计算机科学 2005Vol. 32№. 12

一种基于分发树切分的分布式聚集多播协议

刘志峰　吴国福　窦文华

（国防科学技术大学计算机学院　长沙 410073）

摘　要　随着网络组通讯应用的广泛开展，IP 多播将由于路由状态信息爆炸以及控制信息爆炸而面临严重的扩展性问题。在主干网中，这种状态可扩展性问题尤为严重。为了提高主干网中多播状态的可扩展性，本文提出了一种基于数据分发树切分的聚集多播协议——BEAMBTS（Bi-dirEctional Aggregated Multicast Based on Tree Splitting）。BEAMBTS 是一种简单而易于实现的、使用双向树的分布式协议。仿真试验显示，BEAMBTS 可以更好地改善状态可扩展性。

关键词　多播，聚集多播，分发树，切分，状态可扩展性

BEAMBTS: A Distributed Bi-dirEctional Protocol of Aggregated Multicast Based on Tree Splitting

LIU Zhi-Feng　WU Guo-Fu　DOU Wen-Hua

（School of Computer, National University of Defense Technology, Changsha 410073）

Abstract　With the enormous group communication applications, IP multicast confronts a severe scalability problem due to state explosion and control explosion. In backbone networks, this state scalability problem is exacerbated, since there are potentially enormous multicast groups crossing backbone domains. To improve the state scalability of multicast in backbone domains. This paper proposes a scalable protocol, called BEAMBTS(Bi-dirEctional Aggregated Multicast Based-on Tree Splitting), which uses the concept of aggregated multicast based on tree splitting. The analyses and simulations show that BEAMBTS can greatly improve state scalability: the number of aggregated trees is bounded in a small fixed number, and the multicast routing entries in transit nodes can be dramatically decreased.

Keywords　Multicast, Aggregated multicast, Spanning tree, Split, State scalability

1　引言

随着 Internet 中组通讯需求的不断增加，IP 多播作为一种高效的组通讯数据分发技术得到了越来越多的研究。然而，IP 多播距离广泛配置于 Internet 还有很远的路要走。在延缓 IP 多播广泛应用的问题当中，状态的可扩展性是其中一个重要的问题[1]。状态的可扩展性包括两个方面的问题：一是单个组中大量组成员的可扩展性问题，二是存在大量并发多播组时的可扩展性问题。本文主要研究第二个问题的解决方案。

在传统的 IP 多播协议中，路由器需要维护的多播转发状态信息随着多播并发组数目线性增长。由于每个报文的转发操作都包含了地址查找过程，因此转发状态数目的增加除了需要更多的存储空间之外，还导致了更慢的转发过程。另外，传统的 IP 多播协议要为每个组（或者组/源）构建和维护单独的分发树，大量的多播会话数目意味着需要构造和维护大量的分发树。在骨干网中，同时并发的多播会话组数目非常大时，传统的 IP 多播方案将面临严重的可扩展性问题。

为了解决转发状态的可扩展性问题，研究者提出了一系列的解决方案，这些方案可以分成四类：第一类方案将路由器中的多播转发状态完全地消除，代之以应用层多播 ALM（Application Layer Multicast）[2]。这类方案将复杂性交给了端节点，克服了传统 IP 多播面临的一些障碍，但是传输效率等另一些问题随之产生。第二类方案试图减少非枝节点中的状态，这些技术均假设目标网络中包含大量的稀疏组[3,4]。第三类方案试图通过类似 unicast 中的技术进行转发状态的聚集，从而达到减少状态的目的[5,6]。这类方案的效果和地址分配方案密切相关。第四类方案中多个多播组被强制共享同一棵分发树，这样网络中的多播分发树数目以及多播转发状态数目可以显著地减少[7]。A. Fei 和 J.-H. Cui 等人通过大量的仿真试验证明，聚集多播方案是一种有前景的解决传输域多播可扩展性问题的方案。然而，如果不同多播会话的规模尽可能不同，并且组成员尽可能分散的时候，基本的聚集多播 AM（Aggregated Multicast）方案性能较差。为此，本文作者在文[8, 9]中提出了一种基于分发树切分的聚集多播方案 AMBTS。理论和试验分析表明，AMBTS 性能明显优于 AM 方案，是一种对骨干网非常有效的多播实现方案。

本文第 2 节简单介绍基于分发树切分的聚集多播方案；第 3 节给出一种基于分发树切分的分布式聚集多播协议；第 4 节给出仿真试验结果及分析；最后进行总结。

2　基于分发树切分的聚集多播

AMBTS 的基本思想是：①预先将骨干网中的叶节点分成不同的组；②当一个多播组应用加入到主干网中，根据预先分配好的方案将这个组的叶节点进行分解，得到若干个子分发树；③分别对这些子分发树进行组-树匹配，获得一组聚集

刘志峰　讲师，博士研究生，主要研究方向为 IP 多播、应用层多播；吴国福　博士研究生，主要研究方向为计算机网络、多播数据通讯；窦文华　教授，博士生导师，主要研究方向为计算机网络、多媒体数据通讯。

子树；④用这组聚集子树完成原始多播组的数据分发任务。

每个多播会话有一个覆盖所有的组成员而不会将数据发送到非成员节点的“基本树”$T_0(G)$，$T_0(G)$可以通过一个多播路由算法例如 PIM-SM 或 CBT 直接计算出来。$T_0(G)$的一个划分可以表示为：

$$T_0(G)=\sum_{i=1}^{c}T_0^i(G) \tag{1}$$

其中 c 是对基本树 $T_0(G)$的划分数。如果一组聚集树 T_j，$j=1,2,\cdots,c$ 被分别作为 $T_0^i(G)$的代替，那么带宽浪费代价可以定义为：

$$\delta(G,\sum_{j=1}^{c}T_j)=\frac{C(\sum_{j=1}^{c}T_j)-C(T_0(G))}{C(T_0(G))} \tag{2}$$

其中，$C(T_0(G))$是树 $T_0(G)$的代价。从直观上看，$\delta(c_0,T)$反映了通过聚集树 T_j，$j=1,2,\cdots,c$ 发送数据时额外浪费的带宽的百分比。

随着网络中并发多播组数目的不断扩大，尽管进行了聚集，网络中需要维护的树的数目仍然是一个庞大的数字。如果预先对叶节点进行分类，那么网络中需要维护的树的数目将显著地减少。骨干网络中需要维护的分发树数目可以表示为：

$$\sum_{i=1}^{c}(2^{n_i}-1),n_1+\cdots+n_c=n \tag{3}$$

其中，c 表示分类数，n_i 表示第 i 类中叶节点数目。

从(3)式可以看出，分类数越多，需要维护的分发树数目将越少。但是，分类数越多，可能引入更多的带宽代价，并且给 RP 节点带来更复杂的报文地址封装处理负担。

3 BEAMBTS 协议

BEAMBTS 是为传输域定义的多播路由协议，它可以构建在任何域间路由协议之上，该协议仅仅需要从域间路由协议上获得组成员信息(MASC/BGMP 可以提供这些信息)。实现 BEAMBTS 协议的传输域称为 BEAMBTS 域。在 BEAMBTS 域中，每个聚集树均被指定一个域中唯一的多播地址，这个地址对其他域是透明的。数据报文在入端边界路由器节点进行封装，通过聚集树在域中分发数据，然后在出端边界路由器节点解封装。

为了适应单源和多源多播会话，BEAMBTS 使用核基多播方案。每个聚集树和某个 RP 节点关联。为了简化聚集树地址分配，这里可以使用简单多播的思想，每个聚集多播地址使用核心(RP)节点 IP 地址和多播地址二元组表示。当边界路由器收到加入组 G 的信息后，通过 Hash 函数(group-to-core)决定缺省 RP 节点 c_0。c_0 收到从边界路由器来的请求之后，将为组 G 构造适当的聚集多播子树集合。

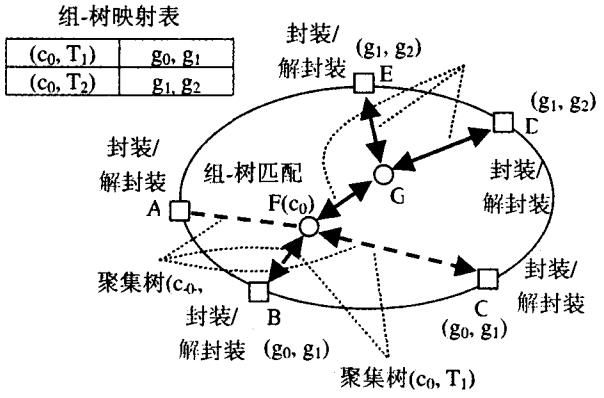


图 1 包含一个 RP 节点及两个子树的骨干网

为了进行组-树匹配过程，每个 RP 节点要维护下列信息：多播会话组表(包含每个组的组成员信息)、聚集树表(包含每个树的叶节点信息)以及组-树映射表。当 RP 节点 c_0 收到组 G 的加入或退出信息时，它要引发组-树匹配过程。

如果 G 变空(最后一个成员离开)，在多播会话组表中删除相应的表项；如果为 G 分发数据的聚集多播子树不为其他组分发数据，那么还要更新组-树映射表；如果 G 是一个新的组(第一个加入成员)，在多播会话表中插入相应的选项。

边界路由器中需要维护一个组-树映射表，表中包含该节点参与的多播会话以及相应的聚集多播子树。

为了便于讨论，协议消息分为两类：M-Type 和 B-Type。所有底层多播路由消息为 M-Type 消息，而帮助实施组-树匹配的消息属于 B-Type 消息。

下面以图 1 所示的域为例，说明 BEAMBTS 协议工作过程。域中边界节点包括(A,B,C,D,E)，预先分成两组(A,B,C)和(D,E)。聚集树(c_0,T_1)包含边界节点 B 和 C，聚集树(c_0,T_2)包含边界节点 D 和 E，聚集树(c_0,T_3)包含边界节点 A、B 和 C。组 g_0 初始包括组成员 B 和 C，组 g_1 初始包括组成员 B、C、D 和 E，组 g_2 初始包括组成员 D 和 E。

3.1 成员加入过程

当边界路由器 r 收到从其他域发来的加入组的请求时，首先使用 Hash 函数 group-to-core 得到组 G 的缺省 RP 节点 c_0，然后向 c_0 发送加入消息 B-JOIN(G,c_0)。c_0 引发树管理模块寻找或构造适当的聚集子树集合(c_0,Σ_jT_j)，$j=1,2,\cdots,d$，d 为叶节点分类数目。c_0 将给节点 r 发送消息 B-JOIN-ACK$(G,(c_0,T_j))$，如果 r 没有加入分发树(c_0,T_j)，那么通过发送 M-JOIN(c_0,T_j)消息即可加入到该分发树，并更新组-树映射表项。如果分发树(c_0,T_j)不在现有聚集分发子树集合中，则要将其加入|MTS|。

3.2 成员退出过程

如果边界路由器 r 希望离开组 G，它将向 RP 节点 c_0 发送退出消息 B-LEAVE$(G,(c_0T_j))$。RP 节点收到 B-LEAVE(G)消息后要运行组-树匹配算法。如果使用当前的聚集子树集合(c_0,Σ_jT_j)导致带宽负担代价过高，那么将引发树迁移过程。如果子树(c_0,T_j)中没有其他组 G 的成员，c_0 要更新组-树映射表，删除相应的表项。如果这个节点不属于其他组，那么这个信息将引发一个 M-LEAVE(G_{T_j},c_0)消息。如果该聚集子树的最后一个节点发送 M-LEAVE(G_{T_j},c_0)消息，则该聚集子树将从聚集多播子树集合|MTS|中删除。

3.3 树迁移过程

当组 G 的成员发生改变，初始的聚集子树集合(c_0,Σ_jT_j)不能覆盖组 G，或者带宽负担代价超过给定的限度，那么就需要引发树迁移过程。假设需要进行迁移的基本子树是 T_j^0，根据 3.2 节的算法，c_0 将为基本子树 T_j^0 找到另一个聚集子树(c_0,T_j')，然后沿着树(c_0,T_j)发送消息 B-TREE-SWITCH$(G,(c_0,T_j),(c_0,T_j'))$，以通知组 G 的基本子树 T_j^0 上的其他成员加入(c_0,T_j')并退出(c_0,T_j)。组 G 的成员节点根据需要发送 M-JOIN 或 M-LEAVE 消息。

3.4 RP 迁移过程

在上述树迁移过程中，迁移仅仅局限在一个 RP 核心节点。为了使更多的组能共享同一棵树，BEAMBTS 允许 RP 核心迁移。

在组 G 的生存周期内，初始聚集子树集合$(c_0,\Sigma T)$可能太大或太小，除了通过单 RP 节点树迁移操作之外，还可以将

整个组向其他 RP 节点迁移。当 c_0 没有合适的聚集子树集合覆盖组 G，c_0 将通过预定义的连接所有 RP 节点的双向多播分发树(c_{core}，T_{core})发送 RP 迁移请求消息 B-CORE-SWITCH-REQ，消息中包含了组 G 的成员信息。其他 RP 节点收到这个消息后将运行树管理模块，如果某个 RP 节点 c' 处存在聚集子树集合(c'，$\Sigma T'$)可以覆盖组 G，该节点将单播发送请求确认消息 B-CORE-SWITCH-ACK 给 c_0，该消息中包含了 RP 节点 c' 的标识以及相应的带宽负担代价。c_0 从收到的确认消息中选择一个带宽代价最小的 RP 核心 c'' 及聚集子树集合(c''，$\Sigma T''$)作为组 G 的新 RP 核心及聚集分发子树集合。然后，一个树迁移过程将引发，使得其他组成员加入(c''，$\Sigma T''$)并退出(c_0，ΣT)。c_0 还要记录组 G 的新 RP 核心信息，这样当新成员节点 r 试图加入组 G 时，c_0 将发回消息 B-CORE-CHANGE(G，c'')，消息中包含组 G 的新 RP 核心信息 c''，这样 r 将发送另一个 B-JOIN 信息给 c''。

RP 核心迁移对于减少需要维护的聚集树数目及减少多播转发状态有一定积极意义，但是操作比较复杂，实际应用中可以考虑仅仅将相对稳定的多播会话进行迁移操作。

3.5 RP 协作过程

在 RP 迁移操作中，如果组 G 的缺省核心 RP 节点 c 没有能够完全覆盖 G 的所有成员的聚集子树集合，而另一核心 c' 中则包含一个能完全覆盖 G 的所有成员的聚集子树集合，那么为了提高聚集效果，G 的 RP 节点将由 c 迁移到 c'，在 RP 节点迁移的过程中所有的组成员都要重新开始加入过程，这是不必要的，RP 协作就可以很好地解决这个问题。

如果 RP 节点 c 的一个叶节点分类(c，$\{\{r_{11}\},\cdots,\{r_{1m}\}\}$)和 RP 节点 c' 的一个叶节点分类(c'，$\{\{r_{21}\},\cdots,\{r_{2n}\}\}$)中叶节点存在交集(crossset)。对于交集的某个子集 subcrossset $=\{\{e_1\},\cdots,\{e_d\}\}$，$\{e_i\}\in$ crosset，$i=1,\cdots,d$，当 RP 节点为 c 时分发树为(c，T_1)，当 RP 节点为 c' 时分发树为(c'，T_2)，其费用分别为 COST1 和 COST2。如果 COST1≤COST2，那么缺省核心为 c'，组员包括 subcrossset 的子组可以用子树(c，T_1)分发数据。如果 COST2≤COST1，那么缺省核心为 c，组成员包括 subcrossset 的子组可以用子树(c'，T_2)分发数据。数据以隧道的方式在核心 c 和 c' 之间传递。

RP 协作的过程包括下面几步：①希望其他 RP 节点提供协作的 RP 节点通过预定义的连接所有 RP 节点的双向多播分发树(c_{core}，T_{core})发送 RP 协作请求消息 B-CORE-COOP-REQ，消息中包含了组 G 的某个分类的成员信息$\{\{r_1\},\cdots,\{r_m\}\}$；②其他 RP 节点收到这个消息后，将运行树管理模块，如果某个 RP 节点 c' 处存在聚集子树(c'，T')，其叶节点包含$\{\{r_1\},\cdots,\{r_m\}\}$，该节点将单播发送请求确认消息 B-CORE-COOP-ACK 给 c_0，该消息中包含了 RP 节点 c' 的标识以及相应的树费用；③c_0 从收到的确认消息中选择一个树费用最小的 RP 核心 c'' 及聚集子树(c''，T'')作为组 G 的分类成员的$\{\{r_1\},\cdots,\{r_m\}\}$分发树分发数据；④$c_0$ 和 c'' 之间通过单播隧道的方式传递组 G 的分类成员$\{\{r_1\},\cdots,\{r_m\}\}$与其他成员之间的信息。

显然，RP 协作更好地提高了聚集效果，网络中需要维护的聚集树数目得到进一步减少，这对于在具有较多 RP 节点的传输域应用基于分发树切分的聚集多播来说是一个很好的结果。

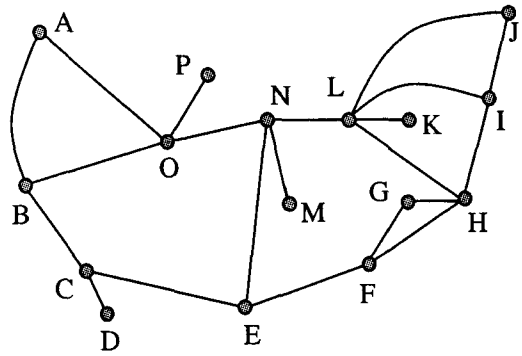


图 2　vBNS 骨干网络拓扑结构图

4　仿真试验及结果分析

在仿真试验中，为了对比 BEAMBTS 与 BEAM 的性能差异，我们使用了一个实际的骨干网络拓扑结构——美国下一代 Internet 试验网 vBNS IP 主干网(图 2)。在 vBNS 主干网络中，有 16 个核心路由器(它们不会是任何多播组的端节点)，假设每个核心路由器都与一个边界节点连接。这 16 个叶节点可能参与到任何的多播组会话当中。由于缺乏大规模多播应用的运行轨迹，在仿真试验中，使用文[7]中提出的带随机权重的节点模型组模型。这种模型中，每个节点都被指派一个权重，以描述该节点加入一个组的可能性，从而控制了组的规模，即会话密度。

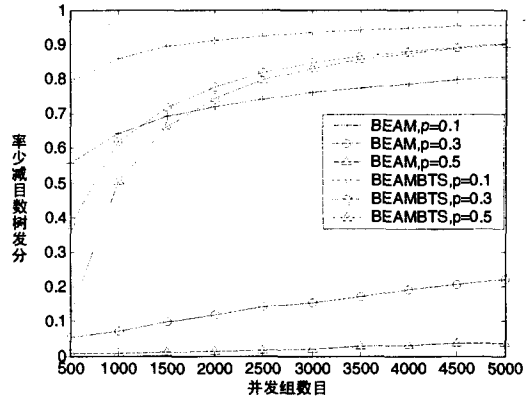


图 3　单 RP 节点时，TSORR vs 并发组数目

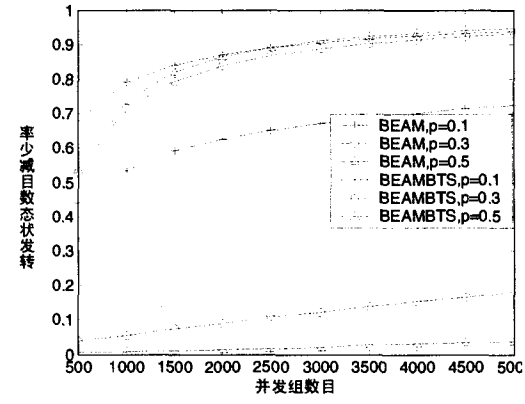


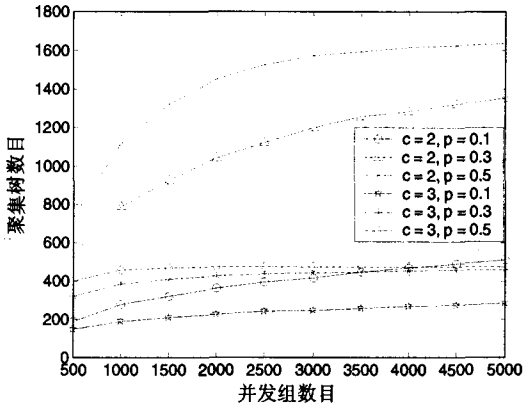
图 4　单 RP 节点时，SRR vs 并发组数目

图 3 中比较了 BEAM 和 BEAMBTS 都进行完全匹配时分发树数目减少率 TSORR(Tree Setup Overhead Reduction Rate)与并发组数目及会话密度之间的关系。从图中可以看出，TSORR 随着并发组数目的增加而变大，并且随着会话密

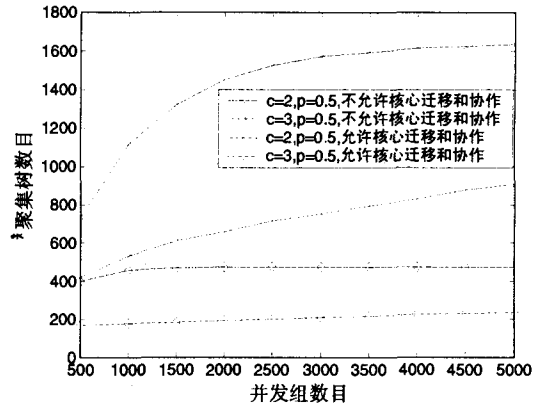
度的不同而有明显差异。随着组密度的增加，分发树数目减少率增加，会话密度相同时，BEAMBTS 获得的状态减少率明显高于 BEAM。当会话密度达到 0.5，并发组数目为 5000 时，BEAM 获得的 TSORR 仅为 0.38%，而此时 BEAMBTS 则获得 89.80%。

图 4 中比较了 BEAM 和 BEAMBTS 都进行完全匹配时，传输节点转发状态减少率 SRR(State Reduction Rate)与并发组数目及会话密度之间的关系。从图中可以看到，SRR 和 TSORR 曲线有相同的趋势，随并发组增加而加大。采用 BEAMBTS 协议时，SRR 受组密度的影响明显小于 BEAM 协议。

图 5 给出了骨干网中包含多个 RP 节点时 BEAMBTS 协议的性能分析。从图中可以看到，当分类数 c=2，并且并发组数目较小时，BEAMBTS 要维护较多的分发树数目。如果将分类数目 c 加大到 3 时，BEAMBTS 需要维护的聚集分发树数目将大为减少。图 5(b)，给出了允许 RP 核心迁移和协作时聚集分发树数目与并发组数目之间的关系。从图中可以看出，如果以一定的管理负担为代价，那么允许核心迁移和协作时，BEAMBTS 仅仅需要维护很少的聚集分发树。



(a) 不包括核心迁移和核心协作的情形



(b) 包括核心迁移和核心协作的情形

图 5　包含 3 个 RP 核心时核心迁移和协作对聚集分发树数目的影响

结论　为了解决多播状态的可扩展性问题，本文在基于分发树切分的聚集多播的基础上，提出了一种基于分发树切分的分布式聚集多播协议 BEAMBTS，详细描述了工作过程，BEAMBTS 协议简单，易于实现，从仿真结果可以看到，当骨干网络中并发组数目较大时，BEAMBTS 协议可以取得比 BEAM 更好的聚集效果。允许核心迁移和协作的 BEAMBTS 协议极大地缩减了网络中可能需要维护的分发树数目，充分发挥了聚集多播的优势，对大规模并发的多播组应用而言具有良好的可扩展性，是一个对传输域非常有效的多播实现方案。

参 考 文 献

1 Almeroth K. The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment. IEEE Network, 2000,14(1):10～20

2 El-Sayed A, Roca V, Mathy L. A survey of proposals for an alternative group communication service. IEEE Network, 2003 (special issue on Multicasting: an enableing technology)

3 Costa L H M, Fdida S, Duarte O C M. Hop-by-hop multicast routing protocol. In: SIGCOMM'01. San Diego, CA. 2001. 249～262

4 Tian J, Neufeld G. Forwarding state reduction for sparse mode multicast communications. In: IEEE INFOCOM'98. 1998. 711～719

5 Thaler D, Handley M. On the aggregatability of multicast forwarding state. In: IEEE INFOCOM'00. 2000. 1654～1663

6 Radoslavov P I, Estrin D, Govindan R, Exploiting the bandwidth-memory tradeoff in multicast state aggregation. Technical report, USC Dept. of CS Technical Report 99-697 (Second Revision). July 1999

7 Cui Jun-Hong, Kim Jinkyu, Maggiorini D, et al. Aggregated Multicast — A Comparative Study. In: the special issue of Cluster Computing: The Journal of Networks, Software and Applications, 2003

8 刘志峰，窦文华. 一种基于分发树切分的聚集多播方案. 计算机研究与发展，2004,41(11)：1895～1901

9 Liu Zhifeng, Dou Wenhua, Liu Yajie. AMBTS: A Scheme of Aggregated Multicast Based on Tree Splitting. In: IFIP Networking'04. Athens, Greece, 2004. 829～830

(上接第 29 页)

簇算法的网络结构的稳定性和拓扑管理性能的高低。从图 3(a)和图 3(b)可以看出，相关度算法比节点度算法有着更好的表现，尤其是在 pause time 比较小即节点移动性比较大的情况下，相关度算法在端到端延时上性能提高显著。

结束语　移动自组网的移动特性导致了节点间拓扑关系的变化，在分层次的网络结构中节点间的相互关系直接影响着网络节点的角色和簇结构。本文首先提出了节点间相关度的定义，并在相关度的基础上提出了节点相关密度的概念。利用相关度和相关密度作为分簇的依据，提出了一种分布式的分簇算法，并对此算法进行了分析和模拟测试。由于充分考虑了节点周围的节点分布情况，测试结果表明，同节点度算法相比，该算法具有更好的分簇均衡性，也在应用的吞吐率、延时方面具有更好的性能。

参 考 文 献

1 Kawadia V, Kumar P R. Power Control and Clustering in Ad Hoc Networks. In: Proc. Infocom, 2003

2 王海洵，张学涛. ad hoc 网络中的分簇算法. 数据通讯，2003,3:32～34

3 Gerla M, et al. Mobile, Multimedia Radio Network. Wireless Networks,1995,1(3):255～265

4 Chatterjee M, Das S K, Turgut D. WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking, 2002, (5):193～204