

应用层组播优化方法

沈晔^{1,2*}, 冯径¹, 印敏¹, 马玮骏¹, 蒋磊¹

(1. 解放军理工大学气象海洋学院, 南京 211101; 2. 95871 部队, 湖南 衡阳 421002)

(* 通信作者电子邮箱 shenye200708@sina.com)

摘要:应用层组播在不同业务领域的性能要求不完全相同,而应用层组播的网络环境也更加复杂,如:组播节点多元化、通信信道复杂化、节点规模庞大化、数据通信量扩大化等,因此需要结合已有应用层组播技术,针对特定业务的优化目标,对组播方案进行优化改进。通过分析应用层组播评价指标,将应用层组播优化方法分为编码特性优化、分层分簇优化、节点性能优化、选择最优父节点优化、路由信息维护优化,比较了不同类型优化方法的性能指标,给出了各自的适用环境,最后讨论了进一步的研究方向。

关键词:应用层组播;编码特性优化;分层分簇优化;节点性能优化;分发效率

中图分类号: TP393.032 **文献标志码:** A

Optimization methods for application layer multicast

SHEN Ye^{1,2*}, FENG Jing¹, YIN Min¹, MA Weijun¹, JIANG Lei¹

(1. Institute of Meteorology and Oceanography, PLA University of Science and Technology, Nanjing Jiangsu 211101, China;

2. 95871 Troops, Hengyang Hunan 421002, China)

Abstract: The performance requirements of application layer multicast are not identical in different business areas, and the network environment is more complex as follows: the multicast node is diversified, the communication channel is complex, the node scale is large, the amount of data is magnified and so on. The multicast programs should be optimized by analyzing the existing application layer multicast and combining new applications demands. By analyzing the evaluating indicator of application layer multicast, application layer multicast optimization method would be divided into the coding features optimization, the hierarchical clustering optimization, the node performance optimization, the optimal parent selection optimization and the routing information maintenance optimization. Through comparing the performance indicators of different types of optimization methods, the applicable environments were introduced separately, and further research directions were discussed finally.

Key words: application layer multicast; coding features optimization; hierarchical clustering optimization; node performance optimization; distribution efficiency

0 引言

应用层组播在终端之间运用单播方式传输数据,这样不必改变原有网络的基础设施,也不需要路由器维护组播组的路由表,相比 IP 组播更容易实现组播。然而正因为应用层组播屏蔽了底层的路由信息,在研究推广中存在以下问题:可靠性差、可扩展性不好、延迟比较大,传输效率不如 IP 组播。围绕应用层组播的这些问题,学者们提出了很多改进优化算法:根据数据的编码特性来提高系统可靠性和减少数据延时^[1-3],采用分层分组的方式提高系统的可扩展性^[4-6],考虑节点性能并通过最优节点转发数据充分利用节点资源^[7-9],维持邻居节点信息并选择最优父节点,提高数据分发效率^[10-11],维护全局路由信息,保证构造出全局最优树^[12-14]。近年来应用层组播的网络环境更为复杂:1)组播终端呈现出多元化的趋势,除传统服务器,手持移动终端、机载终端等大量涌现,这些终端性能较低但移动性很强^[15];2)通信信道呈现异构性,除传统单一的有线信道,出现了业务网与专线网混合信道、卫星信道、短波信道等^[15];3)组播节点规模更加庞

大,传统的组播节点是固定的分发接收服务器,随着云服务的到来,云服务器、云终端成为了新的组播节点,这些节点性能差异大、动态性强^[15];4)随着大数据的到来,组播分发的数据量呈指数级增长,导致组播分发的数据信息剧增,同时终端用户对所需信息的准确性、全面性要求越来越高,从而控制信息也同步增多^[16]。

已有的应用层组播方案不能完全解决以上新问题,而设计一种新的通用的组播方案会非常复杂,因此需要根据具体业务的优化需求研究相应的解决方案。本文在分析应用层组播基础上,对已有的应用层组播优化方法进行综述,并将优化方法分类为编码特性优化、分层分簇优化、节点性能优化、选择最优父节点优化和路由信息维护优化,研究了不同优化方法的应用环境及性能指标,为应用层组播业务的优化提供理论依据。

1 应用层组播性能优化方法分类

1.1 应用层组播性能评价指标

应用层组播的性能评价指标是组播性能的衡量标准,不

收稿日期:2013-07-16;修回日期:2013-07-30。 基金项目:国家自然科学基金资助项目(61070174)。

作者简介:沈晔(1985-),男,湖南湘潭人,助理工程师,博士研究生,CCF 会员,主要研究方向:计算机网络、网络数据分发;冯径(1962-),女,江苏南京人,教授,博士生导师,博士,CCF 会员,主要研究方向:计算机网络、分布式系统;印敏(1978-),女,江苏泰兴人,讲师,博士,主要研究方向:气象海洋信息传输;马玮骏(1980-),男,江苏南京人,讲师,博士,主要研究方向:大数据存储、计算机网络;蒋磊(1987-),男,江苏南京人,硕士研究生,主要研究方向:计算机网络。

同应用场景的性能指标评价标准不同。

1) 数据分发路径的质量。主要有下面三个指标:

强度(Stress) 是指在一条物理链路中发送相同数据包的数量。显然 IP 组播进行转发时并没有进行多余的复制,所以 IP 组播的链路强度是最优值,即 $Stress = 1$;而应用层组播是在主机之间复制数据,链路上可能存在数据包的重复发送,因此链路强度 $Stress \geq 1$ 。强度与网络带宽关系很大,在强度一定的情况下,带宽越大,强度对组播性能的影响越小,反之亦然。

伸展度(Stretch) 将路径长度 L_{A-B} 定义为 N_A 与 N_B 之间的跳数,那么伸展度(用 SH_{A-B} 表示)是指 N_A, N_B 沿应用层组播传播方向的路径长度 L_{A-B} 与 N_A, N_B 沿单播路径传输的 L_{A-B}' 之间的比值,即 $SH_{A-B} = L_{A-B}/L_{A-B}'$ 。单播传输路径长度 L_{A-B}' 是由网络层的路由策略保证的,应用层将该长度理解为最优的,而在应用层组播中,源与接收成员之间可能会通过中间节点进行转发,增加了节点间跳数,因此 $L_{A-B} \geq L_{A-B}'$,即 $SH_{A-B} = L_{A-B}/L_{A-B}' \geq 1$ 。

延时(Delay) 是指沿应用层组播路径,数据从源节点到接收节点的所需时间,表示为 $D = D_{Transmission} + D_{propagation} + D_{Process}$ 。其中: $D_{Transmission}$ 为数据的传输延时,是由链路的传输速率造成的; $D_{propagation}$ 为传播延时, $D_{propagation} = d/s$,其中 d 是距离, s 是传播速率; $D_{Process}$ 是节点处理延时,由节点复制转发造成。应用层组播延时的性能指标可分为最大延时最小、平均接收延时最小两类。

2) 节点的性能。

鲁棒性(Robust) 是指节点失效或链路故障时的故障恢复能力及延时。鲁棒性一般通过组播策略来保证,将因故障而断裂的链路重新连接到组播树中。

收到第一个包的延时($D_{firstpackage}$) 是指当成员加入到组中,收到第一个包的延时,该延时体现了组播树的响应能力,减少成员首次接入时的等待时间。

出度(Out-Degree) 是指沿组播分发路径,节点需要下发数据的节点数,即孩子节点数。出度与节点的处理性能相关性很强,出度越大,对节点的并行处理能力及上行速率要求越高。

深度(Depth-Degree) 是指从源节点到接收节点所经过的节点数,源节点的深度为 0。深度与延时紧密相关,深度越大,数据在节点上的转发次数越多,那么节点接收数据的延时也就越大。一般来说,组播树的最大延时节点出现在深度最大的节点中,可以通过规定组播节点的最大深度来保证节点的最大延时最小。

3) 控制负荷比重(Control Head)。指组播控制信息量占整个组播数据量的比重,为了有效地利用网络资源,必须减少控制负荷比重。

4) 组成员接纳能力(group member capacity)。指组播系统的服务提供能力。在系统仿真实验中,将系统的组成员接纳能力定义为在一定的网络条件和成员服务定制条件下系统最大可容纳的组成员数。在大规模组播通信中,组成员接纳能力是一个很重要的衡量指标。

5) 可扩展性(scalability)。指组播成员的增加对整个组播性能的影响情况。可扩展性好说明组播对成员节点的加入有良好的控制机制。

1.2 优化方法及其分类

目前应用层组播研究的优化源可分为基于编码特性优

化、基于分层分簇优化、基于节点性能优化、选择最优父节点优化和基于路由信息维护的优化共五类。

基于编码特性优化是在终端节点上对组播信息流进行编码、解码,将编码后的信息在节点间的多条传输路径同时传输,提高链路的整体吞吐率^[1-3]。

基于分层分簇优化是根据节点的地理位置、隶属关系、所需数据类型等,构造节点性能评估函数,将节点分到不同的层或簇中,构造节点性能评价函数是关键^[4-6]。

基于节点性能优化是充分考虑组播成员的出度、深度、节点处理能力等,性能越高的节点承担越多的组播分发任务,性能低的节点尽量安排在叶子节点^[7-9]。

基于选择父节点的优化是从选择离本节点距离最近且性能最好的节点作为父节点,并定义一组备选父节点集,在链路出现断裂时可以及时恢复链路,保证组播树的健壮性^[10-11]。

基于路由信息维护的优化是采用拓扑感知和跨层技术,在应用层上获取路由层的信息,构造的组播树分发路径尽量与节点的网络拓扑结构一致,减小链路的强度与伸展度^[12-14]。

2 应用层组播研究的优化分类

2.1 基于编码特性的优化

文献[1]中提到利用两节点在 IP 层的多路径和过剩能力,在节点中进行信息的编码与解码,提高节点间的通信效率。如图 1 所示, a, b 分别是数据源 s 产生的两个独立信息流,节点 u_3 将接收到的 a, b 通过编码规则组合成 $a \oplus b$, 通过节点 u_4 传输给 t_1 与 t_2 , t_1 与 t_2 可以从 $\{a, a \oplus b\}$ 与 $\{b, a \oplus b\}$ 中将 $\{a, b\}$ 还原出来。假设每条链路的传输能力为 C , 采用网络编码方式,节点的吞吐率能达到 $2C$, 而不采用网络编码方式,节点的吞吐率只有 $3C/2$ 。

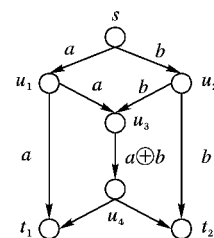


图1 采用网络编码的组播分发路径

文献[2]分析出在传统光组播中,节点所得到的最大传输速率均小于由最大流最小割定理确定的最大传播速率,提出采用网络编码的光组播无需建立多个会话就能达到组播传输速率的上限,实现组播最大容量,对均衡网络负载起到了重要的调节作用。

文献[3]提到为适应组播节点的异质性,降低动态环境中节点失效对组播性能的影响,采用多描述编码技术对流媒体系统中的视频进行编码,并构造层次结构模型来实现描述之间的同步问题。该模型根据多描述的编/解码工作方式,在组播节点中采用分布式算法来实现描述资源的均匀分布,实现了系统的快速收敛,并提出了最大覆盖算法,在为节点获得最大描述集的同时,实现节点对描述请求的均匀分布,提高其健壮性。

2.2 基于分层分簇的优化

针对大数据量或不同需求的节点,可以将具有相同属性(如地理位置、隶属关系、所需数据)的节点组合成一个簇,在

簇内部选择一个节点作为簇首,簇与簇之间通过簇首节点进行交互,通过这种化整为零的方法,将整个组播成员节点集合分散成多个容易管理的簇,使得组播树结构清晰、管理便利。

文献[4]中在单组应用层组播的基础上提出多组合作的应用层组播(Multi-Group Forest, MGF),将具有相同数据源(source)的组播节点定义成一个组播域(session),一个组播域节点可以是另一个组播域的转发节点,如图2所示。

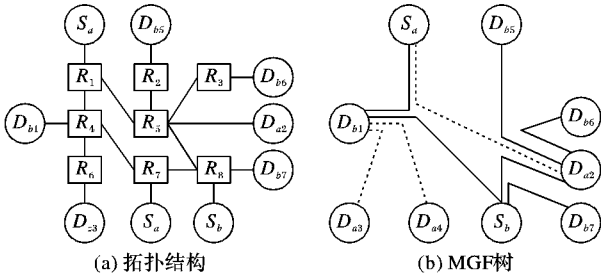


图2 两个组播拓扑结构与对应的 MGF 树

图2表示一个网络拓扑结构与对应的 MGF 树,矩形框代表路由器,环形框代表终端节点,两个组播组 G_a 与 G_b , S_a 是 G_a 的数据源, S_b 是 G_b 的数据源, D_{a2} 、 D_{a3} 、 D_{a4} 是 G_a 的接收端, D_{b1} 、 D_{b5} 、 D_{b6} 、 D_{b7} 是 G_b 的接收端,假设有以下路径:

$$\begin{aligned} path(S_a, D_{a2}) &= \langle R_1, R_5 \rangle, path(S_a, D_{a3}) = \langle R_1, R_4, R_6 \rangle, \\ path(S_a, D_{a4}) &= \langle R_1, R_4, R_7 \rangle, path(S_b, D_{b1}) = \langle R_8, R_7, R_4 \rangle, \\ path(S_b, D_{b5}) &= \langle R_8, R_5, R_2 \rangle, path(S_a, D_{b6}) = \langle R_8, R_5, R_3 \rangle, \\ path(S_b, D_{b7}) &= \langle R_8 \rangle \end{aligned}$$

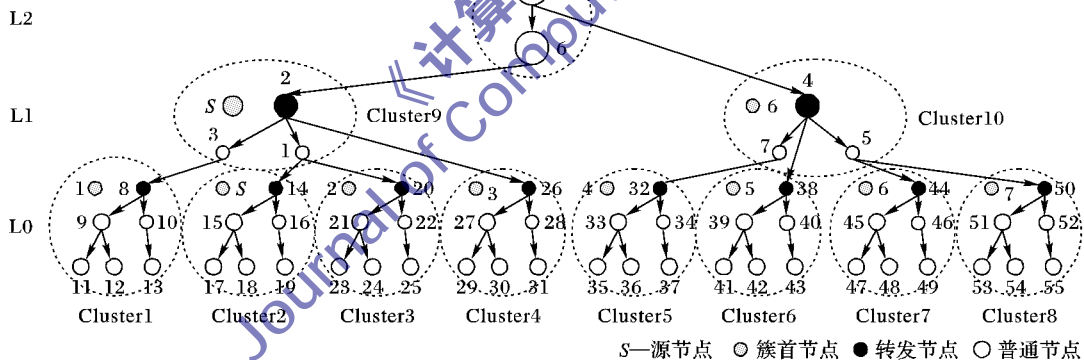


图3 PH-ALM 算法构造的组播树

文献[6]提出根据节点在线时间、可用带宽和数据延时将参与组播的节点划分成不同优先级,据此,一部分节点被选为高级节点,其他为普通节点。每一个高级节点与邻近的普通节点形成一个节点集合,这样整个组播节点划分成了分等级的节点管理域。

其中节点 i 的在线比率计算公式如下:

$$a_i = \sum_k \frac{\tau_k d_{i,k}}{\beta} \quad (1)$$

其中: a_i 是节点 i 的在线比率, β 是计算在线时长的采样周期, k 是最后一个周期中节点间的会话数, $d_{i,k}$ 是第 k 次会话时第 i 个节点的在线时间, τ_k 是平滑参数。

可用带宽比率 b_i 的计算公式如下:

$$b_i = 1 - \frac{(m/x_i)^s}{1 - (m/v)^s} \quad (2)$$

其中: s 为普轮廓参量, m 、 v 分别为最小带宽和最大带宽, x_i 是第 i 个节点的带宽,计算得到的 b_i 代表节点 i 的带宽优先级。

得出的最优覆盖层树是:对 G_a 来说, S_a 是 $\{G_{b1}, G_{b2}\}$ 的父节点, D_{b1} 是 $\{D_{a3}, D_{a4}\}$ 的父节点;对 G_b 来说, S_b 是 $\{D_{b1}, D_{b2}, D_{b7}\}$ 的父节点, D_{a2} 是 $\{D_{b5}, D_{b7}\}$ 的父节点。

MGF 中的每一个节点都维持着一张家族表(Family Table, FT),用来定义组播树中的父子关系,由于添加了组的概念,FT 的一条记录用(address, spath, group)表示, address 是 IP 地址, spath 是由根路径到当前节点的最短路由, group 是当前所在的组。

每个组用(S, G)标识,其中 S 表示源节点, G 表示组播编码,节点 N 只能属于一个组播域,称为该节点的 apply-session,用 $AG(N)$ 表示,但节点可以作为转发节点(relay-node)加入其他域中,称为该节点的转发组(relay-group)。

MGF 方案对节点进行分域后,优点是能充分利用节点之间的路由关系,借用非本分域域的节点作为转发节点,提高数据的转发效率;也存在以下缺点:1)一个节点只能属于一个组播域,即节点只能接受一个源节点的数据,所需数据源约束性比较大;2)维持 FT 表,需要路由器的拓扑信息,因此在路由探测和构建上需要额外的开销。

文献[5]提出了 PH-ALM 算法,在分层的第 0 层包括所有节点,所有节点根据数据延时划分到不同的簇中,簇的大小为 $[k, 3k - 1]$ (k 为大于等于 1 的常量),在每个簇中选取与其他簇节点延时最短的节点作为簇首节点,簇内部建立以簇首节点为根的本簇组播树,构造的组播树如图 3 所示。这种构造方法简单快速,能减少维持组播树的控制信息负荷。

分四种情况来计算节点延时,用矩阵表示如下:

$$M = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{bmatrix} a_i & b_i & t_{i,1} \\ a_i & 1 - b_i & t_{i,2} \\ 1 - a_i & b_i & t_{i,3} \\ 1 - a_i & 1 - b_i & t_{i,4} \end{bmatrix} \quad (3)$$

其中: M_1 表示节点在线且节点可用带宽能满足需求, M_2 表示节点在线且可用带宽不能满足需求, M_3 表示节点不在线且节点可用带宽能满足需求, M_4 表示节点不在线且节点可用带宽不能满足需求。统计节点在各种情况下的延时得到节点的总的延时如下:

$$Y_i = \sum_p (m_{p,1} \cdot m_{p,2} \cdot m_{p,3}) = a_i \cdot b_i \cdot t_{i,1} + a_i \cdot (1 - b_i) \cdot t_{i,2} + (1 - a_i) \cdot b_i \cdot t_{i,3} + (1 - a_i) \cdot (1 - b_i) \cdot t_{i,4}$$

2.3 基于节点性能的优化

应用层组播中,节点除了是数据接收方,还承担着数据转发的任务,因此,考虑节点性能来调整组播策略,能保证整个

组播树的高效性与鲁棒性。节点的性能包括节点出度、深度、主机所在位置、主机性能等。

文献[7]提出的 CFTB 算法考虑了节点出度的大小,在候选父节点中,只有该候选父节点当已有子节点数小于出度阈值时,才可以继续添加子节点。

文献[8]中提到从 WHOIS database 中查询到节点的网络服务名,通过网络服务名携带的 ISP 与本地域信息,将物理距离相近的节点作为相邻节点。

文献[9]提出的 DINCast 算法设计了专门的逻辑数据环来代替集中式组播分发树的汇聚节点(Rendezvous Point, RP)节点(如图 4 所示),与采用单一汇聚节点的应用层分发树相比,该方法具有以下优点:1)减少内部节点延时,环能更深入组播网,离成员节点更近;2)增加系统鲁棒性,环能实现负载均衡,并能缓解单点失效;3)保证了数据持久性,在存活期内信息能不断在环中循环,减少了节点向汇聚点发送信息量。如图 4 所示,节点 0、1、2、3 组成环状汇聚节点,子节点只要向环的一个成员发送请求信息即可。

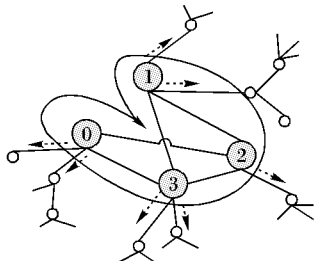


图 4 DINCast 算法的逻辑数据环结构

2.4 基于父节点选择的优化

应用层组播中通信的基本单元是父子节点之间的通信,因此选择性能最优的节点作为父节点,能减少组播延时,提高组播性能。

HTMP 算法^[10]中节点加入组播组的思路为:新加入节点先获取根节点的位置,然后逐渐向下查找子节点,直到寻找到离自己最近的备选父节点作为父节点。而文献[7]提出的 CFTB 算法给出了一种新的节点加入组思路:定义一个关联节点集,关联节点集包括已认可的候选父节点及其子节点,新加入节点通过 CFS 算法,在关联节点集中寻找出距离最短的节点作为父节点。

如图 5 所示,节点 7 的待加入节点,对 HTMP 算法来说,候选节点的变化依次是:(1)→(1,2)→(1,2,6),最后是在(1,2,6)中寻找最佳父节点,选择节点 6 作为父节点;而在 CFTB 算法中,节点 7 的候选父节点变化依次是:(1)→(1,2,3)→(1,2,3,6),最后是在(1,2,3,6)中寻找最佳父节点,选择了比节点 6 性能更好的节点 3 作为父节点。

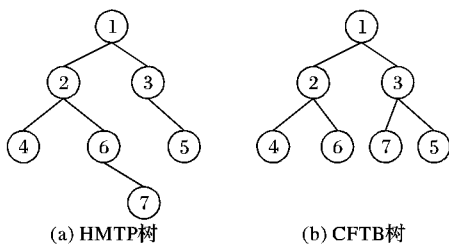


图 5 HMTP 树与 CFTB 树

这种方法寻找到的父节点能保证:

1) CFTB 寻找到的父节点 P 是所有关联节点集中与待加入节点距离最近的;

2) CFTB 寻找父节点时,距离测量的度恒等于关联节点集的数目。

文献[11]提到在传统的基于 Chord 环的覆盖层 P2P 网络中,节点同时与多个节点有直接连接,因此一个节点可以拥有多个可能父节点,然而一个节点在接收子节点请求消息后,会放弃之后请求子节点的消息,因此并不能达到真正的最优。为了消除这种弊端,提出了采用邻接矩阵(Adjacency Matrix, Adj_Mat)来存储节点之间的路由信息,由以下公式计算矩阵值:

$$Adj_Mat[i, j] = \begin{cases} 1, & i \text{ 与 } j \text{ 之间有直接链路} \\ 0, & i \text{ 与 } j \text{ 之间没有直接链路} \end{cases}$$

Adj_Mat 为非对称链路,邻接矩阵如表 1 所示,可得出以下结论:

$$\sum_j Mat_Adj[i, j] = O(\ln(n)); \forall i$$

$$\sum_j Mat_Adj[i, j] = O(\ln(n)); \forall j$$

表 1 邻接矩阵表结构

节点	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_0	—	1	1	0	1	0	0	0
N_1	0	—	1	1	0	1	0	0
N_2	0	0	—	1	1	0	1	0
N_3	0	0	0	—	1	1	0	1
N_4	1	0	0	0	—	1	1	0
N_5	0	1	0	0	0	—	1	1
N_6	1	0	1	0	0	0	—	1
N_7	1	1	0	1	0	0	0	—

邻接矩阵代表了参与组播节点的全部信息,因此在全网节点共享该邻接矩阵。

2.5 基于路由信息维护的优化

应用层组播的效率比 IP 组播差的最主要原因是由于屏蔽了底层的拓扑信息,增加了链路的强度,因此通过获取底层路由信息,能有效提高应用层组播树的分发效率。

文献[12]每个节点维持邻居信息表(Neighbor Information Table, NIT), NIT 记录用<Address; Hop Count>表示, Address 为邻居节点 IP 地址, Hop Count 为与邻居节点间的跳数。发现有新的邻居节点时,更新 NIT,通过 NIT 来维持整个组播树的拓扑结构。

文献[13]提出的组播服务覆盖网络(Multicast Service Overlay Network, MSON)将 OSI 模型中的会话层与表示层用运行有应用层协议的中间件层代替(如图 6 所示),中间件层分为基础覆盖服务(Basic Overlay Services, BOS)与服务覆盖网(Service Overlay Network, SON)。BOS 为覆盖网提供节点定位、邻居节点发现服务;SON 层提供组播、服务质量以及安全服务,SON 层提供骨干支撑网络。

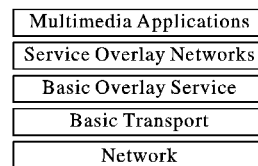


图 6 覆盖层的中间件参考模型

文献[14]提出采用距离启发式算法,通过与父节点、祖父节点、备用父节点延时的比较,对 NICE、HMTP、YOID 组播分发树表现出的不良拓扑结构(如 V 状、Z 状及螺旋状)进行优化并重构组播树,使应用层分发树路径与网络层路由最大

程度接近。

3 应用层组播优化方法的对比分析

将应用层组播的各种优化方法对性能指标的影响进行归纳,如表2~4所示:表2从对数据分发路径质量的影响进行比较;表3对节点性能的影响进行了比较;表4对控制负荷比重、组成员接纳能力、可扩展性的影响进行了比较。

表2 对数据分发路径质量影响的比较

优化类型	链路强度	伸展度	延时
特征编码	减弱	—	减少
分层分簇	—	减弱	减少
节点性能	—	—	减少
父节点选择	—	减弱	减少
路由信息维护	减弱	减弱	减少

表3 对节点性能影响的比较

优化类型	鲁棒性	收到第一个包延时	出度	深度
特征编码	—	—	—	—
分层分簇	增强	减少	受约束	受约束
节点性能	—	减少	受约束	受约束
父节点选择	增强	减少	受约束	受约束
路由信息维护	增强	减少	—	—

表4 对其他性能影响的比较

优化类型	控制负荷比重	组成员接纳能力	可扩展性
特征编码	较大	—	—
分层分簇	较小	增强	增强
节点性能	—	增强	增强
父节点选择	—	增强	—
路由信息维护	较大	—	增强

通过分析,可得出不同组播优化源的应用场景:基于特征编码的优化适用于流媒体分发、大规模成员;基于分层分簇的优化适用于大规模、成员有层级关系的组播业务;基于节点性能的优化适用于节点性能差异较大、节点性能可获取的组播业务;基于父节点选择的优化适用于动态性较强、组播节点频繁加入退出的组播业务;基于路由信息维护的优化适用于路由层信息可获取的组播业务。

4 结语

本文分析了应用层组播的性能评价指标,从近年来提出的应用层组播研究方案中归纳出主流的优化方法,并进行了总结比较。针对应用层组播出现的组播节点多元化、通信信道复杂化、节点规模庞大化、数据量扩大化的新挑战,我们提出了进一步的研究工作:

- 1) 在大规模大数据量的应用层组播系统中,将多路径分发、邻居信息恢复融入应用层组播数据分发中,来提高组播分发效率,减少网络瓶颈对数据分发的影响;
- 2) 维持尽量少的控制信息,减少组播节点间的通信量;
- 3) 考虑组播数据的安全性,防止信息的窃取与丢失。

参考文献:

- [1] ZHU Y, LI B C, GUO J. Multicast with network coding in application-layer overlay networks [J]. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 107-120.
- [2] 肖昊明,张敏,阳小龙.一种基于分布式网络编码的共享树光组播算法[J].计算机工程,2009,26(19),4719-4721.
- [3] 张有为,吴黎兵,吴产乐.一种基于多描述编码的应用层组播模型[J].计算机应用研究,2010,27(10):3932-3934.
- [4] PI R J, MA Y, GUO P, et al. Application layer multicast with multi-group cooperation [C]// Proceedings of the 2006 1st International Symposium on Pervasive Computing and Applications. Piscataway: IEEE, 2006: 71-76.
- [5] HU Y S, CHEN L B, LI Q, et al. A priority-based hierarchical application layer multicast model [C]// WiCOM'09: Proceedings of the 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing. Piscataway: IEEE, 2009: 1-6.
- [6] SU Z, AWIPHAN S, OGURA K, et al. Hybrid application layer multicast with hierarchically distributed nodes [C]// Proceedings of the 2010 7th IEEE Consumer Communications and Networking Conference. Piscataway: IEEE, 2010: 1-5.
- [7] ZHANG X C, LI X D, WANG Z, et al. A delivery tree building approach for application layer multicast [C]// ISCSCT '08: Proceedings of the 2008 International Symposium on Computer Science and Computational Technology. Piscataway: IEEE, 2008, 1: 630-634.
- [8] WANG M, PENG G, ZHANG Y J, et al. Selfishness-aware application-layer multicast [C]// GLOBECOM 2010: Proceedings of the 2010 IEEE Global Telecommunications Conference. Piscataway: IEEE, 2010: 1-6.
- [9] GUO H Q, LEK H N, WONG W C, et al. DIncast: optimizing application-level shared-tree multicast [C]// Proceedings of the 2009 International Conference on Advanced Information Networking and Applications. Piscataway: IEEE, 2009: 270-277.
- [10] ZHANG B C, JAMIN S, ZHANG L X. Host multicast: a framework for delivering multicast to end users [C]// INFOCOM 2002: Proceedings of the 2002 Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Washington, DC: IEEE Computer and Communications Societies, 2002, 3: 1366-1375.
- [11] AMAD M, MEDDAHI A. A scalable approach for application layer multicast in P2P Networks [C]// PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications. Washington, DC: IEEE Computer Society, 2008: 498-503.
- [12] HUANG C-H, KE K-W, WU H-T. An application layer multi-source multicast with proactive route maintenance [C]// TENCON 2009: Proceedings of the 2009 IEEE Region 10 Conference. Piscataway: IEEE, 2009: 1-6.
- [13] UCHOA D C, KOPP S, PIMENTEL H M, et al. An overlay application-layer multicast infrastructure [C]// AINA'09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications. Piscataway: IEEE, 2009: 233-240.
- [14] ZHANG X C, LI X D, LUO W M, et al. On improving the multicast performance of application layer multicast [C]// ETCS'09: Proceedings of the 2009 First International Workshop on Education Technology and Computer Science. Piscataway: IEEE, 2009, 1: 903-907.
- [15] ZHANG Y, LIU H, JIAO L, et al. To offload or not to offload: an efficient code partition algorithm for mobile cloud computing [C]// Proceedings of the 2012 1st IEEE International Conference on Cloud Networking. Piscataway: IEEE, 2012: 80-86.
- [16] DIXIT A, PAWAN P, HU Y C, et al. On the impact of packet spraying in data center networks [C]// Proceedings of the IEEE INFOCOM'13. Piscataway: IEEE, 2013: 2130-2138.