

基于区分服务的蛛网结构应用层多播模型

殷安生¹ 许建真^{1,2} 张福炎²

(南京邮电大学计算机学院 南京 210003)¹

(南京大学计算机科学与技术系 计算机软件新技术国家重点实验室 南京 210093)²

摘要 提出一种高效的蛛网结构的应用层多播模型,该模型以环形结构为基础,具有混乱情况下自愈性强、传输数据速度快等优点;在数据转发时根据实际应用采取 push 或 pull 的策略,同时取消了分层多播模型中的层中控制节点,降低了结构的复杂性;并根据网络状况动态优化系统的拓扑结构。通过仿真试验得知,应用该模型的应用层多播其性能得到提高。

关键词 应用层多播,区分服务,蛛网,服务质量

Application Layer Multicast Model with Cobweb Architecture Based on DiffServ

YIN An-sheng¹ XU Jian-zhen^{1,2} ZHANG Fu-yan²

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)¹

(State Key Laboratory of Novel Software Technology, Department of Computer Science & Technology, Nanjing University, Nanjing 210093, China)²

Abstract This paper presents an efficient application layer multicast model with cobweb architecture. It is based on ring architecture. The model can transmit data with high speed and resume itself in a disordered situation. It chooses pulling or pushing policy according to the application during transmission. The model cancels the leader node to reduce the structure complexity. And also, it optimizes the topology dynamically corresponding to the status of the current environment. Through network simulation, better performance compared to other relative models is achieved.

Keywords Application-layer multicast, DiffServ, Cobweb, Quality of service

1 引言

随着互联网中多媒体应用的发展,应用层多播技术得到越来越多的应用,它能够满足集群用户的服务要求,人们为此建立了很多模型:NICE^[1],Narada^[2],Chord^[3],CAN^[4]等等。也出现了很多应用软件如下载工具 BitTorrent 等,在连接越多时,用户越能快速地得到数据。但同时也带来了一些很严重的问题,例如 BT 对网络资源的无节制的掠夺与浪费等。于是如何提高应用层多播的性能就成了迫在眉睫需要解决的问题。

为了解决多播过程中所带来的这些问题,目前提出很多解决办法。包括一些避免拥塞和拥塞控制算法,例如 TCP 友好,分层编码原理、资源预留、层间冗余链路^[5],还有若干协议的组合^[6]等等。但是这些方法都是在网络恶化时或是即将恶化时才采取措施,这是治标不治本的方法。

本文提出一种高效的蛛网结构的应用层多播模型,使用一种区分服务的基于 Flooding 算法的分层多播数据传输协议 DFLM(DiffServ and Flooding-based Layered Multicast data transmitting)。应用 Flooding 算法构造出一个十字架结构的传输模型,以提高传输性能,从拓扑结构上解决多播的性能问题。

2 模型的设计

应用层多播由于取消了 IP 多播中的路由协议,主要以系

统构造的拓扑结构决定传输策略,简单可靠,主要的多播拓扑有以下几种:采用树优先方式^[7],关注的是怎样直接提高这棵共享树的效率;网状优先^[8]的多播模型采用分布式的算法使得多播结构的构建速度快;采用分层结构的应用层多播协议^[1]引入域和代理等概念。采用环状结构^[9]能够快速传播数据,但控制复杂。

本文提出的蛛网结构模型其实是一种分层环状结构的应用层多播模型 HR(Hierarchical-Ring multicast model)。

2.1 HR 拓扑设计

分层结构考虑到不同接入网络的异构性,每一层均采用环的结构,其中环的维护由终端执行。

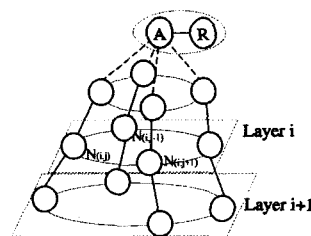


图1 系统的结构

2.1.1 HR 的拓扑结构

当节点处于平衡态时系统拓扑结构如图1所示。最高层为资源节点,即服务器节点,它包括两个部分,资源代理 A 和控制代理 R, A 向外发布资源, R 了解系统的拓扑并维护系

统。节点依次分布在由上向下的各层环中,通常情况下,待上层环中节点满了之后再向下层添加,并且节点之间的连接将尽量依照实际网络拓扑。 $N_{(i,j)}$ 表示为第 i 层第 j 个节点, A 节点将数据分成若干段,将这若干段数据按序传送给第一层的主干节点^[10],在环中还有起备份数据作用的节点,这些节点一般和主干节点并联。规定相邻层上互联的节点上游为父节点,下游为子节点,左右相邻即为邻居节点。

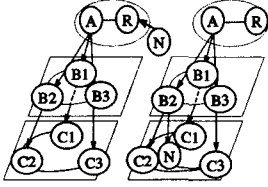


图2 节点请求加入

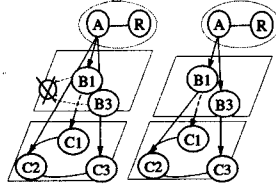


图3 节点非正常退出

2.1.2 拓扑结构的维持

考虑到应用层多播系统中所有的节点都可能随时退出和加入,所以为了保持结构的稳定,必须能够动态实时地处理节点的加入和退出^[11]。

节点的加入:新节点 N 加入,发送 Request-join(state) 请求到 R (如图2),其中 state 描述的是自身例如物理位置,所在网络状态等信息, R 根据 state 信息和系统状态安排节点的位置。新节点加入时若环中主干节点满,则新节点和 $C2$ (根据位置映射得到最近节点) 并联,最终形成一个双环结构^[12]。并联时上层同时向下层的这两个节点传送,并联结节点同时起备份作用。

节点的丢失:节点在正常退出时发送 Request-leave (state) 给 R 。若节点非正常退出,例如 $C2$ 发现无法从上游节点获取数据则连往 A 并向 R 通报 (如图3), R 查询是否有备份节点可供使用,若没有,则根据邻近原则取某一主干节点承担父节点作用。

环的合并和拆分:同一环中最多容纳 $2m$ 节点,所以当环中节点数多于 $2m$,则环将被拆分,外面的一层将脱落下来成为下一层;若环中节点少于 $m/2$ 个,则将该环与其相邻环合并为一个环。

2.1.3 区分服务的数据传输算法 DFLM

DFLM 有两个特点,第一是区分应用,HR 的应用主要有两种,分为非实时性的文件数据下载和实时性要求较高的多媒体应用^[13]。对不同的应用系统采用不同的数据传输策略。第二是基于 Flooding 算法,根据拓扑结构的特点,实现一种泛洪式的数据传播方式。

因此为每个节点建立属性。首先建立一个链表结构体存储与之相邻节点的信息。然后建立资源矩阵用来表示节点占有资源信息。

将源节点的资源进行分割,组成资源矩阵 $\{s[1], s[2], s[3], \dots, s[k], \dots, s[n]\}$ (n 为资源分片的总数, $1 \leq i \leq n, s[k] \in \{0, 1\}$, 表示是否拥有该数据),源节点的资源总量为:

$$S = s[1] + s[2] + s[3] + \dots + s[k] + \dots + s[n] \quad (1)$$

节点结构定义如下:

Struct Member {

```
int layer; //ring number
int sequence; //position in a ring
int s[]; //resources matrix
...;
}
```

这里使用 $\text{Node}(i,j)$ 替代 $(\text{Node}, \text{layer} = i \ \&\& \ \text{Node}, \text{sequence} = j)$

```
Procedure:DFLM for Node(i,j)
App_type=Judge_App(central node);
//judge the type of application
if (App_type==Document downloading) //pull the data
{
for(k=0;k<n;k++)
if(s[k]==0)
{
NodeWithData=RequestData(Node(i,j),s[k]);
SendData(NodeWithData,Node(i,j),s[k]);
}
}
elseif(App_type==Realtime_Media) //push the data
{
for(k=0;k<n;k++)
if(s[k]==1)
{
SendData(Node(i,j),Node(i,j+1),s[k]);
SendData(Node(i,j),Node(i,j-1),s[k]);
SendData(Node(i,j),Node(i-1,j),s[k]);
}
}
```

若为第一种应用则采用 pull 策略,转发数据前先查询对方是否需要。节点传送数据的时候,根据 Flooding 算法的原理,数据由 A 开始依次传输,表现为向下优先的原则。对单个节点来说,采用一种十字架模型传输数据,即节点总是先从父节点接收数据,接收数据后马上向子节点传播数据,然后向邻居节点传播,传播的时候采用查询或中断方式^[14]。

```
Procedure:transmit data with query
RequestData(Node(i,j),s[k])
//return the node which contain s[k]
{
if(Node(i,j+1).s[k]==1)
return Node(i,j+1);
elseif(Node(i,j-1).s[k]==1)
return Node(i,j-1);
else for(level=i;level>0;level--)
{
if(Node(level,j).s[k]==1)
return (Node(level,j));
}
}
```

```
Procedure:transmit data using interrupt
SendData(NodeWithData,Node(i,j),s[k])
//send data s[k] to the Node(i,j)
{
//transmit data from Node with data to Node(i,j)'s
neighbor or parents
TransmitData(NodeWithData,Node(i,j),s[k]);
ReceiveData(Node(i,j),s[k]); //the Node(i,j) receive data from neighbor or parents
}
```

DFLM 能有效保证资源耗费与用户需求的平衡。数据传输时自上而下,有序转发,使得数据能很快到达所有的节点而又不会杂乱无章。在环内数据沿两个方向同时传输,数据在某节点相遇后即停止传播,所以不会无限制地重复转发,这就使得有限的带宽得到了有效的利用,取得很好的资源利用率。

3 HR 动态规划策略

在 HR 中,总是希望拥有数据多并且转发能力比较强的节点处在拓扑的高层。然而由于网络状态的不断改变和新节点的加入使得系统的结构不是很合理,因此采用一种动态规划的方法^[15],根据各节点的资源占有量、时延、带宽、抖动在系统中的存在时间,采用加权平均的方法计算出节点的优先级,据此在数据传输过程中,通过 heart-beat 算法动态改变拓扑中各节点的位置。让拥有大量资源和有效带宽的节点向上游浮动。

设节点在系统中的存在时间 $t = T_{nvw} - T_{arrive}$ (T_{nvw} 为现在时刻, T_{arrive} 为节点达到时间),数据的传输时延指数据从源节点传到目的节点的时间,这里简单使用 hop 衡量,用 R 表示。

设 $D_{a \rightarrow b}$ 为节点 a 传给节点 b 的数据信息,则在 T_{nvw} 时刻节点 $N_{(i,j)}$ 接受的数据为:

$$D_r = D_{N_{(i-1,j)} \rightarrow N_{(i,j)}} + D_{N_{(i,j-1)} \rightarrow N_{(i,j)}} + D_{N_{(i,j+1)} \rightarrow N_{(i,j)}} \quad (2)$$

发送的数据为:

$$D_s = D_{N_{(i,j)} \rightarrow N_{(i+1,j)}} + D_{N_{(i,j)} \rightarrow N_{(i,j-1)}} + D_{N_{(i,j)} \rightarrow N_{(i,j+1)}} \quad (3)$$

则在 t 时间内, $N_{(i,j)}$ 接受的数据可从自身资源表中获得为:

$$D_r(N_{(i,j)}) = S(N_{(i,j)}) \quad (4)$$

忽略传输过程中的速率变化,通过(4)式可近似得到节点发送的数据:

$$D_s(N_{(i,j)}) = \int_0^t D_r dt \quad (5)$$

由(4)式和(5)式可得出通过节点的数据流量总和为:

$$D(N_{(i,j)}) = D_r(N_{(i,j)}) + D_s(N_{(i,j)}) \quad (6)$$

则 T_{new} 时刻节点转发效率为:

$$\rho = D_s(N_{(i,j)}) / D(N_{(i,j)}) \quad (7)$$

t 时刻内效率为:

$$\rho' = D_s(N_{(i,j)}) / (D_r(N_{(i,j)}) + D_s(N_{(i,j)})) \quad (8)$$

判断一个节点是否应该向上提升,首先要考察该节点的转发效率。主要根据该节点的发送效率 ρ 和 ρ' 来决定。 ρ' 是节点存在过程中的效率,是为了弥补节点突发性传播大量数据造成的系统误认。但光有效率还不够,节点若想向其他节点传送数据,还需要本身拥有大量数据,所以可用节点的资源总量 $S(N_{(i,j)})$ 作为第二个参数,以决定节点上浮的层数。

再考虑节点在系统中的存在时间,让在系统中长时间存在的节点充当上游的主干节点有利于提高系统的稳定性,根据节点抖动对系统的影响,多次加入退出系统后只考虑最后一次。

由于上述的参数都具有即时性,考虑 T_{new} 时刻,节点 $N_{(i,j)}$ 的优先级为:

$$P(N_{(i,j)}) = e^{\alpha \rho} + \beta \frac{S(N_{(i,j)})}{S} + \gamma \frac{t}{T} + \delta \frac{1}{R} \quad (9)$$

($0 < \alpha, \beta, \gamma, \delta \leq 1$)

其中 T 为系统稳定运行的时间。将优先级划分与系统分层对应,即一定优先级的节点它们应该属于同一层。整个系统稳定后的状况一定是上层节点的优先级比下层高,同层之间则属于同一范围。

为了减轻 R 的负担,假设所有加入系统的节点都是友善的,将节点的优先级由节点自身计算后再传输给控制代理 R 以决定节点的位置。系统并不需要频繁地刷新所有的节点。

4 模型性能分析

考虑被动发送、主动接受的情况。用户总是希望能得到快速的服务,所以使用时间复杂度来衡量 HR 和其他模型之间的性能,着重比较 HR 和 NICE。假设系统中节点总数为 N 。

4.1 确定环中节点

对于每层环应包含的节点个数,以数据最终到达各节点所消耗的平均延迟时间为参数^[9],设节点总数 $N=10000$,则数据在正常传输过程中到达目标节点 $N_{(i,j)}$ 的平均时延总和为:

$$R_{i,j} = \begin{cases} (i+j)d & ① \\ (i+m/4+1/4)d & ② \end{cases} \quad (10)$$

其中①式表示数据由源节点发送到该环中第一个节点,再由该节点发送到目标节点的时延,②式则为由非零号节点传送到目标节点,表达式为节点的平均时延。则由(10)式可得系统的总传输时延为:

$$R_{total} = \sum_{m=1}^{5000} \sum_{i=1}^m \sum_{j=0}^{5000/m} [j(j+1)+i]d \quad (11)$$

其中 i 为节点的分层, j 为节点在环中的位置, d 为节点平均传输时延, m 为环中节点的个数。

仿真结果如图4所示,以总传输时延为参考标准,若要取得最小值,只要得到图形的拐点值即可,如图所示,出现了若干个拐点,这是由于参数 i, j, m 的相互影响所致。三个参数相互制约,在环的大小改变过程中,由于传输数据的不确定性和模型的不稳定性,使得曲线趋势出现多处变化。由图可知,在本试验中取 $m=300$,延迟最小。

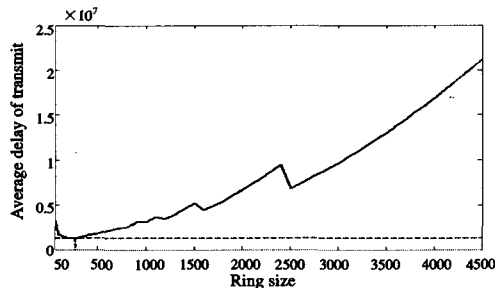


图4 节点处理时延

4.2 主要性能参数

一个好的应用层组播协议,应该有适中的控制负载、良好的网络负载平衡,并且要能满足接收者的性能需求。但目前还没有一个用于衡量应用层多播协议的效率和性能的统一的标准。常用的指标包括链路压力、路径伸展率、控制开销等。

建立了一个软件环境来模拟 HR,在一台 Linux 主机上应用 NS2 仿真平台。按照要求修改节点属性、包头结构和控制路由,通过逐渐添加节点的方式生成拓扑,最终达到稳定状态。

4.3 实验结果

网络的状态以学校校园网为参考,仿真结果如下。

在图5中平均连接压力 HR 始终要比 NICE 小 1/3 以上,这是由于 HR 的目标就是通过分片和有序转发减少链路中的重复传播;而对 NICE 来说当节点数增加时,由于更多覆盖路径共享了物理连接,压力很大。但是随着基数的变大,系统的分层增长缓慢,所以链接压力反而有变小的趋势。

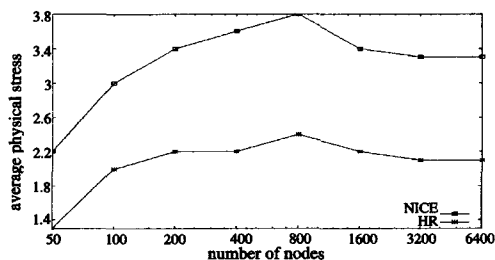


图5 平均链路压力

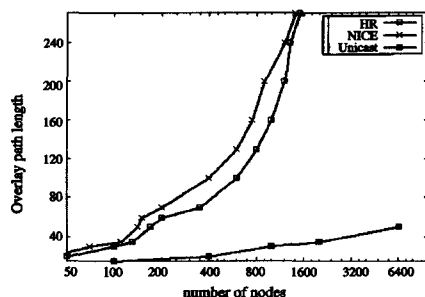


图6 路径长度分布

在图6中,由于 HR 的环状结构呈波浪形向外扩散,每一层的节点数量大致相同,这是为了满足十字架传输模型,当节

(下转第 80 页)

sion tolerant Systems using a state transition model// DARPA Information Survivability Conference and Exposition, vol. 2, 2001;211-221

- [5] Wu T, Malkin M, Boneh D. Building intrusion tolerant applications//DARPA Information Survivability Conference and Exposition, 2000, DISCEX '00, Volume 1, Jan. 2000;74-87
- [6] Liu P. Architecture for Intrusion Tolerant Database Systems// Computer Security Applications Conference, 2002, Proceedings, 18th Annual, Dec. 2002;311-320
- [7] Stroud R, Welch I, Warne J. A qualitative analysis of the intrusion-tolerance capabilities of the MAFTIA architecture// Dependable Systems and Networks, 2004 International Conference, June-July 2004;453-461
- [8] Huang Y, Arsenault D, Sood A. Incorruptible system self-cleansing for intrusion tolerance // Performance, Computing, and Communications Conference, 2006, IPCCC 2006, 25th IEEE International, April 2006;4
- [9] 郭渊博, 马建峰. 入侵容忍的国内外研究现状及所存在的问题分析. 信息安全与保密通信, 2005(7)

- [10] 彭文灵, 王丽娜, 张焕国, 等. 基于角色访问控制的入侵容忍机制研究. 电子学报, 2005, 33(1);91-95
- [11] 俞艳苹, 郭渊博, 马建峰. 基于自适应大数表决机制的容忍入侵模型. 系统工程与电子技术, 2005, 27(6);1098-1101
- [12] 荆继武, 冯登国. 一种入侵容忍的 CA 方案[J]. 软件学报, 2002, 13(8);1417-1422
- [13] 邹立新, 丁建立. 基于拜占庭协议的入侵容忍系统模型设计. 计算机工程, 2005, 31(增刊);88-90
- [14] 季称利, 杨晓元, 胡予濮, 等. 二方共享与 t_n 门限方案相结合的容侵 CA 方案. 计算机工程, 2005, 31(21);138-142
- [15] 殷丽华, 方滨兴. 入侵容忍系统的安全属性分析. 计算机学报, 2006, 29(8);1505-1512
- [16] 李伟生, 王宝树. 基于贝叶斯网络的态势评估. 系统工程与电子技术, 2003, 25(4);480-483
- [17] 黄光球, 孙周军, 刘兆明. 基于贝叶斯置信网的日志服务系统容侵方法研究. 微电子学与计算机, 2006, 23(12);53-60
- [18] 陈健, 李忠民, 王永明, 等. 基于贝叶斯网络的装备部件战斗损伤评估模型. 系统工程与电子技术, 2007, 29(2);329-332

(上接第 77 页)

点数目增加时, 导致层次的不不断扩大, 路径长度增长速度在一定比例时要高于 NICE, 这是为了节省带宽和提高效率而作的牺牲, 可能造成了节点接收数据的延迟, 但是这也只对组员较多的情况。

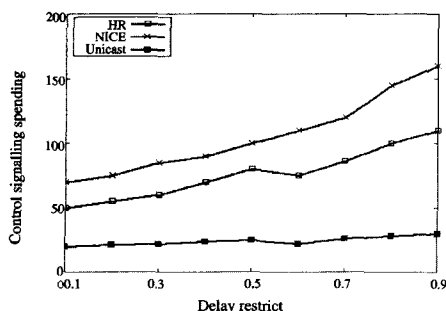


图 7 延迟和控制开销

图 7 表示的是多播组成员数量为 128 时不同延时约束下平均控制信令开销, 对于 NICE 和 HR 来说, 虽然在约束增加时, 平均开销都有所增加, 但是从图中我们可以看出 HR 的性能总要优于 NICE, 尽管实际的网络中协议的寻径还受网络带宽、延时抖动和包丢失率等的影响。

结束语 多播是一种重要的组通信模式。目前的 Internet 中, 由于 IP 多播部署的问题, 应用层多播 (ALM) 逐渐得到了广泛应用。文中提出了一种分层环状新型多播模型, 实现了分段传输和有序转发, 避免了资源在网络中无限制的相互发送和接收, 节省了带宽, 重点解决了在带宽很小的情况下如何提高多播机制系统性能问题。

模型采用有序泛洪算法, 使数据在层间单向传播, 在层内双向传输, 传输速率要比 NICE 快, 且只占用固定的带宽。同时由于有序传播, 使得数据出现差错时, 易于恢复, 下层节点只要向上一层发送重发请求就可以了。

采用动态规划方法, 使系统结构随时间增长不断趋于合理, 同时消除了分层结构中的 leader 节点, 将计算工作放在终端执行, 使系统更加稳定。

在安全性方面由于双环结构的自愈性而使系统更加健壮, 而消除了 leader 节点也使得系统不再受 leader 丢失的影响。

通过和 NICE 的比较以及仿真结果可知 HR 模型在性能方面优于以往的多播模型。下一步工作的重点在于动态规划过程中的优先权公式参数选择以及在数据传播过程中的内容

安全和节点安全问题, 另外应用层多播的差错恢复也是需要认真考虑的一个方面。

参考文献

- [1] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast // Proceedings of ACM Sigcomm, Aug. 2002;205-217
- [2] Chu Y-H, Rao S G, Zhang H. A Case for End System Multicast // Proceedings of ACM SIGMETRICS, 2002, 20(18);1456-1471
- [3] Stoica I, Morris R, Karger D, et al. Chord: a scalable peer-to-peer lookup protocol for Internet applications // Proceedings of ACM Sigcomm, 2003, 11(1);17-32
- [4] Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network. Ph. D. Thesis, University of California, Berkeley, October 2002
- [5] 龙白滔, 孙立峰, 陈文萍, 等. 具有层间冗余链路的应用层组播及其性能分析[J]. 电子学报, 2004, 32(11);1844-1848
- [6] 张冰, 原冰, 刘增基. 一种新的固定速率分层组播拥塞控制协议[J]. 中国科学, 2006, 33(10);23-28
- [7] Pendarakis D, Shi S, Verma D, et al. ALMI: An application level multicast infrastructure // Proc. of the 3rd Usenix Symp. on Internet Technologies and Sys. (USITS 2001), San Francisco, CA, Mar. 2001;49-60
- [8] Chawathe Y. Scattercast: an architecture for internet broadcast distribution as an infrastructure service [D]. USA: University of California, Berkeley, 2000
- [9] Sobeih A, Yurcik W, Hou J C. VRing: a case for building application-layer multicast rings (rather than trees); Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004;437-446
- [10] Zhao J, Yang F, Zhang Q, et al. LION: Layered Overlay Multicast With Network Coding Multimedia. IEEE Transactions, 2006, 8(5);1021-1032
- [11] Yiu W-P K, Chan S-H G. SOT: secure overlay tree for application layer multicast // Communications, 2004 IEEE International Conference, Volume 3, June 2004;1451-1455
- [12] Yiu W-P K, Wong K-F S, Chan S-H G, et al. Lateral error recovery for media streaming in application-level multicast Multimedia. IEEE Transactions, 2006, 8(2);219-232
- [13] Tian Ruixiong, Zhang Qian, Xiang Zhe, et al. Robust and efficient path diversity in application-layer multicast for video streaming Circuits and Systems for Video Technology. IEEE Transactions, 2005, 15(8);961-972
- [14] Gao L T, Sze T W, Crawford D, et al. A Cross-Functional Service-Oriented Architecture to Support Real-Time Information Exchange in Emergency Medical Response Hauenstein, Engineering in Medicine and Biology Society, 2006 // EMBS '06, 28th Annual International Conference of the IEEE Volume Supplement, 2006;6478-6481
- [15] Jin X, Wong W-C, Chan S-H G. Serving dynamic groups in application-level multicast High Performance Switching and Routing, 2005 // HPSR, 2005 Workshop, May 2005;432-436