(*s*,*M*) and *T* denotes time.

The reason for the decrease in the number of nodes joined to the multicast tree with the increase in the average number of users per domain is as follows. With the increase in expected number of users in each domain the probability that any user initiates a call so as to add the domain the multicast tree decreases. In Fig.2 and Fig.3, the numerical results have been shown for several different values of Q, which should be the average number of nodes in every domain.

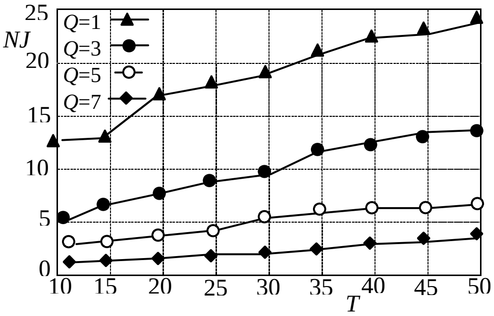


Fig.3 The joined nodes’ number vs. ti m

IV CORRECTNESS AND COMPLEXITY

4.1 proof of correctness

We first give the proof of correctness of the routing update correctness, then give the proof of correctness of the routing decision process and loop-free.

Theorem 1. *If changes of link delay/topology occur between time* τ_0 *and* τ_1 *in MANET, and no changes occur after time* τ_1 *, then after some finite time the routing tables (intracluster or intercluster) stored at the node will be correct and consistent.*

Proof. Case of updates for intracluster routing tables is first considered. Since the changes of network status occur between time τ_0 and τ_1 , and there are impacts of the broadcast speed of update messages, computation and modification speed of routing tables for local nodes, thus the intracluster routing tables are dynamic and unstable. But there are no changes in MANET after time τ_1 , every update message sent can reach each reachable node. Thus, the routing tables stored at each local node have the most up-to-date information about network status after time τ_1 （some finite time, say τ_2 and $\tau_2 > \tau_1$ ）the value of τ_2 is relative to the transportation τ_2 delay of update messages between a pair of the remotest nodes after receiving the update messages, *i.e.*, the intracluster routing tables is correct. Meanwhile, since routing tables stored at each local node contain identical routing information with same network status, the routing table is considered to be consistent. Then, case of updates for intercluster routing tables is considered. The intercluster routing tables would contain routing information with optimal link delay estimates at each bridge node of first-level (second-level or third-level) cluster in MANET. It can be implemented by update procedure of intercluster routing information. Intercluster updates can broadcast an intracluster updates to other clusters via the bridge node. Thus, routing tables stored at each bridge node will have the most up-to-date information about intercluster network status after time τ_2(some finite time, say $\tau_3$and $\tau_3 > \tau_2$) , *i.e.*, the intercluster routing table is considered correct. Meanwhile, since routing tables stored at each bridge node contain identical routing information with same intercluster network status, the intercluster routing table can be considered to be consistent. □

We now prove correctness of the above routing decision process. In routing decision process some principles of the following theorem are used. Thus, the key to proof of correctness for routing decision process lies in proof of correctness for the following theorem.

Theorem 2. *If in N-stage routing decision process at initial state x (0), optimal routing sequence is* $u^*(0)$, $u^*(1)$, $u^*(2)$, ⋯, $u^*(N-1)$, *then in (N-1) stage routing decision process at initial state x (1), sequence* $u^*(1)$, $u^*(2)$, $u^*(3)$, ⋯, $u^*(N-1)$ *is also optimal routing sequence.*

Proof. Suppose $v^*(1)$, $v^*(2)$, $v^*(3)$,⋯, $v^*(N-1)$ is optimal routing sequence and $u^*(1)$, $u^*(2)$, $u^*(3)$,⋯, $u^*(N-1)$ is not optimal routing sequence, then we have

$$D[x(1), v^*(1),\cdots,v^*(N-1)] < D_{N-1}[x(1),u^*(1),\cdots,u^*(N-1)] \qquad (18)$$

Using routing sequence $u^*(0),v^*(1),\cdots, v^*(N-1)$ to routing region, we get

$$D_N[x(0), u^*(0), v^*(1),\cdots,v^*(N-1)] = D[x(0), u(0)] + D[x(1), v^*(1) + \cdots + D[x(N-1)]\, v^*(N-1)]$$

From formula (18), we have

$$\begin{aligned} D_N [x(0), u^*(0), v^*(1),\cdots,v^*(N-1)] &= D[x(0), u (0)] + \{D[x(1), v (1) + \cdots + D[x(N-1),v (N-1)]\} \\ &= D[x(0), u (0)] + D_{N-1}[x (1), v^*(1) ,\cdots, v^*(N-1)] < D[x(0), u (0)] + D_{N-1}[x (1), u^*(1) ,\cdots, u^*(N-1)] \\ &= D_N [x(0), u^*(0)], u^*(1) ,\cdots, u^*(N-1)] \end{aligned}$$

This result is contradicting assumption that $u^*(0)$, $u^*(1)$, ⋯ ,$u^*(N-1)$ is optimal routing sequence. Thus, $u^*(1)$,$u^*(2)$,$u^*(3)$, ⋯, $u^*(N-1)$ must be also optimal routing sequence. □

Lemma 1. *Whenever during the routing process, all paths being searched form a T(s,M) structure .*

Proof. The paths being searched will be marked by the routing entries at the nodes. In HQMRP, any routing entry has a single out interface and one or multiple in interfaces. Hence, the nodes will form a searching tree structure. This tree is just a $T(s,M)$. □

Theorem 3. *An available and feasible path found by HQMRP is loop-free.*

Proof. This Theorem follows directly from the above Lemma 1. □

4.2 Complexity Analysis

There are three steps involved establishing a session between new node and a $T(s,M)$. They are unicasting a JOINreq *msg* from the new node to on-tree nodes via bridge node, sending *SF* messages from on-tree nodes towards the new node and sending a join message from the new node to an on-tree node. Let the time taken by the JOINreq and join *msg* to traverse a link including the buffering and processing time at nodes be one unit of time. Then the time taken by the JOINreq and join *msg* together is O(h_1+h_2), there h_1 is the number of links of the path followed by the JOINreq *msg* and h_2 is the number of links of the join *msg* path. For most cases, h_1=h_2, hence the time is O($2h$) . The path of the JOIN *msg* is opposite of the path followed by the *SF msg* that is used to initiate the join *msg*. Therefore, the time taken by the *SF msg* is the sum of the delays at each node in the path of the join *msg*, i.e. the sum of $(nd)_i$ ($1 \leqslant i \leqslant h_2$) . In all the cases, the time required by the *SF msg* is O(h_2). Therefore, the total connection time for the protocol is O($3h$). To estimate the message overhead, sending a message over a link is counted as one message. The number of message per join request depends on the number of on-tree nodes, size of the flooding domain and QoS requirements. The number of JOINreq and join *msg* per join request is h_1+h_2. For bandwidth requirement and delay requirement, the HQMRP sends at most one *SF* message per link for each (*multicast, host*) pair. The total number of *SF msg* is thus bounded by *f*, where *f* is the number of links in the flooding domain. Therefore, the worst case message overhead is O (f+$2h$).

We now analyze the complexity of during routing decision process of HQMRP.

Let L_{ij} denote link length from node *i* to node *j* and d_{ij} denote the current estimate of the shortest distance from node *i* to the given destination node. The estimate is stored at node *i*. The routing decision process is then given by

1) $d_{i1}^{(0)} = \infty, i \neq 1$

2) $d_{i1}^{(k+1)} = \min_{j \in N(i)}[L_{ij} + d_{j1}^{(k)}], i \neq 1$

$$d_{11}^{(0)} = 0;\ \ d_{11}^{(k+1)} = 0$$

where *N* (*i*) denotes the set of current neighbors of node *i*, and *k* is the iteration count. Let T_1 and T_2 denote computation complexity and communication complexity, respectively. Computation complexity and communication complexity can be given by T_1=O($\overline{d}_{nh}$) and T_2=O($\overline{n}_h$), respectively, where *d* denotes maximum node degree, *n* denotes maximum number of nodes in MANET and $\overline{n}_h$ denotes maximum number of hops along any of the *n*-1 shortest paths. It is obvious that in general $1 \leqslant \overline{n}_h \leqslant n$-1. Thus, computation complexity and communication complexity of routing decision process would be O($\overline{d}_n$) and O(n), respectively.

V. SIMULATION

The following assumptions and parameters are predetermined.1) The message arrivals to a node follow the Poisson distribution. 2) The service time follows exponential distribution and FCFS rules. 3) Each node has infinite storage capacity. 4) Pairs of nodes are connected by a bi-directional full duplex link of same capacity, 1000 kbps. 5) Message length is 5000 bits, retransmissions and acknowledgments are not considered, and update period is 20 seconds.

The network graphs used in the simulations are constructed by the Waxman’s random graph model [9]. In this random graph, the edge’s probability can b

$$P_e(u,v) = \beta \exp\left(-\frac{d(u,v)}{\alpha L}\right),$$

Where *d* (*u*,*v*) is geometric distance from node *u* to node *v*, *L* is maximum distance between two nodes, parameter *α* can be used to control short edge and long edge of the random graph, and parameter *β* can be used to control the value of average degree of the random graph. In the simulation, *β*Âand *α*Áare chosen such that in average each node has a degree between 4 and 5. Geometric distance is used as delay on a link, and a random cost between 0 and 1 is generated for each link. For simplicity, links are assumed to be bi-directional and symmetric.

The average success ratio (W_{req}) of join requests can be given by

$$W_{req} = \frac{Q_{rpl}}{Q_{req}}$$

Where Q_{rpl} is the average number of join requests success, and Q_{req} is the total number of join requests.

We have taken the call blocking rate as the measure of performance since it is the proper manifestation of all the parameter of optimization (i.e., delay and bandwidth guarantees). In order to determine blocked calls, we first estimate the minimum available bandwidth $b_{avail}^{\min}$ for the multicast tree computed for the incoming call as follows:

$$b_{avail}^{\min} = MIN_{l \ni M}\,(b_{avail}^{l})$$

Where $b_{avail}^{l} = c_l - b_l$, is the residual bandwidth on a network, link *l* belonging to multicast *M*. Any multicast session request is considered to be blocked if its bandwidth requirement is more than $b_{avail}^{\min}$ at the time of its arrival. A call is also dropped if the computation time for the routing protocol exceeds the end-to-end delay requirement for the corresponding multicast session. The percentage of blocked calls is taken as a measure of routing performance. Fig.4 shows the relative performance of the two strategies with the increase in the session request arrival rate. The peak data rate for this comparison has been taken as 20 Mbps. Although the performance of both the schemes degrades with the increase of session arrival rate, HQMRP gains consistently and substantially over the Kou algorithm[17]. Here also, the results for HQMRP has been presented for multiple QoS parameter optimization (i.e., bandwidth, delay and delay jitter guarantees). As expected HQMRP performs better when multiple objective optimization is considered. The results presented here clearly suggest that HQMRP has an edge over the existing route selection strategies in terms of computation cost besides having better routing performance. This feature makes it deployable in a dynamic network for routing in an on-demand basis (i.e., whenever a request for multicast service arrives, HQMRP is employed to find the most suitable multicast tree based on the current network conditions). This helps in capturing the dynamism of the network which is evident from the performance comparison of HQMRP with Kou algorithm. Because of this on the fly multicast tree computation for a multicast request, the resource released by any session which terminates before the request is also taken into consideration and hence can be used for the new request. In Fig.4, *CAR* denotes the rate of call arrival and *CBR* denotes the percentage of call blocked.

The growth of the number of terminal nodes in the multicast tree with the increase in the number of domains considered, has also been studied. This measure gives an estimate for the total cost of the multicast tree. In the steady state, (i.e., when $T\rightarrow\infty$), the multicast group size is given by

$$\frac{N}{(1+\frac{\lambda}{\mu})^{un}},$$

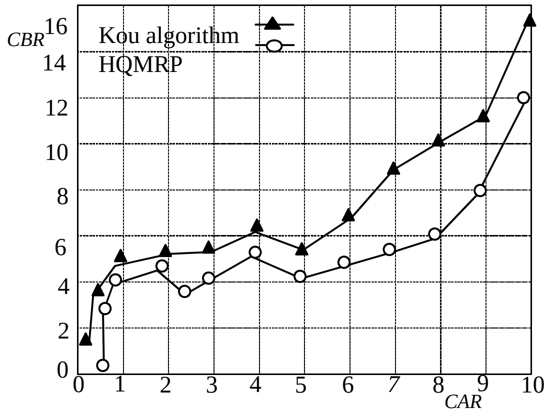


Fig.4 Comparisons of call blocking rate

where *N* is the number of domains and *un* is the expected number of user in each domain. λ and μ are the arrival rate and departure rate for a user in a domain, respectively. The values of the arrival rate and departure rate are assumed to be same expected number of users in a domain is taken to be 1. Simulation has been done for 10^5 secs and the result has been shown in Fig.5, where *NOD* is the number of added nodes, and *DOM* is the number of domains. As the plot shows, the simulation result matches closely with the numerical result. Perfect match was not obtained as infinite time is only an ideal case and practically infeasible.

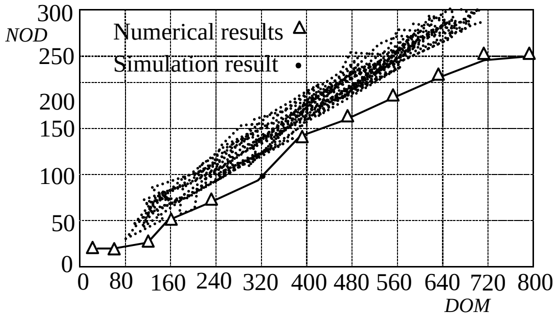


Fig.5 *NOD* vs. *DOM*

The proposed QoS multicast routing protocol was implemented in the simulation for hierarchical and flat network for both handw0idth and delay QoS constraints. The performances metric measures of HQMRP mainly include average success ratio for joining request and average message overhead.

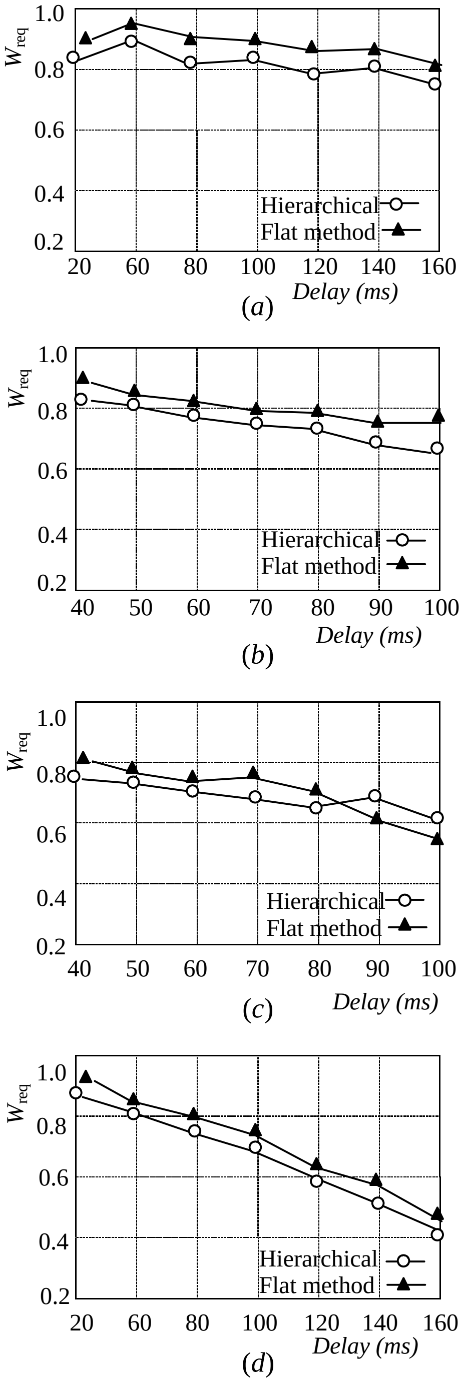
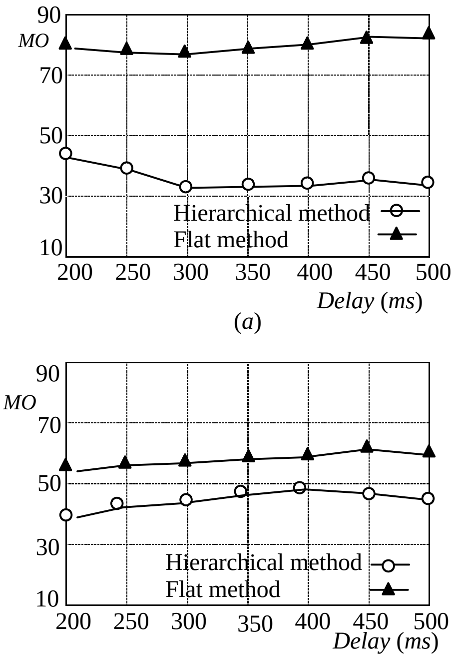


Fig.6 Average success ratio vs. delay

In this simulation experiment, the average degree of the node is four. Random networks with 200 nodes are used. In each simulation run, a random multicast tree with 30 (120) nodes is generated and a new member out of the tree is randomly selected. The source of the multicast group is also randomly selected. The delay constraint *D* of the new receiver is evenly distributed within [20,160] ms and bandwidth constraint *B* is evenly distributed within [10, 60]. Fig.6 shows the average success ratio for different delays and multicast group sizes.

As the same simulation parameters are used for the flat as well as hierarchical routings, the average success ratio has similar behavior in both cases. The path delay of *SF* messages increases with the node delay. Therefore, for a given tree size, the number of messages rejected by the QoS forwarding condition increases and, hence, the success ratio decreases with an increase in the node delay. Since the average path length of *SF* messages is higher for fewer on-tree nodes, this effect is more prominent for 10-node trees than in the other cases. In Fig.6, multicast group size is 50 for (*a*), the group size is 30 for (*b*), the group size is 20 for (*c*) and the group size is 10 for (*d*).



(*a*)

(*b*)

Fig.7 Message overhead vs. delay

Fig.7 shows the message overhead for the delay QoS constraints, where *MO* denotes the message overhead. In Fig.7, the group size is 50 for (*a*) and the group size is 10 for (*b*).

As shown in Fig.7. The advantage of hierarchical routing in terms of lower message overhead as compared to the flat routing scheme can be clearly seen from the figure for the delay.

VI. CONCLUSIONS

In this article, we have discussed the multicast touting problem for mobile ad-hoc networks, which may deal with multiple QoS constraints, such as delay, delay jitter, bandwidth and packet loss metrics, and mainly focuses the delay and bandwidth QoS constraints. This paper has presented an hierarchical QoS multicast routing protocol (HQMRP) for mobile ad-hoc networks. In HQMRP, each local node only needs to maintain the local multicast routing information and/or summary information of other clusters (or domains), but does not requires any global network states to be maintained. The HQMRP also allows that an ad-hoc group member can join/leave the multicast group dynamically. The paper presents formal description and main procedures for realizing routing decision process of HQMRP, and analyzes the dynamics of MANET. It has given the proof of correctness and complexity analysis of the protocol. The performance measures of HQMRP are evaluated using simulation. The studies show that HQMRP can provide an available approach to QoS multicast routing of mobile ad-hoc networks.

Acknowledgements

The work is supported by National Natural Science Foundation of China, NSF of Hubei Province, National 863 project, and Wuhan key Project.

References

[1] M. Faloutsos et al., QoSMIC: Quality of Service sensitive multicast internet protocol, SIGCOMM’¯in Proc of 98 September 1998, pp 56-63.

[2] L. Zhang et al., RSVP: A new Resource ReSerVation Protocol , IEEE Network (9) (1993). 62-70.

[3] X. Jia, A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks, IEEE/ACM Transactions on Networking 6 (6) (1998) 828-837.

[4] Li Layuan, Li Chunlin, A QoS-guaranteed multicast routing protocol, Computer Communications, Elsevier, Vol 27/1 pp. 59-69,2004

[5] Li Layuan et. al., The QoS routing algorithm for ATM networks, Computer Communications 24 (3/4) (2001) 416-421.

[6] Q. Zhu et al, A source-based algorithm for delay-constrained minimum-cost multicasting, in: Proc. IEEE INFOCOM 95, Boston, MA, April 1995, pp. 86-96.

[7] S. Chen et al., Distributed QoS routing in ad-hoc networks, IEEE JSAC, special issue on ad-hoc networks, Aug. 1999, pp.72-92.

[8] Li Layuan et al., A semantics-based approach for achieving self fault-tolerance of protocols. Journal of computer, Sci. and Technol. 15(2) (2000) 176-183.

[9] B. M. Waxman, Routing of multipoint connections. IEEE Journal of Selected Area in Communications, Dec. 1998, pp. 1617-1622.

[10] C. Perkins. IP mobility support. Internet Request for comments. RFC 2002, Oct. 1996.

[11] R. G. Busacker et al., Finite Graphs and Networks: An introduction with applications, McGraw-Hill, 1965.

[12] D. G. Thaler et al., Distributed center- location algorithms, IEEE JSAC 15(4) (1997) 291-303.

[13] I. Cidon, et al., Multi-path routing combined with resource reservation,

IEEE INFOCOM’¯97. April 1997, pp 9-100.

[14] Li Layuan, Li Chunlin, A distributed QoS-Aware multicast routing protocol, Acta Informatica , Springer-Verlag Heidelberg, Vol 40/3, pp. 211 – 233,November 2003

[15] Jeffrey P. Sutton, Ian M. D. Jamieson. Reconfigurable networking for coordinated multi-agent sensing and communications. Information Sciences. 151(5)(2003)103- 111.

[16] Ben-Jye Chang, Ren-Hung Hwang. Analysis of adaptive cost functions for dynamic update policies for QoS routing in hierarchical networks. Information Sciences. 151(5)(2003) 1-26.

[17] L. Kou, G. Markowsky and L.Berman. A fast algorithm for Steiner threes in graphs. Acta In formatica 15(1981) 141~145.

[18] R.A. Guerin et al., QoS routing in networks with inaccurate information: Theory and algorithms, IEEE/ACM Trans. On Networking 7 (3) (1999) 350-363.

[19] Li Layuan, Li Chunlin. A QoS multicast routing protocol for dynamic group topology, Information Science. 2004 to be published.

[20] P. V. Orlik and S. Rappaport. A model for teletraffic performance and channel holding time characterization in wireless cellular communication with general session and dwell time distribution. IEEE JSAC, 16(5) (1998) 788-803.

LI Layuan was born in Hubei, China on 26 February 1946. He received the BE degree in Communication Engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in Communication and Electrical Systems from Huazhong University of Science and Technology, China in 1982. He academically visited Massachusetts Institute of Technology, USA in 1985 and 1999, respectively. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a Professor and Ph.D. tutor of Computer Science, and Editor in Chief of the Journal of WUT. He is Director of International Society of High-Technol. and Paper Reviewer of IEEE INFOCOM, ICCC and ISRSDC. He was the head of the Technical Group of Shaanxi Lonan PO Box 72, Ministry of Electrical Industry, China from 1970-78.His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over one hundred and fifty technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.

Li Chunlin was born in 1974. She received the M.E. in computer science from Wuhan Transportation University in 2000, and Ph.D. degrees in Computer Software and Theory from Huazhong University of Science & Technology in 2003. She now is a lecturer of Computer Science in Wuhan University of Technology. Her research interests include mobile agent, distributed computing and computational grid. She has published over 15 Papers in international journals. She received Wu Fu award by Ministry of Communications, China in 1997.