文章编号:1001 - 9081(2005)10 - 2255 - 03

应用层组播中间件技术研究

冯睿江,窦文华
(国防科技大学 计算机学院,湖南 长沙 410073)
(rjfeng@163.com)

摘 要:目前的研究成果表明,以中间件的形式为组播应用提供应用层组播的基础功能是完全可行的,关键在于明确这些基础功能并确定出一种实用的中间件框架。在分析研究典型的应用层组播系统、协议和中间件的基础上,总结出了应用层组播的基础功能,并提出了应用层组播中间件的一种基本架构。

关键词:应用层组播;中间件
中图分类号:TP393.04 **文献标识码**:A

Research on application layer multicast middleware technology

FENG Rui-jiang, DOU Wen-hua
(*School of Computer, National University of Defense Technology, Changsha Hunan* 410003, *China*)

Abstract: The actual researches have shown the feasibility to provide the fundamental functions of ALM to multicast application as the form of middleware. The key is describing these functions and defining one practical framework for middleware. Based on the analysis on the classical ALM systems, protocols and middlewares, this paper summarized the fundamental functions of ALM, and further presented a basic architecture of ALM middleware.

Key words: ALM(Application Layer Multicast); middleware

0 引言

IP层组播和应用层组播(Application Layer Multicast, ALM)是实现组播的两种机制。应用层组播将端主机(端系统)组织成覆盖网络,并构造组播数据分发树,通过端主机在应用层以单播方式来相互复制并向接收者转发报文。相比IP层组播,应用层组播并不需要改变下层网络基础设施,可以直接利用现有的网络设备和传输技术在应用层达到组播的目的。因为应用层组播不可避免地要在某些链路(也包括发送方链路)重复传输相同的报文,其效率要低于IP层组播,所产生的网络传输负载也高于IP层组播。但是,在应用层实施组播的可靠传输、QoS、拥塞控制、安全性显然要比在IP层现实得多。

应用层组播应用广泛,尤其适用于需要大规模分发数据的应用,比如视频/音频点播、网络会议、信息发布、分布式交互仿真、网络游戏、远程教学等等。从目前实际的研究和应用来看,由于不同领域对组播的需求不尽相同,大多数的应用层组播系统主要还是面向领域的。而在类似的领域或业务模型中,应用层组播的公有特性和基础功能应该是可以共用的,而且可以将这些公有特性和基础功能实现为可重用的中间件,以中间件的形式为多种应用提供组播功能。

目前,国内对应用层组播中间件的研究和实现还很少,本文在对比研究各种典型的应用层组播系统、协议和中间件的基础上,概括出了应用层组播中间件的基础功能和一种基本架构。

1 应用层组播系统和协议的分类

根据控制拓扑和数据拓扑的构造顺序,可将应用层组播技术分为三类[1]:网优先(mesh first)组播,如End System Multicast[2]、Scattercast[3];树优先(tree first)组播,如ALMI[4]、Yoid[5]、Host Multicast[6]、Overcast[7]、HostCast[8];隐式(implicit)组播,如Bayeux[9]、Scribe[10]、NICE[11]。网优先组播易于支持多源组播应用;树优先组播利于实现有高带宽需求的数据传输,但不适合对延迟敏感的实时应用;隐式组播具有良好的可扩展性,可以支持大规模的组播应用,但大多数此类协议的设计比较复杂。

此外,按照组播结点的选取原则还可将应用层组播系统划分为三种结构:基于端主机的结构、基于代理的结构,以及基于端主机和基于代理相结合的结构。

在基于端主机的结构中,组播结点是端主机,覆盖网络和组播树的构造通过端主机间的相互协作来进行。采用此类结构的应用层组播系统有ALMI、Overcast、Hostcast、Bayeux、Scribe、NICE等。

基于代理的结构是集中式和分布式的结合。属于此类结构的应用层组播系统首先策略性地部署或选择一些结点作为组播代理,并只基于这些组播代理来构造骨干覆盖网络和组播树;然后,其他组播成员连接到就近的组播代理,通过组播代理发送和接收数据。基于代理的应用层组播系统可以支持大规模的组播应用,尤其是广域网上的组播应用。如果为支持IP层组播的网络设置一个代理网关,则此类组播系统还可以桥接应用层组播和IP层组播。这类系统的典型代表有Scattercast和Amcast[12]。

基于端主机和基于代理相结合的应用层组播系统综合了上述两种结构,它的组播结点即可以是端主机,也可以是组播代理,具有良好的灵活性。End System Multicast、YOID、Host

收稿日期:2005 - 04 - 13;**修订日期**:2005 - 07 - 05

作者简介:冯睿江(1975 -),男,云南昆明人,工程师,硕士研究生,主要研究方向:应用层组播; 窦文华(1946 -),男,山西人,教授,博士生导师,主要研究方向:计算机网络.

Multicast 属于此类结构。

2 典型应用层组播中间件的比较研究

2.1 ALMI

ALMI 是美国华盛顿大学 St. Louis 分校计算机系 2001 年发布的研究成果，是最早开发的应用层组播中间件之一。ALMI 实现了应用层组播的基础服务功能，包括组播树构造、成员管理、组播数据分发和自组织组网，支持在小规模的只有少量成员（几十个）的组之间进行可靠通信。

ALMI 基本上可以划分为两个模块：一个是会话控制器模块，属于控制平面，功能包括会话管理、组播树构造和自组织组网；另一个是成员管理模块，属于数据平面，功能包括成员操作和组播数据分发。ALMI 的会话控制器只属于控制平面，并不影响会话成员间的数据分发，即使会话控制器失效，只要失效前所形成的成员关系不发生变化，成员间仍然能够进行通信。

ALMI 使用集中式控制方法来维护组播树的一致性和有效性。ALMI 的一个会话包括一个会话控制器和多个会话成员，会话控制器和会话成员间的通信依靠 ALMI 的控制协议进行。会话控制器集中处理成员注册，周期性地计算和维护组播树，并将结果以（parent，children）列表的形式通知所有成员。ALMI 的组播树是带度约束的双向共享树，树中的成员之间具有父子关系。一旦构造好组播树，会话成员就可以发送或接收组播数据，同时也负责向邻居结点转发组播数据。

ALMI 的原型是一个使用 Java 开发的 Java 包（package）。基于 ALMI，该研究小组开发了 PGP Key Server Synchronization 系统。

2.2 RelayCast

RelayCast[13] 是日本东京大学在 2002 年开发的应用层组播中间件，开发平台和编程语言分别是 Linux 和 C ++。RelayCast 集成了应用层组播的一些主要功能，可以满足包括流媒体业务在内的多种应用的基本需求。RelayCast 的研究人员认为覆盖网络构造和组播路由是应用层组播的基础功能。

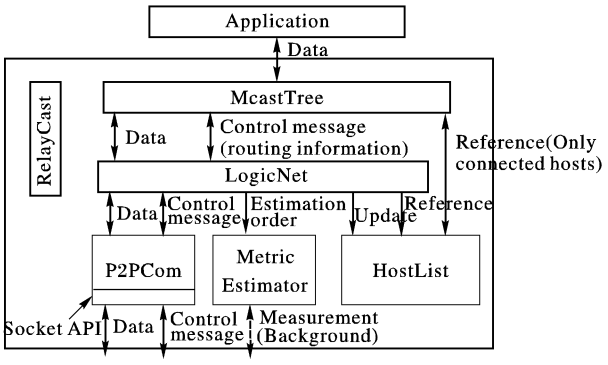


图 1 RelayCast 架构

RelayCast 的总体架构如图 1 所示，单向箭头表示 API 操作，双向箭头表示由 API 操作引起的报文流，方框内的矩形框即是 RelayCast 的各个功能单元。其中，McastTree 的功能是构建组播树和执行路由，将应用数据和控制消息（比如路由信息）经 LogicNet 发送给下游主机。LogicNet 的功能是维护和优化覆盖网络，它从 HostList 中选择要通信的端主机，并向 P2PCom 发送控制消息，比如端主机列表交换、加入/离开、链路连接/断开等；对于应用的数据，LogicNet 将其直接转发给 P2PCom 或 McastTree。MetricEstimator 的功能是根据带宽和延迟来评估网络状态，评估结果被传递给 LogicNet，作为维护和优化覆盖网络的参数。P2PCom 的功能是与 P2P 网络中的端主机直接通信，将控制消息和应用数据传输给由 LogicNet 指定的端主机。HostList 的功能是管理端主机信息列表，列表中的信息由 LogicNet 和 McastTree 引用，但只由 LogicNet 负责更新。

RelayCast 采用网优先组播方案，基于每个源构造组播树。LogicNet、P2PCom、Metric Estimator 和 HostList 共同组成了 RelayCast 的覆盖网络构造模块，建立于其上的 McastTree 则是 RelayCast 的组播路由模块。RelayCast 将应用层组播功能实现为一些单独的构件，这些构件分布在不同的功能单元。实际应用时，用户可以任意选择和组合构件来支持不同的应用；应用也只须和 RelayCast 中间件直接通信，由 RelayCast 负责构建分发树并将应用的数据传输给会话中的所有端主机。在 McastTree 功能单元中，RelayCast 还实现了一种多路径路由（multi-path routing）算法，当组播树因为端主机离开或失效而被分割时可以使用冗余路径快速恢复。

2.3 Emma 中间件

Emma[14] 中间件为实时多方视频通信提供应用层组播基础功能，由日本大阪大学于 2003 年完成开发。Emma 是该研究小组设计的一种应用层组播协议，全称为 End-user Multicast for Multi-party Applications。Emma 协议的覆盖网络是带度约束的，构造方法和 ALMI 非常相似。与 ALMI 不同的是每个结点都有以自己为源的组播树。组播树的构造受限于会话定义的最大延迟或跳数，此外，还要考虑覆盖网络的链路容量。

Emma 中间件使用 Java 开发，由一个主机控制器（host controller）和一个 lobby 服务器（lobby server）组成。主机控制器运行于每台参与组播会话的端主机上；而 lobby 服务器实际上就是一个汇聚点（Rendezvous Point，RP），只在指定的端主机上运行，负责保存和管理所有参与组播会话的结点以及由这些结点所发送的视频源的描述信息（profile）。

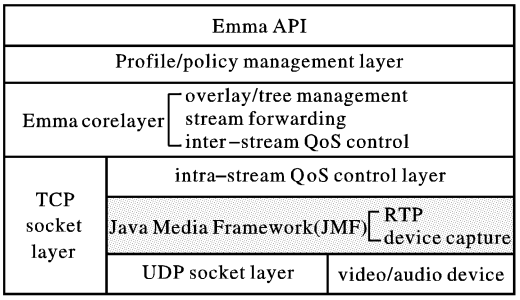


图 2 Emma 中间件主机控制器配置

主机控制器的配置如图 2 所示。其中，profile/policy management layer（描述信息/策略管理层）的功能是保存和管理会话的策略（比如组播树的延迟约束、最大优先值等）以及结点和视频源的描述信息。这些信息都是从 lobby 服务器中获取，主机控制器根据这些信息来监控用户的行为。Emma core layer（Emma 核心层）是 Emma 中间件的核心引擎，它通过交换控制消息来管理覆盖网络和组播路由表；此外，该层还实现视频流转发和流内 QoS 控制（inter-stream QoS control）。流内 QoS 控制功能根据视频流的优先级控制覆盖网络链路上的视频流数量。intra-stream QoS control layer（流内 QoS 控制层）建立在 JMF 之上，利用 JMF 提供的 RTP/RTCP 功能在每条覆盖网络链路上实现拥塞控制和速率控制。

3 应用层组播中间件设计

通过对 ALMI、RelayCast、Emma 等应用层组播中间件的分析，可以看出，将应用层组播的公有特性和基础功能实现为可复用、可配置、易于开发和部署、支持多种组播业务的中间件是完全可行的。本节进一步总结应用层组播的基础功能，

并设计出应用层组播中间件的基本架构。

3.1　基础功能

应用层组播的基础功能可以归纳为以下六点:

1）覆盖网络构造和动态优化

覆盖网络连通所有参与组播的端主机,是构造于端主机之上的逻辑拓扑。由于应用层组播路由是根据覆盖网络的逻辑拓扑来计算,而实际的传输又是在物理拓扑上进行,所以覆盖网络必须最大程度地和端主机的物理拓扑相匹配,否则,物理拓扑上相邻的两个结点在逻辑拓扑上可能相距甚远,进而,基于这种逻辑拓扑所计算的组播路径对于逻辑拓扑而言是最优的,而在实际的物理网络上可能产生大量的冗余传输。

覆盖网络属于控制平面,提供可靠性、鲁棒性和冗余控制。自组织、动态优化,与物理拓扑相匹配是覆盖网络设计的主要指标。

2）组播树的构造和动态优化

应用层组播数据分发通过组播树进行,因而其性能很大程度上取决于组播树的构造。在组播树的构造方法中,共享树为同组的所有成员所共享,树的管理开销小,可扩展性好,适用于大规模组播应用,但无法对每个发送源进行优化;源定义树为每个发送源构造一棵优化的组播树,多棵树的管理开销大,只适用于小规模的组播应用。最短路径树(SPT)和最小生成树(MST)是常用的两种组播树构造算法。

由于端主机的处理能力和带宽有限,通常要根据应用对组播树加入带宽、延迟或度约束。此外,组播成员的加入和退出通常较为随意而且频繁,对于规模较大的组播应用,组播树的计算和动态优化需要在性能和开销上进行权衡。当某些结点失效或退出时,组播树可能被分割,所以组播树的失效恢复机制必不可少。

3）组播路由

组播路由实现组播数据分发。如果成员关系在组播会话中始终不变,则组播路由可以进行针对性的优化,这就是静态组播路由。大多数情况下,成员关系是随机变化的,也必须允许这种随机变化,进而导致组播树需要动态更新,这就要求组播路由要及时适应这种动态性,这就是动态组播路由。此外,组播路由还要保证组播数据的可靠分发。

4）组播组成员管理

组播与单播和广播的主要区别之一是数据收发只在同组的成员间进行。对于商用系统,比如视频点播、视频会议,组成员管理直接与计费和安全相关。组成员管理主要包括成员加入组播组、退出组播组和成员状态维护。当组的规模较大,组成员管理所需的开销是必须要考虑的因素。如果管理开销过大,将防碍系统的性能和可扩展性。

5）会话管理

在实际应用中,系统中同时存在多个组播会话的情况是很常见的,而一个组播成员也可以同时加入多个组播会话中。比如在网络会议系统中,一次主题发言可以表示为一个会话,一个成员可以参与多个主题的讨论,也即可以加入到多个会话中。一般来说,会话和组播组是对应的,但会话管理的主要任务是标识会话、定义和维护会话策略、管理组播组的创建和退出。而组成员管理则具体到成员加入和退出组播组的操作。

6）链路状态监测

链路状态监测是有效构造和维护覆盖网络及组播树的根本机制。应用层组播系统在整个运行期间都需要周期性地监测网络链路状态,根据状态参数来构造和动态调整覆盖网络和组播树,以及对组成员进行管理。所以,链路状态监测的精度和所产生的开销对应用层组播系统有直接的影响。

最常使用的链路状态参数有两个:链路延迟和可用带宽。链路延迟一般就是计算两个结点间的 RTT 值,以 RTT 值的大小来估计两结点间的距离。可用带宽是通过测量发送或接收固定数量的报文所需的时间来进行计算。

3.2　基本架构

通过明确应用层组播的基本服务和基础功能,本文确定的应用层组播中间件基本架构如图 3 所示,其中,会话管理和组成员管理共同组成一个功能单元。如图 3 所示,应用直接利用中间件来实现组播功能;中间件基于 TCP 或 UDP 与其他端主机通信。所有模块都可以利用链路状态监测结果来判断自身的状态或进行相应的操作。应用可以通过会话管理功能来定义会话策略;组播路由、数据分发和组播树根据会话策略适时调整。较为理想的情况是,中间件的某些功能可以实现为多种子构件,应用通过定义会话策略,组合不同的子构件,实现不同的组播方案。

为支持多源组播应用,中间件采用网优先组播方案。首先构造覆盖网络,其次是组播树,而组播路由和数据分发基于组播树进行。至于组播树的构造可以根据组成员规模采用源定义树或双向共享树,前者的性能优于后者,但维护开销也大于后者。

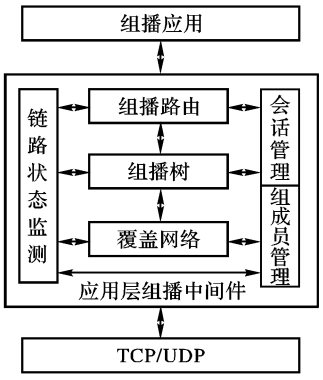


图 3　应用层组播中间件基本架构

一般来说,网优先方法易于提供冗余控制,但不太适合大规模的组播应用,因为在这种情况下,构造和维护覆盖网络的开销比较大。但是,对于大规模的组播应用,极少直接基于端主机构造覆盖网络和组播树,有效的方法是基于组播代理和基于端主机相结合来构造覆盖网络和组播树。也就是说,应用层组播中间件即可以作为一个组播代理来连接和管理其他端主机,也可以支持单个端主机与其他成员直接进行组播通信。这样一来,中间件就可以灵活支持组播应用,使组播应用具备良好的可扩展性。

4　结语

应用层组播中间件的最大优点是能够为多种应用提供可重用、可配置、可组合的应用层组播基础功能,缩短应用开发周期,利于组播应用的大规模部署。因而,研究应用层组播的各个功能机制,建立一个标准化的中间件框架,以中间件的形式来支持组播应用,将是应用层组播的一个发展方向。

参考文献:

[1] BANERJEE S, BHATTACHARJEE B. A Comparative Study of Application Layer Multicast Protocols[EB/OL]. http://www.cs.umd.edu/~suman/publications.html, 2002.

[2] YANG H-C, RAO SG, SESHAN S, *et al*. A Case for End System Multicast[A]. Proceedings of ACM SIGMETRICS[C], June 2000.

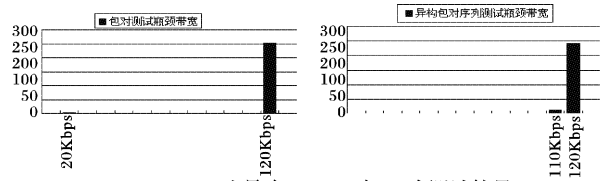
[3] CHAWATHE Y. Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service[D]. Ph. D. Thesis, University of California, Berkeley, December 2000.

(下转第 2260 页)

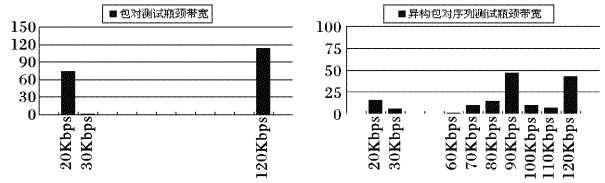
表 1 无背景流量影响的实验过程

步骤	操作内容	测试包	Δ_j, r	$B_{1,j}$
1	测子路径瓶颈带宽 $B_{1,1}$	$[P_1P_2\{P_1P_3\}^3P_1P_2]$, P_1 和 P_3 在 N_1 退出	Δ_1 = 0.192 5, 1	256Kbps
2	测子路径瓶颈带宽 $B_{1,2}$	$[P_1P_2\{P_1P_3\}^3P_1P_2]$, P_1 和 P_3 在 N_2 退出	Δ_2 = 0.192 5, 1	256Kbps
3	测子路径瓶颈带宽 $B_{1,3}$	$[P_1P_2\{P_1P_3\}^3P_1P_2]$, P_1 和 P_3 在 N_3 退出	Δ_3 = 0.192 5, 1	256Kbps
4	测子路径瓶颈带宽 $B_{1,4}$	$[P_1P_2\{P_1P_3\}^3P_1P_2]$, P_1 和 P_3 在 N_4 退出	Δ_4 = 0.385 0, 4	128Kbps
5	测子路径瓶颈带宽 $B_{1,5}$	$[P_1P_2\{P_1P_3\}^3P_1P_2]$, P_1 和 P_3 在 N_5 退出	Δ_5 = 0.385 0, 4	128Kbps
6	测试结论为, $B_{1,n} = B_{1,4}$ = 128Kbps, 瓶颈链路为 L_4			

通过调节 8 个 Pareto 流量发生器,可以模拟出 256 个不同背景流量场景。本文用包对方法和异构包对序列方法分别对 $B_{1,5}$ 进行了测试,这里用柱状图描述了瓶颈带宽的分布情况。



(a) Pareto流量为16Kbps时 256次测试结果



(b) Pareto流量为128Kbps时 256次测试结果

图 3 瓶颈带宽测试情况

图 3 中,横轴表示了可能出现的瓶颈带宽值,纵轴表示实验中某带宽值的测试次数。(a)是 Pareto 流量为 16Kbps 时 256 次测试结果,此时的背景流量比较小,反映了网络负荷较轻时,包对和异构包对序列都可以精确地测试瓶颈带宽。(b)是 Pareto 流量为 128Kbps 时 256 次测试结果,此时的背景流量非常大,两种方法都出现了丢包的情况,因丢包导致的测试错误情况为,包对 67 次,而异构包对序列 101 次。论文把这些有误的测试排除在统计之外,反映了网络负荷很重时,包对和异构包对序列都很难精确地测试瓶颈带宽,异构包对序列测试的带宽分布在比较宽的范围内,而且均值较实际值偏小。看来,采用异构包对序列的方法仍然会受到背景流量的影响,但它可以方便的定位网络瓶颈位置,这是基本包对方法做不到的。

3 结语

本文提出了基于异构包对序列的瓶颈测试方法,它具备包对测试的特点,能够较快地测试出网络瓶颈带宽,发现路径上的瓶颈链路。网络仿真结果表明该方法比较有效。下一步将应用基于异构包对序列的瓶颈测试算法设计测试程序,开展实际网络下的实验,了解实际背景流量对异构包对序列方法的影响程度,并提出算法改进思路。

参考文献:

[1] HARFOUSH K, BESTAVROS A. Measuring Bottleneck Bandwidth of Targeted Path Segments[A]. Proceedings of IEEE INFOCOM'01 [C]. Alaska, 2003. 2079 - 2089.

[2] PAXSON V. End - to - end Internet Packet Dynamics. IEEE / ACM Transaction on Networking[J], 1999, 7(3): 277 - 292.

[3] JACOBSON V. Pathchar - a tool to infer characteristics of Internet paths[Z]. Presented at the Mathematical Sciences Research Institute, 1997.

[4] LAI K, BAKER M. Measuring Link Bandwidths using a deterministic model of packet delay[A]. ACM SIGCOMM2000 Proceedings [C]. Sweden, 2000. 283 - 294.

[5] CARTER RL, CROVELLA ME. Measuring bottleneck link speed in packet switched networks [J]. Performance Evaluation, 1996, 27&28: 297 - 318.

[6] NS. http://www.isi.edu/nsnam/ns/, 2005.

(上接第 2257 页)

[4] PENDARAKIS D, SHI S, VERMA D, *et al.* ALMI: An Application Level Multicast Infrastructure[A]. Proceedings of the 3rd Usenix Symposium on Internet Technologies & Systems[C], March 2001.

[5] FRANCIS P. Yoid: Extending the Multicast Internet Architecture [EB/OL]. http://www.aciri.org/yoid/, 1999.

[6] ZHANG B, JAMIN S, ZHANG L. Host multicast: A frameworkfor delivering multicast to end users[A]. Proceedings of IEEE Infocom [C], June 2002.

[7] JANNOTTI J, GIFFORD DK, JOHNSON KL, *et al.* Overcast: Reliable Multicasting with an Overlay Network[A]. Proceedings of the 4th Symposium on Operating Systems Design and Implementation [C], October 2000.

[8] ZHI L, MOHAPATRA P. Hostcast: A new overlay multicasting protocol[A]. Proceedings IEEE 2003 Intl Conf on Communications (ICC 2003)[C], May 2003.

[9] ZHUANG SQ, ZHAO BY, JOSEPH AD, *et al.* Bayeux: An architecture for scalable and fault-tolerant widearea data dissemination [A]. In Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video(NOSSDAV 2001)[C], 2001.

[10] CASTRO M, DRUSCHEL P, KERMARREC A-M, *et al.* SCRIBE: A large-scale and decentralized application-level multicast infrastructure[J]. IEEE Journal on Selected Areas in communications (JSAC), Sp. Issue on Network Support for Group Communication 20, October 2002.

[11] BANERJEE S, BHATTACHARJEE B, KOMMAREDDY C. Scalable application layer multicast[A]. Proceedings of ACM SIGCOMM [C], August 2002.

[12] SHI S. A Proposal for A Scalable Internet Multicast Architecture [R]. Technical Report WUCS-0103, Washington University in St. Louis, May 2001.

[13] MIMURA N, NAKAUCHI K. RelayCast: A middleware for application-level multicast services[A]. Proceedings of the 3rd Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed System(GP2PC 2003)[C], May 2003.

[14] YAMASHITA T, YAMAGUCHI H,. Emma Middleware: An application-level Multicast Infrastructure for Multi-party Video Communication[A]. Proceedings of the 15th IASTED Intl. Conf. on Parallel and Distributed Computing and Systems(PDCS2003)[C], November 2003.