Multicast Routing for Multimedia Applications

Chang-Ben Wang and Ren-Hung Hwang
Department of Computer Science & Information Engineering
National Chung Cheng University
Chia-Yi, 621, Taiwan
Email: rhhwang@cs.ccu.edu.tw

ABSTRACT

Due to new techniques of Internet, many novel multimedia applications have emerged. These applications include video on demand (VOD), video/audio phone, distance learning, interactive games, etc. There are two major characteristics of these applications. One is the use of group communication. The other is the requirement of more network bandwidth and quality of service guarantee. As a consequence, whether the underlying routing protocols can provide efficient multicast service and quality of service guarantee becomes a very important issue. In this paper, we study many performance issues of routing multimedia applications using existing Internet multicast routing protocols. Our numerical results show that the routing performance can be improved by careful core placement, appropriate link cost function, and multiple multicast trees routing.

1. Introduction

In the current Internet, rapidly hardware revolution accompanies numerously emerging brand-new software applications. For example, a live lecture of distance learning needs to deliver packets of audio, video and text from one or more senders to many recipients. In other words, multimedia data transmission on the upcoming G-bps high-speed network is practical and possible. An inefficient transmission scheme wastes more network resource and lets transmission costly. Thus, researching and obtaining a more efficiently group communicating scheme is the key issue for providing such a multimedia application services.

In the past research, the goal of multicast routing is to find a tree with minimum cost. However, the least cost multicast tree problem, which is also known as the Steiner tree problem has been shown to be an NP-complete problem [1]. Therefore, many heuristic algorithms have been proposed. For example, most existing Internet multicast routing algorithms are based on shortest-path tree instead of Steiner tree. Currently, there are four well-known routing protocols in the Internet [2], namely, Distance Vector Multicast Routing Protocol (DVMRP) [3], Multicast extensions to OSPF (MOSPF), Core-Based Tree (CBT) [4], and Protocol-Independent Multicast (PIM) [5]. DVMRP is the first multicast routing protocol used in the Internet. It based on the Bellman-Ford distance vector algorithm and use reverse path forwarding technology to construct a “source-base tree”. MOSPF is based on OSPF (Open Shortest Path First) and use link-state algorithms for path selecting. CBT is the first group-share protocol that constructs a single delivery tree root at core and shared by all group members. PIM is proposed to possess the advantages of both source-based and group share protocols. It allows receivers to choose between source-based trees or group share tree. The sparse mode and dense mode of PIM are similar to CBT and DVMRP, respectively. In summary, we may classify current multicast routing protocols into two categories: source-based and group-shared.

Many researchers have addressed on different issues on multicast routing. For example, Kompella and Pasquale presented heuristics for multicast tree construction with goal of minimizing cost of the multicast tree under the end-to-end delay constraint [6]. Liu *et al.* proposed a central core-manager for core selection [7]. However, many issues remain open. For example, for group shared tree protocols, how to elect cores remains an open issue. CBT version 2 suggests two options for core discovery: “manually configure” and “hash function”. However the hash function only focus on the consistence of core selection but not tree cost minimization. In some literature, e.g., [7,8], the core of CBT is selected randomly on topology. Another open issue is which category of multicast routing protocols yield better performance for multimedia applications which require QoS guarantee? Yet, another issue is how to select state-dependent link cost function [9] for routing multimedia connections.

For routing multimedia connections, there is another interesting issue that has not been well studied. Literally, multimedia is the combination of two or more continuous media [10,11]. Therefore, when routing multimedia connections, we have a choice between routing all media streams on a single route (tree) or routing different media separately. While routing different media separately, the inter-media synchronization problem also becomes an issue.

Therefore, in this paper, we will focus on the above mentioned issues. We will adopt a state-dependent link cost function and use fractional reward loss (FRL) as the performance metric [12]. We first study the effect of core placement on the performance of routing for group-shared protocols. We then compare the routing performance of source-based protocols and core-based protocols from the network bandwidth utilization standpoint. Finally, we investigate some issues of routing multimedia applications on multiple multicast trees.

The remainder of the paper is organized as follows. Section 2 describes network model and gives the definition of multicast routing problem. In Section 3, we will state how to define the cost of a link. Two well-known multicast routing protocols on the Internet today will be provided in Section 4. In Section 5, experimental results are presented. Finally we summarize our finding and discuss some interesting future works in Section 6.

2. Network Model and Problem Definition

2.1 Network model

Consider a network contain a set of routers *V* that are connected by some transmission links *E*. The routers perform multicast communication with each other and route according to some multicast routing protocol. Such a network topology can be modeled as a weighted graph $G=(V,E)$ with the node set $V=\{v_1, v_2, \dots v_m\}$ and the edge set $E \subseteq V \times V$. An edge element, $e(v_i, v_j) \in \mathrm{E}$, indicates there is a link connecting the node pair (v_i, v_j), for $1 \le \mathrm{i}, \mathrm{j} \le \mathrm{m}$.

Definition 2.1 Definition of Path

A path from v_1 to v_k is a vertices sequence $v_1, v_2, \dots, v_k$ where

(a) $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k) \in E$ are links on G.

(b) Each vertex in the sequence appears only once.

In considering the characteristics of different transmission media, each medium has distinct properties such as capacity, delay etc. We depict these properties below. For each link ℓ on G, function $Cap(\ell) \to Z_0^+$ gives the capacity of link ℓ, which is a positive integer number. Initially, edge link ℓ has $Cap(\ell)$ units of usable resource. After a successful route passes through this link, usable resources are decreasing according bandwidth reservation of the incoming request. We assume that edges and their associated capacity are full duplex. That is, if there exists a link from node v_1 to v_2, then there must be another link with the same capacity from v_2 to v_1, see Figure 2.1.

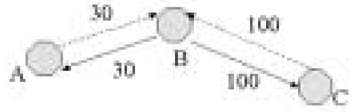

Figure2.1 Link symmetric assumption

2.2 Problem Definition

In this section, we will describe the multicast problem first. Multicasting refers to transmit a single data packet to a set of destinations.

Definition 2.2 Multicast connection request

A multicast connection request is described by three parameters (s, D, b), where

$s \in V$ is the source node of the connection,

$D \subset V$ is the set of destination nodes to be connected,

b is non-negative integer of bandwidth request.

A QoS multicast routing problem is to find a multicast tree $T=\{V_T, E_T)$ from source *s* to all other group members *D* that satisfies a set of constraints. Therefore, we have the following definition.

Definition 2.3 QoS multicast routing problem

$$\sum_{\forall \ell \in E_T} Cost(\ell) \le C,\ C \in R_0^+,\ C : Cost\ constraint \quad (1)$$

$$\sum_{\forall T \in T_n} (allocate\ \ bandwidth\ \ on\ \ \ell\ \ by\ \ T) \le\ Cap(\ell)\ \ \forall \ell \in \mathrm{E} \quad (2)$$

Let $Cost(\ell)$ denotes link cost of ℓ, R_0^+ be the set of non-negative real numbers. We define $Cost(\ell) \to R_0^+$, i.e., the cost of link ℓ is positive real number. We denote a multicast tree by $T=\{(V_T, E_T) \mid V_T \subseteq V, E_T \subseteq E\}$. Clearly there might be lots of multicast trees sending data on the network at the same time; each of them occupies appropriate bandwidth for group communicating.

Furthermore, let W be the set of all (s, D) pairs within multicast connection requests, $W=\{w=(s,D) \mid s \in V, D \subset V\}$, and suppose that a request of $s-D$ pair of request occupies *b* units of resource. From another point of view, the service provider's (ex: Internet Service Provider's - ISP or telephone exchange) revenue r_w should be proportional to the resource provided to their customers. Furthermore, from equation (1), in order to guarantee QoS for established connections, a new connection request may be either accepted or rejected according to the cost constraint *C*, which is referred to as the call admission function. In our simulation, we set the constraint *C* to be proportional to connection reward r_w. Besides, we assume that multicast connection requests arrive at $s-D$ pair w according to a Poisson process with mean rate λ_w. The call holding time is assumed to be exponentially distribution with mean 1/μ.

In the past research on multicast routing, a multicast tree with minimum cost is preferred. The performance of a heuristic multicast routing algorithm was evaluated by comparing the cost of heuristic multicast routing algorithm and cost of the optimal Steiner tree. In this paper, we adopt another performance metric for evaluating multicast algorithm [12,13], which is referred to "*fractional reward loss*".

Definition 2.4Fractional reward loss (FRL)

$$Fractional\ Reward\ Loss = \frac{\sum_{w \in W} r_w \lambda_w B_w}{\sum_{w \in W} r_w \lambda_w} \quad (3)$$

In equation (3), B_w is the traffic blocking probability of $s-D$ pair w, the denominator is the expected revenue of all connection requests are accepted the numerator is revenue loss due to connection rejected. Minimizing the fractional reward loss is equivalent to maximizing the expected revenue

From now on, our goal is to minimize the fractional reward loss. Because the connection blocking probability affects FRL, the cost of a link must be carefully defined to avoid blocking. In the following section, we examine a quantitative approach of link cost function that called Competitive On-Line (COL) cost function [9].

3. Link Cost Function

Routing is to determine a feasible path from source to destination. Thus, we need a quantitative approach for cost calculating, which gives assistance for path finding. In this paper, each link associated with a link cost that is exponential of its current bandwidth utilization. The accumulation of these link costs forms a path cost. The connection is then routed along the path with minimum cost.

In addition, in order to provide quality of service guarantee, the network needs to reserve resources during connection set up. Admission control is also needed since network resources are limited. In general, the link cost function plays as a role of routing decision that decides what path should be used while the call admission control decides whether the request should be accepted or rejected. In the following, we describe a state-dependent link cost function based on Competitive On Line (COL) approach and its call admission control policy.

3.1 COL Approach

Gawlick, *et al.*, first proposed COL routing algorithm [14] which assigns each link a cost that is exponential in its bandwidth utilization. Formally, we define $Cost(\ell_i)$, the cost of link ℓ with occupancy of *i* units of bandwidth, as follows:

$$Cost(\ell_i) = \mu^{\frac{i}{Cap(\ell)}}, \qquad (4)$$

where μ is a parameter to be determined. In our simulation, μ is chosen to be the same as link capacity $Cap(\ell)$. Zhang *et al.* [15] presented a method for setting the cost threshold to 1, by

$$Cost(\ell_i) = \mu^{\frac{i}{Cap(\ell)}-1}, \qquad (5)$$

and we propose to define the link cost when receiving a request reserve *b* units of capacity on link ℓ, $p_\ell(i)$, by

Definition 3.1 Link cost assignment under a new incoming request

$$p_\ell(i) = \begin{cases} \sum_{j=i}^{i+b-1} \mu^{\frac{i}{Cap(\ell)}-1} & If \quad i+b \le Cap(\ell), \\ \infty & otherwise \end{cases} \qquad (6)$$

Equation (6) provides the cost assignment of link ℓ where *i* units of resource were occupied. That is, $p_\ell(i)$ is the cost for reserving *b* units of bandwidth on link ℓ when there are *i* units of bandwidth have occupied, already. Naturally, the allocated bandwidth *i* plus new reserving *b* must be less or equal to the original link capacity $Cap(\ell)$ otherwise the equation result in infinite cost. Furthermore, the path cost is the sum of the link cost along the path:

$$Path\_Cost(P) = \sum_{\ell \in P} p_\ell(i) \qquad (7)$$

3.2 Call admission control of COL approach

After defining path cost, we introduce another chosen parameter ρ in COL. In the COL routing algorithm, a threshold parameter ρ is used for admission control which is defined as:

Definition 3.2 Designated admission threshold ρ

$$\begin{cases} \sum_{\ell \in P} p_\ell(i) \le \rho \times r_w & \text{accept this request,} \\ otherwise & \text{reject this request,} \end{cases} \qquad (8)$$

where r_w is the reward of the connection. A multicast routing algorithm first finds the multicast tree with minimum cost. If the total cost of the tree is smaller than the threshold $\rho \times r_w$, the request is accepted; otherwise, the request is rejected. Therefore, two reasons result in a new incoming call to be rejected; one is insufficient available residual capacity, the other is that total tree cost exceeds the admission threshold.

Note the path cost in equation (7) is the cost for a unicast path. When an incoming multicast request contains more than one destination, the cost of a multicast tree will be the sum of all link costs of the links on the tree. In the following section, we shall present several multicast algorithm that are used for constructing multicast trees.

4. Multicast Routing Protocols

We have defined graph topology and multicast connection request to describe multicast routing problem in Section 2. In Section 3, a quantitative approach to define link cost has been introduced. Thus for a given multicast request (s,D,b), we have enough information to construct a multicast tree, which contains a root at source node, *s*, and a group of members, *D*. In this section, we describe some basic concepts about our multicast algorithms; this is a preliminary works of our simulation. First, we discuss the basic concept of multicast tree constructing. Then a brief introduction of current Internet protocol will be provided.

4.1 Multicast-tree algorithms

The objective of multicast routing is to find a tree of links that attaches all of the necessary routers on the tree. Sometimes, we need to involve some routers that none of the attached hosts is group members for tree constructing, which referred to “intermediate node”. Figure 4.1 depicts a multicast tree contains five routers D-A-B-C-E, router D has a group member host 6, and another non-group member host 5; all hosts attached to the intermediate node router C are not group members

Different protocols construct multicast tree in different way. However, based on the concepts behind them, we may classify these protocols into two categories: *Group-shared protocol* and *Source-based protocol*. An example of Group-shared protocol is CBT and an example of source-based protocol is DVMRP. In current Internet, two categories of multicast routing protocol were used for tree constructing. In the follow section, a brief overview of these two categories will be discussed.

4.2 Source-based trees (Distance Vector Multicast Routing Protocol)

DVMRP is the first routing protocol used in the Internet and the most widely supported multicast routing protocol. It uses reverse path forwarding approach to construct a multicast tree. Routing of DVMRP contains two phases: “forwarding” and “pruning”. Once a multicast data packet arrives at one of the interface of a DVMRP router, if the reverse path from incoming interface to it’s predecessor is the shortest path on it’s own to upstream, the packet will be transmitted on all of it’s outgoing link. Otherwise, the router discards the incoming packet.

As shown in Figure 4.2(a), let the bold lines indicate the least cost paths (shortest paths) from each node to source X. When router A received a packet from X, it then broadcasts to all of A’s outgoing links. Router C does the same as router A, as shown by the arrows in Figure 4.2(a). After router C has sent packets to leaf routers D and E that do not have any attached hosts joining multicast group. Leaf routers D and E will issue a prune message to its upstream, as shown by the dashed lines in Figure 4.2(b). The dashed links with a bold circle in Figure 4.2(b) show the final multicast tree built by DVMRP.

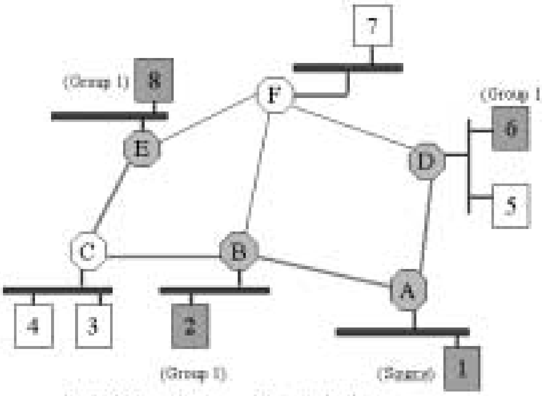

Fig 4.1 Multicast Routers and their attached hosts

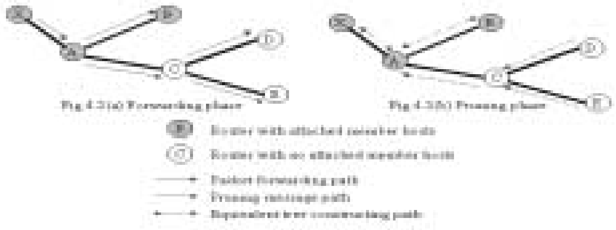

Fig 4.2(a) Forwarding phase

Fig 4.2(b) Pruning phase

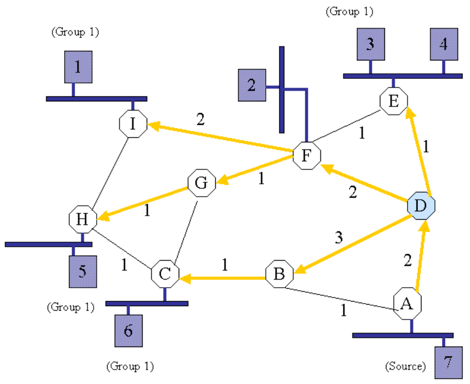

Figure 4.3 Sketch of CBT protocol

4.3 Group-shared trees (Core-Based Trees)

Shared tree architecture offers an improvement in scalability over source-based tree architectures. The CBT (Core-Based Tree) multicast routing protocol builds group-shared tree with a single "core". Figure 4.3 illustrates the basic concept of CBT, where an octagon represents a router and a rectangle represents a host. As shown in Figure 4.3, the source node host 7 sends data to core router D through its designated router A first. Promptly, core router D constructs multicast trees to all group members and forwards data from source. We assume that group membership behavior is static in our simulation, i.e. hosts don't change their state of grouping relation which may comes from a new others *Join*, or a exist one *Leave group* for the duration of the call.

However, the crux of the algorithm about CBT is how to determine the core router. Therefore, we should concentrate on the core selection procedure. From RFC 2189[4], CBTv2 uses "bootstrap" mechanism for core obtaining, which includes three major elements; bootstrap router (BSR), the Candidate Cores (CCs), and the Core-Set (CS). Normally, there are also a few set of routers configured as Candidate BSRs (CBSRs) in a CBT multicast domain. D. Estrin *et al.* suggested to configure the set of CCs the same as CBSRs within a domain [5]. The following section gives a precise description of these three elements fit together to realize the bootstrap mechanisms.

4.3.1 Obtaining Core information of CBT

A CBT multicast domain is a neighboring set of routers that all supported CBT protocol and are configured to operate within a common boundary.

(a) BSR election by bootstrap message

A Bootstrap router (BSR) is dynamically selected within a CBT domain. It is responsible for collects and maintains the set of Candidate Cores (CCs), and distributes the resulting set of Cores (CS) to all the CBT routers in the CBT domain through bootstrap message. The bootstrap message not only informs CS to each CBT router but also elects a BSR among CBSRs by comparing BSR priority.

(b) CS constructing using Candidate-Core-Advertisements

There is a set of routers that are configured as Candidate Cores (CCs) within a CBT domain, typically these are the same routers that configured as CBSRs. Once a BSR for the domain is determined, a periodically unicast Candidate-Core -Advertisement messages is sent form Candidate Cores to the elected BSR. Then, BSR chooses a subset of the living Candidate Cores to form the Core-Set, which it then distributes in the bootstrap message.

From the above properties, the routers in CBT domain could obtain Core-Set by receiving a bootstrap message. In our simulation, three proposed core placement methods would be compared, namely *Random_Core*, *One_Hop_Core* and *Min_Hop_ Core*. Detail of the three methods and experiment results may be found in Section 5.2.1. Furthermore, nodes with more outgoing link are preferred to be selected as Candidate Cores.

5. Simulation Result

Certainly, experiment helps us further understanding QoS routing problem. In this section, extensive simulations are performed to evaluate several mechanisms that proposed in this paper. All of the simulation results were carried out on a random graph $G=(V,E)$ in which the $|V|$ nodes are randomly distributed on a $|V|\times|V|$ Cartesian coordinate [16]. In Section 5.1, we shall discuss the simulation models, including graph topology and traffic pattern we used in the simulation. Next, in Section 5.2, the numerical results of four issues in our experiments will be presented.

5.1 Simulation Model

Topology model

A 20-node random graph shown in Figure 5.1 is used in our simulation; each link is full duplex with a bandwidth capacity of 6.312 Mbps (the T2 carrier).

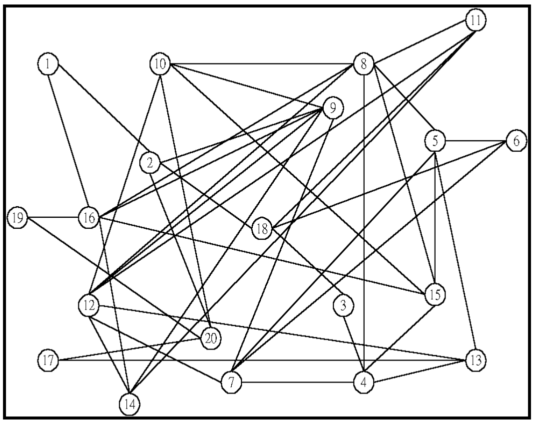

Figure 5.1 A 20-nodes random graph in our simulation

Traffic Model and Other Assumptions

In this section, we examine the traffic load on our network topology. A presentation of multimedia data could be characterized by several kinds of media stream transmitting to multiple recipients at appropriated time [11]. For example, first, a movie displays its program topic (*text data*); background music comes after (*audio data*), the principal part of the movie then played out soon (*video, audio and text sending simultaneously*). Each class of data requires different bandwidth. Table 5.1 gives the requirement detail. In Table 5.1, video data stream follows the H.323 standard and audio/text class data is based on 64K bps PCM (Pulse Code Modulation). In our simulations, we normalize the bandwidth unit to 64 K bps. Therefore, the bandwidth requirement to transmit video data and audio/text data is 6 and 1, respectively. All multicast connection requests arrive as a Poisson process with a mean rate λ. Let W_i denote the set of all multicast requests which have exactly *i* destinations and λ^{W_i} denote the arrival rate of W_i. The reward of a multicast request (s,D,b) is set to $|D|\times b/\mu$ which proportional to the number of destinations $|D|$, reserve bandwidth *b*, and the mean of call holding time. In our simulations, we let $\lambda^{W_1}/\lambda^{W_i}=i$, i.e. arrival rate of unicast connection is higher than multicast sessions. All of the holding time associated with a connection is exponentially distributed with a mean of 1 unit of time. The 95% confidence interval of each numerical point is also calculated.

(a) Video Class	384K Bits
(b) Audio + Text Class	64K Bits

Table 5.1 Bandwidth requirements for each class of data stream

5.2 Simulation Results

Before introduce our simulation results, we define an n-element parameter array, which describe the simulation environment and input parameters. It makes the parameters description clearly. We use $\{T,S,\lambda,nruns,Th\}$ for short in the rest of this paper.

Definition 5.1 Input parameter array

$Parameter=\{Transient, Stop, \lambda, nruns, Threshold\}$ indicates

Transient period

Simulation period for a single run

λ: Unicast request arrival rate, multicast requests are inverse proportion to it.

$nruns$: The independent runs amount we have

Th: Cost threshold used for call admission control

Definition 5.1 only shows the general case of the input parameters. Additional, parameters are given in each case below, if they are necessary.

5.2.1 Effect of Core Placement

The CBT establishes a single shared tree for a multicast connection. However, the location of the core may affect the cost and performance of the CBT. First, we comprise three methods for core election.

The first called *Random_Core,* which selects cores on the topology randomly, i.e., each node on the topology may be elected as a core router. Second is *One_Hop_Core*, We are interested to know whether the performance becomes better or not, when core router is next to source node. Therefore, we make a node is chosen to be core if it is a close neighbor of the source under the *One_Hop_Core* policy. The last one referred to as *Min_Hop_Core*. In this policy, we first elect some *Candidate Cores* (CCs) from a given area. Recall the description in Section 4.3.1 that nodes with more edges took the initiative in announcing as Candidate Core routers. Then, entire Candidate Cores form a core pool that is about 25% amount of the total nodes. Once a multicast connection request arrives at source node and accompanies with a set of destinations. For each Candidate Core in the core pool, the node with minimum hop count to the source node is selected as the core.

Table 5.2 shows the average reward loss value for these placement policies with a parameter array $\{T=1000, S=6000, \lambda=12-24, nruns=10, Th=2.0\}$. Experiment results are also presented in Figure 5.2. The x-axis represents arrival rate and the y-axis represents the average *fractional reward loss* values. Figure 5.2 shows the FRL values is increasingly proportional to traffic load. The *One_Hop_Core* policy performs slightly better than *Random_Core*. As the traffic load decreases, the performance difference among these core placement policies becomes smaller. Overall, *Min_Hop_Core* has the best performance no matter what traffic load is because it combines the advantage of load balancing (due to the CCs have more outgoing link and state dependent cost function is used) and less hop count number to source.

From above experiments, we found that *Min_Hop_Core* policy yields the best performance. Therefore, we make it as the core placement policy of our CBT protocol, except when explicitly stated.

Arrival rate	Random_Core	One_Hop_Core	Min_Hop_Core
λ =12	3.031913E-03	2.544707E-03	8.344264E-04
λ =14	1.424095E-02	1.146666E-02	5.581438E-03
λ =16	4.006108E-02	3.349241E-02	2.031180E-02
λ =18	7.755656E-02	6.804512E-02	4.853319E-02
λ =20	1.225408E-01	1.091666E-01	8.776384E-02
λ =22	1.696888E-01	1.540966E-01	1.330442E-01

Table 5.2 FRL of 3 core placement methods

5.2.2 Performance Comparison of CBT and DVMRP

Recall the difference between *Source-based tree (DVMRP)* and *Group-shared tree (CBT)* is that DVMRP constructs tree from the destinations' view while CBT constructs a group-shared tree that is the shortest-path tree rooted at core. In this section, we focus on the performance of the different tree constructing mechanisms. We set the parameter array to $\{T = 1000, S = 6000, \lambda = 12 - 22, nruns = 10, Th = 2.0\}$.

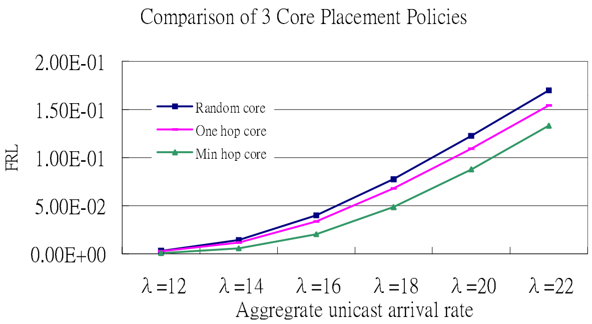

Fig 5.2 Performance comparison of 2 core placement

The results are shown in Fig 5.3. As we can see that DVMRP yields lower FRL than CBT. This result that shown in Fig 5.3 is intuitive because without concentrating traffic on one group-shared tree, DVMRP allows more requests to be accepted. By contrast, *Group-shared tree (CBT)* protocol constructs a tunnel from source node toward core router for each distinct multicast group and the core router may become bottleneck easily. This is also the reason that why we preferred the Candidate Cores (CCs) with more outgoing link in *Min_Hop_Core* policy.

Average FRL	CBT	DVMRP
λ =12	8.344264E-04	2.791697E-04
λ =14	5.581438E-03	2.049930E-03
λ =16	2.031180E-02	8.596170E-03
λ =18	4.853319E-02	2.455533E-02
λ =20	8.776384E-02	4.997987E-02
λ =22	1.330442E-01	8.449274E-02

Table 5.3 Performance comparison of different multicast protocols.

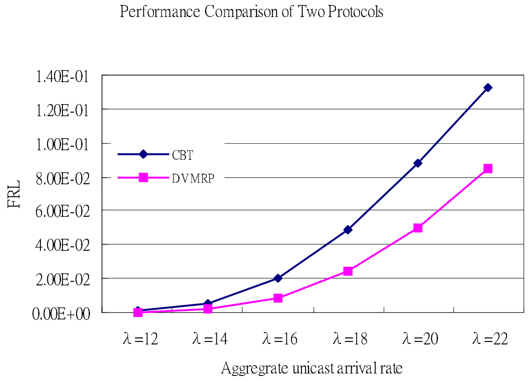

Fig 5.3 Performance comparison of CBT and DVMRP.

5.2.3 Multiple Multicast Trees Routing

In the above experiments, multimedia data are transmitted on a single multicast tree, i.e., video, audio and text streams are all carried on the same multicast tree. In this section, we are interested on dispersing traffic to multiple multicast trees. The idea is dispatching multimedia streams to different trees. As shown in Table 5.1, we classified two categories of multimedia stream: video and audio. Different type of data is transmitted separately. For example, video tree is responsible for transmitting the video data and audio streams are transmitted on the audio tree.

However, synchronization between two media streams becomes a new issue in trying multiple-trees transmission. Thus, we limit the delay variance between two media streams to less than a threshold. In our simulations, media streams from source to each destination may travel different paths, but the length difference of each path is less than two hops for distinct multicast tree. Figure 5.4 shows the FRL values as a function of traffic load for the given topology. The simulation parameters are set to $\{T = 1000, S = 6000, \lambda = 12 - 24, nruns = 10, Th = 2.0\}$. We observe that despite the synchronous restriction, traffic dispersing on multiple trees performs much better when network load is heavy.

Arrival rate(lam)	One_tree	Two_tree
λ =12	8.344264E-04	2.254513E-04
λ =14	5.581438E-03	1.809067E-03
λ =16	2.031180E-02	8.314307E-03
λ =18	4.853319E-02	2.460639E-02
λ =20	8.776384E-02	5.274180E-02
λ =22	1.330442E-01	8.898198E-02
λ =24	1.776979E-01	1.293233E-01

Table 5.4 Effect of multiple multicast trees routing

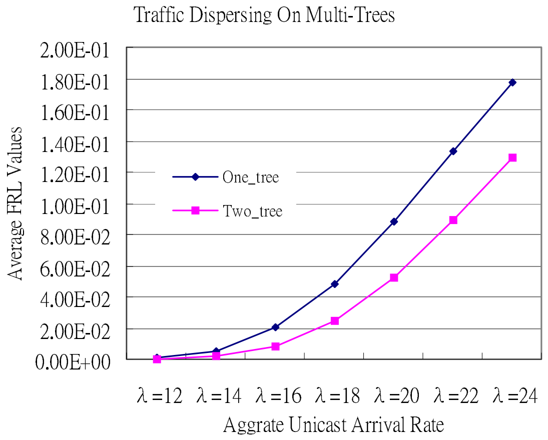

Figure 5.4 Effect of multiple multicast trees routing

5.2.4 Condition for Efficient Traffic Dispersing

We have seen the advantage of dispersing traffic to different trees in the previous section. Intuitively, the bandwidth requirement proportion of video and audio streams will affect performance by load balancing. In other words, we are interested in what the proportion is worth of trying traffic dispersing. Intuitively, closer video/audio ratio earns more profit from this scheme. Following simulation results try to verify this conjecture.

Recall the traffic model shown in Table 5.1, video and audio streams require network bandwidth of 384 Kbps and 64 Kbps respectively. After normalize the bandwidth unit to 64 Kbps, the bandwidth reservation ratio is 6:1, hence the sum of total bandwidth requirement is 7. We preserve this property and modify video/audio ratio to 5:2. Figure 5.5(a) verifies it is worth dispersing traffic into different multicast tree when the proportion of two media streams is closer. For the same reason, media streams with 12:2 ratio performs better than 13:1 ratio does. Figure 5.5(b) gives the indication of this.

Arrival rate(lam	7:0	6:1	5:2
λ =12	8.344264E-04	2.254513E-04	2.009187E-04
λ =14	5.581438E-03	1.809067E-03	1.176826E-03
λ =16	2.031180E-02	8.314307E-03	5.156780E-03
λ =18	4.853319E-02	2.460639E-02	1.594777E-02
λ =20	8.776384E-02	5.274180E-02	3.730148E-02
λ =22	1.330442E-01	8.898198E-02	6.781833E-02
λ =24	1.776979E-01	1.293233E-01	1.042278E-01

Table 5.5(a) Performance evaluation under distinct media proportion

Arrival Rate(lan	14:0	13:1	12:2
λ =6	3.112829E-02	1.340667E-02	1.193110E-02
λ =7	6.441512E-02	3.168451E-02	2.784232E-02
λ =8	1.067605E-01	5.907328E-02	5.302311E-02
λ =9	1.545367E-01	9.525949E-02	8.771109E-02
λ =10	1.995743E-01	1.368786E-01	1.270045E-01
λ =11	2.469441E-01	1.787459E-01	1.680433E-01
λ =12	2.898364E-01	2.213412E-01	2.090314E-01

Table 5.5(b) Performance evaluation under 12:2 vs. 13:1 proportion

We have observed that closer video/audio ratio yields more profit under multiple multicast tree routing from above simulations. In the following, we shall investigate when it is worthy of multiple-tree routing. We first define the performance improvement ratio as follows:

Definition 5.2 Performance improvement ratio

$$\frac{original\ \mathrm{FRL} - improved\ \mathrm{FRL}}{original\ \mathrm{FRL}} \qquad (10)$$

In equation (10), the *original* FRL is the FRL yielded by routing on a single multicast tree while the *improved* FRL is that of multiple-tree routing. We will study the performance improvement ratio with respect to traffic load. Two configurations were used for comparison. In our first case, the bandwidth reservation ratio of video/audio streams is 7:0 for single-tree routing and 6:1 for multiple-tree routing. In the second case, bandwidth reservation ratio is 14:0 for single-tree and 13:1 for multiple-tree routing. The results are shown in Figure 5.5(c) which indicates that we may obtain more benefit from multiple-tree routing when the video/audio reservation ratio is smaller.

5.2.5 Tree Constructing Order

The above experiments indicate that multiple trees routing yields better performance than single tree routing. In this section, we investigate the tree constructing order of multiple trees. Two policies are compared: one is *Video First*, the other is *Audio First*. As implied by the name, *Video First* policy constructs video tree first. Corresponding bandwidth requirement of video tree is deducted from available network resource after video tree has successfully created. Then, audio tree routes according to the newly network status and obeys synchronous restriction. By contrast, *Audio First* policy constructs audio tree first, after it has constructed, video stream is routed immediately. Fig 5.6 shows the FRL of different tree constructing order. Simulation results indicate that *Audio First* policy has lower average FRL values than *Video First*.

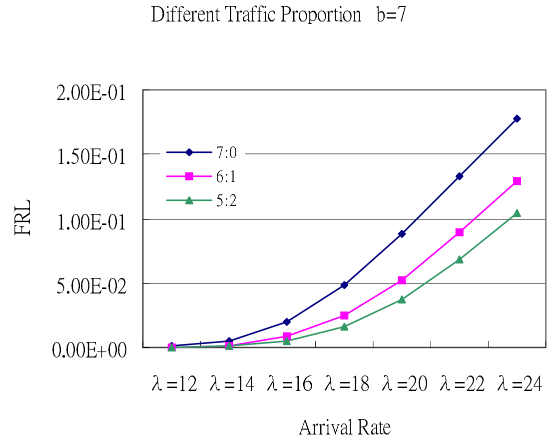

Figure 5.5(a) Efficiency of multiple trees routing(b=7)

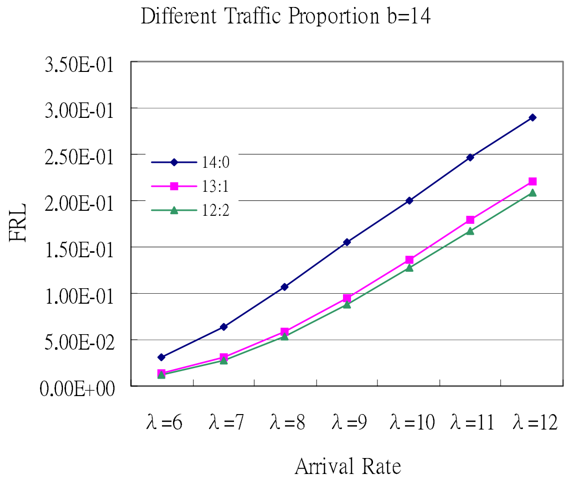

Figure 5.5(b) Efficiency of multiple trees routing(b=14)

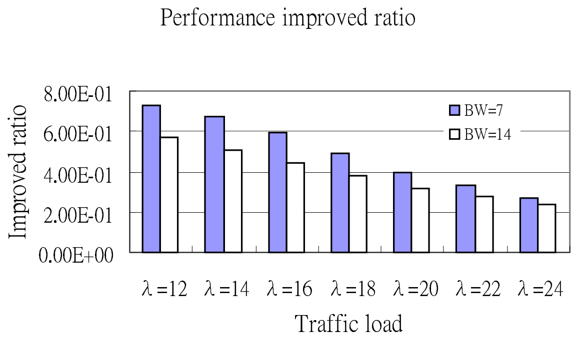

Figure 5.5(c) Performance improvement ratio of multiple tree routing

6. Conclusion and Future Work

Multimedia is the rising star in the networking field. In order to support such continuous media transmitting to a group of receivers, multicast with QoS guaranteed is the essential issue. In this paper, several issues about multicast routing problem were studied.

Average FRL	Video First	Audio First
λ=12	2.254513E-04	1.654110E-04
λ=14	1.809067E-03	5.915542E-04
λ=16	8.314307E-03	1.738982E-03
λ=18	2.460639E-02	4.462088E-03
λ=20	5.274180E-02	1.015796E-02
λ=22	8.898198E-02	1.980970E-02
λ=24	1.293233E-01	3.412119E-02

Table 5.6 RFL of different tree constructing order

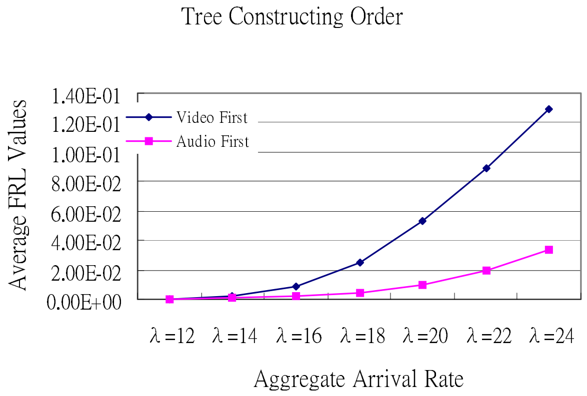

Fig 5.6 FRL of different tree constructing order

DVMRP and CBT use different strategies to meet their different design goal. Within a routing domain, CBT performs very efficiently in terms of the amount of states that routers need to keep. Only routers on the multicast tree for a group keep forwarding state for that group, and none of them need to keep information about any source because only one tree for corresponding group communication. Thus CBT scales much better than Source-based tree protocols, especially for sparse groups communicating. However, CBT never properly solved the problem of how to map a group address to the address of a core [17]. We proposed a *Min_Hop_Core* scheme for CBT core placement. The proposed method just needs the distance vector information for core selecting. Such distance vector information could be obtained from underlying unicast routing table easily.

In addition, media streams dispersing into distinct delivery tree is also an interesting idea we have investigated. Our simulation results indicate that this scheme yields best performance when traffic load is moderate or heavy and the bandwidth reservation ratio of two media streams is small.

As for our future works, more quantitative analysis on the performance among these multicast protocols are under investigation. It includes the message overhead evaluation (both control message and group membership maintaining involved) and dynamic group membership. Besides, due to the low computational complexity of the COL approach, it is appropriate for on-line routing. However, how to set parameters of the COL approach, such as μ and admission threshold ρ, is still lack of systematical approach.

REFERENCES

[1] R. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, (New York: Plenum Press), Pages :85-103, 1972.

[2] Jhyda Lin, Ruay-Shiung Chang “A Comparison of the Internet multicast routing protocols”, *Computer Communications* 22, Pages:144-155, 1999.

[3] T. Pusateri, “Distance vector multicast routing protocol”, Internet draft, <draft-ietf-idmr-dvmrp-v3-04>, February 1997.

[4] A. Ballardie, “Core Based Trees (CBT) Multicast Architecture”, RFC 2189, September 1997.

[5] D. Estrin, D. Farinacci , A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification”, RFC 2362, June, 1998.

[6] V. P. Kompella, J. c. Pasquale, and G. C. Polyzos, “Multicast routing for Multimedia communication”, *IEEE/ACM Transactions on Networking,* vol. 1, no. 3, pp. 286-292, June 1993.

[7] Changdong Liu, Myung J. Lee, Tarek N. Saadawi, “Core-Manager Based Scalable Multicast Routing”, I*EEE Communications, vol.2*, pp. 1202 – 1207, 1998.

[8] Hwa-Chun Lin and Shou-Chuan Lai “A Near Optimal Core Placement Method for the Core Based Tree Multicast Routing Architecture”, *Proceedings of the 14th International Conference on Information Networking,* Hsinchu, Taiwan, R. O. C., Jan., 2000.

[9] Serge Plotkin, “Competitive Routing of Virtual Circuits in ATM networks”, *IEEE Journal on Selected Areas in Communications*, Volume: 13, Issue: 6, pages 1128 –1136.

[10] Andrew S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, Pages: 723, 1996.

[11] Chung-Ming Huang and Hsu-Yang Kung, “A Multicast Multiple-Stream Multimedia Presentation System Over Internet”, *IEEE, Autonomous Decentralized Systems*, Pages: 353 – 360, 1999.

[12] Z. Dziong and L. Mason, “Control of Multi-Service Loss Networks”, Proc. Of the 28th Conference on Decision and Control, (Tampa, Florida), December 1989.

[13] R.-H. Hwang, “Adaptive Multicast Routing in Single Rate Loss Networks”, INFOCOM ’95, Page 571-578, April 1995.

[14] Rainer Gawlick, Anil Kamath, Serge Plotkin, and K.G. Ramakrishnan, “Routing and Admission Control in General Topology Networks”, technical report stan-cs-tr-95-1548, February 1995.

[15] William Aiello, Matthew Andrews, Sandeep Bhatt, and K.R. Krishnan, Lisa Zhang “A Performance Comparison of Competitive On-line Routing and State-Dependent Routing”, GLOBECOM ’97, Nov. 1997.

[16] B. Waxman. “Routing of Multipoint Connections”, IEEE J. on Selected Areas in Commun, vol. 1,no.9, pp. 1617-1622, December 1988.

[17] Mark Handley, ACIRI and Jon Crowcroft, “Internet Multicast Today”, The Internet Protocol Journal, vol. 2, no.4, pp. 2-19, December 1999.