一种基于终端的多源应用层组播系统

惠 飞,黄士坦
(西安微电子技术研究所,陕西 西安 710071)

摘 要:针对实时小规模多源视频会议的需求,提出了一种基于终端的多源应用层组播系统EMSALM。EMSALM是一个完全分布式系统,采用资源确定多播树来发送多媒体数据。系统中所有成员是逻辑平等的,每一个终端包含一个完整的成员列表和对自己多播树的完全控制。结果证明对于小型的实时通讯来说,系统是有效的。

关键词:视频会议;分布式;应用层;组播
中图分类号:TN919.85;TP393 **文献标识码**:A **文章编号**:1673-629X(2006)05-0143-03

An End-Based Multi-Sources Application-Level Multicast System

HUI Fei, HUANG Shi-tan
(Xi'an Institute of Microelectronics Technology, Xi'an 710071, China)

Abstract: Propose an end-based multi-sources application-level multicast system(EMSALM) tailored to small scale video conference. This is a fully distributed system. All members are equal: each end has a complete member list and control of multicast tree. It has been proved that the system is efficient for small scale real-time communication.

Key words: video conference; distributed system; application-level; multicast

0 引 言

随着远程教学、异地医疗和全球商务中对实时交互性视频会议需求的不断增长,小规模多源视频会议的研究也日益得到重视。区别于一对多的分布式系统,这种类型的视频会议比较小,一般少于十个参与者,参与成员更换也相当快,任何一个成员可自由地加入、退出或者邀请其他成员。此外,经常是数据源和与会成员的数量相当,对每一个数据源来说,至少包含两种类型的多媒体流:语音和视频。

更为关键的挑战来自于对实时性的要求,众所周知,视频会议是一种双工通讯的实时应用系统。它对终端之间的反应时间要求很苛刻。这不同于单向传输且允许接收端有几秒缓冲时间的视频流应用。目前,随着应用层组播技术的研究的深入,它相对于IP组播的优点越来越多地显示出来。

应用层组播有两种实现方式:一种是将所有的功能放在实际参加群组通信的主机中;另一种则是由分布在网络中的多个网关系统完成组播功能,每个网关可以为多个客户端同时服务。第一种方法可以做到完全分布;第二种方法则可以提高组的规模。文中介绍的应用层组播系统采用了第一种方法。

收稿日期:2005-08-18
作者简介:惠 飞(1982-),男,安徽濉溪人,硕士研究生,研究方向为网络多媒体技术、嵌入式系统;黄士坦,研究员,博士生导师。

1 相关研究

多播路由技术的发展经历了两个阶段。早期是IP多播。这种架构首先由Steve Deering提出[1],他提出多播相关的功能应该实施在网络层。尽管如此,一些固有的结构缺陷限制了IP多播的发展,例如高复杂度、缺乏可分级性以及针对恶意攻击的安全措施。在这种情况下,一些研究人员提出多播的相关功能在网络层实施。

基于应用层的多播(ALM)通常包括协议和架构两部分的设计[2]。后者集中于以前的工作如分播(Scattercast)和覆播(Overcast),两个ALM系统被设计针对网络分布式内容。不同于其他ALM系统的是,他们在网络中的关键位置放置了结点的集合。对于海量的分布式系统来说,额外的开销是值得的。尽管如此,对于小型的对等的视频会议来说,代价太大。

多数的ALM系统建立在终端的基础上而不需其他额外的设施。主要定位于大规模应用,包括上千个终端。其中的一些,例如NICE[3]和SpreadIT[4],实现了对结点的管理和建立视频流的多播树。其他的一些使用已有的P2P系统提供对象管理和路由功能。这类系统[5]包括SCRIBE, Bayeux和CAN-多播,分别建立在Pastry, Tapestry和CAN基础上。对于以上提到的所有多播系统来说,主要的目标是减少网络资源使用和平衡潜在的传输路由的连接压力。尽管这样,对于实时系统非常重要的结点间的延迟,这些系统并没有做太多优化。

小型ALM系统与大型ALM系统的差异较大。它们

不依赖于硬件,也没有太多基于基础结构的设置。由于规模比较小,组状态的维持可以简单地用指定结点或每一个组成员保持完整的成员列表。系统设计的关键在于多播路由。区别主要在以下几个方面:

(1)基于成员和树维持的集中或分布式控制。

(2)网优先或树优先结构策略。

(3)针对数据传送的共享树或资源确定树。

在终端系统多播中,终端系统使用一种名为 Narada[6]的完全分布协议自组织成为一个覆盖图结构。Narada 采用网点优先策略建立多播树。它形成了一个丰富的连接图(叫做一个网点),然后产生基于网点的确定资源数据分布树。Narada 的缺点是没有对于给定网点的结果生成树的控制。这来自两方面的关系:一个高质量的网不能对于所有的资源的有效多播树生成;Narada 不考虑同一个网的多个树的影响。这样,在后来的使用 Narada 的视频会议的研究中,采取在某一时间点单个数据源的方式。

与 Narada 不同,ALMI 是一个集中式协议。每一个 ALMI 会议有一个会议控制器,会议控制器负责成员注册和多播树维持[7]。多播树是一个由树优先策略建立的共享树。会议控制器阶段性的重新计算一个由会议成员收集的端到端距离的新树。尽管共享树易于管理,但它不像资源确定树那样有良好的延迟特性。集中设计也引发了两个问题:如果控制器失效,多播树将保持不变,因此当网络变化时易受攻击;在新旧多播树交替期间,系统明显不稳定。

在针对多发送源的 3D 视频会议的协议中使用了以上两种系统的混合方式[8]。它采用了类似于 ALMI 的集中方式管理树并采用类似 Narada 的网优先策略生成树。在协议中使用了一种对于成员加入的双算法方式的新理念。如果本地算法在添加一个新接收者时无效,则全局算法将研究一个对于所有树的新的安排。尽管这样,仍然有以下缺点使得它不能实际应用:它也有和 ALMI 中的单点失败类似的问题;它没有将网络的动态资源考虑在内,采取了使用静态有效带宽来计算多播树的方式;第二算法在重新安排所有树的时候没有考虑它们原来的拓扑结构,这样,在树的交替期间,长时间延迟和网络阻塞将不可避免。

针对以上系统的特点,文中提出了一种适合小型多源视频会议的系统。

2 EMSALM 系统

文中提出了 EMSALM,一种基于终端的小型多源应用层组播系统。

EMSALM 是一个完全分布式系统。所有成员是逻辑平等的:每一个终端包含一个完整的成员列表和对自己多播树的完全控制。

为了处理多个数据源,传统观念是采用共享覆盖作为数据发送树组织多个会议。管理花费的成本也比使用多个资源确定树少的多。尽管如此,共享覆盖没有资源确定树那样良好的延迟特性。在小型的实时会议应用中,更注重延迟特性而不是管理复杂度。因此,EMSALM 选择资源确定多播树来发送多媒体数据。

在构建多播树时,应该考虑诸如延迟、带宽、阻塞和成本等很多因素。另外,使用单制指示路径选择器明显不能满足多媒体应用需求。因此,设计了基于延迟和可用带宽度量的两级树构成算法。

事实上,网络是动态变化的。端对端的延迟和可用带宽也不是固定的。因此,需要一个机制去适时测量这两种指示器以便建立有效的多播树。在 EMSALM 中,会议成员定时探测相互之间获得的端到端的延迟。这种探测也用来探测用户是否失效:如果 A 在一段时间后没有收到 B 发出的探测信号,就认为 B 网络连接失败。

3 多播路由算法和协议

EMSALM 采用树优先策略构建多播树。在多播树维持中采用了两级处理:使用本地贪婪算法首先建立资源确定树,然后用全局改善算法逐步改善。两种框架都建立在延迟和带宽测算的基础上,它们共同的目标是优化数据传输在给定带宽限制下的延迟。

3.1 多播树建立算法

EMSALM 是一种完全分布式系统,其中每个数据源管理自己的数据分布树。当数据源 S 接收一个从成员 M 发出的预定请求时,它尝试添加 M 到自己的多播树中。以下为几种可能使用的添加方式:

结构 0:S 直接添加 M;

结构 1:添加 M 到一个指定的树结点(内结点或者叶子结点),并保持其他连接不变;

结构 2:M 取代已存在结点 C 的位置并添加 C 到它自己的子集中。

这三种结构的拓扑如图 1 所示。

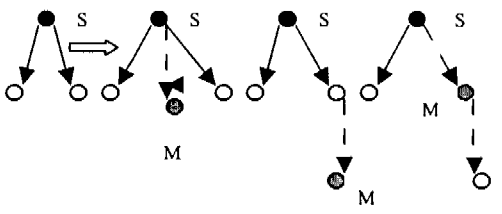


图 1 三种可能的添加方式拓扑图

为了更好地描述,如果 S 的可用带宽允许的话,建立算法将添加 M 到树的根结点。否则,它将比较其他两种结构的开销并选择开销小的一个。这里,结构的开销被定义为这种结构生成的多播树中 S 到 M 或其子孙的最大反应时间。

在实际中,结构 1 和结构 2 优先选择前者,因为结构 1 不会带来已存在传输路由的改变,能保持较好的稳定性。因此,只有在结构 2 的性能比结构 1 超越某个极限时才选择它。图 1 显示了 M 加入根结点为 S 的多播树时的三种可能的拓扑。

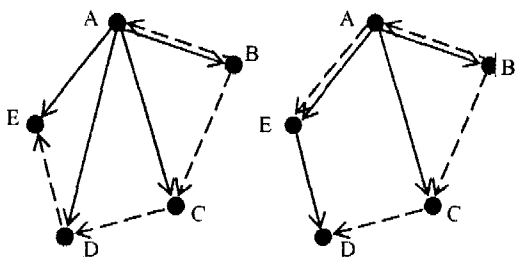
如果没有成员有足够的带宽,那么预定请求将在等待列表中排队,多播树将保持不变。通过不同的流提供不同的服务并优先音频流。因此,当一个音频请求没有满足时,将牺牲视频请求来满足它。

由于多播树使用分布式规则来维持,它必须定义一个规则的集合,或者协议,以规范路由变换行为:当一个数据源要变化它的多播树时,它发送一个新子集到子集需要更新的成员 P。收到这个消息后,P 接受它带宽允许的子结点。如果有些结点未完成,则它向 S 发送饱和信息,包括它的带宽能力和未添加结点列表,然后 S 可以根据更新的带宽信息来重新安排这些结点。

与预定请求相关,如果成员 M 不再愿意接收特定的流,它可以发送结束预定请求到数据源 S。S 将把 M 从多播树中删除并重新安排 M 剩余的子结点。

3.2　多播树改善算法

树的建立算法是本地贪婪算法。每个数据源用它自己的带宽资源去传输它自己的数据。这将导致各多播树在网络资源的使用上不均衡。图 2 显示了一个 5 成员会议多播树在细化前后的过程。实线代表根结点为 A 的多播树,虚线代表根结点为 B 的多播树。在改善前(如图 2a 所示),成员 A 以单播方式进行数据分布,使用了它最大的带宽资源。因此它不能将 B 的多播树转播给 E。结果是从 B 到 E 的传输路由(B－C－D－E)非常的长。



(a) 改善前　　(b) 改善后

图 2　改善算法调整多播树示例

此外,每一个多播会议的成员关系变化相当快,网络也是动态的。这也需要持续的细化多播树以获得不带来明显紧张的更好的性能。细化框架需要数据源之间的合作,那么更多的预定请求可以被满足并且有向边端对端的反应时间可以缩短。

改善的过程分为五步。当等待列表中存在未被满足的请求或者存在由于不合理拓扑造成的过长传输路由时进行改善。全部的改善过程可被描述如下:

第一步:数据源 S 发送一个针对 L 的父参考列表的询问帮助消息。对每一个列表中的结点 N,如有足够的带宽接受 L 作为孩子,则能改善 S 到 L 的传输反应时间。

第二步:接收到询问帮助信息后,其他树控制器 T 将检测是否可以减少其中某些结点的带宽使用并发回带有重新安排开销的应答帮助信息。

第三步:节点 S 收到其他成员的所有回答后,比较建议框架并选择最小开销的那一个。S 将发送帮助消息到提供最好框架的成员 B。

第四步:B 检查提供的框架是否仍然有效,如果有效,在它重新安排分布树后发送一个应答帮助消息;如果无效,它将发送一个指示有些情况已经变化它不能维持原有承诺的拒绝帮助消息。

第五步:如果得到应答帮助消息,S 重新安排 L。否则 S 拒绝改善当前的会议并等待下次改善性能的机会。

从图 2b 中可见,改善过程后,在有向边(如:反应时间 B－C－D－E)的端到端的反应时间大大缩短。新的传输路由 B－A－E 达到了多播树间网络资源利用的平衡。

为了避免频繁改变数据路由树所造成的负担,只有当带宽和延迟的改变超过一定范围,而负载接近满负荷时才会启用树的改进算法。

4　实验与测试

在 EMSALM 中,终端间探测间隔为 5s,连接失效指示时间为 20s。多播路由的另外一个重要机制是,传输路径的可用带宽也要被检测。根据上面提出的协议设计,在 Windows 平台上利用 WinSock 编程接口和多线程的机制实现了该协议。

在现在的试验中,用户之间互相侦测使用每 10 秒钟一次的一对 1kb 的 UDP 数据包。因此,在一个典型的 5 用户会议中,每两个用户之间可用带宽的决定时间为 40s。

每 5s,每个用户将广播它在会议中的更新测量值。与三项循环工作相关的网络开销如表 1 所示。可以看出所有的开销少于 4 kbps,这可以被大多数的网络用户接受。

表 1　系统性能测试结果

工作	包大小	频率	开销
延迟测试	50 byte	8 pck/5 sec	0.7kbps
带宽测试	1000 byte	2 pck/10 sec	2.2kbps
更新测试	80 byte	4 pck/5 sec	0.6 kbps

5　结论和未来的工作

描述了 EMSALM,一种针对多方实时通讯的应用层多播系统的系统架构设计。系统针对源采用了资源确定树发送多媒体数据,结果证明对于小型的实时通讯来说,系统是有效的。

下一步的工作是继续改进数据路由树的构建和维护算法,提高组的规模,改进传输效率,为不同能力的网络和客户端提供适合的数据,并结合工程开发一个多媒体应用系统,将系统付诸实施。

参考文献:

[1] Deering S. Multicast routing in internetworks and extended

(下转第 148 页)

统中的另一个定时事件 timer_sendmsg,用于完成对报警信息队列的扫描,如果队列不为空,则发送报警信息。后台监控核心处理系统的内部工作机制如图 5 所示。

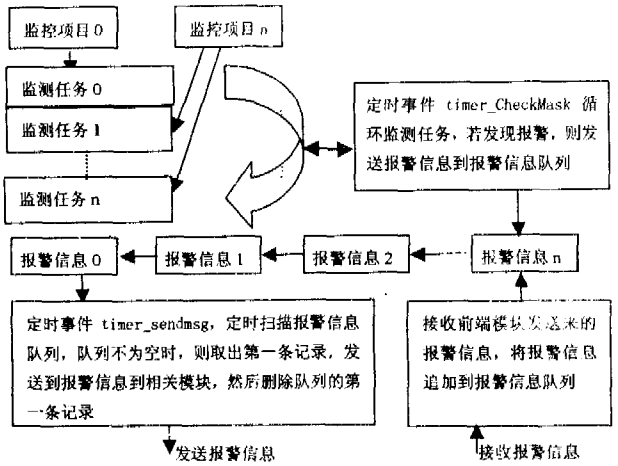


图 5 后台监控核心处理系统的内部工作机制

2.2.4 系统中的报文驱动模块协同工作机制

文中将服务器和客户端通过 Socket 套接字传输的字符串称为通信报文,简称报文。各个模块之间是通过预先约定规则的报文来默契配合进行驱动整个体系来工作的。以下说明主要模块间的报文约定[4]。

(1)前端机房处理机监控系统→后台监控核心处理系统的报文:报警主机 IP,报警信。息

(2)后台监控核心系统→前端机房处理机系统的报文:命令标识,数据 或命令标识。

(3)后台监控核心处理系统→报警信息终端的报文如下:

①显示报警信息的报文:命令标识,事件号,报警信息。

②播报报警语音的报文:命令标识,是否播报,事件号,语音文件名。

(4)后台监控核心系统→总部集中监控系统通信报文:营业部标识,事件号,报警信息。

2.2.5 系统中的通用数据库监测技术实现原理

通用数据库检测内容包含如下几个方面[6]:

(1)任意表变化的检测,在一定时间阈值内,检测表记录数是否变化。

(2)任意表字段内容的检测,在一定时间阈值内,检测表字段等于或不等于设定值。

(3)任意表条件检测,在一个时间点上,计算出的数据记录数和设定的阈值进行对比。

(4)任意两个表,在一个时间点上,记录数的对比。分相等和不等两种情况。

(5)对 SQL 表,系统允许用户自主选择要连接的服务器、数据库、数据表、表字段,允许用户自己定义 SQL 语句。系统按设置的时间范围实时监控扫描,执行用户定义的 SQL 语句,将输出结果和阈值进行对比。

3 系统的总结与展望

提出了集中监控体系的整体架构和设计思路,成功开发出了具有扩展功能的证券行业通用智能监控系统。系统在多家证券公司总部机房和营业部投入使用,取得了良好的效果,对证券信息系统的安全稳定运行发挥了重要作用,越来越成为机房人员的智能助手。诚然,本系统还处于起步阶段,仍然存在着许多不足,还需要进一步补充和完善。可以按照文中提出的集中监控体系的整体架构和设计思路,利用跨平台的 Java 技术,研发出与操作系统无关的、先进的、基于 Web 的、下一代证券行业通用智能监控系统。

参考文献:

[1] 雷震甲.计算机网络管理及系统开发技术[M].北京:电子工业出版社,2002.

[2] Krishna C M,Shin K G. Real-time Systems[M]. USA:MC Graw Hill,2004.

[3] 求是科技.Delphi 7 程序设计与开发技术大全[M].北京:人民邮电出版社,2004.

[4] 钟 军,汪晓平.Delphi 网络通讯协议分析与应用实现[M].北京:人民邮电出版社,2002.

[5] 龙启明,刘 斌,程 捷.Delphi7 高级编程范例[M].北京:清华大学出版社,2004.

[6] 求是科技.SQL Server 2000 数据库管理与开发技术大全[M].北京:人民邮电出版社,2004.

(上接第 145 页)

LANs[A]. Proceedings of the ACM SIGCOMM88[C].[s.l.]:[s.n.],1988.55-64.

[2] Wu D, Hou Y T, Zhang Y Q. Transport Real time Video over the Internet: Challenges and App roach[J]. Circuit and Systems for Video Technology,2001,11(3):282-300.

[3] 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),2002,7:364-378.

[4] Zhuang S Q,Zhao B Y,Joseph A D,et al. Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination[A]. In Proc of NOSSDAV'01[C].[s.l.]:IEEE,2001.58-67.

[5] Waxman B. Routing of Multipoint Connections[J]. IEEE J on Selected Areas in Comm,1988,6:1617-1622.

[6] 陈庆吉.支持实时多媒体传输的应用层组播系统[J].计算机工程,2005(2):136-139.

[7] 方 奕,张 卫.一个单源的应用层组播协议的设计和实现[J].计算机应用,2005,25(2):859-862.

[8] 李 伟,沈长宁.应用层组播协议的研究[J].计算机科学与工程,2004,24:156-159.