第34卷 第17期 计 算 机 工 程 2008年9月
Vol.34 No.17 Computer Engineering September 2008
·网络与通信· 文章编号：1000—3428(2008)17—0082—03 文献标识码：A 中图分类号：TP393

基于NICE协议的应用层组播可靠性研究

周国伟，陈 越，邵 婧
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：目前许多应用层组播协议缺乏明确的数据可靠传输保障机制。该文将NICE协议中原有的控制拓扑和数据拓扑改为环形的控制拓扑和最小延迟树的数据拓扑，在新的拓扑结构上建立差错控制机制，以保证组播的可靠性。仿真实验表明，改进后的组播协议在节点正常工作时具有完全的可靠性，在节点失效的情况下同样具有良好的健壮性。

关键词：应用层组播；差错控制；可靠组播；覆盖网

Research on Reliability of Application-layer Multicast Based on NICE Protocol

ZHOU Guo-wei, CHEN Yue, SHAO Jing
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004)

【Abstract】 Many application-layer multicast protocols are short of definite reliable transmission mechanisms. This paper changes the topology of control and data in NICE protocol, uses ring and minimal delay tree, and adds the corresponding error control mechanism to improve the reliability of NICE. Simulations show that the improved protocol has complete reliability in case all nodes work well and achieves a high delivery ratio when some nodes fail.

【Key words】 application-layer multicast; error control; reliable multicast; overlay

1 概述

目前许多组播应用不同程度地要求数据的可靠传输，如视频会议、网络游戏、交互式仿真。可靠组播正是针对不同的应用实现不同程度、不同要求的数据可靠传输[1]，没有可靠性保障的组播通信将无法在Internet中普及应用。以往可靠组播的研究都是针对IP组播的，而IP组播本身由于技术和非技术因素目前仍无法普及，这就导致基于IP层的可靠组播的应用存在很多局限性。

应用层组播在应用层构建主机之间的逻辑树，不依赖于下层网络设备，所以，与IP组播相比更容易配置，便于在目前的网络条件下推广使用，是一种很有潜力的技术。但在可靠性方面，其下层采用单播方式，传输层使用UDP协议，只提供尽力而为的服务。已有的许多应用层组播协议如NICE, Narada仅重点解决主机间的互连，而没有考虑组播的可靠性[2]。因此，如何提高应用层组播的可靠性是一个亟待解决的问题。

许多文献对应用层组播的可靠性进行了研究，但侧重点不同。文献[2]提出了一种差错恢复机制LER(Lateral Error Recovery)，方法是将主机随机分配到不同的层面中以减小主机之间的错误关联度。本文采用这种设计思想构造合适的覆盖网，使所用的差错控制机制在此覆盖网下最为有效。文献[3]使用数据恢复机制PRM(Probabilistic Resilient Multicast)来提高数据传输率，同时维持一个低的终端间的延迟，并将其应用于NICE协议。本文借鉴了上述2种方法，采用环形控制拓扑和最小延迟树的数据拓扑，使差错检测和恢复分别在环中与树中进行，节省了控制开销，避免了反馈风暴问题；选取能力较强的节点担任领导节点，增强了组播的性能；充分利用了环形拓扑对单点失效的免疫，提高了NICE应用层组播的健壮性。

2 NICE协议[4]及可靠性分析

NICE协议主要采取了层次化的节点集群思想。它可以支持大量不同的数据转发树，有较强的可扩展性。

2.1 NICE中的层次性拓扑结构

如图1所示，NICE的层次结构是把成员节点分到不同的层次中，把每个层的主机分到不同的集群中。每个集群的规模在K~3K-1之间，其中，K为常数。集群由邻近的主机组成，而且每个集群有一个领导节点。NICE协议将根据集群的拓扑结构选择中心节点作为领导节点，领导节点到其他所有节点的距离之和在这个集群中是最小的。

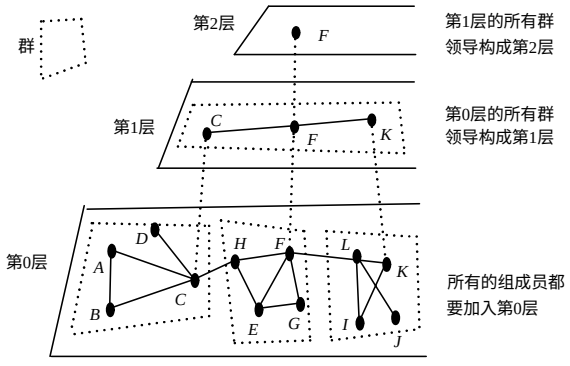

图1 层次性拓扑结构

作者简介：周国伟(1983－)，男，硕士研究生，主研方向：应用层组播技术；陈 越，副教授、博士；邵 婧，硕士研究生

收稿日期：2007-10-10 **E-mail：**zgw029022l@126.com

2.2 控制拓扑和数据拓扑

节点的层次结构同时用于控制拓扑和数据拓扑。在控制拓扑中，每个集群的成员之间相互发送周期性的更新消息。一个集群中的成员构成一个随机的网状结构，如图 1 中第 0 层的 3 个网状集群。数据拓扑基于转发规则从控制拓扑中直接得到，是一个呈中心散射状的基本路径树。

2.3 节点的加入和退出

新节点将根据距离度量选择加入离自己最近的 *L*0 中的某个集群。加入过程从最顶层的节点开始，顺序探测每个层次的集群直到找到最近的集群为止。

2.4 基于 NICE 协议的组播结构可靠性分析

NICE 协议采用了一种基于隐含组播转发的拓扑结构策略，领导节点是数据转发的关键。在组播数据包时，由于主机的缓存溢出、拓扑结构的变化或节点的失效等情况都会导致数据包的丢失，影响组播的可靠性，因此本文重点解决以下 2 个问题：

(1)如何恢复丢失的数据包。在组播树正常工作时，由于缓存溢出或拥塞等原因而出现数据包丢失时，采用何种差错恢复策略是可靠性保证的关键。一般差错恢复机制中都会使用反馈机制，但无论是 ACK 反馈还是 NACK 反馈都存在反馈风暴问题，这直接影响到组播的可扩展性。

(2)节点失效问题。应用层组播利用终端主机实现路由器的复制与转发功能，而主机的稳定性显然不如路由器，会随时失效，甚至存在非法离开和恶意破坏等情况。领导节点的失效更会导致集群中所有成员接收不到数据，出现区域性失效的情况。在数据传输树重构以前，受影响的节点将无法接收到后面的数据。

3 改进的可靠 NICE 协议

与以往 IP 组播的可靠性研究相比，应用层组播的最大优势在于覆盖网拓扑构造的灵活性。对环和树形结构的进一步研究发现：在控制拓扑中使用环结构而在数据拓扑中使用最小延迟树结构便于进行差错控制，从而可以提高协议的可靠性。因此，对 NICE 协议的改进主要是对控制拓扑的改变，但仍保留分层和分群的结构模式。另外增加了相应的差错控制机制和节点失效恢复机制来解决影响 NICE 协议可靠性的 2 大关键问题。

3.1 覆盖网的组织

相对 NICE 协议原有的网状拓扑，采用环形拓扑有着如下优势[5]：(1)每个节点的节点度恒定为 *O*(1)且独立于组规模的扩展；(2)在环中很容易实现安全、可靠和全序的消息传递；(3)环形结构对单点失效免疫，且节点失效的恢复算法简单有效。采用环形拓扑后 NICE 协议的层次拓扑如图 2 所示。

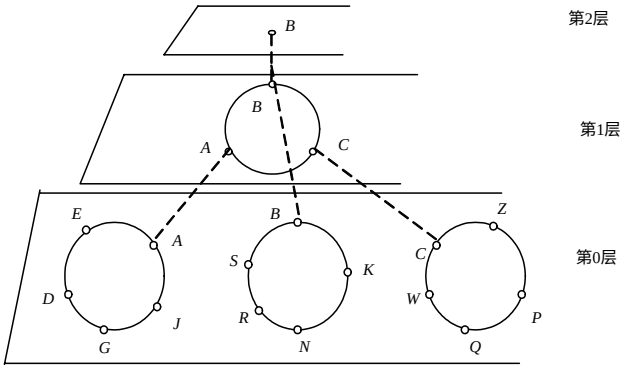

图 2 改进的环形层次拓扑

环的初始化与维护过程与 VRing 环[6]的协议描述类似。在 NICE 协议中，节点在加入和分群时都有一个 Rendezvous Point 管理者，简称 RP(NICE 协议中假设所有成员都事先知道存在这样一个特殊主机)。同样这个 RP 也参与环的形成，而且每个集群的领导节点就是环的领导者，由此与 VRing 环的协议中的元素对应起来。

领导节点是 NICE 协议中的重要元素。原 NICE 协议是根据集群的拓扑结构选择中心节点作为领导节点的，这就忽视了实际应用中主机之间能力的差异。而领导节点恰恰是集群中负担最重的节点，如果中心节点是一个处理速度很慢的主机，那么整个集群的性能会受到严重影响。因此，要选择集群中能力最强的主机担任领导节点，这个过程由 NICE 协议中的 RP 管理者完成。

为了避免在数据分发时出现循环的情况，改进后的数据拓扑仍采用基于源的树形结构。原 NICE 协议使用的是基本路径树，这种数据传输树不是一棵基于约束的优化树，而应用层组播中数据传输的路由优化是影响组播性能的关键因素。同时由于领导节点不再是集群拓扑结构的中心节点，它到其他所有节点的距离之和可能不是这个集群中最小的。

综合以上因素，为了减小传输时延和差错恢复时延，本文采用最小延迟树作为数据拓扑。因为在每个集群中，领导节点通过信息交换了解集群中所有节点之间的延迟信息，所以采用集中式算法，由领导节点执行 Dijkstra 算法形成最小延迟树。

3.2 局部的差错控制机制

目前差错控制采用的基本技术有 3 种：自动重复请求(ARQ)，前向纠错(FEC)和混合差错控制(HEC)。根据差错控制组成员的参与程度，可分为集中式差错恢复和分布式差错恢复。

充分利用 NICE 协议集群的思想，将差错恢复范围局限在一个集群中。因为集群中成员的数量保持在 *K*~3*K*-1 之间，只要选取适当的 *K* 值就可以限制成员的数量，所以改进后的 NICE 协议仍具有良好的可扩展性。

NICE 协议的特点是一个集群中成员的数量较少，领导节点是距离其他成员最近的节点，因此，可以采用集中式局部差错控制方式。差错控制包括差错检测和差错恢复 2 个部分，本文将 2 种功能分别在环形和树形结构中实现，即在环中进行差错检测，在树中进行差错恢复。其基本思想是：发送者沿最小延迟树发送一个数据包后，立刻在环形控制拓扑中发送一个 check 检测消息，由于最小延迟树中每个节点到发送源的延迟是最小的，因此数据包必然先于 check 消息到达每个节点，环中的成员收到 check 消息后检测自己是否收到数据包，若收到，则继续向下游传递 check 消息，若丢失，则在 check 中进行标记，最终 check 消息回到发送者，发送者对返回的 check 消息进行统计判断，决定如何重传数据包，随后沿最小延迟树重传数据包。

现在以一个具体的集群为例进行分析。控制拓扑结构及其最小延迟树如图 3、图 4 所示。领导节点 *A*0 在发送了一个数据包后，就在环形控制链路上传播一个 check 消息，用于检测是否每个节点都收到正确的数据包。每个接收到 check 消息的节点如果需要重传数据包，就在消息中标记出自己的 ID，最后 *A*0 根据返回的 check 消息计算出需要重传的成员数目 *N*，并根据 *N* 的大小和发生包丢失的组成员的位置来决定重传的方式。

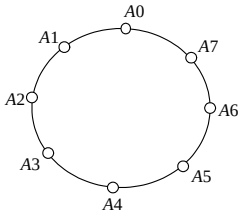

图 3　环形控制拓扑

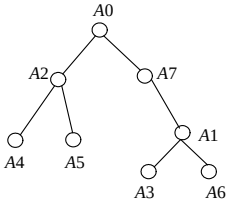

图 4　最小延迟树

例如在图 4 中，若 *A*4 和 *A*3 有包丢失，*A*0 通过路径 *A*0-*A*2-*A*4 和 *A*0-*A*7-*A*1-*A*3 重传数据包；若 *A*4, *A*5, *A*3, *A*1 都有丢失，则 *A*0 沿最小延迟树重新组播数据包。

因为每个集群的组成员数目不同，所以重传方式由领导节点根据丢失包的成员数目和集群的大小来决定。如果领导节点自己丢失数据包，那么它将由其所在的更高层集群恢复。

3.3　节点失效

数据是依靠数据拓扑即最小延迟树进行传输的，因此，中间节点(非叶子节点)的失效必然导致树的中断，它的孩子节点无法接收到正常数据，在重构树以前，这种情况将持续。在应用层组播中，这种节点失效经常发生。这时，应采用如下差错控制机制：领导节点在发现环中有节点失效时，不再使用 check 消息进行差错检测，但依旧沿最小延迟树传播数据。节点在时间 *T*(*T* 为 check 消息在环中循环一周所用的时间)内未收到下一个数据包以及来自领导节点的 check 消息，就认为父节点失效，从而转向环形拓扑，向上游和下游节点发送 NACK 消息要求恢复未收到的数据。上游和下游节点收到请求后，如果有所需的数据，就沿环形拓扑传输给请求节点，节点执行重复抑制机制丢弃重复的数据包。如果上下游节点均失效，则等待，不过这种情况出现的概率非常低。

环形控制拓扑结构具有良好的防单点失效能力，在集群中采取集中式的管理策略，领导节点是管理者。在环形拓扑中传递一个“心跳”消息 HEARTBEAT()来检测是否有节点失效，领导节点在限定时间(*timeout*)内没有收到发出的 HEARTBEAT()消息就认为有节点失效，随后，领导节点发送 REPAIR()消息进行环的修复。每个节点都有一个前向和后向节点列表，列表存储了与它相邻的前 *m* 个和后 *m* 个节点信息，*m* 的大小仍旧根据 *K* 值和具体要求而定。如图 3 所示，若节点 *A*4 失效，*A*0 作为领导节点发现存在节点失效后，随即发出 REPAIR 消息，每个收到 REPAIR 消息的节点依据它的相邻节点列表向后向邻节点发送 HELLO 消息，若收到邻节点的回应 FINE 消息就继续向下游转发 REPAIR 消息，直到 *A*3 收到。*A*3 向 *A*4 发送 HELLO 消息而没有回应，则执行节点恢复算法，依据节点列表向 *A*5 发送 link_repaire 消息，并将后向节点域设置为 *A*5，*A*5 回复一个 Link_repair_ack 消息，将自己的前向节点域设置为 *A*3。

单点失效恢复算法的代码可以参考文献[6]。由于每个节点存储了相邻 *m* 个节点的信息，因此即使连续多个节点同时失效，只要节点数不超过 *m*，环都可以修复。

若领导节点失效，对更高一层的集群来说可能只是非领导节点的失效，所以，算法仍然适用于高层集群。从失效领导节点所在的集群中选出新的领导节点后，再形成新的控制拓扑关系。

4　仿真实验

采用 NS2 网络仿真工具对 NICE 协议在改进前后的可靠性进行对比分析。首先使用 GT-ITM 生成 transit-stub 类型的物理网络拓扑，拓扑节点为 512 个。NICE 协议的参数设定如下：*K*=5，最大层数为 10。在所有节点正常工作和有节点失效 2 种情况下分析改进前后 NICE 协议的可靠性变化，使用数据传递率作为数据可靠传输的判定依据。数据传递率为成功接收到的数据包数与源发送的总数据包数的比率。如图 5、图 6 所示。

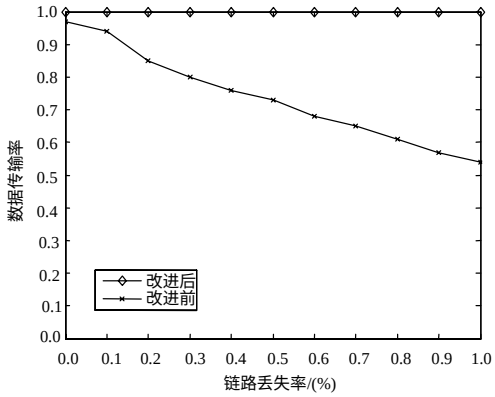

图 5　不同链路丢失率下的数据传输率

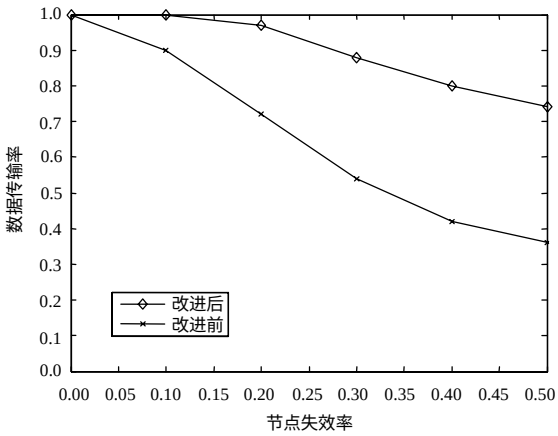

图 6　不同节点失效率下的数据传输率

图 5 是组播节点都正常工作的情况下，人为地在仿真中设置包在链路中的丢失率。由于原 NICE 协议没有差错控制机制，是一种尽力传递的服务，因此随着丢失率的增加，传输率也迅速下降，而改进的协议则实现了完全可靠的传输。

图 6 是在所有组播成员中随机选取部分失效节点，其上限取为 50%。这时组播基本呈现严重失效状态，在实际中这种情况极少出现。可以看出，在失效节点数达到 20%以前，改进的 NICE 协议仍具有良好的数据传输率。

5　结束语

本文对 NICE 协议进行了改进，增加了差错控制机制，提高了基于 NICE 协议的应用层组播数据传输的可靠性，并且保持了 NICE 协议的可扩展性。从目前的可靠性应用来看，本文所采用的差错恢复机制可以实现完全可靠的传输，适用于文件传输、软件升级等要求无误传输而对时延要求不高的应用。但在一些具体问题上仍需进一步研究，比如，如何根据具体的应用选取合适的 *K* 值来限制集群规模；如何使覆盖网拓扑与下层实际链路较好地吻合以提高构建应用层网络的性能。

（下转第 87 页）