Overlay Multicast Update Strategy Based on Perturbation Theory

Ye Shen[1], Jing Feng[1], Weijun Ma[2], Lei Jiang[1] and Min Yin[1]
[1] Institute of Meteorology and Oceanography,
PLA University of Science and Technology, Nanjing, 211101 - China
[e-mail: jfeng@seu.edu.cn]
[2] Institute of Field Engineering
PLA University of Science and Technology, Nanjing, 211101 - China
[e-mail: 13981671@qq.com]
*Corresponding author: Jing Feng

Abstract

The change of any element in the network is possible to cause performance degradation of the multicast network. So it is necessary to optimize the topology path through the multicast update strategy, which directly affects the performance and user experience of the overlay multicast. In view of the above, a new multicast tree update strategy based on perturbation theory Musp (Multicast Update Strategy based on Perturbation theory) is proposed, which reduces the data transmission interruption caused by the multicast tree update and improves user experiences. According to the multicast tree's elements performance and the topology structure, the Musp strategy defines the multicast metric matrix and based on the matrix perturbation theory it also defines the multicast fluctuation factor. Besides it also demonstrates the calculability of the multicast fluctuation factor presents the steps of the Musp algorithm and calculates the complexity. The experimental results show that compared with other update strategies, as for the sensitivity of the multicast fluctuation factor's energized multicast tree to the network disturbance, the maximum delay of the Musp update strategy is minimal in the case of the local degradation of network performance.

Keywords: Overlay multicast, measure matrix, fluctuation factor, perturbation theory, update strategy

1. Introduction

Multicast does the distribution processing with the help of the intermediate nodes. The content has only to be transmitted once on each network link and only at the time of link forking the content will be copied [1], so as to achieve the highly efficient concurrent transmission of the content. Due to technological and economic reasons [2], the IP multicast service has not been yet been fully deployed in the early time and even now it is the same. Compared with IP multicast, the overlay multicast maintains the original simple, unreliable and unicast forwarding model of the Internet. The multicast forwarding function is achieved by the end system, and thus the flexible and convenient deployment can be realized in the network.

The overlay multicast has an extensive coverage of application fields, including real time multimedia applications (such as web conferencing, IPTV, online broadcast and network classroom, etc.), and file distribution applications (such as data distribution transmission, product subscription service, etc.). The above applications have a common feature: need to provide continuous and stable service. Different from the stable and reliable forwarding nodes (routers) of the IP multicast, the overlay multicast data's copying and forwarding are achieved in end nodes (servers). The possible forwarding nodes are not always dedicated forwarding servers with consistent performance and single task, and they can also be common terminal hosts in temporary requisition. Even the multicast end hosts may also undertake the data forwarding task and there may also be other burst applications occupying the bandwidth. Therefore the disturbance exists all the time in the network, including forwarding node load changes, node exiting, node joining, and local link available bandwidth narrowing and so on. Whether these factors will lead to a sharp decline of the multicast network performance, and furthermore how to deal with network disturbance through multicast tree update strategy to ensure the continuity of data distribution and stability of multicast system.

The multicast tree update strategy is an indispensible part of multicast protocols, and its goal is to guarantee the delay of the reconstruction of the multicast tree is minimal at the minimal network cost in the case of network failures and member changing. This paper mainly focuses on the multicast tree update problems caused by network failure resulted from various network element (network elements refer to links and nodes in this paper) failures, traffic congestion problems and other factors. Summarizing the relevant research results in recent years, we can divide the multicast tree update strategy into two categories according to the updating time:

One is the passive update strategy, which means that after the node or link failures in the multicast tree are detected, the multicast tree will be updated. Passive update strategy can be divided into two categories: response type and reservation type. Response type update strategy seeks and establishes a new path to complete the service transmission again. Peercast seeks in the affected grandparent nodes or the root nodes for a new parent node [3]. The affected nodes of SpreadIt algorithm send data interrupt information to the source node , and the parent nodes are redistributed by source nodes [4]. This kind of strategy will lead to that the ancestor nodes will become a bottleneck in the process of transmission and at the same time the recovery time will be longer. Reservation update strategy is the alternate path maintaining redundancy in the multicast distribution path. After the failure occurs, the alternate path is adopted. Redundant path method maintains an alternate parent link for each node [5]. By using traversal path to establish the links among all the nodes in the overlay network, the redundant tree method

constructs two disjoint trees [6][7], one of which is treated as the main tree distributing data from the source nodes. If failure happens on any position in the main tree, the affected nodes can rapidly receive data through the alternate trees. The reservation update strategy can reduce the data terminal time, but the maintenance is of a great cost.

The other one is the active update strategy, which means the cycle detection of the performance of the multicast tree topology construction and the multicast tree will be updated when the update conditions are triggered. The active multicast update strategy can also be divided into two categories: local update and global update. Local update strategy means when links with the following characteristics like heavy congestion [8][9], higher bandwidth [10], more stable parent nodes [11][12] and a closer network distance [13][14] are detected, we will start the local updating of the multicast tree, but the local update cannot guarantee a global optimal multicast. Global update means that the overall performance of the multicast tree is unable to meet the needs of business applications, for example the maximum delay exceeds the threshold [15] and the data retransmission rate is high [16] and so on. At this time, the entire multicast tree needs to be reconstructed. Active update strategy can quickly switch to the alternate path when the fault occurs. The recovery time is shorter than that of the passive update strategy, but active update strategy will consume more network resources.

The characteristics of node and link instability is inherent in the overlay multicast. It is the network random disturbance that the reason to start multicast tree update strategy, but the existing multicast strategy still has the following problems: (1) Lack of the network model characterizing the stability of the multicast tree. (2) The existing researches' evaluation on system stability is based on the measurement results of the distributed tasks, such as data retransmission rate, maximum delay, delay variation and so on, rather than based on multicast topology structure. (3) The existing researches update the multicast tree according to link fracture or better link, but cannot guarantee the multicast tree is global optimal.

In view of the above problems, this paper proposed the overlay multicast update strategy Musp based on the perturbation theory, defines the multicast metric matrix according to the node link performance of the overlay multicast tree and the topology structure, defines the multicast fluctuation factor according to matrix perturbation theory, demonstrates the calculability of the multicast fluctuation factor and studies the multicast tree update opportunity and strategy. The experimental results show that the multicast tree based on perturbation theory can describe multicast stability well. Compared with other update strategies, the maximum delay of the Musp update strategy is minimal under the circumstance of the local degradation of network performance.

2. MULTICAST METRIC MATRIX MODELING

First defines the overlay network model as $G = (V, E)$, $v \in V$ is the node in the overlay network. $e \in E$ is the link between nodes in the overlay network. $n = |V|$ is the number of nodes in the network G. $T(V', E')$ is the multicast tree constituted by the data distribution node V' and the distribution edge E', $V' \subseteq V, E' \subseteq E$. Under the circumstance of not to cause confusion, both v_i and i in this paper refers to node i.

The followings are some relevant definitions needed for the introduction of the multicast metric matrix.

Definition 1 Link cost: the function H from $V \times V$ to $R+$ is called the link cost on $V \times V$. If it meets the following conditions:

1) Non-negativity: $H_{ij} > 0, i \neq j$.

2) Asymmetry: $H_{ij} \neq H_{ji}, i \neq j$.

where $i, j \in V$, H_{ij} is the link cost from node i to node j, indicating the cost of transferring data between nodes. In particular, when i=j, it indicates the data processing ability of the nodes.

Definition 2 Overlay network cost matrix: the link cost of any node $v_i \in V$ to other nodes in the overlay network makes a n dimensional link cost vector $H_i = (H_{i1}, H_{i2}, H_{i3}, \cdots, H_{in})$, the space $H = (H_1, H_2, H_3, \cdots, H_n)^T$ formed by the link cost vectors of all the link cost is called overlay network cost matrix.

Definition 3 Alternate parent node vector: If the node v_i's current parent node is v_j, then vector $P_i' = (v_k, v_l, \cdots, v_m)$ ($\{k, l, \cdots, m\} \subset N \wedge 1 \leq k, l, \cdots, m \leq n \wedge \{i, j\} \bigcap \{k, l, \cdots, m\} = \varnothing$) is called the alternate parent node vector.

The alternate parent node vector P_i' has the following characteristics:

1) The elements of P_i' are sequenced in priority order, and the priority level of the elements are reduced from the former to the latter.

2) $P_i' = \varnothing$ means node v_i has no alternate link, or there is no relevant definition in overlay multicast protocol.

3) P_i' changes in real time according to the overlay network's properties. The alternate parent node with the highest priority is selected as the forwarding node. Nodes exiting and nodes joining in the alternate parent node vector will cause the update of P_i'.

Definition 4 Possible parent node vector: The combination of the current parent node v_j and the alternate parent node P_i' of node v_i is called the possible parent node vector of node v_i, which is indicated as P_i. So $P_i = \{v_j\} \bigcup P_i'$. Obviously, $v_i \notin P_i$, and $|P_i| \leq n-1$. $|P_i|$ is the elements number of the vector P_i

The change rule of the possible parent node P_i is: when it is confirmed that v_j can no longer be the current parent node of node v_i, the current parent node of node v_i will be updated as the node of the highest priority among the alternate parent nodes, and meanwhile we will update the possible parent node vector.

Definition 5 Coefficient vector: The vector $A_i = (A_{i1}, A_{i2}, A_{i3}, \cdots, A_{in})$ is called the coefficient vector of node v_i, and

$$a_{ij} = \begin{cases} |S_j| + 1 & , i = j \\ (\sum_{k \in S_j} a_{kk}) + 1, i \neq j \wedge v_i \in P_j \wedge v_i \notin P_j' \\ \lambda \cdot (2 - e^{1-r}), v_i \in P_j \bigcap P_j' \\ 0, & otherwise \end{cases} \quad (1)$$

is called the coefficient of link e_{ij}. S_j in formula (1) indicates the child nodes set of v_j. $|S_j|$ is the set number of S_j. $\lambda \in R_+$ is the link sensitivity factor, whose value range is $(0,1]$. r is the ordering of node v_i in P_j by priority starting from 1. The analysis of formula (1) according to different conditions is shown in the following:

1) When $i = j$, then A_{ii} is the out degree performance of node v_i.

2) When $i \neq j \wedge v_i \in P_j \wedge v_i \notin P_j'$, then node v_i is node v_j's current parent node. Let $S_j = \varnothing$, then $(\sum_{k \in S_j} A_{kk}) = 0$.

3) When $v_i \in P_i \cap P_i'$, then node v_i is the alternate parent node of node v_j. The link sensitivity factor λ indicates the importance degree of the alternate link e_{ij} as a data distribution link. When the network link performance is not stable, the value of λ is larger.

4) In other cases, $a_{ij} = 0$, which means link v_{ij} is neither a multicast path nor selected as an alternate path.

Intuitively, when $i = j$, then a_{ij} refers to the number of data needs to be copied and forwarded in the process of multicast, that is, the node's out degree. The larger value of the node a_{ij} is, the higher demand for the data processing ability of node v_i is. When $i \neq j$, a_{ij} indicates the importance degree of link e_{ij} in the process of multicast. The larger value of a_{ij} is, it means link e_{ij} is located in the core network of the multicast tree's distribution tree.

From the analysis of a_{ij}'s value range, we can get the following piecewise inequation:

$$\begin{cases} a_{ij} \geq 1, & i = j \\ a_{ij} \geq 1, & i \neq j \wedge v_i \in P_j \wedge v_i \notin P_j' \\ \lambda > a_{ij} > \lambda \cdot e^{(2-n)/n} > \lambda \cdot e^{-1}, & v_i \in P_j \cap P_j' \\ a_{ij} = 0, & otherwise \end{cases} \quad (2)$$

Definition 6 Coefficient matrix: If A_i is the coefficient vector of node v_i, then the matrix $A = (A_1, A_2, A_3, \cdots, A_n)^T$ formed by the coefficient vectors of all the nodes in the overlay multicast is called Coefficient matrix.

Definition 7 Multicast metric matrix: Assuming the cost matrix in the overlay multicast is H, and the coefficient matrix is A, then $D = H * A$ is called multicast metric matrix. The symbol * refers to the Hadamard product of the matrix, that is, the product of the corresponding elements of the two matrices.

3. Analysis of the Multicast Fluctuation Factor

3.1 The Definition of multicast fluctuation factor

Assuming the equation set $Ax = b$, A is the non-singular matrix, b is the constant vector, and x is the solution of the equation set.

Definition 8 Ill conditioned equations [17]: If slight changes of A or b will cause obvious changes in the solution of $Ax = b$, then it is called ill conditioned equations. A is called ill conditioned matrix.

Definition 9 Condition number [17]: Assuming the matrix $A \in R^{n \times n}$, $\| \cdot \|$ is the matrix norm defined on $R^{n \times n}$, then

$$cond(A)_k = \| A \|_k \| A^{-1} \|_k, \ k = 1, 2 \ or \ \infty$$

is A's condition number of norm $\| \cdot \|$.

In general, if matrix A's condition number is large, then it is "ill-conditioned" for the linear equation set to be solved, otherwise it is known as "well-conditioned". Since the matrix norm numbers calculated through different norms are constrained by each other, and all the matrix condition numbers are equivalent, therefore this paper uses the most common spectral condition number in order to easy to calculate, denoted by $cond(A)_2$, namely

$$cond(A)_2 = \| A \|_2 \| A^{-1} \|_2 = \sqrt{\lambda_{\max}(A^T A) / \lambda_{\min}(A^T A)} \tag{3}$$

When A is the real symmetric matrix,

$$cond(A)_2 = \frac{| \lambda_{\max} |}{| \lambda_{\min} |} \tag{4}$$

In formula (3), (4), $\lambda_{\max}$ and $\lambda_{\min}$ are the largest eigenvalue and the smallest eigenvalue of the corresponding matrix. The matrix condition number plays an important role in numerical analysis and the stability of the whole system. It presents the system's sensitivity to disturbance. This paper defines multicast fluctuation factor to measure the stability of the multicast tree according to multicast tree topology structure in the overlay network construction.

Definition 10 Multicast fluctuation factor: Assuming D is the multicast metric matrix constructed by overlay network based on multicast tree T, called

$$k(D) = \begin{cases} Cond(D)_2, & | D | \neq 0 \\ \infty, & | D | = 0 \end{cases} \tag{5}$$

is the multicast fluctuation factor of the multicast tree T.

Multicast fluctuation factor $k(D)$ reflects the stability performance of the multicast tree constructed in the overlay network, and provides the network model characterizing the stability of the multicast tree. The smaller the $k(D)$ is, the more stable structure of the current distribution multicast tree is. And trivial changes of nodes or links in the network will not have great impact on multicast distribution. The larger the $k(D)$ is, the more instable the structure

of the current distribution multicast tree is. Slight fluctuation in the network may lead to the fracture of the multicast distribution tree or even the crash of the multicast network.

The multicast fluctuation factor has the following properties:

$$k(\mu D) = Cond(\mu D) = \| \mu D \|_2 \cdot \| (\mu \cdot D)^{-1} \|_2 = k(D) \tag{6}$$

And $\mu \in R, u \neq 0$. The above mentioned relevant properties of the multicast fluctuation factor will be verified in the simulation experiment in the fifth section.

3.2 The calculability of the multicast fluctuation factor

From the definition 9 we know that, in order to get the solution of matrix A's condition number, it has to meet the condition that matrix A is non-singular, that is, the constructed multicast metric matrix must be reversible. First, we have to prove the non-singularity of the multicast metric matrix under two special circumstances. And then discuss the non-singularity of the multicast metric matrix under normal circumstances.

Theorem 1 If $P_i' = \varnothing$, then the multicast metric matrix D is the non-singular matrix.

Demonstration: Assuming multicast distribution tree is T, we layer the multicast tree's nodes according to T's depth. From the top to the bottom, the layers are from L_0 to L_m (m is the depth of the multicast tree). And then we number the nodes according to the stratification. The stratification and numbers are shown in **Fig. 1**. The fixed number of multicast root node is 1, and the parent node number is smaller than child node number.
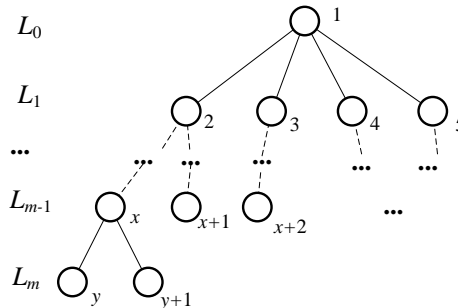

Fig. 1. Schematic of node stratification numbering rule

From $P_i' = \varnothing$ we can see, all the nodes of the multicast tree have no alternate parent nodes. From definition 5, we can see that the element a_{ij}'s value of the coefficient matrix $A = (a_{ij})_{n \times n}$ is

$$a_{ij} = \begin{cases} |S_j| + 1 & , i = j \\ (\sum_{k \in S_j} A_{kk}) + 1, & i \neq j \wedge v_i \in P_j \wedge v_i \notin P_j' \\ 0 & , otherwise \end{cases} \tag{7}$$

According to the depth numbering rule, if node v_i is the parent node of node v_j, then $i < j$. And formula (7) can be simplified as

$$\begin{cases} a_{ij} \geq 1 & , i \leq j \\ a_{ij} = 0 & , otherwise \end{cases}$$

Then we can get that $A = (a_{ij})_{n \times n}$ is sparse upper triangular matrix. Matrix A with n steps has $2n - 1$ non-zero value, and the diagonal element is $\alpha_{ii} \geq 1$. Therefore the multicast metric matrix $D = (d_{ij})_{n \times n} = C * A$ is the sparse upper triangular matrix. From the non-negativity of link cost in definition 1, the diagonal element of D $d_{ii} > 0$. From the properties of the diagonal matrix we can know that $|D| = \prod_{i=1}^{n} d_{ii} \neq 0$, therefore the multicast metric matrix D is non-singular matrix.

QED

Theorem 2: If O_i is the set of node v_i's child nodes (including the child nodes of the alternate links) and if $P_i \cap O_i = \varnothing$ is set up, then the multicast metric matrix D is the non-singular matrix.

Demonstration: the condition $P_i \cap O_i = \varnothing$ can be understood as: node v_i's parent node or alternate parent node cannot be node v_i's child nodes at the same time. Intuitively to say, there is no loop in the multicast topology map composed by the multicast distribution links and the alternate links.

Assuming the multicast coefficient matrix is $A = (a_{ij})_{n \times n}$, then $|A|$ of matrix A can be gained through the determinant definition, then we can get:

$$\begin{aligned} |A| &= \begin{vmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{vmatrix} \\ &= \sum_{j_1 j_2 ... j_n} (-1)^{\tau(j_1 j_2 ... j_n)} a_{1 j_1} a_{2 j_2} ... a_{n j_n} \\ &= a_{11} a_{22} ... a_{nn} + \sum_{j_1 j_2 ... j_n} (-1)^{\tau(j_1 j_2 ... j_n)} a_{1 j_1} a_{2 j_2} ... a_{n j_n} \big|_{j_1 \neq 1 \vee j_2 \neq 2 \vee \cdots \vee j_n \neq n} \end{aligned} \qquad (8)$$

Among them $\tau(j_1 j_2 ... j_n)$ is the inversion number of the arrangement $j_1 j_2 ... j_n$.

In formula (8), $|A|$ is decomposed into the sum of the elements' product on all the diagonal lines and other elements' product. Then each item in the accumulation will be proved by the apagoge in the following:

$$a_{1 j_1} a_{2 j_2} ... a_{n j_n} \big|_{j_1 \neq 1 \vee j_2 \neq 2 \vee \cdots \vee j_n \neq n} = 0 \qquad (9)$$

Assuming an inverse proposition

$$a_{1 j_1} a_{2 j_2} ... a_{n j_n} \big|_{j_1 \neq 1 \vee j_2 \neq 2 \vee \cdots \vee j_n \neq n} \neq 0$$

is set up, from the condition $j_1 \neq 1 \vee j_2 \neq 2 \vee \cdots \vee j_n \neq n$ it can be known that there is an element a_{mk_1} in $a_{1j_1} a_{2j_2} ... a_{nj_n}$ making $m \neq k_1$. For the convenience of describing, i is called the first index of element a_{ij} and j is called the tail index of element a_{ij}. Among elements $a_{1j_1} a_{2j_2} ... a_{nj_n}$, there must be such a product arrangement must be able to be found: a_{mk_1} is the first element, and the first index of each following element equals to the tail index of the element before it, until the tail index of the last element is m. The product of all these elements is not 0, that is $a_{mk_1} a_{k_1k_2} \cdots a_{k_{p-1}k_p} a_{k_pm} \neq 0$. Each element is not 0, which is:

$$a_{mk_1} \neq 0, a_{k_1k_2} \neq 0, \cdots, a_{k_{p-1}k_p} \neq 0, a_{k_pm} \neq 0 \qquad (10)$$

In the multicast topology structure, the meaning of $a_{ij} \neq 0$ can be understood as: node v_i is node v_j's parent node or alternate parent node. From the $a_{mk_1} \neq 0, a_{k_1k_2} \neq 0, \cdots, a_{k_{p-1}k_p} \neq 0$ in formula (10) it can be known that k_p is the child node of m, while from $a_{k_pm} \neq 0$ it can be seen that k_p is m's parent node of alternate parent node. It is inconsistent with the known condition $P_i \bigcap O_i = \varnothing$, so the inversion proposition cannot be set up, but the original proposition (9) can be set up.

Combine the conclusions of formula (2) and (4) and continue to derive formula (8) we can get:

$$|A| = a_{11} a_{22} ... a_{nn} \geq 1$$

According to the $D = H * A$ in definition 7, the multicast metric matrix D and matrix A have the non-zero element with the same position. From $|A| \geq 1$, $|D| \neq 0$ can be derived. D is the non-singular matrix.

QED

Further analysis shows that in the multicast topology structure with any alternate parent node being empty, there must not be loops. **Theorem 1** is the special case of **theorem 2**. When the multicast topology matrix is being constructed, the multicast fluctuation factor can be directly calculated if **theorem 1** and **theorem 2** are satisfied.

Without the loss of generality, if matrix D is the near singular matrix, then the matrix's minimum eigenvalue $\lambda_{\min} \to 0$. From definition 9 it can be known that the matrix's condition number $k(D) \to \infty$, therefore it can be known that the condition number of the near singular matrix is very large. When the matrix D is the singular matrix, $\lambda_{\min} = 0$ cannot calculate the exact value of $k(D)$ in theroy. However in terms of engineering application, $k(D)$ can be regarded as relatively large when the matrix D is judged as singular matrix.

4. MUSP Updating Algorithm Research

4.1 Musp updating strategy

The multicast fluctuation factor threshold interval can be defined as $[Q_1, Q_2)$. When $k(D) < Q_1$, the current multicast tree structure can be considered as well-conditioned. The

slight fluctuation in the network will not cause abrupt changes of multicast performance, and the current overlay multicast tree will continue to complete the multicast distribution task steadily. When $k(D) \in [Q_1, Q_2)$, the current multicast tree's structure is regarded as near "ill-conditioned", and slight network fluctuation will cause abrupt changes of multicast performance. At this time, update the multicast tree during the idle time of the multicast distribution tasks. When $k(D) \geq Q_2$, the current multicast tree structure is "ill-conditioned", so the multicast tree should be updated immediately.

The value of $Q_i (i = 1, 2)$ is closely related to network performance, multicast scale and multicast request. Q_i can be described as follows:

$$Q_i = f_i(P_{network}, S_{multicast}, T_{multicast})(i = 1, 2)$$

where $P_{network}$ indicates the network performance, such as maximun delay between nodes, minimun bandwidth in the network, etc. $S_{multicast}$ is the scale of the multicast. $T_{multicast}$ indicates the multicast request, like retransmission rate, judder, etc. The experimental data is used to calulate Q_i in the simulation experiment.

To better understand, here's an example. If the network performance is poor and the multicast scale is large, the multicast fluctuation factor $k(D)$ must be large. At this time, if we set Q_1 and Q_2 to a low number, the multicast tree structure will be mistakenly judged as "sick", thus it will start to update multicast tree strategy and waste system resources. Under normal circumstances, when the network performance is relatively stable, the multicast scale is relatively small and the multicast task having a higher demand for the stability of performance, smaller Q_1 and Q_2 can be set to create the multicast distribution tree with higher stability and vice versa.

4.2 Musp updating algorithm

Musp updating algorithm is shown in **Fig. 2**. Its basic idea is as follows:

(1) Initialization parameters: Set the interference factor $\rho \in R_+$, testing cycle t and the multicast fluctuation factor threshold interval $[Q_1, Q_2)$. Calculate the multicast fluctuation factor $k(D)$.

(2) If $k(D) < Q_1$, it is not necessary to update the multicast tree.

(3) If $k(D) \in [Q_1, Q_2)$, first we locate the potentially affected nodes. Then go through the elements in the multicast metric matrix $D = (a_{ij})_{n \times n}$ in order. When it comes to element a_{ij}, plant $a_{ij} = a_{ij} - \rho$ into D, and calculate the multicast fluctuation factor $k'(D)$ after the a_{ij} being affected. If $k'(D) < Q_1$, add node into the set of affected nodes $\widehat{E}$. If a traversal is completed and the potentially affected nodes are empty, double the interference factor and continue to go through the elements in the multicast metric matrix until after a traversal, the set of potentially affected nodes is not empty. Then update the parent nodes of the potentially affected nodes into the alternate parent nodes. End the cycle.

(4) If $k(D) \geq Q_2$, then reconstruct the whole multicast tree. End the cycle.

Algorithm: Musp	
Input: D	//Multicast metric matrix

Begin
1: Initialize ρ, t, Q_1, Q_2 //Initialization parameter
2: While(true)
3:   If (k(D) Q1 && k(D)<Q2) //Meet updating conditions
4:     k=1;
5:     While (true)
6:       For each xij in D
7:         xij = xij - $k\rho$;   //Add simulated disturbance factor
8:         If (Cond(D)2 < Q1)
9:           $\widehat{E} = \widehat{E} \cup \{v_j\}$;
10:      EndFor
11:      k++;
12:      If ($|\widehat{E}|$>0)
13:        updateNodeParent($\widehat{E}$);   //Locale update parent node
14:        break
15:      EndIf
16:    EndWhile
17: ElseIf (k(D)$\geq$ Q2) updateTree();//Reconstruct multicast tree
18: EndIf
19: Sleep(t) ;
20: EndWhile
End

Fig. 2. Musp updating strategy core algorithm

4.3 Musp updating algorithm example

In order to facilitate the understanding, this section gives the example of the Musp updating algorithm based on the specific network topology. **Fig. 3**(a) is shown as the initial network, and the nodes for constructing the multicast tree are v_1, v_2, v_3, v_4 . **Fig. 3**(b) shows the constructed multicast path, in which the solid line represents the multicast distribution link and the dotted lines represents the alternate link. The steps of the Musp updating algorithm are divided into construction cost matrix, construction coefficient matrix and the calculation and analysis of the multicast fluctuation factor. The detailed steps are as follows:

(1) Constructing cost matrix

To construct the cost matrix, the cost of nodes and links needs to be calculated firstly. The cost elements of the network element considerd by different overlay multicast protocols are different [8][9]. And their calculation methods and steps are also different. The emphasis point of this paper is to solve the multicast updating opportunity, so it is not necessary to do too much analysis on cost calculation. In order to facilitate the simulation experiment of the algorithm, the following calculation methods are designed:

Let link e_{ij} be the link from the parent node v_i to the child node v_j, v_i's unit data processing delay be t_{node}, and v_i's unit data transmission delay be t_{link}. Define the network element cost from v_i to v_j as

$$H_{ij} = \begin{cases} \alpha t_{node} + \beta t_{link}, & i \neq j \\ \alpha t_{node}, & i = j \end{cases} \tag{11}$$

Among them, α, β are the weight coefficients, and $\alpha > 0 \wedge \beta > 0 \wedge \alpha + \beta = 1$, which can be adjusted according to specific multicast service requirements. In the experiment of this paper, make $\alpha = \beta = 1/2$. t_{node} is the delay of v_i's reading 1K files. t_{send} is the delay of node v_i sending 1K files. Cyclically update each delay value. Intuitively speaking, t_{node} reflects nodes' ability. t_{send} reflects the ability of network links.

Let **Fig. 3**(c) be the cost information of the network element (links and nodes) calculated based on the initial network in **Fig. 3**(a). The value at the node is the value of t_{node}, the value on the link is the value of t_{link}. The constructed cost matrix is

$$H = \begin{Bmatrix} 6 & 300 & 90 & 29 \\ 309 & 15 & 65 & 43 \\ 95.5 & 61.5 & 11.5 & 76.5 \\ 33 & 38 & 75 & 10 \end{Bmatrix}$$
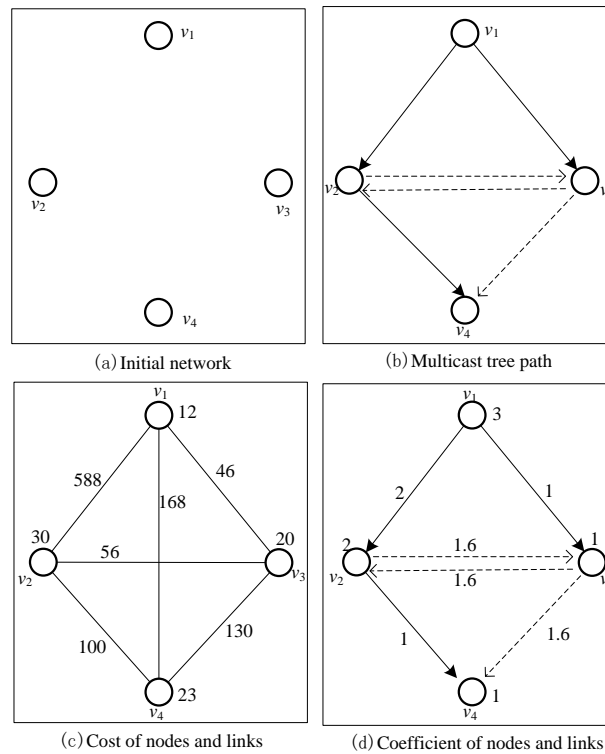

(a) Initial network

(b) Multicast tree path

(c) Cost of nodes and links

(d) Coefficient of nodes and links

Fig. 3. The construction process of multicast metric matrix

(2) Constructing coefficient matrix

According to the multicast path in **Fig. 3**(b) and the coefficient matrix element in formula (1), make the link sensitivity factor $\lambda = 1$, then we can get the nodes and link coefficient information. As shown in **Fig. 3**(d), the value on the link represents link coefficient, and the value on the nodes indicates nodes' coefficient.

According to the information in **Fig. 3**(d), we can get the multicast coefficient matrix as follows:

$$A = \left\{ \begin{matrix} 3 & 2 & 1 & 0 \\ 0 & 2 & 0.78 & 1 \\ 0 & 0.78 & 1 & 0.78 \\ 0 & 0 & 0 & 1 \end{matrix} \right\}$$

(3) Calculation and analysis of the multicast fluctuation factor

According to definition 7, we can get the multicast metric matrix as follows:

$$D = H * A$$

$$= \left\{ \begin{matrix} 6 & 300 & 90 & 29 \\ 309 & 15 & 65 & 43 \\ 95.5 & 61.5 & 11.5 & 76.5 \\ 33 & 38 & 75 & 10 \end{matrix} \right\} * \left\{ \begin{matrix} 3 & 2 & 1 & 0 \\ 0 & 2 & 0.78 & 1 \\ 0 & 0.78 & 1 & 0.78 \\ 0 & 0 & 0 & 1 \end{matrix} \right\}$$

$$= \left\{ \begin{matrix} 18 & 600 & 90 & 0 \\ 0 & 30 & 50.7 & 43 \\ 0 & 47.97 & 11.5 & 59.67 \\ 0 & 0 & 0 & 10 \end{matrix} \right\}$$

The multicast fluctuation factor $k(D) = 2.5486 \times 10^3$ is calculated according to formula (5). From the comparison with the multicast fluctuation factor threshold interval $(Q_1, Q_2]$ (let $Q_1 = 10^3, Q_2 = 10^4$) of the current network scale, we can get $k(D) \in (Q_1, Q_2]$. It means that the current multicast structure is near "ill-conditioned" and instable. Slight network fluctuat will cause instability or even crash of the multicast system. So it is needed to judge which node might cause instability of multicast and replace the parent nodes of the potentially affected nodes.

5. Algorithm Simulation and Experiment

5.1 Algorithm complexity analysis

The calculation of the multicast fluctuation factor's complexity is mainly divided into two parts: First is to measure the performance of all the multicast nodes and the performance of the link between nodes when the cost matrix is obtained. The complexity degree is $O(n^2)$. Second

for the common matrices, the calculation complexity of the matrix condition number is $O(n^3)$ [18]. The multicast metric matrix satisfying theorem 1 is triangular matrix. From the diagonal element, $\lambda_{\max}$ and $\lambda_{\min}$ can be got. And its calculation complexity is $O(n)$. Therefore the complexity of common multicast tree condition number is $O(n^3 + n^2)$, and the complexity of the multicast tree condition number satisfying theorem 1 is $O(n^2 + n)$.

It can be known from the analysis of the complexity of traditional multicast algorithm that, the calculation complexity of SMO is $O(\dfrac{n \log n}{\log \log n})$ [12], the calculation complexity of NICE is $O(\dfrac{n \log n}{\log \log n})$ [15], and the maximum calculation complexity of SOM-HS is $O(nk \log n)$ (in which k is the node out degree)[21]. From this it can be seen that even if the multicast metric matrix is a sparse triangular matrix, the Musp algorithm is more complex than traditional algorithm. That is because Musp takes the specific multicast topology structure into account and is more general. The large sample space of Musp increases computation and complex.

5.2 Simulation experiment scheme

In the simulation experiment, this paper adopts the NS2 simulation tool. First we use the GT-ITM tool to generate 5 transmission topologies, each of which includes a transport domain and a nodes domain. Each topology contains 4 domains and 64 node transport domain. Each node domain produces 50 nodes and it is connected to the node through the router.

In order to make the simulation environment is consistent with the real network; Harvard was adopted as the initial data set [19]. Harvard data set is the set of end-to-end average delay of 1895 nodes collected worldwide through the method of King in 2006, whose delay data is of directivity [20]. This paper selects 220 nodes whose connectivity between nodes is over 48% as the experiment sample. Its end-to-end time delay distribution is shown in **Table 1**. During the simulation process, the delay is treated as the initial cost between nodes. The calculation method of node v_k's processing time delay is

$$H_{kk} = \mu \cdot Min(\underset{i=1}{\overset{n}{Min}}\{a_{ik}\}, \underset{j=1}{\overset{n}{Min}}\{a_{kj}\}) \tag{12}$$

Among them $\mu \in (0,1)$ is the ratio of node processing delay and link cost. Make $\mu = 0.1$ in the simulation. *Min* is the minimum value of the set. During the experimental process, the random fluctuation range of the network performance is 10%.

Table 1. End-to-end delay distribution of the experimental sample

Delay interval(m)	(0,20]	(20,40]	(40,60]	(60,100]	(100,∞]	Disconnected
Ratio	3.5%(1675)	6.4%(3079)	6.7%(3213)	11.9%(5735)	24.1%(11631)	47.4%(22847)

In order to verify the validity of the multicast fluctuation factor, this paper chooses three multicast algorithms to analyze the relationship between multicast fluctuation factor and other three factors: data retransmission times, maximum delay and delay variation. The three kinds of overlay multicast algorithms are NICE [15], SMO [12] and SOM-HS [21] respectively. The

multicast's basic idea of the three algorithms is the following respectively: NICE algorithm constructs multicast tree through the stratification clustering method. SMO algorithm first constructs the core network, and puts the noncore network as the leaf node and the core network as the parent node. SOM-HS algorithm is the previous work of the author, and ensures the out degree of the top node is O(n) on the basis of NICE algorithm. In order to increase the comparability, all the three algorithms use formula (9) to calculate link cost and reserve the path construction strategy of the three algorithms.

In order to verify the validity of the Musp updating strategy, this paper compares the multicast maximum delay under three multicast updating strategies. The three multicast updating strategies are Peercast [3], SpreadIt [4] and Musp respectively. Their basic ideas are the followings: Peercast strategy find a node of minimum cost in the ancestor nodes as the parent node when checking the parent node failure by using passive response updating strategy. SpreadIt is the link between cyclic traversal and neighbor nodes. If there is a link of less cost, then switch this node to the parent node. SpreadIt is an active updating strategy. Musp is the active updating strategy proposed in this paper to determine whether to update multicast tree through the multicast fluctuation factor.

5.3 Analysis of simulation experiment

(1) The relationship between multicast fluctuation factor and data retransmission times.

Select 50 nodes from the experimental sample to construct multicast tree by using the three multicast algorithms respectively. The passive response updating strategy of Peercast is used. The distributed data files' size is about 1M. These files are spitted into small files to be sent and the retransmission times are no more than 100 times. Adjust the link or node delay in each experiment, and make the multicast fluctuation factor changes. Record the multicast fluctuation factor in each experiment and data retransmission times. Gather the averages of the data retransmission times of multicast fluctuation factor in different order of magnitude. As shown in **Fig. 4**.
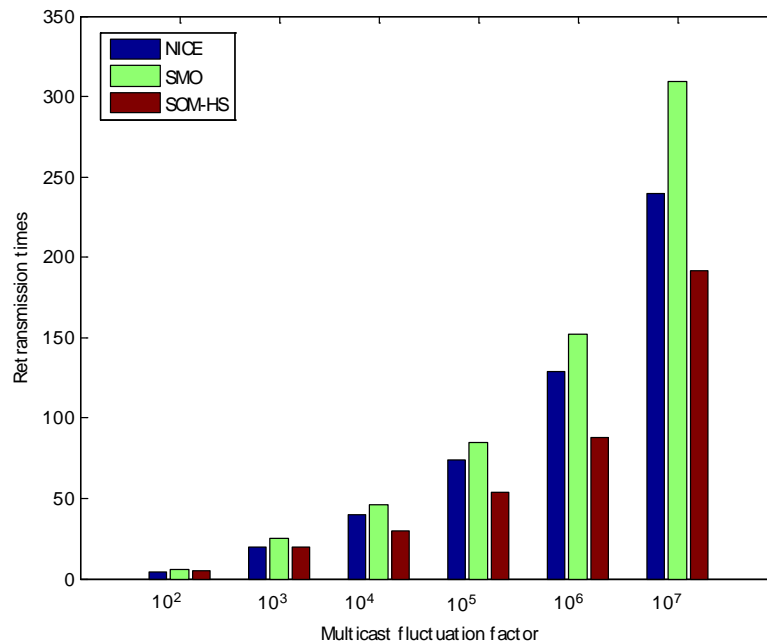

Fig. 4. Data retransmission times changes with multicast fluctuation factor

It can be seen in **Fig. 4** that under the condition of unchanging group scale, data retransmission times increase with the increasing of multicast fluctuation factor. Because the multicast fluctuation factor is large, slight fluctuation of network performance will increase the fluctuation range of network performance, and the instable end-to-end link transmission will cause data packet loss and increase the data retransmission times.

(2) The relationship between multicast fluctuation factor and the maximum delay

The multicast maximum delay can reflect the degree of the bad experience of multicast service. It exists as a parameter index in delay sensitive applications, such as file distribution of delay sensitivity, and online video platform, etc. In order to reflect the relationship between multicast fluctuation factor and the maximum delay, the design scheme is as follows: select 100 multicast nodes and distribute data through three different multicast schemes. Small data file distribution lasts 30 minutes, during which time we set 10 nodes and 10 links with a worsening performance respectively. Collect the node's maximum time delay of the three multicast algorithms under different multicast factors. Each group of experiment is conducted for 10 times, and we take the average of the maximum time delay. The result is shown in **Fig. 5**.
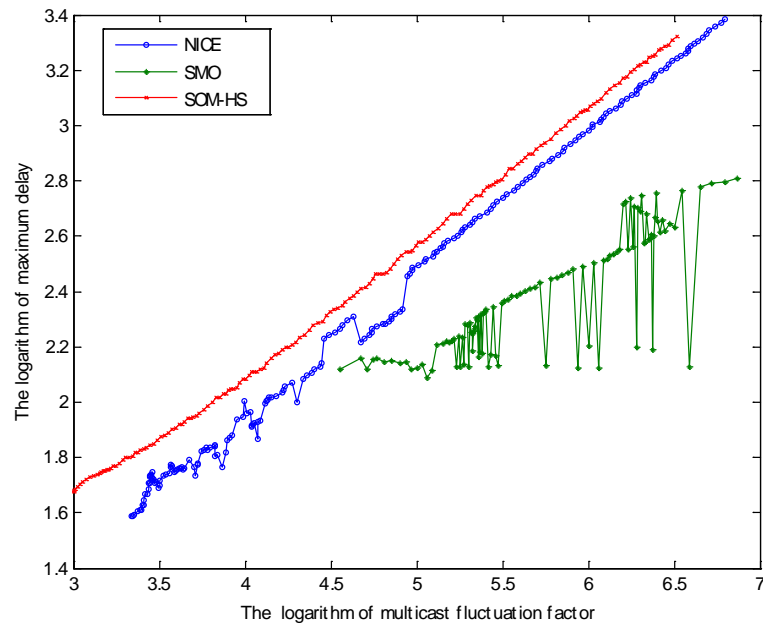

Fig. 5. The relationship between maximum delay and the multicast fluctuation factor

It can be seen from **Fig. 5** that when the multicast scale is invariant, with the increase of multicast fluctuation factor, the maximum delay shows a growing trend, which is due to the degradation of local network performance resulting in the rise of link packet loss and retransmission rate. All these cause a longer time delay of nodes' receiving data, therefore the longer delay is becoming worse. Meanwhile we can see that in the case of the same multicast fluctuation factor, among the three protocols, NICE protocol has the steadiest maximum delay changing trend, but has the longest maximum delay. SOM-HS algorithm has sharp jumping changes of the maximum time delay, but its maximum time delay is relatively shorter. This is

because of the balancing load of the multicast balancing in NICE protocol, whose overall performance is bad but steady. While the core network performance of SOM-HS algorithm is better, but bottleneck is easy to occur in the network.

Multicast fluctuation factor change can reflect the trend of maximum delay. On the contrary, maximum delay change cannot prove multicast fluctuation factor's change. To verify this conclusion, the designed scheme makes some adjustment on the previous experimental scheme: the overall performance of the network becomes worse. The experimental results are shown in **Fig. 6**.
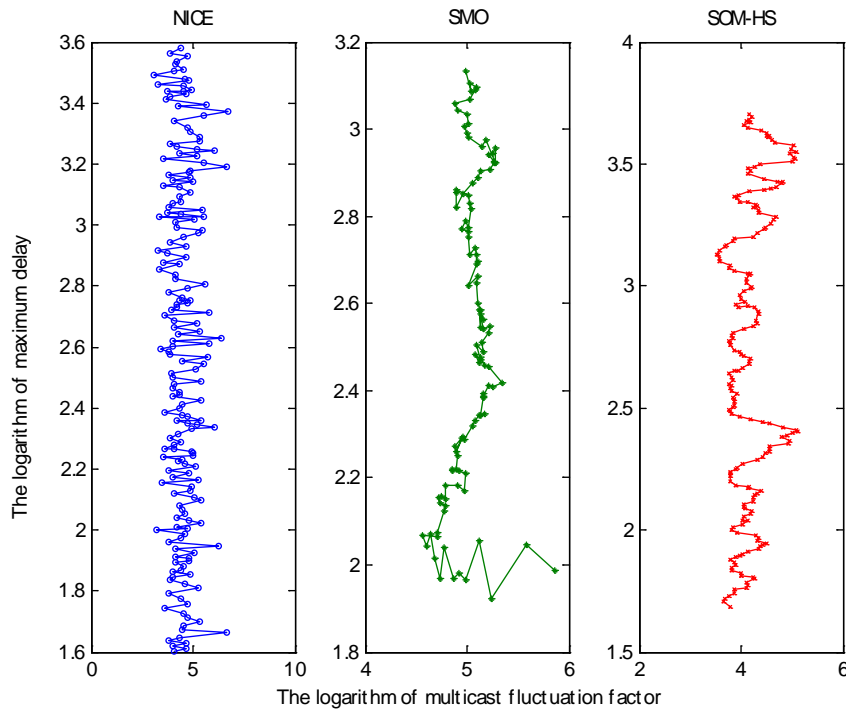
Fig. 6. The relationship between multicast fluctuation factor and maximum delay in the case of the overall degradation of network performance

It can be seen from **Fig. 6** that with the degradation of the overall network performance, the fluctuation factor of the three kinds of multicast are becoming larger within a certain range, and the maximum delay is on the growing trend. It is because the degradation of the overall network performance is equivalent to multicast metric matrix multiplies by coefficient $\mu(u>1)$. It can be known from formula (5) $k(uD)=k(D)$ that multicast fluctuation factor does not change, but due to the randomness of the overall network performance change, the multicast fluctuation factor also changes within a certain range.

(3) The relationship between multicast fluctuation factor and delay variation

Multicast delay variation [22] refers to the delay variation from the source node to each destination node and its application scenarios are such as remote telecommunications conference, distributed database system and distributed gaming environment [23]. Limiting delay variation is to provide fair service for nodes. In order to test the relationship between multicast fluctuation factor and experimental variation, 100 multicast nodes are selected. The

data distribution is conducted through three different multicast schemes. The small data files are distributed for 30 minutes. Collect the average variance value of the three kinds of multicast scheme's node delay under different multicast fluctuation factors. The average variance is smoothly processed by using the moving average filter with a window width of 9 [24]. The results are shown in **Fig. 7**.
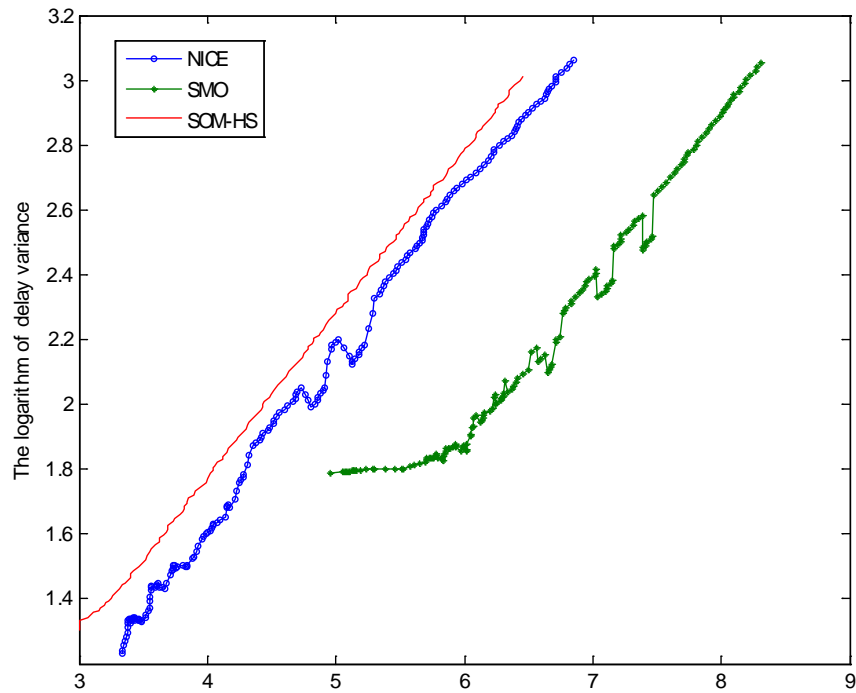

Fig. 7. The relationship between multicast fluctuation factor and delay variance

From **Fig. 7** it can be seen that with the increase of the multicast fluctuation factor, the delay average variances of the three different multicast algorithms are generally on the growing trend, indicating that the more instable of network performance, the more variant the time delay among the nodes is. In the experiment, when the multicast fluctuation factor is in 10^6 order of magnitude, the delay average variance even reaches 10^3 by using SMO-HS algorithm, and the time delay variation is very large. Therefore the multicast fluctuation factor can be used as the index to judge delay variation: the larger multicast fluctuation factor is, the larger variation of the time delay is. From the change of the time delay's average variance of the three different multicast algorithms with the multicast fluctuation factor it can be seen that the time delay's average variance of SMO-HS algorithm is more steady that those of NICE and SMO algorithms. In other words, the multicast fluctuation factor of SMO-HS algorithm has a better corresponding relationship with time delay's average variance. It is mainly because during the multicast tree's construction process, the network load of SMO-HS algorithm will be balanced, and the change of network performance will not cause abrupt changes of multicast time delay's variation.

(4) The situation of Musp updating strategy performance

Select 100 nodes from the test samples. SMO-HS algorithm is adopted in multicast tree construction. Updating strategies are Peercast [3], SpreadIt [4] and Musp proposed in this paper. Cyclically distribute 100 small data files of 1k each. During the distribution process, we randomly choose 20 links or nodes to make their performance continuously decline. Collect the average value of the maximum time delay of many times' file distribution. The results are shown in **Fig. 8**.
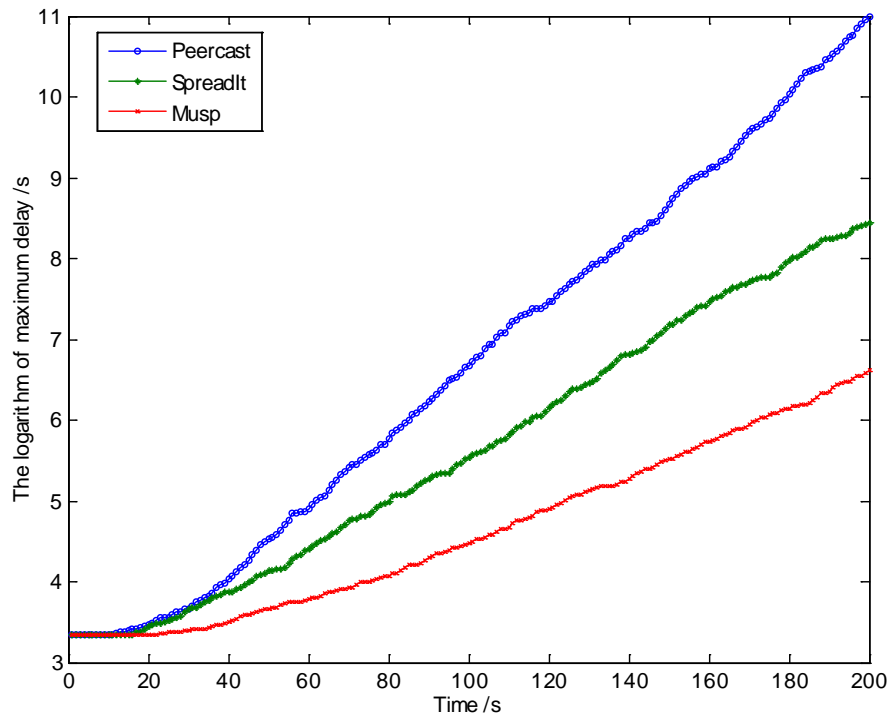
Fig. 8. The comparison between the maximum delay of different updating strategies

It can be seen from **Fig. 8** that with the degradation of the local network performance, the multicast maximum time delay becomes longer. In the early period of the degradation of network performance, the maximum time delay's variation of the three updating strategies changes little, and different updating strategies varies not much from each other at the same time, mainly because little degradation of network performance does not have big impact on the overall maximum delay. With the further degradation of network performance, the multicast maximum delay of Musp updating strategy becomes the shortest, while the multicast maximum delay of Peercast updating strategy becomes the longest. The reason is that the multicast tree is adjusted by Musp updating strategy in time according to the rise of multicast fluctuation factor, making the nodes and links with worsened performance leave the distribution backbone network in advance and reducing the affected child nodes, and at the same time the data that has not been completely transmitted by the original parent node will continue to be transmitted following the original link, which reduces data retransmission caused by switching and also shortens time delay. However, Peercast strategy does not start to seek for parent nodes until the link fracture is found. Its delay is mainly made up of three parts: First, since the detection of link connectivity is periodic, there is time difference from link

fracture happening to link fracture being found. Second, there is time delay of seeking for parent nodes in ancestor nodes. Third is the data retransmission time delay. Therefore, the performance of Peercast strategy is the worst. SpreadIt strategy can always make nodes choose the best parent nodes, but frequent switch of links will increase data retransmission rate, leading to maximum time delay longer than that of Musp strategy.

6. Conclusion and future work

The matrix condition number is initially used to check the stability of the solution of equation set. The larger the condition number is, the less stable the solution is. In this paper, the matrix condition number is applied into the judging of the stability of multicast performance. The properties of the proposed Musp algorithm are as follows: Firstly, the network model is constructed to describe the stability of multicast tree according to the multicast stable factor. Then, the stability of multicast tree can be judged only through the network topology information without the measurement results of the distributed tasks. That is, the Musp algorithm saves network resources when update the network. Finally, the Musp algorithm saves the network resources when updating the network. The experimental results show that, the multicast fluctuation factor can qualitatively evaluate the stability factor multicast network. Compared with the traditional multicast update strategy, the maximum time delay of the Musp update strategy proposed in this paper is the smallest.

The enlargement of network scale will cause an exponential growth of multicast fluctuation factor, and the calculation load then increases. Thus the calculation of multicast fluctuation factor in large scale network will have a higher demand on the performance of the server, and results in calculation delay, causing the failure of timely invoking the multicast updating strategy. Next, it is necessary to research on the quick estimation method suitable for multicast fluctuation factor, quickly estimate whether a multicast tree needs to be updated.

7. Acknowledgements

References

[1] Yang-hua Chu, S. G. Rao, S. Seshan and Hui Zhang, "A case for end system multicast," in *Proc. of IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456-1471, Oct, 2002. Article (CrossRef Link)

[2] Diot C, Levine B N and Lyles B, "Deployment issues for the IP multicast service and architecture," in *Proc. of IEEE network*, vol. 14, no. 1, pp. 78-88, 2000. Article (CrossRef Link)

[3] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," in *Proc. of IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79-89, May, 2007. Article (CrossRef Link)

[4] Deshpande H, Bawa M and Garcia-Molina H, "Streaming live media over a peer-to-peer network," in *Proc. of Technical Report*, 2001. Article (CrossRef Link)

[5] Mengkun Yang and Zongming Fei, "A proactive approach to reconstructing overlay multicast trees," in *Proc. of INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2743-2753, 2004. Article (CrossRef Link)

[6]   Roca V and El-Sayed A, "A host-based multicast (HBM) solution for group communications," in *Proc. of International Conference on Networking. Springer Berlin Heidelberg*, pp. 610-619, 2001. Article (CrossRef Link)

[7]   Li Y, Yu J and Tao D, "Genetic algorithm for spanning tree construction in P2P distributed interactive applications," in *Proc. of Neurocomputing*, vol. 140, pp. 185-192, 2014. Article (CrossRef Link)

[8]   Anardhan Iyengar J, "Concurrent multipath transfer using SCTP multihoming," in *Proc. of Multihomed Communication with SCTP (Stream Control Transmission Protocol)*, pp. 99, 2012. Article (CrossRef Link)

[9]   Wu J, Cheng B and Shang Y, "A novel scheduling approach to concurrent multipath transmission of high definition video in overlay networks," in *Proc. of Journal of Network and Computer Applications*, vol. 44, pp. 17-29, 2014. Article (CrossRef Link)

[10] Kostić D, Rodriguez A and Albrecht J, "Bullet: High bandwidth data dissemination using an overlay mesh," in *Proc. of ACM SIGOPS Operating Systems Review*, vol. 37, no. 85, pp. 282-297, 2003. Article (CrossRef Link)

[11] M. Bishop, S. Rao and K. Sripanidkulchai, "Considering Priority in Overlay Multicast Protocols Under Heterogeneous Environments," in *Proc. of IEEE INFOCOM 2006 - 25TH IEEE International Conference on Computer Communications*, pp. 1-13, 2006. Article (CrossRef Link)

[12] G. Tan and S. A. Jarvis, "Stochastic Analysis and Improvement of the Reliability of DHT-Based Multicast," in *Proc. of IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 2198-2206, 2007. Article (CrossRef Link)

[13] Alekseev S and Schäfer J, "A New Algorithm for Construction of a P2P Multicast Hybrid Overlay Tree Based on Topological Distances," in *Proc. of The Seventh International Conference on Networks & Communications (NeCoM 2015)*, 2016. Article (CrossRef Link)

[14] J. Cao, J. Xie and F. Chen, "DSD-D: A Distributed Algorithm for Constructing High-Stability Application-Layer Multicast Tree," in *Proc. of 2010 Fifth International Conference on Frontier of Computer Science and Technology*, pp. 122-128, 2010. Article (CrossRef Link)

[15] K. Mokhtarian and H. A. Jacobsen, "Minimum-Delay Multicast Algorithms for Mesh Overlays," in *Proc. of IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 973-986, June, 2015. Article (CrossRef Link)

[16] K. Kim, S. Mehrotra and N. Venkatasubramanian, "Efficient and Reliable Application Layer Multicast for Flash Dissemination," in *Proc. of IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2571-2582, Oct, 2014. Article (CrossRef Link)

[17] Stoer J and Bulirsch R, "Introduction to numerical analysis," in *Proc. of Springer Science & Business Media*, 2013. Article (CrossRef Link)

[18] Sankar A, Spielman D A and Teng S H, "Smoothed analysis of the condition numbers and growth factors of matrices," in *Proc. of SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 446-476, 2006. Article (CrossRef Link)

[19] NC Research Group at Harvard, in http://www.eecs.harvard.edu/syrah/nc, 2008. Article (CrossRef Link)

[20] Gummadi K P, Saroiu S and Gribble S D. King, "Estimating latency between arbitrary internet end hosts," in *Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 5-18, 2002. Article (CrossRef Link)

[21] SHEN Y,FENG J and WANG Z F, "Scalable overlay multicast algorithm with high stability," in *Proc. of Journal on Communications*, vol. 37, no. 5, pp. 73-80, 2016. Article (CrossRef Link)

[22] G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," in *Proc. of IEEE INFOCOM 1996 - Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation,* vol. 1, pp. 353-360, 1996. Article (CrossRef Link)

[23] N. Rhodes and S. M. Banik, "Empirical evaluation of designing multicasting network with minimum delay variation," in *Proc. of Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference*, pp. 89-94, 2015. Article (CrossRef Link)

[24] Langan R T, Lerche I and Cutler R T, "Tracing of rays through heterogeneous media: An accurate and efficient procedure," in *Proc. of Geophysics*, vol. 50, no. 9, pp. 1456-1465, 1985. Article (CrossRef Link)
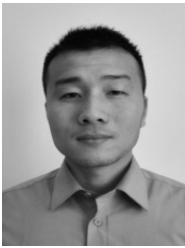
Ye Shen graduated from the PLA University of Science and Technology (PLAUST) with M.A. degree in 2011. Now he is a PhD candidate in Institute of Meteorology and Oceanography, PLAUST, Nanjing, China. His research interests are focus on peer-to-peer network and overlay multicast.
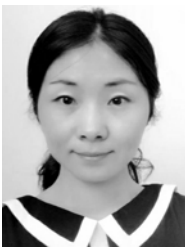
Jing Feng is a professor in the PLAUST, Nanjing, China. She received the PhD degree in 2000 from Southeast University, Nanjing, China. Her research interests are focus on infromation system integration.

Weijun Ma is a lecturer in the PLAUST, Nanjing, China. He received the PhD degree in 2012 from PLAUST, Nanjing, China. Her research interests are focus on infromation system integration.

Lei Jiang graduated from the PLA University of Science and Technology (PLAUST) with M.A. degree in 2013. Now he is a PhD candidate in Institute of Meteorology and Oceanography, PLAUST, Nanjing, China. His research interests are focus on wireless communcation, satellite networking.

Yin Min is a lecturer in the PLAUST, Nanjing, China. She received the PhD degree in 2007 from PLAUST, Nanjing, China. Her research interests are focus on wireless network.