采用 SCTP 设计多路径媒体传输平台

李挺屹[1,2],王劲林[2]
LI Ting-yi[1,2],WANG Jin-lin[2]

1.中国科学院 研究生院,北京 100049
2.中国科学院 声学研究所 国家网络新媒体工程技术研究中心,北京 100190
1.Graduate School of Chinese Academy of Sciences,Beijing 100049,China
2.National Network New Media Engineering Research Center,Institute of Acoustics,Chinese Academy of Sciences,Beijing 100190,China
E-mail:lty@dsp.ac.cn

LI Ting-yi,WANG Jin-lin.Transferring platform for media data based on SCTP.Computer Engineering and Applications,2010,46(4):59-61.

Abstract: This paper designs the architecture of a multi-path transferring platform based on SCTP,proposes the methods to select the overlay-paths and decide the SCTP ports being used.The process of data transferring between two nodes is given.
Key words: Stream Control Transmission Protocol(SCTP);multi-path;overlay path

摘 要:设计了一个基于 SCTP 的媒体数据传输平台,基于 SCTP 的特点,采用多路径的传输方法,提出了重叠路径和传输端口对的选择方案,给出了实现流程。
关键词:流控传输协议;多路径;重叠路径
DOI:10.3778/j.issn.1002-8331.2010.04.019 **文章编号:**1002-8331(2010)04-0059-03 **文献标识码:**A **中图分类号:**TP393

1 引言

在互联网上传输大数据量的内容时,如何提高传输速度是很多研究者关注的焦点。在点到点数据传输中,通常使用的 TCP 协议效率比较低,研究者提出了 XCP、UDT[1]等协议来加快数据的传输,此外也提出了各种不同的 FEC 算法,传输冗余数据,放弃丢包重传。由于互联网具有多路径特性,将数据通过多个路径进行传输也是一个有效的途径[2]。通过对多个路径的带宽集聚,可以提高所传输的数据率。

在实践中,两个主机之间通常只存在一个 IP 路径,所以多路径的实现多基于重叠路径,即通过其他节点进行中转。该文在基于 SCTP 协议的多重叠路径传输方法[3]的基础上,研究了该方法的具体应用,设计了一个多路径的媒体传输平台,给出了传输路径和传输端口的选择方案及流程实现。

2 分布式媒体服务系统

该文设计的传输平台是一个分布式媒体服务系统的重要组成部分。该分布式媒体服务系统结构如图 1 所示。系统采用了"863"计划重大专项"3TNet"[4]所研发的大规模接入汇聚路由器(ACR)[5]提供用户接入,每个 ACR 可以支持数万用户接入,并且为每个用户提供数十兆带宽。每个 ACR 均和一个媒体服务节点直接连接,该媒体服务节点负责为该 ACR 接入的用户提供媒体服务,如流媒体、下载、视频通讯等。为了能满足上万用户的并发流媒体需求,该节点实际是一个服务器集群,将它看成一个节点,不涉及集群的内部细节。所有媒体服务节点组成系统的媒体内容存储和传输网络,系统的所有媒体文件都切片分散存储在媒体服务节点上,媒体数据在节点之间的传输通过多重叠路径的方式进行。

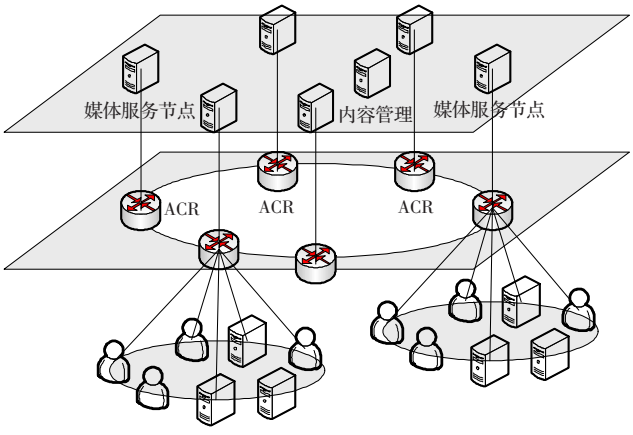

图 1 分布式媒体服务系统结构示意图

所有的媒体内容分布存储于媒体服务节点之上,系统有一

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2008AA01A317);中国科学院知识创新工程青年人才领域前沿项目。
作者简介:李挺屹(1975-),男,博士生,助理研究员,主要研究领域为内容分发网络、P2P 网络;王劲林(1964-),男,主任研究员,博士生导师,主要研究领域为宽带多媒体通信。
收稿日期:2009-03-05 **修回日期:**2009-04-08

个集中式的内容管理服务器,它的功能之一是根据媒体文件的特性(如受欢迎程度、地域性等)来决定媒体的分布存储策略。任何一个媒体服务节点都可以接收获得内容管理服务器许可的媒体文件进入系统。当媒体文件进入系统时,接收该文件的媒体服务节点执行对媒体文件的切分操作,并根据内容管理服务器做出的分布存储策略将切分后的数据块传输到其他媒体服务节点进行存储。每个数据块存储在至少一个节点上,同一个文件的不同数据块被存储的位置和副本数目可能是不一样的。随着媒体文件特性的变化,内容管理服务器可以调整媒体的分布,以更好地提供媒体服务。

媒体数据在系统中的传输主要有如下几种场景:(1)媒体数据进入系统时,从进入节点向其他节点进行分布存储;(2)内容管理服务器对分布存储进行调整;(3)当用户所请求的内容在节点没有存储时,该节点从其他存储有内容的节点获取内容到本地;(4)当用户之间进行具有实时性要求的数据传输(如音视频会话)时,由用户将音视频媒体数据传输给其所属的媒体服务节点,由该媒体服务节点传输到对方用户所属的媒体服务节点,再转发给用户。由于媒体数据量很大,并且高质量的视频码率较高,为了加快媒体数据的传输速率,系统设计中将所有媒体服务节点组织成一个多路径传输平台,采用多路径的方式来提高传输速率。

3 传输平台的设计实现

3.1 平台概述

图 2 给出了一个跨三个城域网的传输平台示意图。

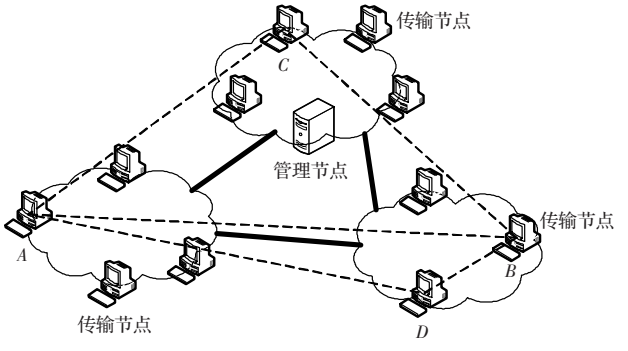

图 2 多重叠路径传输平台

平台由分布在广域/城域网上的媒体服务节点和管理节点组成媒体服务节点都是传输平台的节点(后文称传输节点)。管理节点负责对传输平台进行管理,具有如下功能:

(1)对传输节点进行合法性认证,保证只有认证合法的节点才能作为传输节点存在,防止未经认证的节点盗用本传输平台的资源或者窃取本传输平台所传输的信息;

(2)接收传输节点的状态报告信息,对当前和历史状态信息进行管理和统计分析,提供状态报告;

(3)向传输节点发布传输平台其他节点的信息,如节点的ID、IP、所属城域网编号、当前状态等,供传输节点选择重叠路径时使用;

(4)应传输节点的查询请求向其应答其他节点或节点间路径的信息(为避免管理节点成为瓶颈,传输节点间通常互相直接查询和测量来获得这类信息,当该方式失效时才向管理节点查询);

(5)提供操作人员的管理接口,如增加、删除、暂停、恢复传输节点等。

当传输节点参与不同的传输任务时,节点可以同时充当源节点、目标节点和中转节点这三种角色。图 3 给出了传输节点的主要功能模块及其所使用的传输层协议。

应用层	测量	发送	接收	IP 转发控制
传输层	UDP	SCTP		TCP
网络层	IP(包括 IP 层转发)			

图 3 传输节点的功能与使用的协议

各模块完成如下功能:

(1)测量模块是指测量传输节点之间的链路质量信息,为选择多条传输路径提供信息;

(2)IP 转发控制是根据所选择的传输路径,请求相应的中转节点进行转发配置,以中转本次传输的数据;

(3)发送和接收模块是基于 SCTP 协议进行媒体数据的发送和接收工作;

(4)IP 层转发是当传输节点作为中转节点角色时,根据 IP 转发控制所配置的转发规则,将 SCTP 协议的特定报文进行转发。

当节点之间的测量出现异常导致多路径选择无法完成时,向管理节点报告该异常,并由管理节点提供可作为多路径选择依据的链路质量信息。

该传输平台利用文献[3]所提出的方法在任意两个节点之间进行基于多重叠路径的媒体数据传输。下面首先简介该方法,再对传输平台的设计进行详细说明。

3.2 基于 SCTP 的多路径传输方法

SCTP 支持多穴的特点为基于其进行多路径传输提供了可能,但是需要主机的多个网络接口分别接入不同的网络,这在当前的互联网上很难解决。文献[3]提出一种基于 SCTP 的在单接口主机间的多重叠路径传输方法。

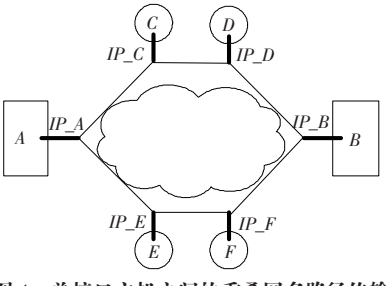

图 4 单接口主机之间的重叠网多路径传输

如图 4 所示,非多穴主机 *A* 和 *B* 之间要进行多路径传输,可以通过中间节点 *C*、*D* 以及 *E*、*F* 进行中转,从而形成两条重叠路径。在主机 *A* 看来,它是在与一个具有三个接口(*IP_C*、*IP_B*、*IP_E*)的主机进行通信,而主机 *B* 也可以认为它是与一个具有三个接口(*IP_D*、*IP_A*、*IP_F*)的主机进行通信。通过修改 *A* 和 *B* 之间建立 SCTP 关联的 INIT 和 INIT_ACK 报文,在报文中携带不属于 *A* 或 *B* 的中转节点的 IP 地址,可以让通信对方认为自己是一个具有多接口的主机,从而将报文发给中转节点。同时,在中转节点配置 IP 转发策略,就可以在 *A* 和 *B* 之间建立重叠路径并进行通信。文献[3]给出了仅配置 IP 层策略,而不修改 SCTP 协议栈的一种实现途径。该方法应用于传输平台时,需解决传输路径的选择和并发传输的端口选择问题,下面

分述之。

3.3　传输路径的选择

每个传输节点都有一个链路性能测量的功能模块，定期测量自己与其他节点之间的 IP 路径的时延、带宽等链路质量指标。

当节点 A 需要向节点 B 传输数据时，A 首先判断 B 是否和自己在同一个城域网内(是否具有相同的城域网编号)，如果是，则 A 将其他在该城域网内的传输节点作为候选中转节点集合 R。如果 A 和 B 不在同一个城域网内，则 A 将传输平台的所有传输节点作为候选中转节点集合 R。其他节点信息是由管理节点发布的。

A 向 R 中所有节点发出请求，查询各节点到 B 的链路质量信息。A 收到这些信息后，结合自己到各个中转节点的链路质量信息，综合得到经过各个中转节点到 B 的链路质量情况，并择优确定 N_{r0} 个节点作为该次传输的中转节点。若 R 中节点数量特别庞大，则 A 可根据自己保存的到这些节点的链路质量信息，选取质量好的前 $k\times N_{r0}$(k=2 或 3)个节点，查询它们到 B 的链路质量信息，再综合择优确定 N_{r0} 个节点。

A 通过与 B 之间的直接 IP 路径以及由 N_{r0} 个中转节点确定的重叠路径(共 $1+N_{r0}$ 个路径)向 B 进行数据传输。B 统计自己在 ΔT 时间内接收到的数据速率，并判断是否满足自己的需求。如果 B 需要更大的数据传输速率，则向 A 发出增加重叠路径的请求。

A 收到增加重叠路径的请求后，重复前面确定中转节点的过程，选出 N_{r1} 个中转节点，并利用 SCTP 动态地址配置的功能[6]，在当前的 SCTP 关联中增加新的中转节点地址，从而增加重叠路径，此时数据通过 $1+N_{r0}+N_{r1}$ 个路径向 B 进行传输。

3.4　传输端口对的确定

根据文献[3]的方法，当只有一个数据传输任务在执行时，中转节点的转发规则很简单，只需对数据包的源 IP 地址进行判断并修改源和目的 IP 地址即可。例如图 2 中，当 C 转发来自 A 的报文时，首先判断所收到的 SCTP 数据包的源地址是否为 A 的 IP 地址，若是，则将数据包的目的地址改为 B 的 IP 地址，将源地址改为 C 的 IP 地址，并转发出去。

但是，由于在传输平台上，同时有很多传输任务在进行。如图 5 所示的情况，共有 7 个节点(A~G)，8 个编号为 1~8 的数据流。其中 1、2 是直接从 B 到 C，3、4 是 A 发送经 B 中转后到 C，5 是 A 发送经 B、C 中转后到 E，6 是 B 发送经 C 中转后到 E，7 是 B 发送经 C 中转后到 F，8 是 B 发送经 C、D 中转后到 G。且 1、2、3、4 分属于节点 C 的不同应用，3、4、5 分属于 A 的不同应用，5、6 分属于 E 的不同应用。

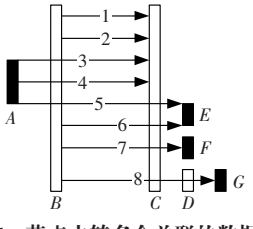

图 5　节点中转多个关联的数据流

对不同的传输任务，节点可能是源节点、中转节点或者目的节点。一个节点收到 SCTP 数据包后，需要判断自己是否是该数据包的目的节点，如果不是，则进一步判断应该向哪个节点进行转发。通常，IP 转发可以通过五元组(源地址、源端口、协议、目的端口、目的地址)来对报文进行分类，然后根据特定策略转发出去。但是在所述传输平台中很可能出现源地址、目的地址、协议都一样的情况。例如图 5 中，从 B 到 C 的 8 个数据流的源地址(B 的 IP 地址)，目的地址(C 的 IP 地址)，协议(SCTP)都一样。此时只有依靠源端口和目的端口来制定转发策略。在整个传输平台中，任何一个中转节点都可能中转平台中当前的所有传输任务的数据，因此，整个平台中不应该出现完全相同的源、目标端口对。可以用管理节点来统一管理和分配 SCTP 关联的端口对，以确保唯一性。假设每个节点可用的端口数目都为 K，则传输平台中可并发支持的传输关联数为：

$$Num\_Association=\frac{K(K-1)}{2} \tag{1}$$

考虑到所有的关联都是双向的，不能区别源端口和目的端口，所以式(1)中除以 2 才是真实可用的端口对数目。

然而，集中进行端口管理会使管理节点成为性能瓶颈，带来失效隐患。设计了一种不需要管理节点分配端口且可以避免系统中出现相同端口对的方法。

前面提到，每个节点都可以作为数据的发送者、接收者，需要考虑每个节点的发送实体和接收实体的端口分配问题。首先将端口集合分成两部分，称为 P_s 和 P_d，集合中的端口数目分别为 K_s 和 K_d，有 $K_s+K_d=K$。将 P_s 平均分配给所有节点，固定这些端口作为发送实体的候选端口。若传输平台有 N 个节点，那么每个节点可以有 K_s/N 个端口。传输平台最大可以容纳 K_s 个节点，此时每个节点的发送实体只有一个端口可供使用。

一个待建立的关联由节点 i 的发送实体和节点 j 的接收实体共同构成，为了避免节点 j 的接收实体随机选择端口而可能带来的混淆，由节点 i 在端口集合 P_d 中排除掉当前已经与自己建立关联的对方节点的端口后，随机选择一个剩余的端口作为该新发起的关联的对方端口，并通知节点 j。这样，所有节点的发送实体的端口是不可能出现重复的，且单个节点所建立的所有关联的接收实体端口不会相同。而且由于发送实体和接收实体的候选端口是互斥的集合 P_s 和 P_d，因此，系统中不会出现相同的端口对。

传输平台可并发支持的关联数目为：

$$Num\_Association=K_sK_d \tag{2}$$

考虑到 $K_s+K_d=K$，且传输层端口数量 K 可以为 6 万的量级，所以，这种情况下所支持的并发关联数目虽然少于集中管理情况下，但仍是相当可观的。以 $K_s=K_d=30\,000$ 为例，则系统支持 9×10^8 个关联，此时，系统最多可以支持 3 万个节点。

3.5　数据传输流程

以图 2 为例，给出 A 与 B 之间进行数据传输的流程如下：

(1)A 向 B 发出数据传输请求；

(2)B 从自己的 K_s/N 个源端口号中随机选择一个端口作为本关联的发送实体端口 p_B，记当前与 p_B 建立了关联的所有端口集合为 Q，并从集合 P_d-Q 中选择一个端口作为本关联的接收实体端口 p_A，若 P_d-$Q=\phi$，则重新执行本步骤，直到得到 p_A。B 将该发送、接收端口对(p_B，p_A)回复给 A，A 在后续建立关联的时候就使用 p_A；

(3)A 向所有候选中转节点发起查询，询问它们到 B 的路径时延；

(4)根据收到反馈报文的时延，A 得出自己到候选中转节点的路径时延，与报文中携带的时延相加，A 得到重叠路径的时延。A 择优选择分别经由 C 和 D 的重叠路径进行传输；

(下转 65 页)