基于受限泛播技术的可伸缩性 QoS 组播路由协议*

黄东军+, 王建新, 陈松乔, 邓清华

(中南大学 信息科学与工程学院,湖南 长沙 410083)

A QoS-Aware and Scalable Multicast Routing Protocol Based on Bounded Flooding Technique

HUANG Dong-Jun+, WANG Jian-Xin, CHEN Song-Qiao, DENG Qing-Hua

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

+ Corresponding author: Phn: +86-731-8836152, E-mail: djhuang@mail.csu.edu.cn, http://www.csu.edu.cn

Received 2003-01-20; Accepted 2003-05-27

Huang DJ, Wang JX, Chen SQ, Deng QH. A QoS-aware and scalable multicast routing protocol based on bounded flooding technique. *Journal of Software*, 2004,15(5):772~782.
http://www.jos.org.cn/1000-9825/15/772.htm

Abstract: The emergence of distributed multimedia applications such as teleconferencing, remote education and distributed interactive simulation etc. prompts the importance of multicast services. The QoS (quality of service) requirements of these applications drive the development of QoS-aware multicast routing. Recently, many QoS-aware multicast protocols have been proposed to meet these requirements. However, few of them can achieve high success ratio, high scalability, and low control message overhead. A new QoS-ware multicast routing protocol is proposed based on bounded flooding technique. It aims at alleviating the memory overhead of routers for setting up multicast trees and improving scalability of the protocol. In this scheme, every node has a two-level forwarding table which contains information about its immediate neighbors (routers reachable in one hop) and its second-degree neighbors (neighbors of an immediate one). By the information about the second-degree neighbors, a router can forward Join_Probe messages intelligently instead of blindly flooding them. This protocol also utilizes multi-path searching to increase the probability of finding feasible branches while connecting a new node to the multicast tree. The details of the data structures in the protocol and the algorithm of building a distribution tree are described. It demonstrates the effectiveness of the proposed protocol through simulation which evaluates its performance in terms of ACAR (average call acceptance ratio) and ACMO (average control message overhead).

Key words: QoS multicast routing; bounded flooding technique; two-level forwarding table; protocol performance evaluation; average call success ratio; average control message overhead

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA112051 (国家高技术研究发展计划(863))

作者简介: **黄东军**(1966－),男,湖南常德人,博士生,副教授,主要研究领域为 QoS 组播路由算法与协议,分布式多媒体系统; **王建新**(1969－),博士,副教授,主要研究领域为 QoS 路由算法与协议,网络性能评价;**陈松乔**(1940－),教授,博士生导师,主要研究领域为软件工程;**邓清华**(1979－),女,硕士生,主要研究领域为计算机网络,软件工程.

摘 要: 随着远程会议、远程教育和交互式仿真等分布式多媒体应用的兴起,组播技术受到网络研究人员的重视.而这些应用的 QoS(quality of service)需求又进一步推动了 QoS 敏感的组播路由协议的发展.在已提出的各种 QoS 组播路由协议中,如何提高呼叫成功率、增强规模伸缩性、降低控制报文开销,仍然是一个有待探索的问题.提出了一个新的 QoS 组播路由协议,其基本思想是使路由器只存储其两层邻居节点的可达性信息以及链路的 QoS 状态信息,以减少路由器存储开销,提高协议的规模伸缩性(scalability).协议采用受限的泛播技术,构造了一个接受节点发起的、采用多路径技术的、分布式路由算法.描述了协议的数据结构、组播树的构造算法,并给出了模拟实验结果.分析表明,基于受限泛播技术的组播路由协议具有节点存储开销小、呼叫接收成功率高等特点.虽然该协议付出了泛播引起的额外带宽开销较大的代价,但是由于协议所需要的控制数据总量不大,加上两层存储结构在一定程度上限制了泛播通信量,因此该方案具有很好的性能.

关键词: QoS 组播路由;受限泛播技术;两层转发表;协议性能评价;平均呼叫成功率;平均控制报文开销

中图法分类号: TP393 **文献标识码**: A

随着网络视频会议、VOD/AOD、Internet-TV、交互式仿真、远程教学等分布式多媒体应用的兴起,支持组播(multicast)功能的网络基础结构研究已引起人们的极大兴趣.组播通过避免数据包重复发送,提高了网络资源利用率.传统组播主要考虑如何通过连接建立组播树.近年来,QoS 组播成为高速网路研究的热点.虽然 IETF(因特网工程组)正在积极推进资源预留协议(如 RSVP[1])、综合服务(IntServ[2,3])、区分服务(DiffServ[4,5])协议的发展,但是由于 IntServ 等不是路由协议,其应用依赖于底层的路由机制,如果路由协议所确定的路由不具有应用要求的 QoS 资源,那么应用的资源预留能力就是一个浪费.因此,找到一棵伸展到群组成员的组播树,使树上的每一个路由器都具有媒体数据流所需要的可用资源,仍然是组播路由协议必须解决的问题.很多研究人员认为,发展具有 QoS 保证的组播路由协议是提高网络资源利用率的关键.

本文提出了一个新的支持 QoS 的组播路由协议.它设想组播路由器只存储其两层邻居节点的可达性及 QoS 信息,以便减少路由器存储开销,从而提高协议的规模伸缩性(scalability).协议采用受限的泛播技术,意在支持多路径搜索以提高呼叫接受成功率,并更好地满足加入者的 QoS 请求.一个想加入群组的成员节点,首先在其两层邻居节点信息库中搜索组播源节点或在树节点(on-tree-node),如果能够成功,立即可以确定一条满足其 QoS 请求的路径,以连接到组播树上;如果不能在两层邻居信息表中确定源节点或在树节点,该新成员就在两层邻居的范围内沿满足其 QoS 需要的路径(path)广播一个加入探测报文(join_probe),当这个加入请求报文到达其第 2 层邻居节点以后,又开始新的一轮两层搜索,直到加入报文被源节点或在树节点截获.

本文第 1 节介绍一些相关工作.第 2 节描述协议的数据结构及组播树的构造算法.第 3 节给出模拟实验结果.分析表明,基于受限泛播技术的组播路由协议具有节点存储开销小、呼叫接收成功率高等特点.当然,它也付出了泛播引起的额外带宽开销较大的代价.但是,由于协议所需要的控制数据总量不大,加上两层存储结构在一定程度上限制了泛播通信量,因此本方案具有很好的性能.

1 相关工作

组播路由的实质是组播树的构造,它是一个随成员加入和离开群组而不断变化的过程.已经提出了很多组播路由协议.根据协议是否支持 QoS 特性,它们可分为 QoS 敏感(QoS-aware)的[6~8]和非 QoS 敏感(QoS-oblivious)的两类[9~11].QoS 组播试图建立一棵从源端到成员节点的满足一定 QoS 请求的优化组播树.非 QoS 组播则采用传统的 IP 尽力而为(best-effort)机制传输组播数据.从组播树的构造过程看,组播协议又可分为单路径路由协议(single path routing,简称 SPR)和多路径路由协议(multi path routing,简称 MPR).前者提供一条单一的路径将新成员连接到组播树上,而后者提供多条路径以供选择,新成员确定其中最能满足其 QoS 要求的一条连接到树上.单路径路由协议多为非 QoS 敏感的,而多路径路由协议则多是 QoS 敏感的.

传统的非 QoS 组播路由协议中最具影响力的是 DVMRP(距离向量组播路由协议)[9]、MOSPF(开放最短路径优先组播路由协议)[12]、CBT(核基树协议)[11]和 PIM(协议独立组播协议,包括密集模式 DM 和稀疏模式

SM)[10].DVMRP 和 PIM-DM(PIM 密集模式)采用反向路径转发技术(reverse path forwarding,简称 RPF),而采用 RPF 技术的路由协议是不能支持面向接受端的 QoS 请求的.其余协议采用了单路径路由技术.例如,MOSPF 协议利用单播路由算法(Dijkstra)计算出一条从源端或在树节点(on-tree-node)到新成员节点的最短路径,新成员用这条路径连接到组播树上.研究证明,这样形成的组播树对于 best-effort 通信量是合适的,但是当需要考虑多媒体数据的 QoS 属性时,最短路径树不一定具有足够的资源,以支持应用请求的服务质量.

为了找到满足 QoS 需要的资源可用组播树,多路径路由协议采取了为新成员提供多条候选路径的策略,新成员选择其中最好的一条路径连接到组播树上.Spanning-Join[8]是这种方案的典型代表.该协议使新成员采用广播方式(反向路径转发技术,RPF[13])发出 join-request 报文以寻找在树节点.当一个在树节点接收到 join-request 报文时,它就返回一个应答报文 ack 给新成员.这个由单播路由协议决定的 ack 传输路径即为一条候选组播路径.在新成员离组播树比较远时,Spanning-Join 协议需要连续广播加入请求报文.总体上看,Spanning-Join 协议使用蛮力搜索,所以会显著增加网络带宽开销.QoSMIC[7]是另一个引起注意的多路径路由协议,其候选路径的搜索由本地搜索和树搜索两个进程构成.本地搜索与 Spanning-Join 相当,差别只在搜索的范围较小.当本地搜索未能找到在树节点时,协议就采用树搜索.树搜索使新成员发送一个 M-JOIN 报文到所谓指定管理节点(designated manager node,简称 DMN),DMN 在组播树中选择一个在树节点子集.这个子集中的节点随后向新成员发送 BID 报文.这些 BID 报文所经历的路径是由单播路由协议决定的,它们构成候选路径.QoSMIC 的规模伸缩性比 Spanning-Join 协议要好,因为它并不需要在整个网络泛播搜索报文,泛播技术只是被用于局部搜索.但是如何选择 DMN 以及由此而引起的通信量增加仍是一个不可忽视的问题.QMRP[6]是又一个新提出的 QoS 组播路由协议,它将单路径路由和多路径路由结合起来.QMRP 先采用 SPR 搜索,如果失败,搜索退回到失败点的前趋节点并转而采用多路径搜索.不难看出,QMRP 协议的效用取决于单路径搜索的状况,如果单路径搜索形成的部分路径具有较高的延迟,则多路径搜索失败的概率会很大.可见,QMRP 协议并不能很好地支持像延迟这样的加性 QoS(additive QoS).

总体上讲,多路径路由算法较之单路径路由算法在支持 QoS、提高连接成功率、缩短加入时间等方面有较大优势,但它的控制报文会有较大的带宽开销.如果能找到有效办法降低这种开销,多路径路由则是更有前途的协议.一些学者基于理论和实验分析,也倾向于发展多路径路由协议[14].

2 基于受限泛播的组播路由协议

为了减少多路径路由的带宽开销同时吸收泛播技术的简单性和健壮性,我们设想路由器存储且只存储其两层邻居节点的可达性和链路 QoS 状态信息.这种存储结构在 D. Ghosh 等人提出的 QoS 单播路由算法[15]中得到了有效应用.它能减少路由器的存储开销,所存储信息的非精确性也远低于全局状态信息,因为一个节点只需要同它的邻居交换信息;同时,利用这些信息转发加入探测控制报文又提高了路径搜索的预见性.在两层转发表的基础上,我们设计了一个以受限泛播技术为核心的组播树构造算法.

2.1 网络模型

本文将互联网看成一个无向赋权图 $G=(V,E)$,V 是节点(路由器)集合,E 是边(链路)集合.G 的一条链路可表示为二元组(v,u),其中 $v,u\in V$.设 M 表示包括源端在内的组播群组,则 M 为 V 的一个子集.在本文中,节点和路由器具有相同的含义.

协议基于如下设定:

① 网络中每个节点具有向其邻居节点广播数据报文的能力;

② 每个节点均了解自身的 QoS 状态,包括每个转发端口的缓存大小、可用带宽、链路的延迟等;

③ 每个节点能够基于一种数据更新策略与邻居节点交换其自身及其邻居节点的 QoS 信息,因此每个节点都能据此建立两层 QoS 状态与可达性信息表,即探测报文转发表;

④ 每一个在树节点均了解从组播源到其自身的 QoS 属性,用 *m_QoS* 表式,它包括源节点到本节点的累计延时 *m_QoS.delay*,以及源节点到本节点路径的带宽 *m_QoS.band*(即分配给该群组的带宽).

2.2 用于多路径搜索的两层QoS转发表

为了实现两层邻居泛播,路由器需要维护一个两层(two-level)转发表,当一台路由器接收到一个探测报文(Join_Probe)时,它就利用这个转发表决定输出链路.

考虑任意节点 v.设 $N^1(v)$表示 v 的邻居节点集,$E^1(v)$表示连接 v 和 $N^1(v)$节点的链路集,$N^2(v)$表示 $N^1(v)$的邻居节点集,$E^2(v)$表示连接 $N^1(v)$和 $N^2(v)$的链路集.

v 的两层转发表包含了 $E^1(v)$和 $E^2(v)$的全部链路 QoS 状态信息;称对应于 $E^1(v)$的表项为第 1 层表项(the-first-level-entry),简记为 R_v^1;称对应于 $E^2(v)$的表项为第 2 层表项(the-second-level-entry),简记为 R_v^2,则 v 的两层转发表如图 1 所示.

Link	Node	Available QoS	Attribute
L_1	V_1	B_1, D_1	*
L_2	V_2	B_2, D_2	*
⋮	⋮	⋮ ⋮	⋮
L_n	v_n	B_n, D_n	*gAddr*
(L_1,l_1)	u_1	Min(B_1,b_1), D_1+d_1	*
⋮	⋮	⋮ ⋮	⋮
(L_i,l_j)	u_j	Min(B_i,b_j), D_i+d_j	*gAddr*
⋮	⋮	⋮ ⋮	⋮
(L_n,l_m)	u_m	Min(B_n,b_m), D_n+d_m	*

Fig.1 Two_Level forwarding table of node v

图 1 节点 v 的两层转发表

在图 1 中,$L_i \in E^1(v)$,$l_j \in E^2(v)$;一个 R_v^2 项表示成一个二元组(L_i,l_j).如果一个节点既属于 $N^1(v)$又属于 $N^2(v)$,则它只出现在 R_v^1 中.V_i,i=1,2,3,...,n,表示第 1 层邻居节点,即 $V_i \in N^1(v)$;u_j,j=1,2,3,...,m,表示第 2 层邻居节点,即 $u_j \in N^2(v)$.B_i 和 D_i 是 $E^1(v)$的可用带宽与延迟;b_j 和 d_j 是 $l_j \in E^2(v)$的可用带宽和延迟.Min(B_i,b_j)表示取 B_i 和 b_j 的最小者;D_i+d_j 是路径(L_i,l_j)的延迟,即链路 L_i 和 l_j 的延迟之和.*gAddr* 是组播地址,当节点 v 或 u 是源节点或在树节点(on-tree-node)时,表的属性域填入该群组地址.

2.3 数据结构

协议将用到以下数据结构:

① Join_Probe(加入探测报文)

Join_Probe 是一个六元组(*G_Addr*,*Req_Band*,*Req_DelayC*,*Call_ID*,*CurNToDestCacD*,*Eligible_paths*),其中 *G_Addr* 是组地址,供节点判断 Join_Probe 要加入哪个群组;*Req_Band* 是发起探测的主机路由器请求的带宽,*Req_DelayC* 是发起探测的主机路由器请求的延迟限制值,二者构成新成员节点的 QoS 请求,它们供节点计算到目前的搜索路径是否满足加入节点的请求;Call_ID 是呼叫标识符(可为发起探测节点的 IP 地址),供本节点区分发起加入的节点,因为一个节点可能接收到多个加入同一群组的不同探测报文,因此有必要区别对待这些 Join_Probe 报文;*CurNToDestCacD* 是接收探测报文的当前节点到发起探测的节点的累计延迟;*Eligible_paths* 是两层范围内 Join_Probe 报文传输的可用链路或路径,可用性含义见第 2.4 节的说明.

② Ack_For_Probe(肯定应答报文)

Ack_For_Probe 是一个三元组(*G_Addr*,*Call_ID*,*m_QoS*),其中 *G_Addr* 是组地址,*Call_ID* 是呼叫标识符,它们的作用与 Join_Probe 报文中的类似;*m_QoS* 是从组播源到当前发送 Ack_For_Probe 报文的节点的路径的 QoS 属性值,包括路径带宽 *m_QoS.band* 和累积延时 *m_QoS.delay*,当一个节点从上游接收到多个 Ack_For_Probe 报文时,它就凭借这些报文的 *m_QoS* 决定哪个最佳.

③ Nack_For_Probe(否定应答报文)

Nack_For_Probe 是一个二元组(*G_Addr*,*Call_ID*),其中 *G_Addr* 是组地址,*Call_ID* 是呼叫标识符.当节点接收到一个从上游传送来的 Nack_For_Probe 时,它就会把这条链路排除在选择范围之外.

④ Prune(剪枝报文)

Prune 是一个二元组(*Call_ID*,*G_Addr*),其中 *Call_ID* 是呼叫标识符,*G_Addr* 是组地址.

一个节点在两种情况下会向其父节点发送 Prune 报文:一种是该节点没有任何成员主机与之相连接时,它通过向其父节点发送 Prune 报文剪除与父节点的连接;另一种是一个节点在接收到一个 Ack_For_Probe 报文时,如果该节点已经有了一条更好的 QoS 的路径,它通过发送 Prune 报文到 Ack_For_Probe 传送来的父节点,告知剪除为 Ack_For_Probe 记录的路径.

2.4 链路和路径的可用性

在两层范围内,如果 v 到 v_i 的链路 l_i:(v,v_i)满足以下条件:

$$B_i \geq Req\_Band \tag{1}$$

$$D_i+CurNToDestCacD \leq Req\_DelayC \tag{2}$$

则称链路 l_i:(v,v_i)是可用的.

如果 v 到 u_j 的路径(L_i,l_j):(v,v_i,u_j)满足以下条件:

$$\mathrm{Min}(B_i,b_j) \geq Req\_Band \tag{3}$$

$$D_i+d_j+CurNToDestCacD \leq R\_DelayC \tag{4}$$

则称路径(L_i,l_j):(v,v_i,u_j)是可用的.

上述两个定义将被用于节点处理加入探测报文的过程中.

2.5 搜索树和组播树

协议运行中将出现两种树:搜索树(search tree)和组播树(multicast tree).

节点维护一个记录搜索树的表,称该表为 *R*.*R* 的一个表项可表示为

$$R\{G\_Addr, In\_Interface, Out\_Interface(O_1,O_2,...,O_j), Call\_ID, Status(O_1,O_2,\ldots,O_j)\},$$

其中 *G_Addr* 是组地址,*In_Interface* 是 Join_Probe 报文输入端口,*Out_Interface*$(O_1,O_2,...,O_j)$是 Join_Probe 报文输出端口.值得注意的是,组播数据包的传输方向与 Join_probe 的输入输出方向正好相反.Call_ID 是呼叫全局惟一标识符.Status 是节点关于 *Call_ID* 呼叫的状态标记,当节点接收到一个 ACK 报文并确定 ACK 传来的路径可用的时候,节点就标记该呼叫 *Call_ID* 在该节点的输入输出端口为 Fixed.

组播树记录在组播转发表中,每个支持组播的节点均维护一个组播转发表,组播树的建立就是由组播转发表最终体现的.我们记组播转发表为 *M*,一个 *M* 表项可表示为

$$M\{G\_Addr, In\_Interface, Out\_Interface(O_1,O_2,...,O_j)\},$$

其中 *G_Addr* 是组地址,*In_Interface* 是组播数据包的输入端口,也就是 *R* 表中的多个输出端口中的最佳者,*Out_Interface*$(O_1,O_2,...,O_j)$是组播数据包的输出端口,即 Join_Probe 报文的输入端口.

2.6 组播树构造算法

(1) Join_Probe 报文处理算法

当节点 *v* 接收到一个加入探测报文 Join_Probe 时,它首先检查该报文的 *Eligible_paths* 集合是否为空,如果非空,表明节点 *v* 只是一个两层转发过程的中间节点,因此它只需简单按 *Eligible_paths* 中所指明的路径来转发 Join_Probe 即可,当然,在转发之前 *v* 还应清空 *Eligible_paths*,同时在其 *R* 表中存储该报文的 *Call_ID*、报文的进入端口(*In_Interface*)、输出端口(*Out_Interface.*O_1,*Out_Interface.*O_2,...)和组地址(*G_Addr*).如果 *Eligible_paths* 为空,表明本节点不是中间节点,而是第 2 层节点.这种情况下,如果 *v* 是在树节点,它首先判断 *m_QoS.delay*+*Join_Probe.CurNToDestCacD*≤*R_DelayC* 是否成立,如果成立,说明该在树节点满足加入请求 QoS,于是返回一个 ACK 报文.如果 *v* 不是在树节点,它便开始检查其两层转发表,如果表中包含有组播源节点或在树节点,*v* 就计算一条最佳可用链路或路径,并将 Join_Probe 中的 *CurNToDestCacD* 修改为当前值与最佳链路或路径延迟之和,

然后将 Join_Probe 转发到组播源节点或在树节点,同时 v 存储该报文的 *Call_ID*、报文的输入端口(*In_Interface*)、输出端口(*Out_Interface*)列表和组地址.如果在两层转发表中不包含任何组播源节点或在树节点,v 就基于其两层转发表计算出一组到第 2 层邻居节点的可用路径 *Eligible_paths*,并将 Join_Probe 中的 *CurNToDestCacD* 作修改后,沿可用路径转发给第 2 层节点.当 v 的第 2 层可用邻居节点接收到 Join_Probe 以后,即开始新一轮探测.重复上述过程,直到 Join_Probe 被组播源节点或在树节点接收为止.节点 v 对加入探测报文 Join_Probe 的处理描述见算法 1.

算法 1. 加入探测报文 Join_Probe 处理算法.

```
INPUT: Join _Probe
1.   Switch (Join_Probe.Eligible_paths)
2.    Case Join_probe.Eligible_paths≠NULL   // 本节点为中间节点
3.        R.in=In_Interface;   // In_Interface 是 Join_probe 的输入端口
4.        R.out=Join_Probe.Eligible_paths;
5.        R.Call_ID=Join_Probe.Call_ID;
6.        R.G_Addr=Join_Probe.G_Addr;
7.        Join_Probe.Eligible_paths="";   // 清空两层内的可用路径表
8.        Forward Join_Probe to R.out;
9.    Case Join_Probe.path=NULL
10.       If (本节点是源节点或在树节点)
11.       if (m_QoS.delay+Join_Probe.CurNToDestCacD≠Join_Probe.Req_DelayC)
12.       M.G_Addr=Join_Probe.G_Addr;
13.       M.out=M.out+In_Interface;
14.       M.G_Addr=Join_Probe.G_Addr;
15.       Send ACK to In_Interface;
16.       Return;
17.   else
18.       Send NACK to In_Interface;
19.       Return;
20.    endif
21. Else
22.     if (本节点的两层转发表中含有源节点或在树节点)
23.         Join_Probe.Eligible_paths=计算可用链路或路径;
24.         if (Join_Probe.Eligible_paths≠NULL)
25.            Forward Join_Probe to  源节点或在树节点;
26.            R.in=In_Interface;
27.            R.out=Join_Probe.Eligible_paths;
28.            R.Call_ID=Join_Probe.Call_ID;
29.            R.G_Addr=Join_Probe.G_Addr;
30.         else
31.            Send NACK to In_Interface;
32.         endif
33.     else
34.         Join_Probe.Eligible_paths=计算到第 2 层节点的全部可用路径;
35.         if (Join_Probe.path≠NULL)
```

```
36.            Forward Join_Probe according to Join _Probe.Eligible_paths;
37.            R.in=In_Interface;
38.            R.out=R.out+Join_Probe.path;
39.            R.Call_ID=Join_Probe.Call_ID;
40.            R.G_Addr=Join_Probe.G_Addr;
41.            Return;
42.          else
43.              Send NACK to In_Interface;
44.              Return;
45.           endif
46.       endif
47. Endif
48. EndSwitch
```

(2) ACK 报文处理算法

节点 v 收到一个 ACK 报文时,它将检查 *ACK.m_QoS* 值,如果在这之前已经有一条路径被确认具有更好的 QoS 属性,v 将向 ACK 到来的端口发送剪枝报文 prune.如果不存在一条已获确认的路径或者 *ACK.m_QoS* 值更好,节点 v 就确认该报文到来的端口为最佳路径端口,并将 Join_Probe 报文通过该节点时所记录在 R 表中的表项标记为 fixed,表示该路径被确认为目前的最佳路径;同时,更新它的组播路由表,即执行 *M.G_Addr*←*R.G_Addr*, *M.In_Interface*←*R.Out_Interface*,*M.Out_Interface*←*R.In_Interface*.如果本节点 v 是发起探测的主机路由器,则加入宣告成功;如果 v 还不是最初的探测发起者,则 v 将向 Join_Probe 进来的端口继续发送 ACK 报文,不过这次发送的 ACK 报文其 *m_QoS* 值加上了 v 的 QoS 属性值.

(3) NACK 报文处理算法

当节点 v 接收到一个 NACK 报文时,它首先要对其 R 数据库作清除工作,即删除 NACK 报文输入端口;如果对应于 *Call_ID* 的 *R.out* 变为空(即没有可用路径时),则 v 将删除 *Call_ID* 的全部表项.如果 v 不是发起探测者,它还必须将 NACK 报文继续发送到它的子节点.

(4) prune 报文处理算法

当节点 v 接收到一个 prune 报文时,它首先在它的组播转发表 M 中删除 prune 报文输入进来的端口号,如果对应于 *Call_ID* 的 *M.out* 为空,v 还将删除 M 表中 *Call_ID* 表项,并向 v 的父节点(即组播数据包输入进来的端口)继续发送剪枝报文 prune.

总的看来,本算法是一个接受节点发起的、采用多路径技术的分布式算法.

3 模拟实验及其分析

我们首先从直观上看一下算法的特点.由于节点只需要存储其两层邻居的可达性信息和 QoS 信息,因此,较之需要全局状态信息支持的组播协议(如 MOSPF,DVMRP,PIM 等),其存储开销大大降低,因而有更好的规模伸缩性(scalability).其次,采用两层泛播技术,一方面简化了探测报文的转发,同时又可避免盲目泛播引起的额外带宽开销.图 2 有助于理解这一点.

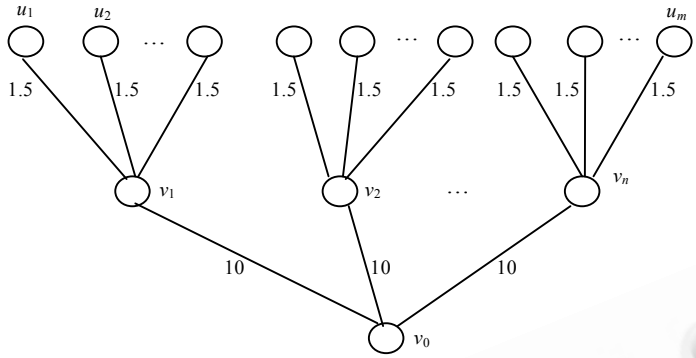

Fig.2 Two_Level forwarding table can void blindly flooding

图 2 两层转发表可以避免盲目泛播

为简便起见,图中仅考虑链路带宽,第 1 层链路带宽为 10.第 2 层链路带宽为 1.5.设 v_0 是发起探测的节点,其带宽请求是 10.当 v_0 准备向其第 2 层邻居节点 u_j, j=1,2,…,m,发送探测报文时,它首先要计算出可用路径,可以看出,尽管 v_0 到第 1 层邻居节点 v_i,i=1,2,…,n 的各链路均满足探测报文的带宽请求,但是第 2 层链路无一满足带宽条件,于是泛播在 v_0 处即停止.显然,如果采用盲目泛播,泛播将在 v_i 处停止,可见,盲目泛播会引起通信量不必要的增加.正是两层转发表使泛播有了更多的“智能性”或者说“预见性”.另外,当树节点处于两层转发表之中时,泛播已无必要,采用的是单播,这也避免了盲目泛播引起的额外开销.

模拟实验采用 Waxman 网络拓扑结构[16].我们随机在一个α*α方阵中选择网络节点.节点 u 和 v 之间存在一条链路的概率 $p(u,v)$为

$$p(u,v)=a*\mathrm{e}^{-d(u,v)/(b*\alpha^2)} \tag{5}$$

其中 $d(u,v)$是节点 u 和 v 之间的几何距离;a 和 b 是两个小于 1 的常数.我们取 a=0.2,b=0.15,这样产生的拓扑图的平均节点度数为 3.5.α=100,得到一个具有 100 个节点的网络拓扑.每次模拟时,随机选择一个源节点和一组新成员,这些新成员随机地以某个 QoS 请求依次加入群组.同时,我们还设网络的每一条链路以 0.5 的概率满足新成员的 QoS 请求.测试的群组大小分别为 5,10,20,30,40 个节点.在每一个群组尺度下,模拟程序运行 100 次.

测试主要集中在两个指标上——平均呼叫接受成功率(average call success ratio,简称 ACSR)和平均控制报文开销(average control message overhead,简称 ACMO).它们分别定义如下:

$$ACSR=\frac{\sum_{i=1}^{N}accepted(Join\_Probe.Call\_ID_i)}{N} \tag{6}$$

$$ACMO=\frac{\sum_{i=1}^{N}messageSent(Jpin\_Probe.Call\_ID_i)}{N} \tag{7}$$

其中 N 是请求总数,$\sum_{i=1}^{N}accepted(Join\_Probe.Call\_ID_i)$ 是被接受的路由请求总数;$\sum_{i=1}^{N}messageSent(Join\_Probe.Call\_ID_i)$ 是在一个群组尺度下为建立组播树而消耗的控制报文总数(包括 Join_Probe,Ack_For_Probe,Nack_For_Probe,prune).值得注意的是,如果一个报文通过了 m 跳(m hops)路径,该报文将记为 m 个.

测试选择了最短路径组播协议(MOSPF),QMRP,Spanning-Join 作为参照.我们的协议简记为 SMRP.

图 3 显示了呼叫成功率的测试结果.

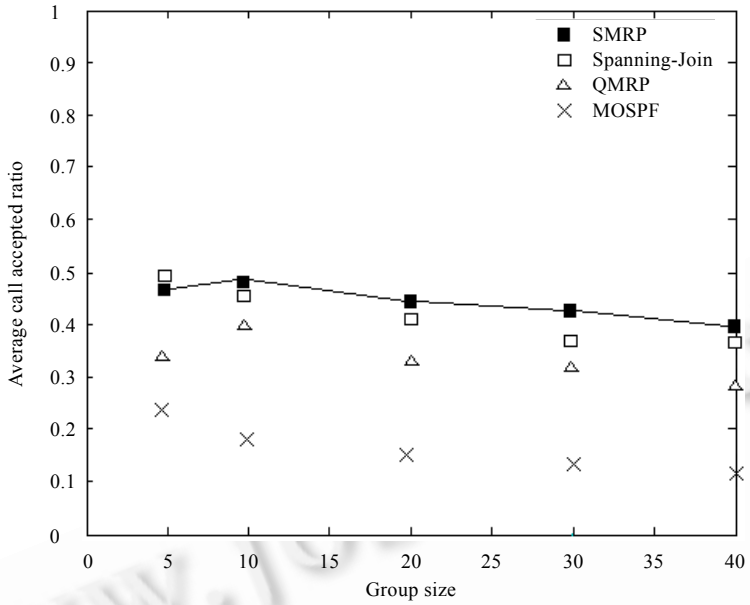

Fig.3 Comparison of average call success ratio

图 3 呼叫成功率比较

可以看出,SMRP 和 Spanning-Join 协议具有较高的呼叫接受成功率,在 50%附近(我们在前面曾假设链路以 50%的概率满足新成员的 QoS 请求),这是因为它们都采用了多路经搜索的策略.MOSPF 的成功率较低,原因在于它并不以 QoS 状态信息作为路由决策的基础.QMRP 协议的成功率介于其间,因为 QMRP 的效用取决于单路径搜索状况,如果单路径搜索形成的部分路径的 QoS 属性不太好,则多路径搜索失败的概率会很大.

图 4 显示了平均报文开销测试结果.

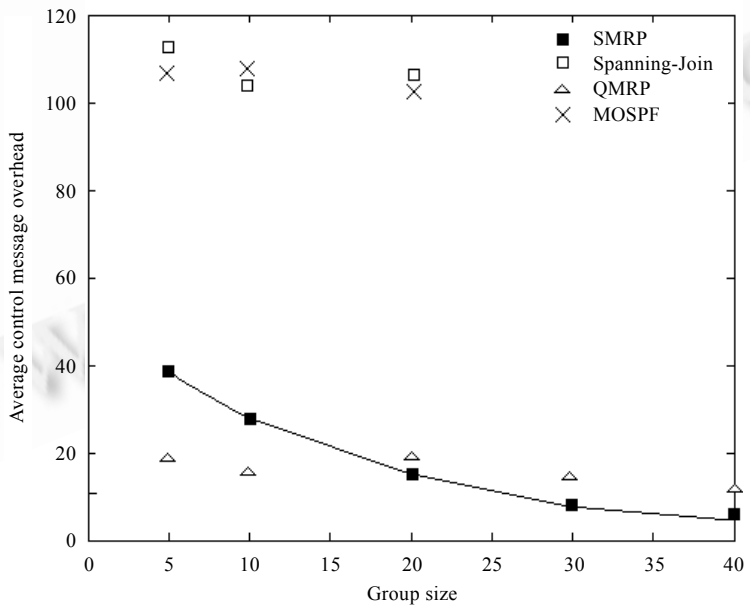

Fig.4 Comparison of average control message overhead

图 4 平均报文开销比较

Spanning_Join 协议的控制报文最多,因为它采用的 RPF 技术是未受限的,仍然属于蛮力搜索的范畴.可以证

明,即便 Spanning_Join 采用扩展 RPF[13]技术,一个新成员节点所产生的 Join 报文也等于网络节点总数减 1,而整个加入过程消耗的控制报文数还要加上具有随机性的 Ack 和 Confirm 报文,所以 Spanning_Join 协议对网络的冲击比较大.MOSPF 协议的平均控制报文开销也非常高.虽然它采用了集中算法(centralized algorithm)计算最短路径树,组播树建立过程本身没有太多的控制报文(剪枝过程需要控制报文),但是在计算组播路径之前,指定路由器(与成员主机直接相联、并负责在网络广播成员关系链路状态广告 group-membership-LSA 的路由器)广播 group-membership-LSA 已经传输了大量报文,在采用扩展的 RPF 技术条件下,一个 MOSPF 指定路由器所传输的报文数等于网络节点总数减 1,所以 MOSPF 的控制报文开销也很高.但是,我们提出的协议 SMRP 协议以及 QMRP 协议在报文开销上比较接近,明显低于 Spanning_Join 和 MOSPF 协议,这是因为 SMRP 采用两层转发表后极大地减少了不必要的报文传输.值得注意的是,SMRP 协议的规模伸缩性比较好,从图中可以看出,随着群组规模(group size)的增长,其控制报文的开销迅速减少.这是因为当群组成员增多时,其两层转发表所包含的在树节点会越来越多.

4 结束语

本文提出了一种新的支持 QoS 特性的分布式多路径组播路由协议.其基本思想是使路由器只存储其两层邻居节点的可达性信息以及链路的 QoS 状态信息.同时,针对 QoS 组播的需要,设计了一套数据结构和组播树的构造算法.在我们的算法中,QoS 路由探测由主机路由器发起,通过探测寻找组播源节点或在树节点来确定一条满足接受端 QoS 请求的路径.本方案是面向接收者(receiver-oriented)的分布式 QoS 组播路由算法,因此较之面向源端的 QoS 路由,更适合于异构网络环境.发起探测的节点(即主机路由器)只需知道组地址而无须事先知道源地址或在树节点地址,因此比需要会话协议(SDP)目录服务器的协议更加健壮.我们的协议是多路径路由协议,它提供多条满足 QoS 的路径供成员节点选择.多路径路由提高了呼叫接收成功率.此外,协议不依赖于任何单播路由协议传播成员关系信息,大大减少了路由器存储开销,提高了规模伸缩性.

本文的后续工作主要是研究采用有向泛播(directed flooding)[8]技术进一步降低控制报文开销.在有向泛播中,Join_Probe 报文新包含一个单播地址和路由距离(routing metric),用于限制报文的传输方向.该单播地址可以为群组源地址或一个熟知根(root)地址,路由距离用来表示当前节点离单播地址的远近程度.在 Join_Probe 报文传输过程中,接收它的节点在转发之前要比较报文所携带的路由距离与该节点到单播地址的距离,如果前者大于后者,节点就用它到单播地址的距离代替报文所携带的路由距离,并继续转发报文,否则就不再转发,因此可以限制报文朝背离组播树的方向传输.在进一步降低路由器存储开销方面,可以考虑对节点信息进行聚集.此外,我们还计划将两层转发表扩充到 $n(n>2)$层,并评估不同 n 值对协议性能的影响.

References:

[1] Yadav S, Yavatkar R, Pabbati R, Ford P, Moore T, Herzog S, Hess R. Identity representation for RSVP. RFC 3182, 2001.

[2] Wroclawski J. Specification of the controlled-load network element service. RFC 2211, 1997.

[3] Shenker S, Partridge C, Guerin R. Specification of guaranteed quality of service. RFC 2212, 1997.

[4] Clark D, Fang W. Explicit allocation of best effort packet delivery service. IEEE/ACM Trans. on Networking, 1998,16(4):362~373.

[5] Nichols K, Jacobson V, Zhang L. A two-bit differentiated service architecture for the Internet. RFC 2638, 1999.

[6] Chen SG, Nahrstedt K, Shavitt Y. A QoS-aware multicast routing protocol. IEEE Journal on Selected Areas in Communication, 2000,18(12):2580~2592.

[7] Yan SQ, Faloutsos M, Banerjea A. QoS-aware multicast routing for the Internet: The design and evalution of QoSMIC. IEEE/ACM Trans. on Networking, 2002,10(1):54~66.

[8] Carlberg K, Crowcroft J. Building shared trees using a one-to-many joining mechanism. Computer Communication Review, 1997,27(6):5~11.

[9] Waitzman D, Partridge C, Deering S. Distance vector multicast protocol. RFC 1075, 1988.

[10] Deering S, Estrin D, Farinacci D, Jacobson V, Liu CG, Wei LM. The PIM architecture for wide-area multicast routing. IEEE/ACM Trans. on Networking,1996,4(2):153~162.

[11] Ballardie A. Core based trees (CBT) multicast routing architecture. RFC 2201, 1997.

[12] Moy J. Multicast extension to OSPF. Internet Draft, Network Working Group, 1992.

[13] Dalal YK, Metcalfe RM. Reverse path forwarding of broadcast packets. Communications of the ACM, 1978,21(12): 1040~1048.

[14] Cidon I, Rom R, Shavitt Y. Analysis of multi-path routing. IEEE/ACM Trans. on Networking, 2000,6(11):885~896.

[15] Ghosh D, Sarangan V, Acharya R. Quality-of-Service routing in IP networks. IEEE Trans. on Multimedia, 2001,3(2):200~208.

[16] Medina A, Matta I, Byers J. On the origin of power laws in Internet topologies. ACM SIGCOMM Computer Communication Review, 2000,30(2):18~28.

2004 年全国软件与应用学术会议(NASAC 2004)

征 文 通 知

2004 年全国软件与应用学术会议（NASAC 2004）由中国计算机学会软件工程专业委员会主办，将于 2004 年 10 月 15 日~17 日在北京召开。届时将进行软件工程等方面的技术与应用交流，会议将出版正式论文集，并将优秀论文推荐到核心学术刊物（EI 检索源）发表。欢迎大家踊跃投稿。

与此同时，为进一步增进学术界与企业界的交流与互动，届时将举办 2004 年中国软件技术与产业互动、计算机软件与应用教育等专题研讨会，热忱欢迎学术界、教育界和企业界的专业人士共同参与承办，共同就软件技术成果的转化与应用、软件产业发展与挑战、软件标准制定与推广应用、软件人才培养与教材出版等热门话题进行深入探讨和交流。在研讨会期间，参会单位可以举办成果和产品的展示与介绍活动。欢迎企业和研发机构踊跃报名。

一．征文范围（包括但不限于）

需求工程	质量度量与质量管理	软件语言
基于构件的软件开发	软件测试、检验与验证	软件标准与规范
面向对象技术	软件再工程	软件工程实践
软件体系结构、设计模式与软件复用	工具与环境	软件工程教育应用软件
软件过程管理与改进	操作系统与中间件	应用软件

二．论文要求

1．论文未曾在其他杂志、会议上发表或录用；

2．论文长度：每篇限定在 6 页（A4）以内；

3．请以 PDF 或者 PS 格式提交论文。有关文章的版心、字号、题目、各级标题、格式及参考文献格式与《软件学报》相同。

三．重要日期

1．文稿截止日期：2004 年 6 月 20 日

2．论文录用通知日期：2004 年 7 月 31 日

3．专题研讨会报名截止日期：2004 年 9 月 10 日

四．联系方式

联系单位：（100083）北京航空航天大学 软件工程研究所

联系人：刘超 金梅 刘平

E-mail：nasac2004@sei.buaa.edu.cn

网址：http://sei.buaa.edu.cn/nasac2004/