

根节点 A, 由于 A 的子节点已经达到自身上限值 3, 然后 N 在 A 的 3 个子节点 (B、C、D) 中寻找最优的节点 D. 但是由于 D 的子节点数目也达到上限值 2,

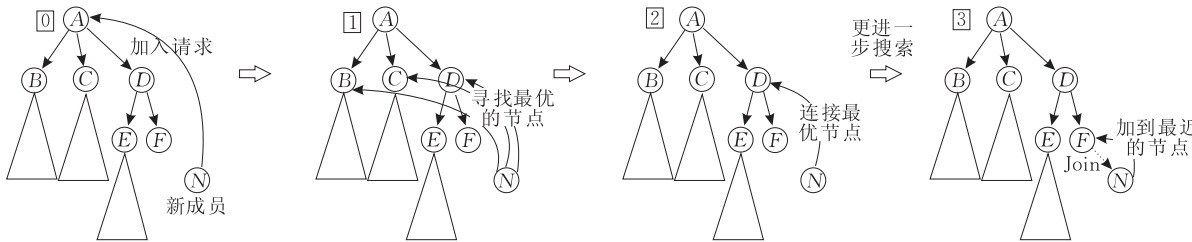


图 7 新成员加入过程

在组播树形成的过程中, 需要保证不产生回路并且不违背子节点数目限制, 以保证组播树结构的稳定. Yoid 协议中采用的是回路检测和避免 (loop detection and avoidance) 机制, 而 HMTP 协议采用的是环路检测和消除 (loop detection and resolution) 策略, 这是两者的不同之处. 此外, Yoid 协议中的覆盖控制拓扑的建立是通过组播共享树中的每个节点随机选择一些非邻居节点交互状态信息实现的, 而在 HMTP 协议中, 成员节点周期性地获得组播树其它部分成员节点的状态信息来提高组播树的鲁棒性, 不会特别产生覆盖控制拓扑.

在树优先组播协议中, 一方面在节点的加入和退出过程中都需要考虑环路检测的问题, 需要增加大量的控制信息, 相应增加了协议的复杂性. 另一方面, 由于网络中始终无法避免由于节点成员的意外离开或者失效造成拓扑结构的突变和传输的暂停, 而树优先组播协议往往不能提供足够的状态信息和健壮的恢复机制来快速修复组播树, 因此可靠性成为了制约其大规模应用的瓶颈.

(3) 隐含式组播协议: NICE^[26], Can-multicast^[33], Scribe^[34] 和 Bayeux^[35].

针对网优先和树优先组播协议的优缺点, 基于可靠性、扩展性、控制开销之间性能和效率的平衡问题, 学术界又提出了一系列隐含式组播协议.

4.1.1 NICE 协议

NICE 协议核心思想是在组播树的基础上“分层”(Hierarchical)和“分簇”(Cluster). 如图 8, 不同于树优先或者网优先协议, 组成员被组织为层次控制拓扑. 从整体上看层次结构是树状的, 但从局部 (每层内节点构成的簇) 又是以网状结构组织的. 由于成员只和少量固定数目的节点联系, 控制开销不大, 同时考虑网络异构的特点, 由选择的领导节点负责簇内的管理和数据分发, 从而很好地融合了网优

于是 N 又递归搜索到 D 的下一层子节点 F 最优并且 F 子节点数目没超过上限, 最终 N 便加入成为 F 的子节点.

先和树优先两种思想的优点, 一定程度上兼顾了网络异构、可扩展性和鲁棒性.

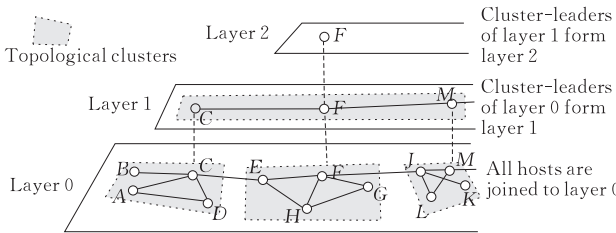


图 8 NICE 层级结构

但我们看到在 NICE 协议中, 由于本质上仍采用组播树的拓扑结构, 处在高层 (比如第 i 层) 的成员, 也是所有相对低层 (从 $0 \sim i-1$) 的领导节点, 这样高层的领导节点往往容易形成系统瓶颈.

4.1.2 CAN-Multicast 协议

P2P 技术最新的研究集中在基于分布式散列表 DHT (Distribute Hash Table) 的分布式发现和路由算法方面. 此类算法通过分布式散列 HASH 函数, 将输入的关键字映射到某个逻辑节点上, 然后通过路由算法查找该节点, 避免了借助中央服务器或者利用广播进行洪泛查找, 可以提高路由的效率, 减少路由表容量和路由延时, 克服了非结构化拓扑中的路由查找存在极大的不可扩展性的问题.

基于 DHT 的结构化拓扑结构能够自适应节点的动态加入/退出, 有着良好的可扩展性、鲁棒性、结点 ID 分配的均匀性和自组织能力精确的发现机制, 正好为应用层组播的实现提供了良好的底层平台. 采用 DHT 结构的应用层组播协议, 一方面能够实现节点的动态加入和退出, 保证节点之间的均匀性和自组织能力. 另一方面也能够实现对目标节点的快速路由发现和路由查找, 减少状态维护开销和转发开销, 比较典型的例子是 CAN, Pastry 和 Tapes-try, 它们之间的差别在于具体的路由策略和发现方

式不同。

在内容可寻址网络 CAN(Content-Addressable Network)协议中,所有的成员节点被组织成一个虚拟的 d 维笛卡尔坐标空间(可以看作一个新的逻辑网络),同时在这个新的逻辑网络上使用散列函数对成员节点进行重新编址(比如对关键值 key 进行散列计算)。由于这个逻辑网络是通过笛卡尔坐标空间的方式构造的,根据坐标值就可以根据最近相邻原则进行路由查找。所以,通过选择合适的散列函数,我们可以对坐标空间的进行合理的分配和管理,实现插入、查询和删除等功能。在文献[33],Ratnasamg 等提出了基于 CAN 架构的应用层组播模式。

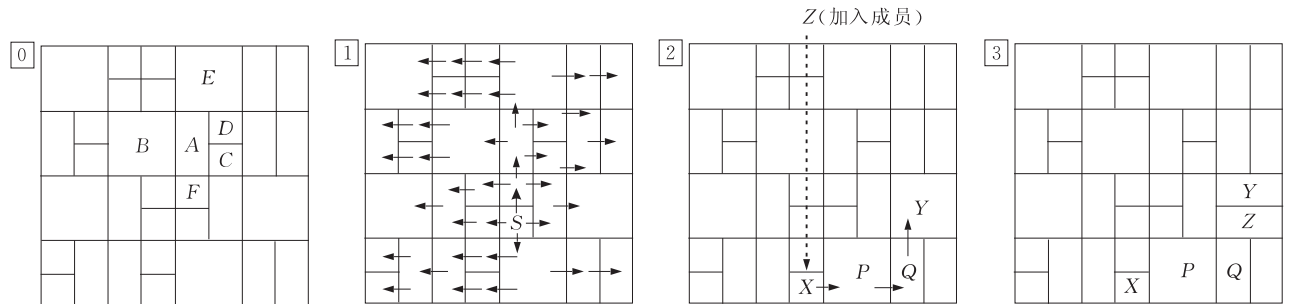


图 9 CAN 坐标空间示意图

在 d 维笛卡尔坐标空间中,任何一个节点的邻居节点只有 $2d$ 个邻居节点,因此它只需要保留最多 $2d$ 个邻近路由信息,维护的路由表信息和网络规模无关,维护开销很小。而数据拓扑结构隐含在控制信息的洪泛过程中。此外,针对笛卡儿坐标空间的路径长度能控制在 $O(d \times N^{1/d})$ 的特点,通过增加维数 d ,可以进一步降低路径长度,降低网络延迟,同时随着维数的增加,邻居节点也相应增加,鲁棒性也得到了提高。

由于加入过程中坐标区域的选择是比较随机的,忽略了对成员节点间相对距离的影响,因此成员节点的分布也没有规律,造成覆盖拓扑网络伸张度(stretch)过大。为此作者又提出了“分布式储存”(distributed binning)的改进思路,使得路径距离较近的节点分配临近的区域空间,以此降低覆盖网络的伸张度。

4.1.3 Scribe 协议

Pastry^[36]作为可扩展的分布式对象定位和路由协议,是另一种典型的 DHT 路由协议,由位于英国剑桥的微软研究院和莱斯(Rice)大学提出,可用于构建大规模的 P2P 系统,而 Scribe 系统正是建立在底层 Pastry 网络上基于主题(topic-based)的大规模

一个新节点要加入 CAN,关键是采用散列函数对 $(key, value)$ 中的 key 进行散列运算,找到其坐标空间中对应的区域,并将 $(key, value)$ 存储在拥有该点所在区域的节点内。如果所对应的区域已经被占用,则已存在的成员节点分割其所在子块的区域空间,把其中的一半区域分配给新加入成员节点。图 9 所示是一个由 34 个成员构成的二维坐标空间,被相应分成了 34 个区域(zone)。当节点 Z 欲加入 CAN 时,首先通过引导程序找到一个已存在的节点 X ,通过 X 路由随机找到属于节点 Y 的坐标区域空间,然后分割 Y 节点的一半区域给 Z ,最后通知邻接区域 Z 的加入。

发布-预订(publish-subscribe)事件通告系统。

加入 Pastry 网络中的每个节点都会被赋予一个唯一的节点标识(node identifier),节点标识一般通过计算成员公钥或者 IP 地址的 Hash 值得到,标识值每位数字取值范围 $0 \sim 2^b - 1$ (b 为很小的常数)。在 Pastry 形成的覆盖网络中,只要能够知道节点标识,就能够通过路由机制找到路径。

对于 Pastry 中的每个成员节点都有一个路由表,一个邻居节点集合和一个叶节点集合。如图 10 所显示的是标识为 2313 的路由表, b 取 2,每位能取值为 0,1,2,3。路由表中每个矩阵框内的值对应其它节点成员的标识值,2313 作为自身的标识值是隐藏存在的。路由表第 i 排的节点标识值和本节点的标识(2313)具有 $i-1$ 个相同的前缀。图中第 3 排的节点标识 2301、2330 就和 2313 的前两位相同。因此我们可以知道所有节点的路由表一共具有 $\log_2 N$, Pastry 路由的复杂度为 $O(\log_2 N)$,其中 N 表示 Pastry 中成员节点的数目。在具体的路由查询中,如果指定一个目标节点的标识,通过标识前缀最大匹配,节点将会把消息路由到在标识值和目标标识最接近的那个节点。

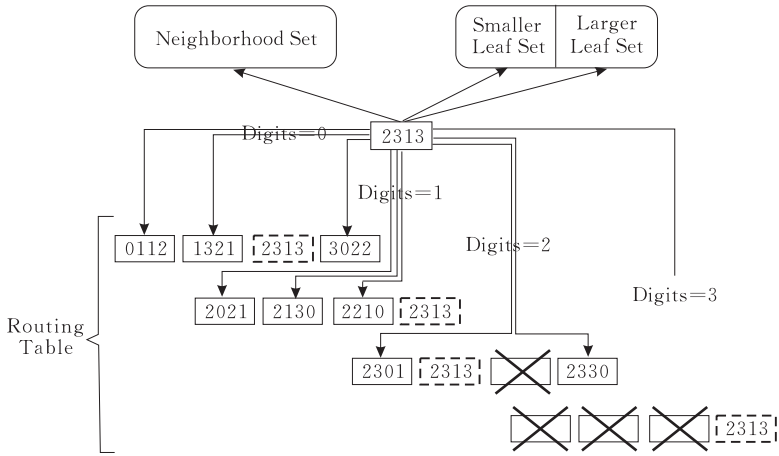


图 10 Pastry 节点路由表

Scribe 应用层组播协议采用 Pastry 网络提供底层路由支持. 因此 Scribe 的控制拓扑和 Pastry 网络的控制拓扑是相同的, 包括了成员节点的路由表信息、邻居节点集合和叶节点集合.

对于数据拓扑结构, 则是 Scribe 协议通过匹配最大前缀的原则, 建立源节点到其它节点的数据传输路径, 即使不能直接查询到目标节点的路径, 总能保证找到一条路径更接近目标节点.

需要指出的是, Scribe 协议中可以支持多个组播组, 因此需要给每个组播组分配一个唯一的标识地址, 也被称为主题标识 (topic identifier). 自身标识和主题标识最近的成员节点成为当前组播组的引导节点, 负责引导其它节点加入组播覆盖网络.

相比 CAN 协议, Pastry^[36] 引入了叶节点集合和邻居节点集合的概念, 能够快速准确地获取路由信息, 大大加快了路由查找的速度.

4. 1. 4 其它应用层组播协议

Bayeux^[35] 应用层组播协议采用的是隐含的方式构造覆盖网络, 底层依靠对等式目标定位系统 Tapestry^[37]. Tapestry 覆盖体系和 Scribe 的底层结构 Pastry 相似, Bayeux 和 Scribe 不同在于组播数据拓扑的产生方式.

Scattercast^[38] 和 Overcast^[39] 协议采用应用层网关方式, 通过部署一些代理节点 (proxy) 组成应用层网络的分布式组播树, 具有比较高的稳定性, 但灵活性比较差.

ALMI^[40] 没有采用分布式设计思想, 而是采用集中式的树优先构造方法. 集中式协议假设服务器知道成员之间的拓扑结构, 然后由服务器按照性能要求构建组播树, 然后转发拓扑信息给相应节点. 由于存在服务器节点性能瓶颈, 鲁棒性很低, 扩展性

不好.

4. 1. 5 应用层组播协议的比较

除了按照创建覆盖网络的方式划分应用层组播协议, 根据构建转发树的策略, 还可以将它们分成集中式和分布式两类: 集中式协议以 ALMI、HBM 为代表; 而分布式协议则有 NICE, Overcast, Yoid, HMTTP, Narada 等等. 由于分布式协议健壮性更高, 可扩展性更好, 因此更多地被采用.

按照覆盖网络成员节点的性质, 可以分为架构式、对等式和混合式协议 3 类: 架构式以 Overcast 为代表, 存在代理节点; 对等式协议以 Narada 为代表, 成员节点的地位相同; 混合式综合上述两者特点, 以 CoopNet 为代表.

一般说来, 除了集中式组播协议因为扩展性上的缺陷外, 其它分布式应用层组播协议都有各自不同的优缺点和适用环境.

流媒体应用有较高的实时需求, 对延迟敏感. 这与协议的最大路径长度有关, 长度越小相应的延迟也会越小. 多媒体应用对网络带宽也有较大的需求, 这与协议的最大子树度数有关, 度越小, 能获得的网络带宽也越大. 另外平均控制开销也希望越小越好.

表 6 给出几类典型的应用层组播协议的性能对比 (N 是组播组成员数, d 是 CAN 组播协议中成员形成的笛卡尔坐标空间的维数), 主要比较最大路径长度、最大子树度、平均控制开销等. 从中可以看出比较适合多媒体应用的应用层组播协议是隐含式方法, 而且在隐含式应用层协议中, 可以看到 CAN 这类分布式结构化协议通过采用多维的标识符空间来实现分布式散列 (DHT) 算法, 具有更加良好的可扩展性, 而且节点维护的路由表信息和网络规模无关, 路径长度也能控制在 $O(d \times N^{1/d})$ 规模上.

表 6 各类应用层组播数据比较

	基本类型	组播树类型	拓扑结构类型	最大路径长度	最大子树度数	平均控制开销
Narada	网优先	特定源端	中心化拓扑	无上界	无上界	$O(N)$
Yoid/HMTP	树优先	共享树	中心化拓扑	无上界	$O(\text{最大节点度数})$	$O(\text{最大节点度数})$
Scribe/Bayeux	隐含式	特定源端	分布式结构化	$O(\log N)$	$O(\log N)$	$O(\log N)$
CAN	隐含式	特定源端	分布式结构化	$O(d \times N^{1/d})$	常数	常数
NICE	隐含式	特定源端	中心化拓扑	$O(\log N)$	$O(\log N)$	常数

4.2 多发送端单接收端方式

在多发送端单接收端传输方式中,考虑到异构网络中的多数 peer 节点性能不稳定,一个发出请求的节点通过接收多个节点发送的数据,以此提高传输的效率和质量.按照具体实现方式不同,又可以分为混合方案(以 CoopNet 为例)和标准方案(以 PROMISE 为例).

4.2.1 混合方案

CoopNet 系统基于中心服务器实现,结合了以 Overcast 为代表的架构式系统和以 Narada 为代表的对等式协议的特点,实现了 Client/Server 和 P2P 两种模式的融合.为了保证整个流媒体系统的健壮性和适应性,一方面采用了多描述视频编码 MDC 技术,另一方面通过在组播成员之间维护多个组播树来实现.

在 CoopNet 系统中,一个或者一组高性能中心服务器负责组织节点建立和管理多个数据分发树,这些数据分发树就构成整个系统的 P2P 对等传输网络.

在一般的流媒体点播服务时,数据分发树所构成的 P2P 对等网络只是为了完善传统的服务器/客户端模式.当服务器没有超过负载阈值时,用户的请求通过服务器直接发送数据来响应.若服务器满载,

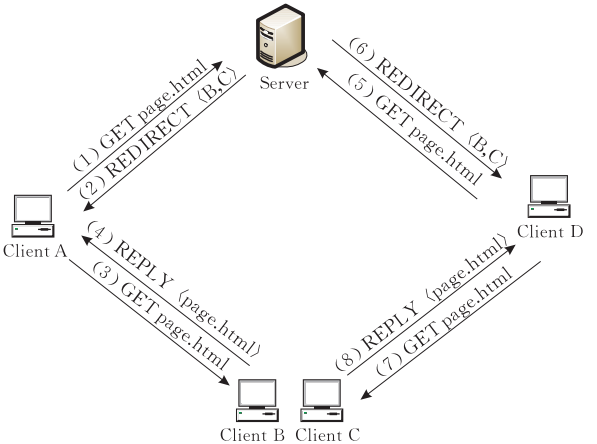


图 11 CoopNet 基本结构

仍有用户发送请求,则服务器通过查询记录找到历史用户列表,选出一定的待选节点转发应答(redirect response)至当前请求的用户,这样发出请求的节点就可以有选择性地与待选节点以 P2P 的方式传输所请求的流媒体内容.

在 CoopNet 系统应用于对实时性要求较高的流媒体直播时,所有参与直播的用户节点构成以多个数据分发树为核心的 P2P 数据拓扑,不同的 MDC 数据流沿着不同分发树路径传输至不同用户节点.通过在网络路径和数据编码两方面引入冗余机制来提高 CoopNet 系统的健壮性.

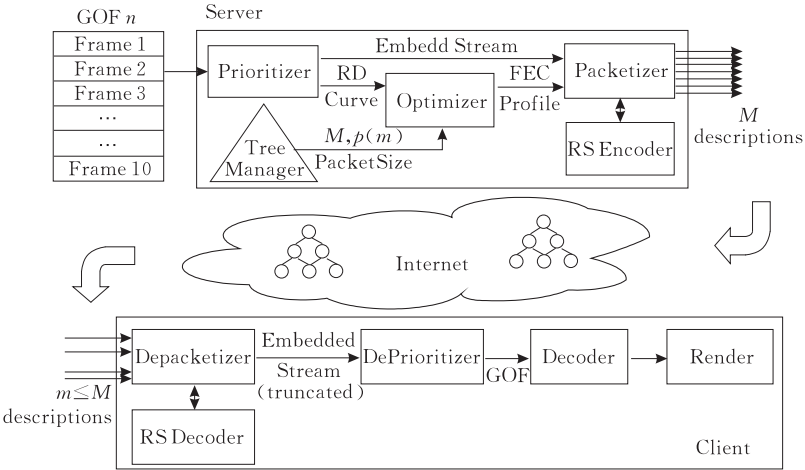


图 12 CoopNet 编码系统

在 CoopNet 系统中,由于需要同时维护多个组播树,控制开销大.同时由于 MDC 采用多个分层编

码,因此还必须考虑解决多路传送时的数据同步问题.

4.2.2 标准方案

相比于 CoopNet 兼有两种传输模式, Promise 系统^[41]属于名副其实的多发送端单接收端的 P2P 流媒体系统. 它的实现独立于底层网络, 可以构建在多种 P2P 底层网络之上(如 Pastry, CAN), 因此灵活性强, 扩展性好.

在 Promise 系统中, 节点之间的连接控制, 对成员节点的管理和对目标节点的查询都由底层网络实现. 当一个节点发出数据请求后, 通过底层网络查询返回一系列满足要求的候选节点, 然后按照基于拓扑(topology-aware)的原则选择传输性能相对较高的节点组成活跃发送集合(active sender set), 其它节点作为备用发送集合(standby sender set). 最后, 由原接收端节点向活跃发送集合的所有节点发起连接, 并行从多个节点接收数据, 如图 13 所示. 在连接建立后, 由接收端来控制每个发送节点的发送速率和数据分配, 发送端只需要按照接收到的控制信息来执行.

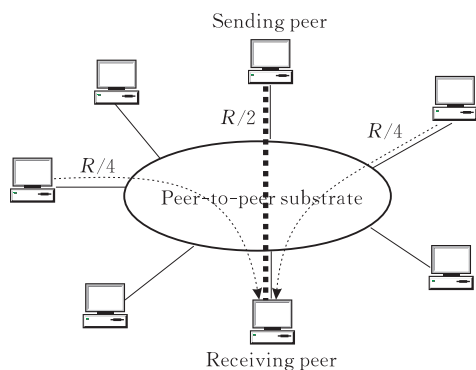


图 13 Promise 结构图

在 Promise 系统中, 当网络或者发送节点出现意外故障时, 接收端通过检测传输的速率判断网络状况和发送节点是否发生故障. 如果是网络的波动造成大范围的传输速率降低, 接收端节点就相应地动态调整总传输率和每个发送端节点的发送速率; 如果认为是发送节点出现故障, 就依据路径原则从备用发送集合选出新的节点进行替换.

同时, Promise 也通过采用前向纠错编码 FEC (Forward Error Correction) 增加视频编码的冗余性, 以此提高健壮性. 具体来说, 就是把视频流分成等长的数据段, 对每段数据进行编码, 接收端对接收到的数据进行纠错处理, 可以一定程度容忍数据包的丢失, 提高信道的传输性能.

4.3 多发送端多接收端方式

作为大规模流媒体应用的解决方案, 单发送多

接收端和多发送单接收端两种模式都具备一定的优点. 但在前两种模式中, 我们可以发现在数据拓扑结构中, 不同的节点(根据发送或接收来区分)的负荷和地位并不完全相同. 比如应用层组播树中的父节点除了接收数据还需要转发数据, 显然比子节点的负荷更大. 所以当父节点成员离开或者失效时, 往往容易造成拓扑结构的突变和传输的暂停. 虽然已经有一些组播树修复算法被提出^[42], 但在动态变化的网络环境中, 组播树的破裂和修补在所难免.

多发送多接收端的 P2P 方式融合了前两种方式的特点, 一定程度克服了前两者的不足. 任何节点既可以接收多个节点的数据, 也可以向多个节点发送数据, 通过建立真正对等互联的体系结构达到了去中心化(decentralize)的效果, 因此被称为纯粹的 P2P 模式(Pure P2P).

在此类协议中, 一个媒体文件会按照一定长度被分为许多块, 块的长度主要由网络环境和客户端性能来决定. 节点通过发送数据请求给中心节点或者通过洪泛至其它节点来获得网络上数据的分布信息, 按照某种规则(比如延时)来选择每个数据块的来源. 而接收到数据请求的节点则负责响应或者转发数据请求. 由于任何一个节点都涉及到同时担任服务器端和客户端, 因此实现机制的设计直接关系到运行的效率.

根据获取数据块信息的获取方式, 我们可以将多发送多接收端协议分成: 中心化获取方式(Centralized Request Way)和分布式获取方法(Decentralized Request Way). 在中心化方式中, 节点无法通过 P2P 网络本身进行目标寻找, 而只能通过目录服务器来查找目标节点. 而在分布式获取方式中, 节点往往通过支持分布式操作的通信协议(如 Gossip 协议)获得其它邻居节点的状态信息, 从而寻找到目标节点.

本节分别介绍多发送端多接收端协议两种方式的特点.

4.3.1 中心化获取方式

作为中心化获取方式的典型代表, BitTorrent 协议是当前最为流行的提供文件和其它内容共享的 P2P 网络协议, 具备了高扩展性、差错容忍性和独立性, 易于部署应用, 得到了大范围的使用.

在典型的 BitTorrent 协议中, 节点通过目录服务器来查找目标节点(这一点与最早的 Napster 系统类似). 当一个节点把文件共享为种子(seed)时, BitTorrent 协议把共享的文件按 256 KB 大小分成

数据块,同时把共享的文件信息发布在目录服务器上.其它对该内容感兴趣的用户节点,只需要点击种子信息,即可通过源节点和所有用户节点构成的 P2P 网络传输数据.其中种子信息包含了所有参与节点已经下载的数据和尚未下载数据的情况.

在用户节点接收数据同时,其已经下载的数据块作为新的种子,同时又可以被其它多个节点下载.这样源文件就通过多个种子的方式分布于整个 P2P 网络之中,即使拥有完整文件的节点离开,只要所有存在节点的数据块能构成一个完整的文件,就能保证每个节点都能获得完整的文件.因此系统的鲁棒性和自适应能力很强,少数节点的故障或者退出都不会对整个结构造成重大影响.对于简单的文件传输,这种策略是容易实现的,且只需要占用较少的网络带宽,因此得到了大范围的推广应用.

但在 BitTorrent 协议中,服务质量(如延迟和抖动)始终很难保证,因此 BitTorrent 协议在对延迟和抖动要求不高的文件共享中显得游刃有余,但不能直接适用于流媒体传输.传统的 CDN 网络通过部署高性能的中心服务器和靠近用户的边缘代理服务器,能够为用户提供高质量的流媒体服务,但限于成本问题,又很难大范围应用.

在文献[43]中,Skevik 等提出了一种混合 BitTorrent 和 CDN 技术的流媒体方案,一方面利用了 P2P 对等网络易于部署,具备高扩展性和鲁棒性,另一方面利用 CDN 的内容分发技术和流量负载均衡技术提供一定的安全保障和满足用户需求的服务质量,能够有效降低主干网络的流量负担.同时提出基于代理的结构(proxy based structure)以解决防火墙对 P2P 网络下载流量的影响.

如图 14,最左边的用户端应用程序(Client app),如视频播放软件,负责播放本地主机(LHC)所接收的视频文件.本地主机和代理服务器,即本地内容缓存 SCC(Site Content Cache)通信,同时从代理服务器和本地网其它主机获得数据,并发送已获得的数据至其它主机.本地的代理服务器又从主内容服务器(main content server)和其它代理服务器获得数据.

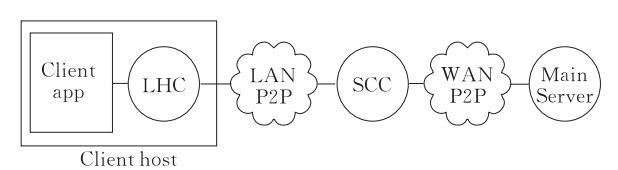


图 14 融合式流媒体方案结构图

在整个结构中,本地的主机在代理服务器的组织下构成一个底层 P2P 网络,所有的代理服务器在主内容服务器的组织下构成一个高层的 P2P 网络.在功能上,代理服务器除了存储容量更大,功能更强,而且还需要根据本地主机的工作情况,进行流量均衡处理,比如缓存注入、缓存替换等.

4.3.2 分布式获取方法

在文献[9]中,作者提出了一种典型属于分布式获取方法的 DONet(Data-driven Overlay Network)协议,通过构建纯粹的 P2P(Peer to Peer)覆盖网络实现数据的传输,无需构建复杂的控制结构.基于 DONet 协议的实时流媒体播放系统 Cool-Streaming,其出色的播放效果、较低的延迟已经在实际运行中得到了证实和肯定.

DONet 的核心思想非常简单:每个节点通过 SCAM(Scalable Gossip Membership protocol)协议周期性和协作节点交互有关数据有效性的信息,从若干伙伴节点那里获得自身没有的数据,同时发送自身拥有的数据给其它需要数据的节点.除了提供节目的数据源节点(origin node),其它节点既可以作为数据接收方也可以作为数据提供方,完全取决于数据的有效信息.由于不需要构造复杂的全局拓扑结构,所以具备很强的可扩展性、高效性和鲁棒性.

从 DONet 的结构图中可以看到其具有 3 个核心模块:(1) 成员管理模块,负责帮助节点获得其它部分覆盖节点的信息.(2) 协作管理模块,负责建立和保存与其它节点的合作信息同时负责周期性地从成员列表中选择一些更优的节点(带宽更高或者有效数据更多)建立协作关系,这样可以提高系统的性能和鲁棒性.(3) 调度模块,采用启发式算法安排视频数据的传输.这 3 个核心模块的设计方法直接关系到整个实时流媒体系统的运行效率.

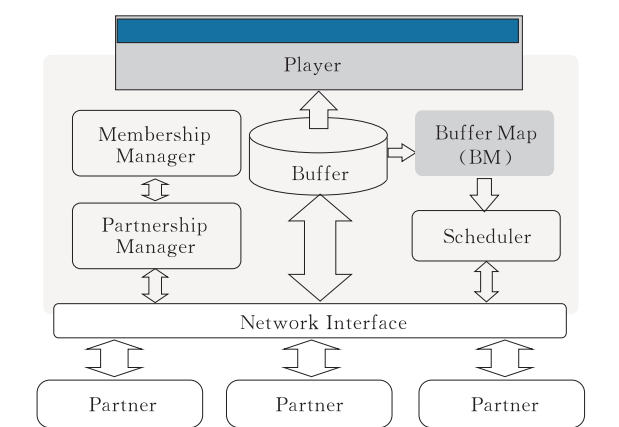


图 15 DONet 系统结构图

在 DONet 协议中,视频都被分成了固定长度的片段.节点缓存中每个片段的有效性通过缓冲图 BM(Buffer Map)来表示.比如采用 120 位来记录 BM,其中 1 表示数据有效,0 表示数据无效,则 120 位的 BM 可以表示 120 个片段的有效信息.节点通过不停地和协作节点交互 BM 获得数据的有效信息,然后确定可以从哪些协作节点获取自身没有的视频片段.

考虑实况流媒体直播对实时性的要求,节点都是半同步的.因此在确定片段长短后,一方面需要节点具有一定的数据缓冲区来缓存一定长度的视频,保证播放的流畅,更重要的就是设计一个高效的调度算法,来满足视频回放的时间要求和节点间的带宽传输限制. DONet 协议中采用的启发式调度算法,通过视频片段的提供者数量来确定优先顺序,然后从少到多来进行处理.针对同一片段的提供者,又按照节点的带宽和延时来确定优先顺序.

根据前文对基于 P2P 技术的流媒体方案的分析可以看到,虽然异构性、网络服务质量和可扩展性的问题得到了一定程度的解决,但安全性问题却始终是研究中的一大难点.首先,由于大规模流媒体服务的成员是频繁变化的,可能每时每刻都有很多的人员加入或退出组播组,这种动态变化下的安全问题是复杂的,同时 P2P 技术的中间节点不可靠性增加了特殊的安全性挑战.其次是高开销的问题,当规模从几个节点到上万个节点甚至更多,保存密钥所占用的节点存储空间、密钥生成所需要的计算量、密钥发送所占用的网络带宽、密钥更新的时间延迟和密钥更新的频率都会相应增加.而且利用基于 P2P 技术的大规模流媒体应用,其对带宽与计算资源的消耗要明显大于基于网络层组播的方案.另外,由于视频流传输的数据量非常大,同时对实时性的要求很高.所以对视频加密时必须考虑到其中的时延敏感性. P2P 技术传输效率低于 IP 层组播,其实现机制本质上是多次单播,时延敏感性问题尤为突出.对于大规模流媒体应用,如何减少密钥传输时延是一个重要问题.此外,现有的多媒体通信通常需要利用转码、应用层码率的自适应控制和速率整形等机制来提高视频通信中的服务质量,加密后的视频由于语法结构的变化可能无法有效地实施上述的操作,所以这就要求加密与数据嵌入算法要保持对多媒体通信中 QoS 控制机制的透明性.

4.4 P2P 和 CDN 的结合

对等网(P2P)技术的出现给解决当前大规模流

媒体应用中的网络与系统瓶颈问题带来新的机遇,现有的 P2P 技术面临的主要问题包括:缺乏中心化的管理和健壮性,其动态性使其缺乏对服务质量的保障;纯粹的 P2P 技术占用了大量骨干带宽资源,耗费大量跨 ISP(电信运营商)的带宽;传统的服务器辅助的 P2P 系统中,松散组织的超级节点的过重负载容易引起 SN 的链式崩溃反应.文献[44]基于大规模 P2P 流媒体系统 CoolStreaming,研究了视频直播系统的工作负载,系统动态性和性能测试,并发现了随机邻居节点选择和多子流能有效解决 P2P 系统动态性问题.在详尽数据结果和理论分析的基础上,作者证明:(1)在视频直播中一个关键问题是在过多用户同时加入(flash crowd)导致的过长初始接入时间和过高的接入拒绝率;(2)系统动态性是影响整体系统性能的最关键参数;(3)不同节点在系统中的上传带宽严重不均衡,极大影响了系统资源的分配;(4)在不同系统参数下,系统需要考虑关键设计的折中.一方面过长的初始接入时间和过高的失效率是基于 P2P 技术的流媒体系统本身带来的问题,这种情况在 NAT 和防火墙后有大量不可利用节点时尤其严重.同时当系统中节点较少时,寻找合适的伙伴节点需要耗用较长的时间.因此在直播中部署适当的服务器节点变得必要^[10,45].但我们不能回避的是纯粹基于服务器模式的方式不具备很好的扩展性,其部署和维护代价昂贵.因此,一个大型的网络视频直播系统需要是一种融合服务器与 P2P 的混和结构.

在产业界作为网络加速技术的 CDN(内容分发网络)得到广泛应用. CDN 采用分布式缓存、负载均衡、流量工程等技术在已有的 Internet 上构筑一个分布式的覆盖网络,通过将内容从信源推送到网络边缘设备,一方面,使得用户得以在“就近”的位置快速访问到所需的内容,降低了端到端时延,提升了用户服务质量;另一方面,突破了中心服务器的性能瓶颈,减轻了骨干网络流量,有效缓解了高吞吐率内容传输对骨干网络的压力,在一定程度上也增加了系统容量.近年来 CDN 得到越来越多的重视并在国内外得到广泛的部署,有代表性的 CDN 服务提供商有 Akamai^①等.

由于 P2P 与 CDN 具有较强的技术互补性,设计新的架构将两者优势相结合,是克服当前大规模

① Akamai. Akamai Technologies, Inc., www.akamai.com, 2008

流媒体解决面临挑战的有效途径. P2P 技术的优点在于低成本、高可扩展性,这是传统 CDN 所欠缺的;而 CDN 的可靠性和可管理性将能解决 P2P 技术许多顽疾. 融合 CDN 与 P2P 的混合式流媒体系统成为大规模应用的体系发展趋势,但将两者结合的研究工作尚处步阶段^{[46-47]①}. 现有研究要么是 1+1 式的 P2P 与 CDN 模式简单相加^[47],要么是服务器支撑 P2P 的模式,它们只能解决大规模流媒体应用所面临的部分问题,不能同时满足服务质量、可扩展性、安全性和异构性需求.

5 安全可扩展的流媒体系统 TrustStream

从 IP 组播、应用层组播到 P2P 技术的发展以及各种技术的综合使用,虽然一定程度上解决了可扩展性的问题,但是这些方案还不能令人满意地解决 QoS、网络异构和安全等问题.

在文献[48-49]中,我们在应用层流媒体组播领域首次将可扩展分层视频编码(PFGS)的可扩展性、可控性与 P2P 的传输思想结合了起来,设计并实现了一套新的安全流媒体系统(TrustStream)和相关的安全应用层组播协议(Secure-ALM)与调度算法.

如图 16,PFGS 编码服务器首先对视频采集的流媒体文件进行编码,产生流媒体基本层和增强层.

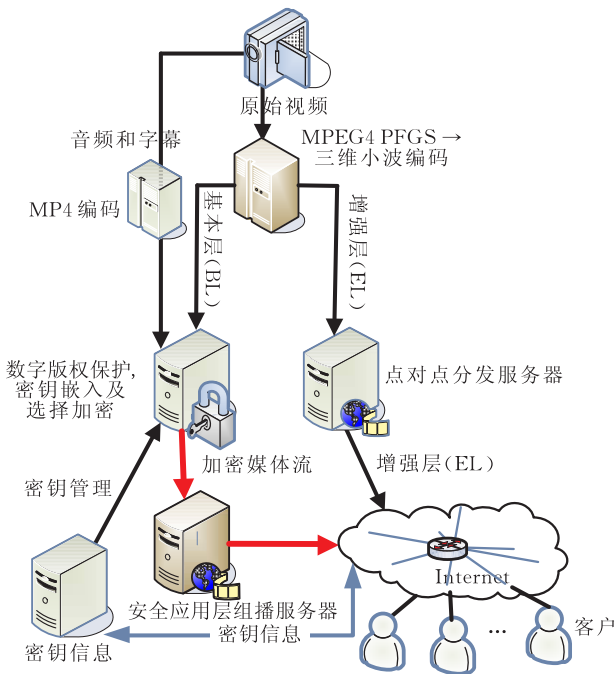


图 16 TrustStream 的系统架构

由于基本层包含了最重要特征的数据,因此需要版权保护及加密服务器对基本层进行加密和版权保护,防止非法用户的接入和非法拷贝. 加密后的流媒体基本层通过单发送多接收端方式的应用层组播传输方式进行传播发布. 对于用作进一步提高视频质量的增强层,直接利用多发送多接收端的 P2P 传输方式进行传播发布.

一方面采用 PFGS 编码,由于利用了多个视频层进行预测,增强了可伸缩性和容错性,能够获得不同输出速率的流媒体,可以很好地适应网络与终端用户的异构性.

另一方面为了实现系统的可扩展性、可控性和安全性,本系统架构创新地提出了针对基本层进行版权保护及选择加密,降低了服务器的负载. 由于 PFGS 的解码是以成功接收基本层为前提,即使未经授权的用户获取到增强层,也无法正常进行解码播放. 同时对增强层采用多发送多接收端的传输方式,进一步增强了系统的可扩展性.

TrustStream 系统不仅解决了大规模流媒体应用中服务器容易成为性能瓶颈和实际应用中的网络与用户异构性问题,更创新地解决了流媒体的加密、密钥分发等一系列安全性问题,对大规模流媒体应用中安全性这一研究的难点问题做出了一定的突破.

6 总结和展望

由于流媒体组播技术可以大幅度提高网络传输的效率,具有良好的应用前景,非常可能成为 Internet 上最受欢迎的应用之一. 但是 Internet 规模庞大,异构性强,视频传输对网络带宽和延迟等 QoS 性能有特殊的要求;同时,流媒体应用中的安全保护和版权保护始终都是非常棘手的问题.

一个完善的流媒体解决方案必须综合编码技术和网络传输技术来解决上述问题的. 一方面,可以结合新的视频编码技术(如可扩展编码 FGS 和多描述编码 MDC 等)和视频加密技术提高视频流对网络异构和安全的支持. 同时,可以综合 CDN 的内容分发技术、网络组播技术到 P2P 技术等网络传输技术进一步提高大规模流媒体系统的可扩展性. 根据网络应用环境和用户需求,我们可以设计不同的流媒体方案. 比如 CoopNet 系统即是为了满足对数据可

靠性要求很高的用户需求,结合 MDC 编码和多组播树的 P2P 技术来实现的典型例子.其它可伸缩编码和最新的 P2P 技术(比如 DHT)的有效结合,成为了流媒体应用的未来方向.

除了 FGS 编码、MDC 编码外,3D 小波编码作为一种可伸缩性的视频编码技术,目前已经成为 MPEG-21 可伸缩视频编码组的重点研究方案.但由于小波编码在处理时需要把整个一帧或一帧中的一大块图像作为一个单元来处理,要占用较大的系统资源,因此需要通过新的技术来改进编码和解码的处理速度使其实用化.

从网络传输技术角度来看,虽然基于分布式散列(DHT)的结构化 P2P 技术得到了迅猛的发展,但要应用于大规模的流媒体应用,我们还面临不小的挑战.目前分布式散列算法一般采用的是 consistency Hash(Consistent Hash),比如 SHA-1 Hash 函数.这类一致性 Hash 函数虽然能够兼顾负载均衡和一定安全保障,但却存在明显的缺陷.比如在构建逻辑结构的时候,并没有太好的方法解决物理地址和逻辑地址不一致的情况,因此在一定程度上降低了在大规模流媒体方案中的实际效率.同时也不能保证相同类型的流媒体资源能够在物理上邻近存放,很可能出现源两个内容相关度很高的多媒体资源由于 Hash 生成了完全不同的散列值,被存放到了完全随机的两个节点.因此针对应用于流媒体应用和其它 P2P 应用的需要,我们需要进一步研究更适合分布式散列的 Hash 函数,使其能够实现负载均衡和一定安全性的前提下,解决逻辑网络和物理网络的不匹配问题,在一定程度上提高内容和语义的耦合度,这对整个 P2P 技术的发展都是具有深远意义的.

参 考 文 献

- [1] Wu D-P, Hou Y-T, Zhu W, Zhang Y-Q, Peha J. Streaming video over the internet: Approaches and directions. *IEEE Transactions on Circuits System Video Technology*, 2001, 11(3): 282-300
- [2] Li Wei-Ping. Overview of fine grannlarity scalability in MPEG-4 video standard. *IEEE Transaction on Circuits and System for Video Technology*, 2001, 11(3): 301-317
- [3] Wu F, Li S, Zhang Y-Q. A framework for efficient progressive fine granularity scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001, 11(3): 282-300
- [4] Wang Y, Lin S. Error-resilient video coding using multiple description motion compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, 12(6): 438-453
- [5] Deering S, Cheriton D. Multicast routing in datagram inter-networks and extended LANs. *ACM Transactions on Computer Systems*, 1990, 8(2): 85-110
- [6] Day M, Gilletti D. Distribution network peering scenarios. *IETF Internet-Draft*, 2001-05
- [7] Chu Yang-Hua, Ra Sanjay, Seshan Srinivasan, Zhang Hui. Enabling conferencing applications on the Internet using an overlay multicast architecture//*Proceedings of the ACM SIGCOMM*. San Diego, 2001: 55-67
- [8] Chu Yang-Hua, Rao Sanjay, Zhang Hui. A case for end system multicast//*Proceedings of the ACM Sigmetrics*. Santa Clara, CA, 2000: 1-12
- [9] Zhang X, Liu J, Li B, Yum T-SP. CoolStreaming/DoNet: A data-driven overlay network for peer-to-peer live media streaming//*Proceedings of the IEEE INFOCOM'05*. Miami, FL, 2005: 2102-2111
- [10] Li B, Yin H. The peer-to-peer live video streaming in the Internet: Issues, existing approaches and challenges. *IEEE Communications Magazine*, 2007, 45(6): 94-99
- [11] Heising G, Marpe D, Cycon H L, Petukhov. A Proposal for ITU-T H.26L: A wavelet-based video coding scheme using OBMC and image warping prediction. *ITU-T*, Mosterey, 1999
- [12] Bolot J-C, Turletti T. A rate control mechanism for packet video//*Proceedings of the IEEE INFOCOM'94*. Toronto, Canada, 1994: 1216-1223
- [13] Bolot J-C, Turletti T, Wakeman I. Scalable feedback control for multicast video distribution in the Internet//*Proceedings of the ACM SIGCOMM'94*. London, UK, 1994: 58-67
- [14] Li X, Ammar M H. Bandwidth control for replicated-stream multicast video distribution//*Proceedings of the HPDC'96*. Syracuse, USA, 1996: 6-9
- [15] McCanne S, Jacobson V, Vetterli M. Receiver-driven layered multicast//*Proceedings of the ACM SIGCOMM'96*. California, USA, 1996: 117-130
- [16] Vickers B, Albuquerque C, Suda T. Source adaptive multi-layered multicast algorithms for real-time video distribution. *IEEE/ACM Transactions on Networking*, 2000, 8(6): 720-733
- [17] Liu J-C, Li B, Zhang Y-Q. An end-to-end adaptation protocol for layered video multicast using optimal rate allocation. *IEEE Transactions on Multimedia*, 2004, 6(1): 87-102
- [18] Shacham N, McKenney P. Packet recovery in high-speed networks using coding and buffer management//*Proceedings of the IEEE INFOCOM'90*. San Francisco, USA, 1990: 124-131
- [19] Biersack E W. Performance evaluation of forward error correction in ATM networks. *Computer Communications Review*, 1992, 22(4): 248-257

- [20] Zhang L, Deering S, Estrin D, Shenker S, Zappala D. RSVP: A new resource reservation protocol. *IEEE Network*, 1993, 7(5): 8-18
- [21] Wu D, Hou Y T, Zhang Y Q. Transporting real-time video over the Internet: Challenges and approaches. *Proceedings of IEEE*, 2000, 88(12): 1855-1875
- [22] Padhye J, Firoiu V, Towsley D, Kurose J. Modeling TCP throughput: A simple model and its empirical validation//*Proceedings of the ACM SIGCOMM 98*. Vancouver, Canada, 1998: 303-314
- [23] Sisalem D, Wolisz A. MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments//*Proceedings of the IWQoS 2000*. Pittsburgh, USA, 2000: 65-74
- [24] Kwon Gu-In, Byers J. Smooth multirate multicast congestion control//*Proceedings of the IEEE INFOCOM'03*. San Francisco, USA, 2003: 1022-1032
- [25] Yin Hao, Lin Chuang, Qiu Feng, Liu Jiang-Chuan, Min Ge-Yong, Li Bo. CASM: A content-aware protocol for secure video multicast. *IEEE Transactions on Multimedia*, 2006, 8(2): 270 - 277
- [26] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast//*Proceedings of the ACM SIGCOMM*. Pittsburgh, USA, 2002: 43-51
- [27] Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users//*Proceedings of the IEEE Infocom*. New York, USA, 2002: 1366-1375
- [28] Banerjee S, Kommareddy C, Kar K et al. Construction of an efficient overlay multicast infrastructure for real-time applications//*Proceedings of the IEEE INFOCOM*. San Francisco, 2003: 1521-1531
- [29] Deshpande H, Bawa M, Garcia Molina H. Streaming live media over a peer-to-peer network. Stanford University, Stanford, CA, USA: Technical Report, 2001
- [30] Padmanabhan V, Wang H, Chou P et al. Distributing streaming media content using cooperative networking//*Proceedings of the ACM NOSSDAV*. Miami, 2002: 177-186
- [31] Castro M, Druschel P, Kermarrec A M et al. SplitStream: High-bandwidth content distribution in a cooperative environment//*Proceedings of the IPTPS'03*. Berkeley, 2003: 292-303
- [32] Banerjee S, Bhattacharjee B. A comparative study of application layer multicast protocols. University of Maryland, College Park, MD, USA: Technical Report, 2002
- [33] Ratnasamy S, Handley M, Karp R. Application-level multicast using content-addressable networks//*Proceedings of the 3rd International Workshop on Networked Group Communication*. London, UK, 2001: 14-29
- [34] Castro M, Druschel P, Kermarrec A-M, Rowstron A. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications (JSAC)*, 2002, 20(8): 1489-1499
- [35] Zhuang S Q, Zhao B Y, Joseph A D, Katz R H, Kubiatiowicz J D. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination//*Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*. New York, USA, 2001: 11-20
- [36] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems//*Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*. Heidelberg, Germany, 2001: 329-350
- [37] Zhao B Y, Kubiatiowicz J, Joseph A. Tapestry: An Infrastructure for fault-tolerant wide-area location and routing. University of California, Berkeley, CA, USA: Technical Report UCB/CSD-01-1141, 2001
- [38] Chawathe Y. Scattercast: An architecture for Internet broadcast distribution as an infrastructure service [Ph.D. dissertation]. University of California, Berkeley, CA, USA, 2000
- [39] Jannotti J, Gifford D K, Johnson K L, Kaashoek M F, James W O'Toole, Jr. Overcast: Reliable multicast with an overlay network//*Proceedings of the OSDI*. San Diego, 2000: 197-212
- [40] Pendarakis D, Shi S, Verma D et al. ALMI: An application level multicast Infrastructure//*Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*. California, USA, 2001: 49-60
- [41] Hefeeda M, Habib A, Botev B, Xu D, Bhargave B. PROMISE: Peer-to-peer media streaming using CollectCast//*Proceedings of the ACM Multimedia 2003*. Berkeley, CA, 2003: 45-54
- [42] Yang M, Fei Z. A proactive approach to reconstructing overlay multicast trees//*Proceedings of the INFOCOM'04*. Hong Kong, China, 2004: 2743-2753
- [43] Skevik K-A, Goebel V, Plagemann T. Analysis of BitTorrent and its use for the design of a P2P based streaming protocol for a hybrid CDN. Department of Informatics, University of Oslo, Norway: Technical Report, 2004
- [44] Li B, Xie S-S, Keung G Y, Liu J-C, Stoica I, Zhang H. An empirical study of the CoolStreaming+System. *IEEE Journal on Selected Areas in Communications, Special Issue on Advances in Peer-to-Peer Streaming System*, 2007, 25(9): 1-13
- [45] Li B, Xie S-S, Qu Y, Keung G, Liu J-C, Lin C, Zhang X-Y. Inside the new coolstreaming: Principles, measurements and performance implications//*Proceedings of the IEEE Infocom'2008*. Phoenix, AZ, 2008
- [46] Tu Y, Sun J, Hefeeda M, Prabhakar S. An analytical study of peer-to-peer media streaming systems. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2005, 1(4): 354-376
- [47] Xu D, Kulkarni S, Rosenberg C, Chai H. A CDN-P2P hybrid architecture for cost-effective streaming media distribution. *Computer Networks*, 2004, 44(3): 353-382

[48] Yin H, Lin C, Qiu F, Liu X, Wu D. TrustStream: A novel secure and scalable media streaming architecture//Proceedings of the 13th ACM International Conference on Multimedia. Singapore, 2005: 295-298

[49] Yin Hao, Lin Chuang, Zhang Qian, Chen Zhi-Jia, Wu Da-Peng. TrustStream: A secure and scalable architecture for large-scale Internet media streaming. IEEE Transactions on Circuits and Systems for Video Technology, 2008



YIN Hao, born in 1974, Ph. D. , associate professor. His research interests include performance evaluation for Internet and wireless network, image/video coding, multimedia over wireless network, and security.

LIN Chuang, born in 1948, Ph. D. , professor. His current research interests include computer networks, performance evaluation, logic reasoning, and Petri net theory and its applications.

WEN Hao, born in 1983, Ph. D. candidate. His research interests include modeling and performance analysis for wireless network.

CHEN Zhi-Jia, born in 1981, Ph. D. candidate. His research area is in computer network and media streaming, especially for architecture design and analytical modeling for P2P media streaming system.

WU Da-Peng, Ph. D. , assistant professor. His research interests are in the areas of networking, communications, multimedia, signal processing, and information and network security.

PFWRR: 能实现比例公平的增强型 WRR

王胜灵¹⁾ 侯义斌²⁾ 黄建辉¹⁾ 黄樟钦²⁾

¹⁾(西安交通大学电子与信息工程学院 西安 710049)

²⁾(北京工业大学软件学院 北京 100022)

摘 要 为了实现比例公平原则,在加权轮循调度(WRR)算法的基础上提出了比例公平 WRR 调度算法——PFWRR. PFWRR 依据各队列的平均分组到达率,调整各队列的调度权值,从而在当队列长度小于等于缓冲长度时,保证各队列的平均分组排队时延符合给定比例;当队列长度大于缓冲时,保证各队列的平均分组丢失率符合给定比例. PFWRR 的计算负荷是合理的,因为它仅当系统超载且平均分组到达率发生变化时,才调整各队列的服务率. 实测性能显示:当系统超载且不出现分组丢失时,PFWRR 实现了比例平均分组排队时延保证,当系统出现分组丢失时,PFWRR 实现了比例平均分组丢失率保证.

关键词 加权轮循调度;比例公平;平均分组排队时延;平均分组丢失率;平均分组到达率
中图法分类号 TP393

PFWRR: An Enhanced WRR Scheduling Realizing the Proportion Fairness Principle

WANG Sheng-Ling¹⁾ HOU Yi-Bin²⁾ HUANG Jian-Hui¹⁾ HUANG Zhang-Qin²⁾

¹⁾(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049)

²⁾(College of Software Engineering, Beijing University of Technology, Beijing 100022)

Abstract To realize proportion fairness principle, a scheduling based on Weighted Round Robin (WRR) is proposed, whose name is Proportion Fairness WRR (PFWRR). PFWRR adjusts each queue's weight in terms of the average packet arrival rate to guarantee that the proportion of average queuing delay accords with the differentiation parameters when each queue's length is smaller than or equals to its buffer length and the proportion of average packet loss rate accords with the differentiation parameters when the system begins to drop packets. The computation load of PFWRR is reasonable because it renews each queue's weight only when the system is overloaded and the average packet arrival rate changes. The experiment shows that PFWRR can realize the average queuing delay proportion fairness principle when the system is overloaded but do not drop packets and the average packet loss rate proportion fairness principle when packet loss happens in the system.

Keywords weighted round robin; proportion fairness; the average packet queuing delay; the average packet loss rate; the average packet arrival rate

收稿日期:2006-11-06;最终修改稿收到日期:2007-08-05. 本课题得到北京市教育委员会科技发展计划重点项目基金(KZ2005100005006)、国家自然科学基金(90407017,60403035)及国家“九七三”重点基础研究发展规划项目基金(2007CB307105,2007CB307100)资助. 王胜灵,女,1978年生,博士研究生,主要研究方向为移动 IPv6、移动性管理、服务质量. E-mail: slwang@mail.xjtu.edu.cn. 侯义斌,男,1952年生,博士,教授,博士生导师,主要研究领域为移动 IPv6、人机交互. 黄建辉,男,1977年生,博士研究生,主要研究方向为移动 IPv6、服务质量、人工智能. 黄樟钦,男,1964年生,博士,教授,主要研究领域为无线局域网关键技术.

1 引言

为了向用户提供服务质量(QoS)保证,互联网工程任务组提出了三种 QoS 模型:集成服务^[1]、差分服务^[2]和多协议标签交换^[3].集成服务和多协议标签交换均是面向连接的,可扩展性差,不适用于大规模网络中;而差分服务是面向无连接的,总体上呈现分布式管理和松散控制的特点,因此在设计、规划和扩充方面都很灵活,这些优点使得差分服务成为近年来网络 QoS 的研究热点.

为了实现差分服务,Dovrolis^[4]提出使用分组调度的方法来实现该 QoS 模型的比例公平原则,其目的是使网络资源按照供应者给定的区分参数按比例地分配,使每个业务类的 QoS 随着负载变化,但类间的 QoS 比值不变,与负载无关.若用 q_i, q_j 代表业务类 i, j 的性能量值,则比例公平原则给每对业务类加入如式(1)的限制,式中, $c_i, c_j (i, j = 1, 2, \dots, N)$ 是 QoS 区分参数, N 是系统提供的业务类别总数.

$$q_i/q_j = c_i/c_j, \quad i, j = 1, 2, \dots, N \quad (1)$$

分组调度的任务是按照既定的策略从待发送分组中选择一个发往输出链路.近年来,主流的调度算法有两大类:基于通用处理机共享(GPS)的调度算法^[5-7]和基于轮循的调度算法^[8-9].前者能体现业务的优先级,而后者能保证业务的公平性,由于这两类算法都无法同时保证业务的优先级和公平性^[10],因此要实现比例公平原则,需要对现有的调度算法进行改进.

WRR-PAD^[11]提出了一种能保证比例平均分组排队时延的方案,该方案在实现时,需要对到来的每个分组打上时间戳,并计算每个测量时段内发送的分组数和每个分组的时延,同时, WRR-PAD 的仿真显示,当参数设置不当时,所提方法可能失效. APDDS^[12]通过动态地调整各队列的瞬时优先级,然后根据各队列的首分组的到达时间和当前时间,确定在系统当前时间各队列的优先级,调度器每次调度时需要比较所有队列的优先级,从而选择优先级最高的队列服务,最终使各队列的平均排队时延比例符合给定的参数. PAD^[4]定义了归一化的平均时延 $\hat{d}_i (\hat{d}_i = \bar{d}_i/\delta_i (i = 1, 2, \dots, N))$, 其中 $\bar{d}_i (i = 1, 2, \dots, N)$ 是第 i 个队列的平均分组排队时延,而 $\delta_i (i = 1, 2, \dots, N)$ 是该队列的平均分组排队时延比例. PAD 在每次调度时,选择归一化的平均时延最大的

队列进行服务,从而实现比例平均分组排队时延公平原则.此外, PLR^[13]提出了一种能保证比例平均分组丢失率的方案,定义了归一化的分组丢失率 $\bar{l}_i (\bar{l}_i = \bar{l}_i/\sigma_i (i = 1, 2, \dots, N))$, 其中, $\bar{l}_i (i = 1, 2, \dots, N)$ 是第 i 个队列的平均分组丢失率,而 $\sigma_i (i = 1, 2, \dots, N)$ 是该队列的平均分组丢失率比例. PLR 每次选择归一化的分组丢失率最小的队列实施分组丢弃,从而实现比例平均分组丢失率公平原则.

本文基于 WRR 调度算法,提出了一种实现比例公平原则的方案——增强型 WRR (PFWRR). PFWRR 将系统时间分为等距的时段,根据时段内各队列的平均分组到达率计算出各队列的服务率,即各队列的权值,从而在队列长度不超过缓冲时,能提供比例平均分组排队时延保证,在队列长度超过缓冲时,能实现比例平均分组丢失率保证.虽然为了实现比例公平原则, PFWRR 为系统引入了一些计算负荷,但与同类方案相比, PFWRR 的实现更为简单,因为它无需为到来的每个分组打时间戳,计算每个分组的时延,无需对分组进行排序等操作.在 PFWRR 中,只要系统不超载, PFWRR 不会引入额外的处理负荷,且仅当分组的平均到达率发生变化时,才会更新队列的权值.如果说同类方案是基于每个分组的特性来实现比例公平原则,那么 PFWRR 则是基于每个流的特性来实现比例公平原则,这使得 PFWRR 比同类方案更适用于大规模的高速网络.

2 PFWRR

由于平均分组排队时延和平均分组丢失率是衡量调度算法的重要参数,因此在本文中,当队长小于等于缓冲长度时,以平均分组排队时延为性能参数;当队长大于缓冲长度时,即网络需要丢弃分组时,以平均分组丢失率为性能参数.

2.1 以平均分组排队时延为性能参数的情况

假设网络提供 N 类业务,第 $i (i = 1, 2, \dots, N)$ 类业务的分组到达率服从参数为 $\lambda_i (i = 1, 2, \dots, N)$ 的泊松分布,于是根据 Little 公式,第 $i (i = 1, 2, \dots, N)$ 类业务的平均分组排队时延 $W_i (i = 1, 2, \dots, N)$ 为

$$W_i = \bar{Q}_i / \lambda_i (1 - l_i), \quad i = 1, 2, \dots, N \quad (2)$$

式(2)中, $\bar{Q}_i (i = 1, 2, \dots, N)$ 为第 i 类业务的平均队长, $l_i (i = 1, 2, \dots, N)$ 为第 i 类业务的分组丢失率.当队长小于等于缓冲长度时, $l_i (i = 1, 2, \dots, N) = 0$,

即有

$$W_i = \bar{Q}_i / \lambda_i, \quad i=1, 2, \dots, N \quad (3)$$

基于平均分组排队时延的比例公平原则要求各类业务的平均分组排队时延满足:

$$W_1 : W_2 : \dots : W_N = d_1 : d_2 : \dots : d_N \quad (4)$$

式(4)中, d_1, d_2, \dots, d_N 是各类业务流的平均分组排队时延的比值, 由式(3)和式(4)可得

$$\bar{Q}_1 : \bar{Q}_2 : \dots : \bar{Q}_N = d_1 \lambda_1 : d_2 \lambda_2 : \dots : d_N \lambda_N \quad (5)$$

将系统时间划分为长度为 t 的若干时段, 设 Q_{ij} ($i=1, 2, \dots, N; j=1, 2, \dots$) 为第 i 个队列在 j 时段的长度, 若式(6)满足, 则式(5)成立.

$$Q_{1j} : Q_{2j} : \dots : Q_{Nj} = d_1 \lambda_1 : d_2 \lambda_2 : \dots : d_N \lambda_N, \quad j=1, 2, \dots \quad (6)$$

又因为

$$Q_{ij} = Q_{i(j-1)} + \lambda_i t - w_i, \quad i=1, 2, \dots, N; j=1, 2, \dots \quad (7)$$

式(7)中, w_i ($i=1, 2, \dots, N$) 为队列 i ($i=1, 2, \dots, N$) 在 t 内被服务的分组数. 由式(6)、式(7)以及系统的处理能力, 可得方程组(8), 其中, P 是调度器在 t 时段内最大可服务的分组总数. 这样, 通过求解方程组(8), 可求出队列 i ($i=1, 2, \dots, N$) 在 t 时段内被服务的分组数 w_i ($i=1, 2, \dots, N$), 并将其作为队列 i ($i=1, 2, \dots, N$) 的权值, 从而最终实现分组的比例平均排队时延保证.

$$\begin{cases} (Q_{1(j-1)} + \lambda_1 t - w_1) : \dots : (Q_{N(j-1)} + \lambda_N t - w_N) = \\ d_1 \lambda_1 : d_2 \lambda_2 : \dots : d_N \lambda_N \\ w_1 + w_2 + \dots + w_N = P \end{cases}, \quad j=1, 2, \dots \quad (8)$$

假设系统的缓冲无限长, 不存在分组丢弃, 同时令 $d_i \lambda_i = k_i$ ($i=1, 2, \dots, N$), 以系统提供三个队列为例, 来分析式(8). 此时, 方程组变为

$$\begin{cases} k_2 w_1 - k_1 w_2 = k_2 Q_{1(j-1)} - k_1 Q_{2(j-1)} + k_2 \lambda_1 t - k_1 \lambda_2 t \\ k_3 w_2 - k_2 w_3 = k_3 Q_{2(j-1)} - k_2 Q_{3(j-1)} + k_3 \lambda_2 t - k_2 \lambda_3 t \\ w_1 + w_2 + w_3 = P \end{cases}, \quad j=1, 2, \dots \quad (9)$$

由于除 $Q_{10} = Q_{20} = Q_{30} = 0$ 外, 当各队列的平均分组到达率不发生变化时, $Q_{1(j-1)} : Q_{2(j-1)} : Q_{3(j-1)} = k_1 : k_2 : k_3$ ($j=2, 3, \dots$), 于是, 可将式(9)化简为

$$\begin{cases} k_2 w_1 - k_1 w_2 = k_2 \lambda_1 t - k_1 \lambda_2 t \\ k_3 w_2 - k_2 w_3 = k_3 \lambda_2 t - k_2 \lambda_3 t \\ w_1 + w_2 + w_3 = P \end{cases} \quad (10)$$

式(10)说明, 在实现比例平均分组排队时延保

证时, 只要获知各队列的平均分组到达率, 便可调整各队列的服务率 w_i ($i=1, 2, \dots, N$), 从而使各队列的分组排队时延服从给定比例, 且当平均分组到达率不发生变化时, 由于式(10)不变, 因此不用重新计算服务率, 直接使用前次的结果即可. 这个结论对于系统提供多个队列(大于3)时, 同样成立. 由于目前已有文献提出了确定分组到达率的有效方法, 如文献[14-15]等, 因此在本文不再赘述.

2.2 以平均分组丢失率为性能参数的情况

当队列长度大于缓冲长度时, 网络需要丢弃分组, 此时, 基于平均分组丢失率的比例公平原则要求各类业务的平均分组丢失率 l_i ($i=1, 2, \dots, N$) 满足:

$$l_1 : l_2 : \dots : l_N = \bar{l}_1 : \bar{l}_2 : \dots : \bar{l}_N \quad (11)$$

式(11)中, $\bar{l}_1, \bar{l}_2, \dots, \bar{l}_N$ 是各类业务的平均分组丢失率的比值. 令 l_{ij} 为第 i ($i=1, 2, \dots, N$) 个队列在时段 j ($j=1, 2, \dots$) 内的分组丢失率, 若在时段 j ($j=1, 2, \dots$) 内丢弃的分组数为 d_{ij} ($i=1, 2, \dots, N; j=1, 2, \dots$), 则分组丢失率 l_{ij} ($i=1, 2, \dots, N; j=1, 2, \dots$) 为

$$l_{ij} = d_{ij} / (\lambda_i * t) \quad (12)$$

于是: $d_{ij} = \lambda_i * l_i * t$. 当式(13)满足, 则式(11)成立, 进而可实现比例平均分组丢失率.

$$\begin{aligned} d_{1j} : d_{2j} : \dots : d_{Nj} = \\ (\lambda_1 * \bar{l}_1 * t) : (\lambda_2 * \bar{l}_2 * t) : \dots : (\lambda_N * \bar{l}_N * t), \\ j=1, 2, \dots \end{aligned} \quad (13)$$

由于在第 j ($j=1, 2, \dots$) 个时段内, 当队列长度超过缓冲长度 L 时, 丢弃的分组数为

$$d_{ij} = Q_{i(j-1)} + \lambda_i t - w_i - L \quad (14)$$

于是根据式(13)、式(14)和系统的处理能力, 可得到方程组(15). 这样, 在每个时段初, 通过求解方程组(15), 可得到实现比例丢失率公平原则的各队列的权值.

$$\begin{cases} (Q_{1(j-1)} + \lambda_1 t - w_1 - L) : \dots : (Q_{N(j-1)} + \lambda_N t - w_N - L) = \\ (\lambda_1 * \bar{l}_1 * t) : \dots : (\lambda_N * \bar{l}_N * t) \\ w_1 + w_2 + \dots + w_N = P \end{cases}, \quad j=1, 2, \dots \quad (15)$$

上述求解 t 时段内丢失分组个数的方法采用了类似文献[11]中介绍的方法, 但不同之处在于文献[11]通过控制在时段 t 内丢弃的分组数来实现比例丢失率公平原则, 而 PFWRR 采用调整各队列的权值, 即各队列的服务率, 来实现这一原则.

式(15)中, 当各队列均出现了丢弃时, $Q_{i(j-1)} = L$ ($i=1, \dots, N; j=1, 2, \dots$), 于是式(15)可简化为

$$\begin{cases} (\lambda_1 t - w_1) : \dots : (\lambda_N t - w_N) = \\ (\lambda_1 * \bar{l}_1 * t) : \dots : (\lambda_N * \bar{l}_N * t) \\ w_1 + w_2 + \dots + w_N = P \end{cases} \quad (16)$$

式(16)说明:当系统超载到使各队列均需要丢弃分组时,可根据各队列的平均分组到达率,来调节各队列的服务率,从而使各队列的平均分组丢失率服从给定的比例.假设并不是所有队列都出现了分组丢弃,则按照式(15),需要根据各队列的长度以及分组到达率来共同决定队列的服务率,但后续的实验表明,当并不是所有队列均出现分组丢弃时,如果仍然选用式(16)来确定各队列的服务率,则系统能在短时间内迅速对后续时段各队列丢弃的分组配额进行重新分配,使系统做到比例公平.考虑到处理负荷,当某个队列出现丢弃时,PFWRR 仍然采用式(16)求解各队列的服务率.

为动态地求解各队列的服务率,系统启动一个时长为 t 的定时器,每当定时器超时,系统根据各队列的平均分组到达率以及比例性能参数要求,按照上述介绍的方法求解各队列的权值 w_i ($i=1,2,\dots,N$),如图 1 所示.从图中可看出,有两种情况无需重新计算各队列的服务率:①当平均分组到达率小于系统的服务率时,各队列均无排队时延,此时系统可无差别地服务各队列,实现最大限度利用系统资源;②当各队列的平均分组到达率均未发生改变时.由于按照上述方法计算出来的权值可能不是一个整数且数值较大,因此,最终的权值经过了取整和约去最大公约数的处理.图 2 示出了 PFWRR 流程.

```
int i, g;
if(时段定时器超时) {
    if (sum_{i=1}^N \lambda_i t < P) {
        //此时平均分组到达率小于服务率,无排队延迟
        for(i=1; i<=N; i++) w_i = 1;
        return;
    }
    if(各类业务队列长度均小于 L) {
        //此时实现比例平均分组排队时延原则
        if(各队列的平均分组到达率均未发生变化) return;
        else 按照 2.1 节的方法计算 w_i(i=1,2,\dots,N);
    }
    else { //此时实现比例平均分组丢失率原则
        if(各队列的平均分组到达率均未发生变化) return;
        else 按照 2.2 节的方法计算 w_i(i=1,2,\dots,N);
    }
    //函数 gcd() 返回[w_i(i=1,2,\dots,N)的最大公约数
    g = gcd([w_1], [w_2], \dots, [w_N]);
    if(g>1) {
        for(i=1; i<=N; i++) w_i = [w_i]/g;
    }
}
```

图 1 权值的求解

```
int A_1, A_2, \dots, A_N;
bool is_pkt2send = false;
bool is_pktinqueue = false;
for(i=1; i<=N; i++) A_i = w_i;
while(1) {
    for(i=1; i<=N; i++) {
        if((N*Q_i>0) && (A_i>0)) { //N*Q_i为队列 i 的长度
            S(Q_i); //函数 S(Q_i)发送队列 i 的一个分组
            A_i--;
            is_pkt2send = true; //标识出已找到符合发送条件的队列
            break;
        }
        else {
            if(N*Q_i>0) is_pktinqueue = true;
            //表明队列中有数据待发送
        }
    }
    //有数据但没有服务配额
    if((is_pkt2send=false) && (is_pktinqueue=true)) {
        //修改各队列的服务配额,使有数据的队列可以发送数据
        for(i=1; i<=N; i++) A_i += w_i;
    }
    if((A_1=0) && (A_2=0) && \dots (A_N=0)) {
        for(i=1; i<=N; i++) A_i = w_i;
    }
    is_pktinqueue = false;
    is_pkt2send = false;
}
```

图 2 PFWRR 流程

3 性能测试

实验环境如图 3 所示.其中,节点 4 是一个 IBM T42 笔记本电脑,该节点安装了芯片为 prism2.5 的 NetGear 无线网卡,在 Linux 环境下使用 hostap^① 驱动使无线网卡工作在 AP 模式下.节点 1~3 为源节点,分别使用流量发生器产生 3 个不同等级的业务流,这些业务流通过节点 4 到达节点 5,每个业务流的分组长度均为 1000 字节,且业务流的优先级按照 1~3 的顺序逐次降低.虽然节点 4 与节点 5 的理论带宽为 11Mbps,但经过反复测试,节点 4 和节点 5 的之间的实际带宽大约为 6Mbps 左右.此外,节点 1~3 与节点 4 通过 10/100Mbps 有线网络相连,在下面的测试设置中,流发送速率均小于有线网络带宽,节点 1~3 与节点 4 之间不存在通信瓶颈,即它们的通信无分组排队及分组丢失.当业务流 1~3 到达节点 4 时,由节点 4 根据给定的比例参数,按照 PFWRR 算法对各队列的业务流进行调度,实现比例平均分组排队时延或比例平均分组丢失率.为评估 PFWRR 的性能,需要对节点 4 和节点 5 之间的时延进行测试.由于直接测试分组的排队时延比较困难,而排队时延可以由最终的时延反映出来,因此我们以分组的时延性能来考察分组的排队时延

① <http://hostap.epitest.fi/>

性能. 在本文的测试中, $t=1\text{s}$, 当实现比例平均分组排队时延时, 各队列的缓冲长度均设为 5000000 个分组长, 较长的缓冲使得分组无丢失, 当实现比例平均分组丢失率时, 各队列的缓冲长度均设为 500 个分组长. 无论以平均分组排队时延还是以平均分组丢失率为性能参数时, 要求其比例均为 2:3:5.

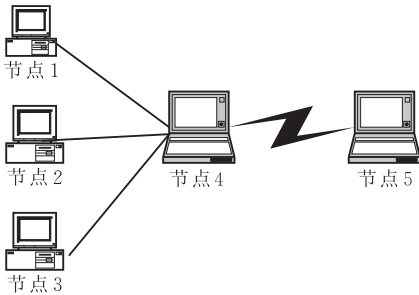
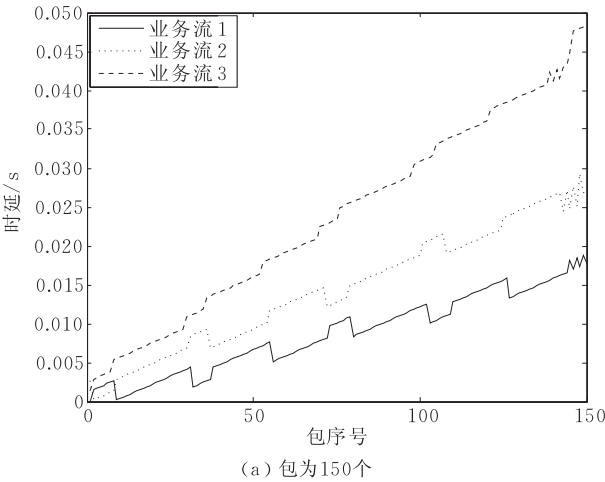


图 3 实验环境

3.1 平均分组排队时延

图 4(a) 显示了当所有业务流均为 2.1Mbps、以



平均分组排队时延为性能参数时, 各业务流的前 150 个分组的时延情况. 由于缓冲长度设置较长, 各业务流均未出现丢失. 从图 4(a) 中可看出, 随着包序号的增加, 各业务流的时延也随之增加, 这是因为在先进先出的队列中, 序号大的分组需等序号小的分组发送完之后才能发送, 所以序号大的分组的时延要比序号小的分组的时延大. 此外, 图 4(a) 显示, 不同业务流的相同序号的分组的时延可能不满足严格的 2:3:5 的关系, 但是从整体的趋势来看, 这种比例关系是满足的, 这一点可以从图 4(b) 中进一步看出. 图 4(a) 和图 4(b) 是同一次测试中的数据, 不同的是, 图 4(a) 仅显示了各业务流的前 150 个分组的时延情况, 而图 4(b) 显示了各业务流的前 2780 个分组的时延, 前者反映了 PFWRR 的短期效果, 而后者反映了 PFWRR 的长期效果. 图 4(b) 说明: 从长期效果来看, PFWRR 在分组无丢失的情况下, 实现了比例平均分组排队时延.

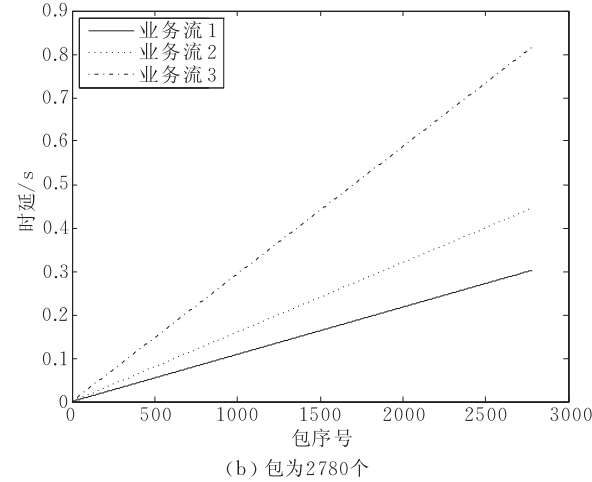


图 4 实现比例平均分组排队时延时各流时延的变化

图 5 显示了各业务流的发送速率从 2.1Mbps 变化到 3Mbps、以平均分组排队时延为性能参数时, 各业务流的前 1.3×10^3 个分组的时延均值情况. 同样, 由于缓冲长度设置较大, 各队列均未出现分组丢弃现象. 从图中可看出, 随着各业务流的发送速率的增加, 其时延也随之增加. 这是因为随着各业务流的分组到达率的不断增加, 有限的系统服务能力也越来越不能及时地处理分组, 从而使得积压在队列缓冲的分组排队时间也越来越长. 此外, 图 5 进一步显示, 各业务流在不同的发送速率下, 其时延大致服从给定的比例, 这使得 PFWRR 在较轻超载或较重超载时, 都能实现比例平均排队时延公平原则. 值得一提的是, 当系统不超载时, 分组无排队时延, 此时的时延完全为线路传输时延, 经测量, 传输时延大

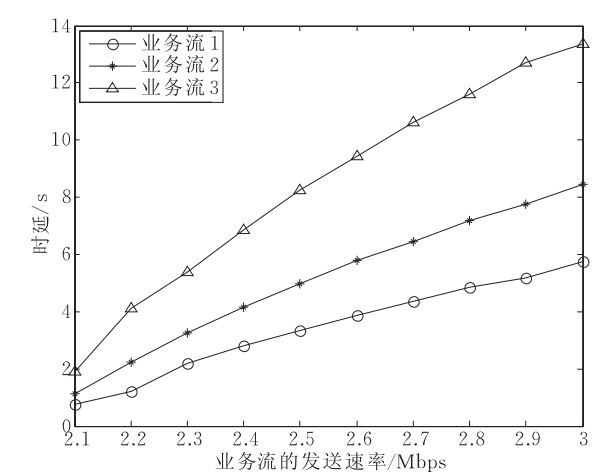


图 5 实现比例平均排队时延时各流的时延随发送速率的变化

约为 1ms 左右;而当系统超载时,分组的时延不仅包含线路传输时延,还包含排队时延,由图 5 显示,分组的时延达到秒级,这说明当系统超载时,排队时延成为延长分组传输的主导因素,因此,在系统超载时,以平均分组排队时延而非平均分组线路传输时延作为性能参数,是合理的.

3.2 平均分组丢失率

图 6 显示了当各业务流的发送速率均为 3.1Mbps,以 1s 为时段间隔来观察的各业务流的丢失率情况.由于队列缓冲长度设置很短,因此从第 7s 开始,各队列的业务分组就相继出现丢失.图 6 显示:由于为各队列选择了恰当的服务率,各队列的平均分组丢失率满足了给定的比例.此外,图 6 进一步显示,从第 7s 开始,第三个队列开始出现分组丢失,到全部队列均出现分组丢失且各队列的平均分组丢失率符合给定比例的时间差只有 4s 的时间,这印证了前面所述的内容,即如果当某一队列开始出现丢失时,仍然使用式(16)求解队列的权值,则各队列的平均分组丢失率失去原有比例的时间很短暂,但用该式求解权值大大减轻了系统的处理负荷.这种现象是由于此时各业务流的平均分组到达率较大,各队列很快会出现分组丢弃,而一旦分组出现丢弃,使用式(16)求出的服务率会实现比例平均分组丢失率公平.

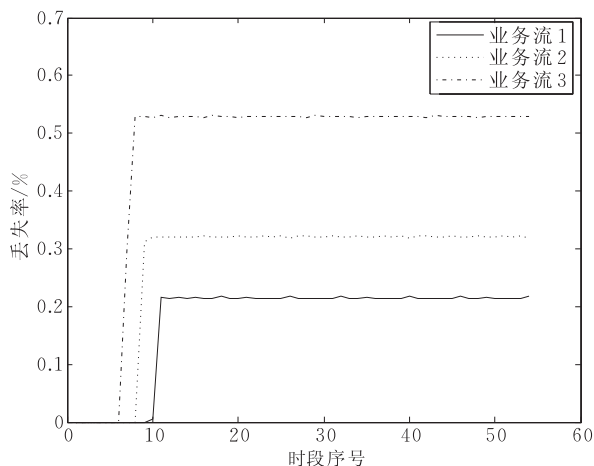


图 6 实现比例平均分组丢失率时各流的平均分组丢失率情况

图 7 显示了当各业务流的发送速率从 3.0Mbps 变化到 4.0Mbps、以 1s 为时段间隔,来观察在 55 个时段内各业务流的分组丢失率均值随各业务流的发送速率的变化.图 7 显示:在不同的业务流发送速率下,各业务流的平均分组丢失率大致满足给定的比例.

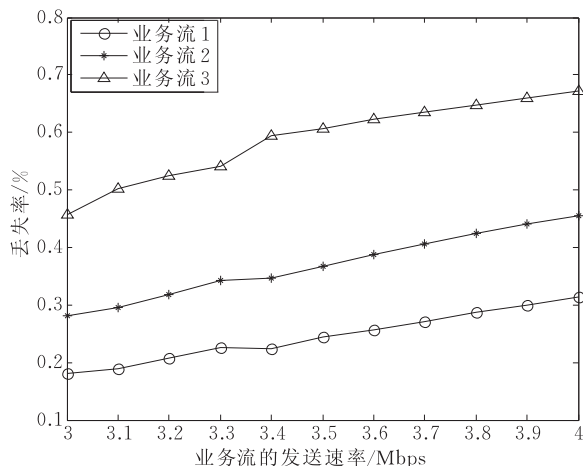


图 7 实现比例平均分组丢失率时各流的平均分组丢失率随流发送速率的变化

3.3 与同类方案的对比

本小节以时间复杂度为衡量性能的标准,对实现比例平均分组排队时延的 PRWRR 及其两个同类方案 APDDS^[12] 和 PAD^[4] 进行对比,然后对实现比例平均分组丢失率的 PRWRR 和同类方案 PLR^[13] 进行对比.

在 APDDS 中,为了选择优先级最高的队列,系统需要进行 $N-1$ 次比较,这将为系统引入额外的计算负荷.此外,为了确定系统当前时间各队列的优先级,APDDS 需要实时地计算各队列的瞬时优先级,而计算该值需要使用 Gauss-Seidel 迭代算法求解一个规模为 N 的非线性方程组,使其计算复杂度至少为 $O(N^2)$;在 PAD 中,为了选择归一化的平均时延最大的队列,系统同样需要进行 $N-1$ 次比较,这也为系统引入了额外的计算负荷.此外,为了得到各队列的平均排队时延,PAD 每发送一个分组就需要完成 N 次除法^[4],这使得 PAD 的计算复杂度为 $O(N)$.在 PRWRR 中,当实现比例平均分组排队时延时,PRWRR 需要按照式(8)计算各队列的服务率.由于求解服务率的方程组是一个线性方程组,且使用直接法时,使用 $N-1$ 次乘法就可以将其系数矩阵变为一个带状的下三角矩阵,这样再使用 $2(N-1)$ 次乘除法即可求出各队列的服务率.因此实现比例平均分组排队时延时,PRWRR 所需的总的乘除法数为 $3(N-1)$,时间复杂度为 $O(N)$.

从上述比较中可以看出,PRWRR 的时间复杂度显然优于 APDDS,因此我们仅进一步对比 PAD 和 PRWRR 在单位时间内的计算量.在 PAD 中,由于系统每发送一个分组就需要完成 N 次除法,因此,在单位时间内的计算量为 $N \times W/t$ (W 为 t 时段

内的系统服务的分组总数);而在 PRWRR 中,只有当各队列的平均分组到达率发生变化时,才进行一次计算,因此,在单位时间内的计算量为 $3(N-1) \times (1 - \prod_{i=1}^N (1 - P_i)) / t$, 式中, $P_i (i=1, 2, \dots, N)$ 是队列 $i (i=1, 2, \dots, N)$ 的平均分组到达率在 t 时段内发生变化的概率. 各队列的平均分组到达率发生变化可由多种因素造成,如网络节点的传输速率发生变化或业务流的源端调整了发送速率,因此 $P_i (i=1, 2, \dots, N)$ 并不好确定,这里我们假定 $P_i (i=1, 2, \dots, N)=1$, 使 PRWRR 在最恶劣的情况下与 PAD 进行对比,在本实验环境中, $W=750, t=1$, 所以 PAD 在单位时间内的计算量为 $750N$, 而 PRWRR 在最恶劣的情况下的单位时间内的计算量为 $3(N-1)$. 根据文献[4], 在当前的商用微处理器 ($\approx 500\text{MHz}$) 中, 浮点数的除法所消耗的时间一般小于 100ns ($\approx 50\text{cycles}$). 为简便起见, 我们令浮点数的除法所消耗的时间等于 100ns , 于是使用 PAD 以及 PRWRR 在单位时间内所需要的计算时间大约分别为 $75N\mu\text{s}$ 和 $3(N-1)\text{ns}$, 显然 PRWRR 要优于 PAD.

PLR 每次选择归一化的分组丢失率最小的队列实施分组丢弃, 因此需要进行 $N-1$ 次比较, 这将为系统引入额外的计算负荷. 由于每次需要丢弃分组时, 均要执行 N 次乘法和 N 次除法, 因此, 其时间复杂度为 $O(N)$. 而与实现比例平均分组排队时延的 PRWRR 一样, 实现比例平均分组丢失率的 PRWRR 所需的总的乘除法次数为 $3(N-1)$, 时间复杂度为 $O(N)$. 与上述分析类似, 由于 PLR 每丢弃一个分组就要执行 $2N$ 次乘除法, 而 PRWRR 仅仅在平均分组到达率发生变化时才执行 $3(N-1)$ 次乘除法, 因此 PRWRR 更加优化.

4 结 语

本文基于比例公平原则在 WRR 的基础上提出了 PFWRR 算法. 实验表明, PFWRR 在队列长度不超过缓冲时, 能实现比例平均分组排队时延保证; 在队列长度超过缓冲时, 能实现比例平均分组丢失率保证. 虽然 PFWRR 为实现比例公平原则在系统中引入了一些计算负荷, 但是这些计算负荷是合理的, 因为当系统不超载的时候, PFWRR 不会引入任何计算负荷, 仅当系统超载且平均分组到达率发生变化时, 计算负荷才被引入. 与同类方案基于每个分组的特性来实现比例公平原则相比, PFWRR 是基于

每个流的特性来实现的, 因此 PFWRR 更为简单.

参 考 文 献

- [1] Braden R, Clark D, Shenker S. Integrated services in the internet architecture: An overview. RFC 1633, 1994
- [2] Nichols K, Blake S, Baker F, Black D. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. RFC 2474, 1998
- [3] Rosen E, Viswanathan A, Callon R. Multi-protocol label switching architecture. RFC 3031, 2001
- [4] Constantinos Dovrolis, Dimitrios Stiliadis, Parameswaran Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. IEEE/ACM Transactions on Networking, 2002, 10(1): 12-26
- [5] Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks: The single node-case. IEEE/ACM Transactions on Networking, 1993, 1(3): 344-357
- [6] Bennett J C R, Zhang H. Hierarchical packet fair queuing algorithm. IEEE/ACM Transactions on Networking, 1997, 5(5): 675-689
- [7] Golestani S. A self-clocked fair queuing scheme for broadband applications//Proceedings of the IEEE INFOCOM'94. Toronto, 1994: 6363-6466
- [8] Shimonishi H, Yoshida M, Ruixue F et al. An improvement of weighted round robin cells scheduling in ATM networks//Proceedings of the IEEE Global Telecommunications Conference. Phoenix, AZ, 1997: 1119-1123
- [9] Shreedhar M, Varghese G. Efficient fair queuing using deficit round robin. IEEE/ACM Transactions on Networking, 1996, 4(3): 375-385
- [10] Chen Jie, Sun Shu-He, Chen Xue. The research and implementation of service scheduling algorithm in APON. Study on Optical Communications, 2002, (6): 5-9 (in Chinese)
(陈洁, 孙曙和, 陈雪. APON 业务调度算法的研究与实现. 光通信研究, 2002, (6): 5-9)
- [11] Zheng Bo, Lin Chuang, Li Yin. A queue management algorithm fit for network processors. Journal of Computer Research and Development, 2005, 42(10): 1698-1705 (in Chinese)
(郑波, 林闯, 李寅. 一种适用于网络处理器的队列管理算法. 计算机研究与发展, 2005, 42(10): 1698-1705)
- [12] Leung Matthew K H, Lui John C S, Yau David K Y. Adaptive proportional delay differentiated services: Characterization and performance evaluation. IEEE/ACM Transactions on Networking, 2001, 9(6): 801-817
- [13] Constantinos Dovrolis, Parameswaran Ramanathan. Proportional differentiated service, Part II: Loss rate differentiation and packet dropping//Proceedings of the 8th International Workshop on Quality of Service. Pittsburgh, PA, 2000: 53-61

- [14] Jaesung Hon, Changhee Joo, Saewoong Bahk. Active queue management algorithm considering queue and load states// Proceedings of the IEEE ICCN 2004. Boston USA, 2004: 140-145

- [15] Cheng Sheng-Tzong, Wu Mingzoo. Performance evaluation of Ad-Hoc WLAN by M/G/1 queueing model//Proceedings of the IEEE ITCC 2005. Washington, DC, USA, 2005, 2: 681-686



WANG Sheng-Ling, born in 1978, Ph. D. candidate. Her research interests include mobile IPv6, mobility management, quality of service.

HOU Yi-Bin, born in 1952, Ph. D. , professor, Ph. D.

supervisor. His research interests include mobile IPv6, human-computer interaction.

HUANG Jian-Hui, born in 1977, Ph. D. candidate. His research interests include mobile IPv6, quality of service and artificial intelligence.

HUANG Zhang-Qin, born in 1964, Ph. D. , professor. His research interests include the key technologies in wireless LAN.

Background

This work is supported by the Key Project Foundation of Technology and Development Program of Beijing Municipal Commission of Education (No. KZ2005100005006), and the National Natural Science Foundation of China (Nos. 90407017, 60403035), and the National High Technology Development Program (863 Program) of China (No. 2006AA01Z205).

As an important solution providing QoS for users, packet scheduling has attracted many attentions. IETF has published many RFCs and drafts in this filed and a lot of top conferences and journals have proposed novel scheduling algorithms. However, most of existing packet scheduling algorithms do not guarantee fairness and priority at the same

time. Thus some researchers improved them according to the proportion fairness principle, but regrettably their scheduling algorithms have high computational complexity. In this scenario, the authors propose a new packet scheduling algorithm realizing the proportion fairness principle, named PFWRR. PFWRR has low computation load because it renews each queue's weight only when the system is overload and the average packet arrival rate changes.

The authors' projects focus on the key technologies in wireless/mobile network including QoS, security and modeling and so on. They have developed a prototype system of wireless access point with QoS, and have published some relevant papers.