Design of a multicast routing algorithm

著者	根元　義章
journal or publication title	IEEE International Performance, Computing and Communications, 1998. IPCCC '98.
page range	442-447
year	1998
URL	http://hdl.handle.net/10097/46911

doi: 10.1109/PCCC.1998.660136

Design of A Multicast Routing Algorithm

Akihiro FUJII, Takayuki YAMAGATA †, Yoshiaki NEMOTO ‡

Department of Information Technology,
School of Project Design, Miyagi University
Gakuen, Taiwa, Kurokawa, Miyagi, 981-32 JAPAN
TEL:+81(22)377-8369, FAX:+81(22)377-8390
fujii@mail.sp.myu.ac.jp

† NTT Data Communication Inc., JAPAN

‡ Tohoku University, JAPAN

Abstract

Multicast communication can support wide range of applications such as digital video distribution. In multicast communication, routing pathes are established in terms of minimizing the total cost of the multicast tree or minimizing the distances of each connections from source to the destinations. In this paper, we also concern about the number of copies in one switching fabric and propose a new algorithm for multicast routing in which (1) it avoids concentration of copying at a specific exchange node and (2) it ensure the optimal path for longest connections among destinations. Time complexity and space complexity of the algorithm are discussed.

Key Words
Multicast, Routing Algorithm, Switching Network

1 Introduction

High-definition Audio and Visual applications are now available in end-user computer communication environment. Multicast communication technology is highly expected as a key for realizing multi-point data sharing and distributions of data. In the current hardware market, it is gradually common that switching fabric has multicast feature that makes it possible to design large scale campus/enterprise intranet/extranet with advanced applications such as VoD.

Establishing multi-point to multi-point logical connection among participants is one of the main issue in multicasting protocols.[1][2] Mbone ,for example, IP tunneling technique is applied for TCP connections.[4] Its routing table maintenance is based on the tree configuration. When reconfigurations such as adding/resuming hosts or topology changes occure, the maintenace is held by manual operation. If there is more demand for multicasting, the dynamic and automatic establishment of multicast connection will be necessary in order to support applications flexibly.

There have been discussion how to design protocols for establishing efficient multicast connections. [7] [11] And the minimum cost spanning tree is the base of the protocol. From the theoretical point of view, however, this kind of protocol may not have practical solution especially when the network size become very large. Among the problems beeing considered, coping feature to support multicast switching is an important issue. In a branching node, switching fabric should produce copies of cell/packet according to the fan-out of the tree. This means that the node has to process as much copies as branches par cell or packet at the exchange device. Surely they will provide high load to the switches and cause the transmission delay and congestion compared with simple point-to-point communications. Substantial solutions should be taken for minimizing the affect of copy. This is the motivation of the study in this paper.

We consider the number of copies in one switch-

0-7803-4468-5/98 $10.00 © 1998 IEEE

ing fabric when a logical multicast connection is establised. [9] [12] We propose a new algorithm for multicast routing algorithm in which: (1)it avoids concentration of copying at a specific exchange node, (2)it ensure the optimal path for longest connections among destinations.

The paper is organized as the followings, In 2 the algorithm is explained. First, rather informal explanation of the algorithm is given, then the formal description of the algorithm is shown. In 3 computational complexity of the algorithm is discussed. Some concluding remarks are in section 4.

2 The Algorithm

2.1 The outline of the algorithm

The outline of the algorithm, SBT (Suppressing Branch Tree), is explained in this subsection. Then the formal description of the algorithm is shown after. First some definition of network and multicast are given below. There are 3 stages in the algorithm. And the explanation is given in each stage.

2.1.1 Preparation

G: Physical configuration of a network, represented as a graph $G = (V, E)$

S: The source node of Multicasting

D: Set of Destinations $D = \{d_i | 1 \leq k\}$

M: Multicast Set $M = \{S, D\}$

Now the outline of proposed SBT which is based on the SPT (Shortest Path Tree) algorithm is explained with the defined notations. The first step of setting up multicast connection, SPT algorithm is used to select original paths from sources to the destinations. Then the branching nodes are adjusted if there are more than two paths running through a single node. As a results, number of branching in a certain node is limited. This adjustment is controlled according to the priorities of each connections which the SPT forms paths for the each paths.

Stage1: SPT forms paths for each destinations

Stage2: Then Established paths are sorted according to the distance and cost.

Stage3: Adjustment of the branching points

The Fig 1 is an example of configuration of a network. We are going to use the example and explain detail of the algorithm in the following.

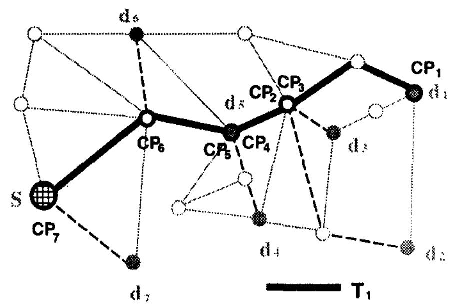

Figure 1: Example of Network Configuration

2.1.2 Stage1 : Sorting paths generated by SPT

This stage consists of two steps of operations:

1. The shortest path is searched for every k destinations from source node.
2. The distance is calculated for each path, then all the destinations are sorted according to the distance. $d_1, d_2, \cdots, d_k, D = \{d_i \mid 1 < i < k\}$

2.1.3 Stage2 : Preparations of Adjustment.

In this stage the preparations of adjustment is done based on the factor for the next stage. There are two factors defined below.

1. T_1, The length of path which is longest among shortest paths, d_i.
2. Copy point CP_i is defined for each destination d_i, that is the node where the path for d_i branches from T_1.

In case there is the destination d_x is on the T_1, $CP_x = d_x$. In fig5 CP_5 is the copy point.

2.1.4 Stage3 : Adjustment

In stage3 according to the sorted order of destinations d_i, and the set of copy point $\{CP_i\}$ final configuration of the multicast path is determined.

1. Among subset of d_i that have not completed, d_i which has highest priority is chosen for adjustment.

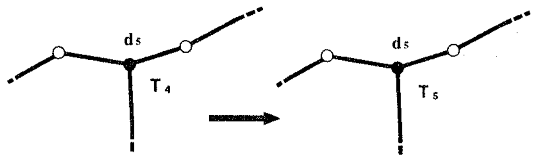

Figure 2: Case 1

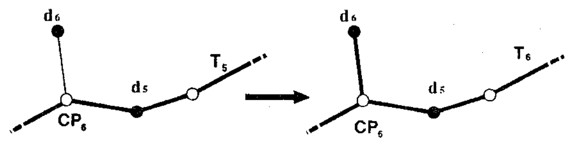

Figure 3: Case 2

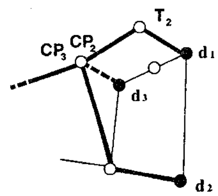

Figure 4: Case 3

2. At this point there are three cases. Decide path for d_i,

Case1: d_i exists on T_{i-1} T_{i-1} becomes T_i without changing configuration.

Case2: copy point of di which is now under estimation is not the same as any of $d_j (j < i)$ The shortest path from d_i to CP_i is merged with T_{i-1} which is already decided. Then the formed tree Ti becomes determined tree T_i of Case 2.

Case3: The set of $\{CP_j\}$ of $\{d_j\}$ that is already decided contains the CP_i. In this situation, there are more tan two paths branching from the node. In order to reduce number of branching, three steps of procedure is taken:

(a) Candidate of copy point of di will be chosen among the set of $CP_j - CP_i$ the shortest path is calculated. from the source to destination d_i under the condition of running through one of CP_j defined above. It will be the temporal candidate of CP_i.

(b) For the candidates obtained in (a), the length of the path from S to d_i is compared, supporsing each became CP_i. The shortest one is chosen as a final candidate for CP.

(c) The candidate found in the previous step is then the candidate is decided as the CP_i and the shortest path from d_i to this CP_i is merged to the Tree T_i. In case the distance exceeds that of S to D_1, the adjustment is not necessary so that the original CP_q is used as CP_1 and shortest path from d_i to CP_i is merged to the tree T_i.

The adjustment process in the Stage 3 should be taken for all the destinations recursively. The resulted T_i is the multicast routing connection that guarantees the shortest distance for the farest destination among the nodes and avoid the concentration of branching at one node.

2.2 The Formal Description of SBT

2.2.1 Preparation

In the following, the formal discription of the proposed algorithm is given. There are several definitions and assumptions for describing the algorithm. First the parameters for SBT algorithm is defined.

SPATH(a, b): The shortest path from node a to b.

path(a): Path from S to a over tree T_i.

DONE: the set of nodes that the route is already decided.

YET: the set of nodes that the route has not yet been decided.

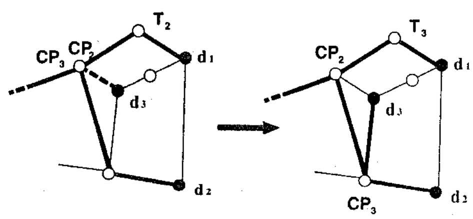

Figure 5: Determined Tree by SBT

2.2.2 Formal Description

In this subsection, the formal description of the algorithm is given using the notation defined in the previous section.

Stage1: Sorting of the destination

1. Forming the shortest path tree T_0 over the network graph G which is associated with the multicast set M.

2. The distance to each destination is measured based on T_0, then the destination is sorted along with the distance, $d_1, d_2 \ldots, Cd_k$
   In the other word, $|path(d_1)| \geq |path(d_2)| \geq \ldots \geq |path(d_k)|$ is satisfied.

2.2.3 Stage2: Preparation for the adjustment

1 Let $T_1 = SPATH(S, d_1)$ and $CP_1 = d_1$
And let $DONE = \{1\}, YET = \{2, 3, \ldots k\}$

2 for $2 < i < k$ **do**

if $d_i \in V_{T_1}$ **then** let d_i itself be CP_i

else Let the node where $SPATH(S, d_i)$ branches from T_1 be CP_i

when CP_i that is $2 \leq i \leq k$ is determined, each distance to d_i is compared with T_1 . The comparison is done along with T_1 . Search continues with the next node in $SPATH(S, d_i), (2 \leq i \leq k)$ Once a node which is not equal to T_1 is found,it becomes CP_i .

Stage3:The adjustment of the path

1 for $2 < i < k$ **do**

if (d_i **exists on** T_{i-1}) **then**

(1) Set $T_i = T_{i-1}$
(2) Set $YET = YET - \{i\}$, $DONE = DONE + \{i\}$

else if (CP_j and CP_i ($j < i$) are not overlapping one another) then

(1) Set $T_i = T_{i-1} + SPATH(CP_i, d_i)$
(2) Set $YET = YET - \{i\}$, $DONE = DONE + \{i\}$

else

(CP_j and CP_i ($j < i$) are overlapping one another)

(1) Let the point over T_{i-1} except $CP_j (j < i)$ be $cp^l_{T_{i-1}} (l = 1 \ldots)$, then the shortest path and its distance from CP_i to $cp^l_{T_{i-1}}$ is measured.

(2) $\exists cp^l_{T_{i-1}}$ *st.* $min(|path(cp^l_{T_{i-1}})| + |SPATH(cp^l_{T_{i-1}}, d_i)|)$

(Suppose $cp^l_{T_{i-1}}$ is CP_i, find the node on T_{i-1} except $CP_j (j < i)$. The candidate CP for CP_i is found as the one which the distance between $path(cp^l_{T_{i-1}})$ and $SPATH(cp^l_{T_{i-1}}, d_i)$ is minimum.)

if $|path(CP)| + |SPATH(CP, d_i)| \leq |path(d_1)|$

then

CP becomes CP_i and $T_i = T_{i-1} + SPATH(CP_i, d_i)$

else

In this cast, the adjustment increase the depth of the path tree, CP_i is not replaced by CP
Set $T_i = T_{i-1} + SPATH(CP_i, d_i)$

(3) Set $YET = YET - \{i\}$, $DONE = DONE + \{i\}$.

2 T_k is the output as the multicast path tree.

3 Computational Compulexity of SBT

3.1 Comparison with SPT

At first it is shown that the depth of the routing tree,$depth(t)$ does not exceed the depth of shortest path tree, $depth(SPT)$.

Preposition Let us define the depth of multicast tree T $depth(T) = max(|Path(di_i)|)$ where $d_i(D)$ is the destination and $path(d_i)$ is the distance from the source to it. It can be claimed that $depth(T) = depth(SPT)$ holds for both created by SBT and SPT by SPT.

Proof In SBT algorithm, SPT is created in Stage1. Then in Stage2,

$$T_1 = SPATH(S, D1)$$

By this, $depth(T_1) = depth(SPT)$ holds. In the adjustment process at the Stage3, if node does not exceeds to $path(d_i)$ of D_i, searching procedure of SBT is terminated, so that $path(d_1 i)(1 < i)$ never exeeds path d_1. This fact result in

$$depth(T) = depth(T_1)$$

.

3.2 Computational Complexity

In the following, computational complexity which is necesary to create routing tree by SBT algorithm is estimated, in terms of both time and spece compulexity. For the discussion below, let us define the following notions.

$n = |V|, m = |E|$ of $G = (V, E)$

$k = |D|$; the size of the set of destinations.

T_{sp} ; Time complexity for establishing spanning tree at the stage1.

S_{sp} ; Space complexity for establishing spanning tree at the stage1.

Stage1 $O(T_{sp} + k \log k)$

1. T_{sp} is needed since the shortest path is searched and establshed from the source S to the destination.
2. For the sorting of k destination, it require $O(k \log k)$ time.

Stage2 $O(kn)$

1. Initial route is determined by cost of $O(1)$ because T_1 is already found it in Stage1.
2. CP_i $(2 \leq i \leq k)$ is determined by comparing each route to d_i with T_1. It requires $O(|V_{T_1}|)$.

Stage3 :$O(k(T_{sp}+n))$ In this Stage3 we assume the worst cast of the procedure. All the remained nodes di $(3 \leq i \leq k)$ other than d_2. Each d_i require the following cost in order to establish path from source.

1. For each $cp^l_{T_{i-1}}$, the shortest path from d_i on T_{i-1} except $CP_j(j < i)$ is determined. It requires time of $O(T_{sp})$
2. In order to determine the candidate for CP_i, the algorithm searches in the set of nodes whose size is $O(|V_{T_{i-1}}|)$. Then it finds the shortest path by comparing $O(|V_{T_{i-1}}| - 1)$ times. Since $|V_{T_{i-1}}| \leq n$, it requires $O(n)$ time.
3. In order to decide final route, the adjustment process has to check whether the route exceeds the distance from S to d_1, it requires time of $O(1)$.

By the above discussion, the determination for each destination requires $O(T_{SP}+n)$ time complexity.

Above all, SBT requires time complexty of;

$$O(k(T_{SP} + n))$$

3.2.1 Space Complexty

Stage1 $O(S_{SP} + k)$

1. In step1, space of S_{sp} is required for determing the shortest path from source S to the each destinations.
2. At the step2, Sroting of k distinations is necesary that require space complexity of $O(k)$.

Stage2 $O(1)$

When $CP_i(2 < i < k)$ is determined, each requires $O(1)$ space, since two nodes are compared at once.

Stage3 $O(S_{SP})$

In this stage, it si considered that how much space complexity the adjustment reauires.

1. S_{SP} space is required in order to determine the shortest path from the d_i to $cp_{T_{i-1}}$ other than $CP_j(j < i)$ on T_{i-1}.
2. When CP_i is chosen among $O(|V_{Ti-1}|)$ candidates, $O(1)$ of space complexity is required to determine the shortst path.

Above all, SBT requires space complexty of;

$$O(S_{SP} + k)$$

3.2.2 Remark on the boundary condition

Next, Let us show that the distance from the source to the farest destinations does not exceed that of the longest path of shortest path tree.

In general, the Dijkstra method is used for searching SPT so that the complexity are the followings:

$$T_{SPT} = O(m \log n), \qquad S_{SPT} = O(m)$$

This means that our algorithm require, $O(km \log n)$ of time complexity, and $O(m)$ of space complexity. That is the proposed algorithm terminates in polynominal time, and can be excuted practical computain cost.

4 Conclusion

In this paper, the branch suppression-type multicast routing path establishment algorithm, SBT is proposed. The algorithm reduce the number of branch in each switching node up to a limit. It ensures the shortest path for the furthest destination as well as common shortest path algorithm.

The analysis of time complexity shows that this algorithm can be executed in a practical time constraint. The algorithm works in the polynomial time so that it complete its task in the practical computation time. The future extension of this algorithm would be distributed algorithm with decentralized control.

References

[1] A. Segall B. Awerbuch "A Reliable Broadcast Protocol" *IEEE Trans. on Communications Vol. COM-31, No.7, July 1983 pp.896-901*

[2] A. S. Tanenbaum "Computer Networks Second Edition" Prentice-Hall,Inc. ,1988

[3] D. Waitzman C. Partridge S. Deering "Distance Vector Multicast Routing Protocol" Request for Comments: 1075, BBN STC, Stanford University, November 1988.

[4] S. Deering "Host Extensions for IP Multicasting" Request for Comments: 1112, August 1989.

[5] T. E. Anderson *et al.* "High-Speed Switch Scheduling for Local-Area Networks" *ACM Transactions on Computer Systems Vol.11, No.4, November 1993, pp.318-352*

[6] Xing Chen and Jeremiah F.Hayes:"Access Control In Multicast Packet Switching," *IEEE/ACM Transactions On Networking, vol. 6, Dec. 1993*

[7] J. Moy "Multicast Extensions to OSPF" Request for Comments: 1584, March 1994.

[8] Xiaola Lin Lionel M.Ni "Multicast Communication in Multicomputer Networks" *IEEE/ACM Transactions on Parallel and Distributed Systems vol.4, no.10, October 1993 pp.1105-1117*

[9] H. Tode, K. Ojiri, M.Yamamoto, H. Okada, and H. Ikeda, "A Routing Method Based on Required Communicaion Qualities in High-Speed Multimedia Network," Proc. 2nd Asia-Pacific Conference on Communications (APCC'95), 8E.5, pp.959-963, June 1995.

[10] Jae W. Byun, Tony T. Lee "The Design and Analysis of an ATM Multicast Switch with Adaptive Traffic Controller", IEEE/ACM Transactions on Networking Vol.2, No.3, June 1994 pp.288-298

[11] The ATM Forum Technical Committee: "Traffic Management Specification Ver. 4" af-tm-0056.000, April 1996

[12] A.Fujii, T. Yamagata, Y. Nemoto" A Multicast Routing Algorithm with Branch Suppression" IEICE, Technical Report, IN96-51, (1996-09)(in Japanese)