A QoS Multicast Routing Protocol for Dynamic Group Topology

Li Layuan and Li Chunlin

Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P. R.China

jwtu@public.wh.hb.cn

Abstract. This paper discusses the multicast routing problem with multiple QoS constraints, which may deal with the delay, delay jitter, bandwidth and packet loss metrics, and describes a network model for researching the routing problem. It presents a multicast routing protocol with multiple QoS constraints (MRPMQ). The MRPMQ attempts to significantly reduce the overhead of constructing a multicast tree with multiple QoS constraints. In MPRMQ, a multicast group member can join or leave a multicast session dynamically, which should not disrupt the multicast tree. It also attempts to minimize overall cost of the tree, and satisfy the multiple QoS constraints and least cost (or lower cost) requirements. In this paper, the proof of correctness and a complexity analysis of the MRPMQ are also given. Simulation results show that MRPMQ is a feasible approach to multicast routing with multiple QoS constraints.

Keywords: Multicast routing; protocol; multiple QoS constraints; QoS routing; NP-complete.

1 Introduction

The traditional multicast routing protocols, e.g., CBT and PIM [1-4], were designed for best-effort data traffic. They construct multicast trees primarily based on connectivity. Such trees may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms [5-7] 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 most 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. Jia's distributed algorithm [2] 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 Carlberg and Crowcroft [3] handles dynamic membership and does not require any global network state. However, it has excessive communication and message processing overhead because it relies on flooding to find a feasible tree branch to connect a new member. QoSMIC [4], proposed by Faloutsos et al., 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.

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 980–988, 2003.

Multicast routing and its QoS-driven extension are indispensable components in a QoS-centric network architecture [5,7,9–10]. Its main objective is to construct a multicast tree that optimizes a certain objective function (e.g., making effective use of network resources) with respect to performance-related constraints (e.g., end-to-end delay bound, inter-receiver delay jitter bound, minimum bandwidth available, and maximum packet loss probability).

2 Network Model

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 network, respectively, Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes [8]. Associated with each link are parameters that describe the current status of the link.

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

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) $bandwidth\ (p(s,t)) = \min_{e \in p(s,t)} \{bandwidth\ (e)\}$.

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 $T\ (s, M)$.

Definition 1. The QoS-based multicast routing problem deals mainly with some elements: network $G=(V,E)$, multicast source $s \in V$, the set of end nodes $M \subseteq \{V - \{s\}\}$, $delay(\cdot) \in R$, $delay\text{-}jitter(\cdot) \in R^+$, $cost(\cdot) \in R$, $bandwidth(\cdot) \in R$, and $packet\text{-}loss(\cdot) \in R^+$. This routing problem is to find the $T\ (s, M)$ which satisfies some QoS constraints for all $t \in M$:

1）Delay constraint: $delay\ (p\ (s,t)) \leqslant D_t$

2）Bandwidth constraint: $bandwidth\ (p\ (s,t)) \geqslant B$

3）Delay jitter constraint: $delay\text{-}jitter\ (p\ (s,t)) \leqslant J$

4）Packet loss constraint: $packet\text{-}loss\ (p\ (s,t)) \leqslant L$

Simultaneously, the $cost\ (T\ (s, M))$ should be minimum. In the above QoS constraints, the bandwidth is a 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 an additive metric. For simplicity, we assume that all nodes have enough resources, i.e., they can satisfy the above QoS constraints. Therefore, we only consider the links' or edges' QoS constraints, because the links and the nodes are equivalent to the routing issue in question.

3 MRPMQ

The Join procedure of MRPMQ can be formally described as follows.

1) if a new member (t_i) wishes to join a $T\ (s,M)\rightarrow$
   the new member sends JOINreq to some neighbor t_j
2) if $(d(s,*)+d(j,i) \leqslant D) \wedge (dj(s,*)+dj(j,i) \leqslant J) \wedge (bw(t_u,t_v) \geqslant B) \rightarrow$
   {where $d(s,*)$ and $dj(s,*)$ are the delay sum and the delay jitter sum from the source s to all downstream nodes of a path, respectively, but except for the last pair of nodes. u and v are the sequence numbers between two adjacent nodes on the path from source to the new member}
   t_j transfers JOINack to t_i
   fi
   if $bw\ (t_u,t_v) < B \rightarrow$
   remove $e\ (u,v)$ from G
   fi
3) if $(d\ (s,*)+d\ (j,i) > D) \wedge (dj\ (s,*)+dj\ (j,i) > J) \rightarrow$
   if the next hop is the immediate upstream node $t_{j'}$ of $t_j \rightarrow$
   t_j transfers JOINreq to $t_{j'}$
   t_j adds JOINpend for t_i to the forwarding entry
   $t_{j'}$ transfers JOINack (or JOINnak) to t_j
   fi
   if the next hop is not the immediate upstream node $\rightarrow$
   if $(d\ (s,*) \geqslant D - d\ (j,i)) \wedge (dj\ (s,*) \geqslant J - dj\ (j,i))$
   t_j transfers JOINreq to t_{j*}
   t_j adds the routing entry
   marks t_{j*} as upstream node
   t_{j*} transfers JOINack (or JOINnak) to t_j
   if t_j receives JOINack $\rightarrow$
   t_j forwards a pruning msg to $t_{j'}$
   fi
   fi
   fi
   fi
4) if $(d(s,*)+d(j,i) \leqslant D) \vee (dj(s,*)+dj(j,i) \leqslant J) \rightarrow$
   t_j computes a new path
   if $(d\ (p\ (s,i)) = \min[d\ (s,*), D - d\ (j,i)]) \wedge$

$(dj(p\ (s,j))=\min[dj\ (s,*),\ J-dj\ (j,i)])\rightarrow$
t_j receives JOINack
fi
t_j receives JOINnak
fi

We can use the following example to show how the MRPMQ works and how the multicast tree is constructed in a distributed fashion. Fig. 1 is a network graph. In this example, node t_0 is the multicast source. t_4, t_9, t_{14}, t_{19} and t_{24} are the joining nodes. Recall that the characteristics of a network's edge can be described by a fourtuple (D,J,B,C). In this example shown in Fig. 1, suppose delay constraint is D=20, delay jitter constraint J=30 and bandwidth constraint B=40. t_4 wishes to join the group, it computes the paths according to the multiple QoS constraints. The path

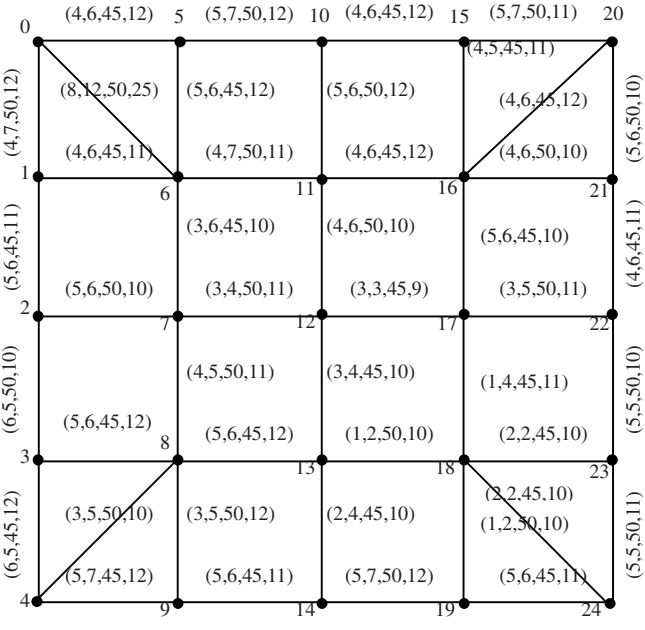

Fig. 1. An example network graph

$(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$ can satisfy the delay constraint, the delay jitter constraint and the bandwidth constraint, and has minimum cost. Therefore, the join path should be the path $(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$. The bold lines of Fig. 2(a) show the tree when t_4 has joined the group. When t_9 joins the group, it computes a path $(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_9)$ which should satisfy the delay, delay jitter and bandwidth constraints, and also have minimum cost. The JOINreq is accepted at t_8. The bold lines of Fig. 2(b) show the tree when t_9 has joined the group. When t_{14} joins the group, it computes the paths with multiple QoS constraints. The path $(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_{12}\rightarrow t_{13}\rightarrow t_{14})$ does not satisfy the delay jitter constraint. The path $(t_0\rightarrow t_5\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_{13}\rightarrow t_{14})$ does not satisfy delay and delay jitter constraints. The path $(t_0\rightarrow t_6\rightarrow t_7\rightarrow t_{12}\rightarrow t_{13}\rightarrow t_{14})$ and path

($t_0 \rightarrow t_5 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{12} \rightarrow t_{13} \rightarrow t_{14}$) satisfy the delay, delay jitter and bandwidth constraints. The latter has the lower cost. Therefore, the join path should be the path ($t_0 \rightarrow t_5 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{12} \rightarrow t_{13} \rightarrow t_{14}$). Meanwhile, t_6 should prune off from the original parent t_1, the resulting tree is shown in Fig. 2(c) (see the bold lines of Fig. 2(c)). The tree after t_{19} joins the group is also shown in Fig. 2(d). When t_{24} joins the group, it computes the join paths. If t_{18} receives JOINreq from t_{24}, it finds out that the existing path ($t_0 \rightarrow t_5 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{12} \rightarrow t_{13} \rightarrow t_{18}$) does not satisfy the delay constraint for the new member t_{24}, while the new path ($t_0 \rightarrow t_5 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{12} \rightarrow t_{17} \rightarrow t_{18}$) does not satisfy the delay jitter constraint for t_{24}. t_{18} computes a new feasible path with delay constraint, which is given by

$$\begin{aligned} d(p(s,j)) &= \min[d(s,*), D-d(j,i)] \\ &= \min[(d(0,5)+d(5,6)+d(6,7)+d(7,12)+ \\ &\qquad d(12,13)+d(13,18)), D-d(18,24)] \\ &= \min[19,18]=18 \end{aligned}$$

and delay jitter constraint, which can be given by

$$\begin{aligned} dj(p(s,j)) &= \min[dj(s,*), J-dj(j,i)] \\ &= \min[(dj(0,5)+dj(5,6)+dj(6,7)+dj(7,12)+ \\ &\qquad dj(12,13)+dj(13,18)), J-dj(18,24)] \\ &= \min[28,28]=28 \end{aligned}$$

Thus, this new feasible path should be path ($t_0 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{12} \rightarrow t_{13} \rightarrow t_{18}$). t_6 should prune off from the old parent t_5, and the final tree can be shown in Fig. 2(e) (see the bold lines of Fig. 2(e)).

The loop-free routing for the above protocol can be achieved by maintaining a searching tree at any time.

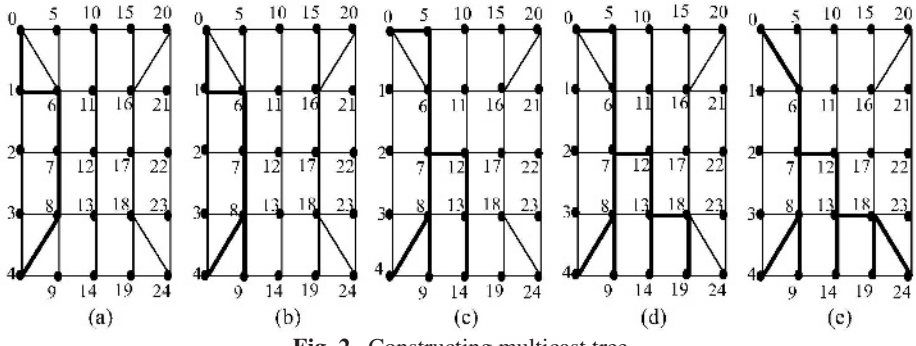

Fig. 2. Constructing multicast tree

4 Correctness and Complexity Analysis

Theorem 1. If a path from a new member to $T(s,M)$ has sufficient resources to satisfy the QoS constraints and has minimum cost, the algorithm searches only one path.

Proof. Note that a necessary condition for multiple paths to be searched is that a single path does not satisfy the QoS constraints, such as ($d(p(s,j)) \neq \min[d(s,*), D-d(j,i)]) \wedge (dj(p(s,j)) \neq \min[dj(s,*), J-dj(j,i)]$). However, if sufficient resources are

available on every link and node of the path, no node forwarding JOINreg will ever enter the multiple paths search state. Thus, the above theorem holds.

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 MRPMQ, 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 2. An available and feasible path found by MRPMQ is loop-free.

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

Theorem 3. MRPMQ can find an available and feasible path if one exists.

Proof. This theorem can be proved by contradiction. Suppose MRPMQ fails while an available and feasible path does exist. Let $e(i,j)$ be the first link in the path that the protocol did not explore. Since $e(i,j)$ is the first unexplored link of the path, t_i must have received a request message from the previous link or t_i is the new member issuing the request message. In either case, t_i is not in the initial state. Therefore, t_i is in the failure state, which requires t_i to explore all outgoing links including $e(i,j)$. It contradicts the assumption that $e(i,j)$ is not explored.

In MRPMQ, route computation can generally be made by the end node. If the join path is computed on-demand, the complexity depends on the unicast protocol. If QoS metrics are delay and bandwidth, there exist QoS routing heuristics which are O ($|V|$ x $|E|$), where $|V|$ is the number of nodes and $|E|$ is the number of edges in a network. For most networks, $|E|=O(|V|)$, hence the complexity is $O(|V|^2)$. For a multicast group with $|M|$ members, the computation overhead is $O(|V|^2|M|)$. The study shows that computation complexities of CSPT and BSMA [5] are O ($|E|$ log$|V|$) and O($|V|^3$ log$|V|$), respectively. The study shows that the average message processing overheads to construct the multicast tree of MRPMQ, Jia's algorithm, and QoSMIC (centralized or distributed) are $K.2|M|$, $K.2|M|$, $|M|$ $(w \bullet (w\text{-}1)^{(y\text{-}1)}+c\text{-}k) \bullet x$ (centralized QoSMIC) and |M| $(w \bullet (w\text{-}1)^{(y\text{-}1)}+|T|) \bullet x$ (distributed QoSMIC), respectively, where the x factor is added to reflect the fact that messages have to be processed at more than one node, w is the average degree of a node, y is the maximum TTL used for a search, $|T|$ is the tree size, c is the number of candidates for a BID-ORDER session and x depends on the topology and y, while $2 \leqslant x \leqslant 1+K$.

5 Simulations

In the simulations, we compare the quality of routing trees by their network cost for constructing a multicast tree ($cost$ (T (s, M))) [10]. The network cost is obtained as the mean value of the total number of simulation runs. At each simulation point, the simulation runs 80 times. Each time the nodes in the group G are randomly picked out from the network graph. The network cost is simulated against two parameters: delay bound D and group size. In order to simulate real situations, the group size is always made less than 20% of the total nodes, because multicast applications running in a wide area network usually involve only a small number of nodes in the network, such

as video conference systems, distance learning, co-operative editing systems, etc. [6–7].

Fig. 3 shows the network cost versus group size. In this round of simulations, the network size is set to 300 and D is d_{max}+3/8d_{max}. From Fig. 3, we can see when group size grows, the network cost produced by MRPMQ, BSMA and KMB increases at a rate much lower than CSPT. MRPMQ performs between BSMA and KMB. BSMA, KMB and the proposed MRPMQ can produce trees of comparable costs.

Fig. 4 is the network cost versus D. During this round of simulations, the network size is fixed at 300 nodes, group size is 20. From Fig. 4, it can be seen that the network cost of the CSPT algorithm is on the top and almost does not change as D increases. This is because the generation of the shortest path tree does not depend on D. Of the remaining three algorithms, the proposed MRPMQ has the lowest cost. From Fig. 4, we can also see that tree costs decrease for the MRPMQ, BSMA and KMB algorithms as the delay bound is relaxed. This shows all three schemes indeed can reduce the cost when the delay bound is relaxed. From Fig. 3 and Fig. 4, one can see that the MRPMQ, BSMA and KMB algorithms can produce trees of comparable costs. However, compared with the BSMA and KMB algorithms, the proposed MRPMQ has the advantage of being fully distributed and allowing incremental tree build-up to accommodate dynamic joining of new members. Furthermore, the MRPMQ is much less costly in terms of computation cost and in terms of cooperation needed from other network nodes compared with other schemes.

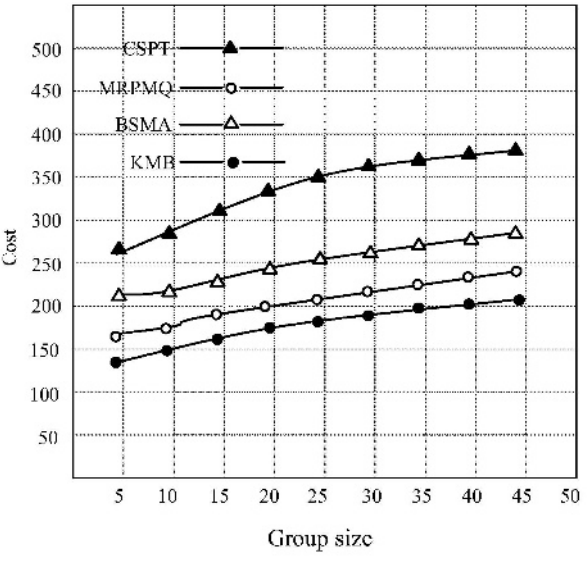

Fig. 3. Network cost vs. group size

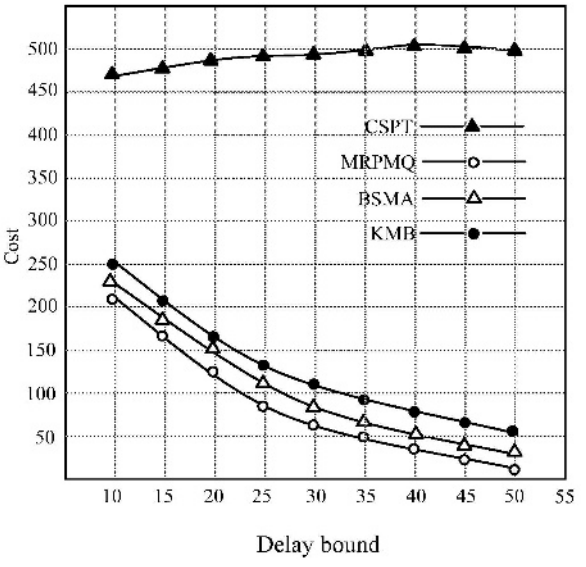

Fig. 4. Network cost vs. delay bound

6 Conclusion

In this paper, we discuss the multicast routing problem with multiple QoS constraints, which may deal with the delay, delay jitter, bandwidth and packet loss metrics, and describe a network model for researching the routing problem. We have presented a multicast routing protocol with multiple QoS constraints (MRPMQ). The MRPMQ can significantly reduce the overhead of establishing a multicast tree. In MRPMQ, a multicast group member can join or leave a multicast session dynamically, which should not disrupt the multicast tree. The MRPMQ also attempts to minimize the overall cost of the tree. This protocol may search multiple feasible tree branches in distributed fashion, and can select the best branch connecting the new member to the tree. The join of a new member can have minimum overhead to on-tree or non-tree nodes. The correctness proof and complexity analysis have been made. Some simulation results are also given. The study shows that MRPMQ is a feasible approach to multicast routing with multiple QoS constraints. Further work will investigate the protocol's suitability for inter-domain multicast and hierarchical network environments.

Acknowledgment. The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

References

[1] Li Layuan and Li Chunlin, "The QoS routing algorithm for ATM networks", *Computer Communications*, Vol. 24, No. 3-4, 2001, pp. 416–421.
[2] X. Jia. "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks", *IEEE/ACM Trans. on Networking,* Vol. 6, No. 6, Dec. 1998, pp. 828–837.
[3] K. Carlberg and J. Crowcroft, "Building shared trees using a one-to-many joining mechanism", *ACM Computer Communication Review*, Vol. 27, No. 1 , Jan. 1997, pp. 5–11.
[4] M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: Quality of Service sensitive multicast Internet protocol", *SIGCOMM'98*, Vol. 28, September 1998.
[5] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting," *Proc. IEEE INFOCOM 95*, Boston, MA, April 1995.
[6] Y. Xiong and L.G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks", *IEEE/ACM Trans. on Networking*, Vol. 7, No. 1, Feb. 1999, pp. 98–110.
[7] B. M. Waxman, "Routing of multipoint connections." *IEEE Journal on Selected Area in Communications*, Dec. 1998, pp. 1617–1622.
[8] R. G. Busacker and T. L. Saaty, *Finite Graphs and Networks: An introduction with applications*, McGraw-Hill, 1965.
[9] R. A. Guerin and A. Orda, "QoS routing in networks with inaccurate information: Theory and algorithms," *IEEE/ACM Trans. on Networking*, Vol. 7, No. 3, June 1999, pp. 350–363.
[10] Li Layuan and Li Chunlin, *Computer Networking*, National Defense Industry Press, Beijing, 2001.