

Ad hoc 网络基于协同的可靠组播丢失恢复算法设计

黄梦洁 冯 钢 张翼德

(电子科技大学通信抗干扰技术国家级重点实验室 成都 611731)

摘 要: 在 Ad hoc 网络中保证组播通信的可靠性要面对 Ad hoc 网络高误码率、带宽受限、节点能量有限和拓扑结构频繁变化等技术挑战。该文将协同的思想引入到组播丢失恢复设计中, 设计了新的基于协同的可靠组播丢失恢复算法(CoreRM)。根据各个节点经历的不同丢失情况, 通过自适应选择本地恢复、全局恢复或发送端恢复, 分布式地处理整个网络的丢失恢复。CoreRM 还设计了恢复路径缓存、NAK 抑制机制, 以及源路由数据包(SPM)发送机制来应对 Ad hoc 网络中的拓扑变化。性能分析和 NS2 的仿真实验表明相对于 UDP 和 PGM 可靠组播通信, CoreRM 算法可以在网络吞吐量和丢失恢复延时方面有显著性能改善。

关键词: Ad hoc 网络; 可靠组播; 协同通信; 丢失恢复

中图分类号: TN915

文献标识码: A

文章编号: 1009-5896(2010)09-2065-07

DOI: 10.3724/SP.J.1146.2009.01054

CoreRM: Cooperative Loss Recovery for Reliable Multicast in Ad hoc Networks

Huang Meng-jie Feng Gang Zhang Yi-de

(National Key Laboratory of Communications, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Reliable multicast in Ad hoc networks is challenging due to high error rate, low bandwidth, limited node power, and highly dynamic topology changes. In this paper, an efficient loss recovery scheme is proposed for reliable multicast (CoreRM). The basic idea is to apply the notion of cooperative communication to support local loss recovery. According different packet loss cases, CoreRM tries to retrieve the lost packet through progressively cooperating with neighboring nodes, upstream nodes or source node. In order to minimize recovery latency and retransmission overhead, CoreRM caches not only data packets but also the path which could be used for future possible packet recovery. In CoreRM, upstream nodes suppress duplicate NAKs to tackle feedback implosion. Both analytical and simulation results reveal that CoreRM significantly improves the multicast performance in terms of throughput and recovery latency compared with UDP and PGM.

Key words: Ad hoc networks; Reliable multicast; Cooperative communication; Loss recovery

1 引言

在 Ad hoc 网络的应用中, 当存在多个接收者需要同时接收数据时, 使用组播技术进行数据传输是很有效的方式。虽然一些应用, 例如一个实时的多媒体组播通信, 因为无法容忍丢失恢复产生的延时, 就容忍组播传输过程中不可靠网络带来的数据丢失或者错误; 但是其它的应用, 例如在单点对多点的组播传输或者是军事应用中, 是无法容忍数据包的丢失的。因此要求组播传输需要具备可靠性。然而 Ad hoc 网络中节点的能量和网络的传输带宽

有限, 并且网络自身具有无结构性、节点频繁移动, 以及无线传输信道环境高的误码率和信道衰落特性, 都影响了组播传输的可靠性。

到目前为止, 已经提出了许多可靠组播协议, 其中的 PGM 组播可靠传输协议是一个著名适用于有线组播网络的可靠传输协议^[1]。PGM 适用于多个源端和多个接收端之间通信的可靠传输协议, 它适用于组成员随时加入或离开, 但对于分组丢失不敏感的应用。另一方面, 也提出了一些适用于 Ad hoc 网络的组播传输通信协议^[2-5]和基于 Mesh 网络的协议^[6]。这些协议通过不同的方式提高了 Ad hoc 网络的组播通信可靠性。根据进行丢失恢复的不同位置, 这些算法可以分为: 基于源节点的丢失恢复、基于接收节点的丢失恢复或者基于额外的恢复节点

2009-07-29 收到, 2009-11-24 改回

国家自然科学基金(60772142), 国家 973 计划项目(2007CB310604)

资助课题

通信作者: 黄梦洁 sophiahmj@163.com

的丢失恢复。Gopalsamy 等人提出的 Reliable Multicast Algorithm(RMA)算法^[2]是依靠发送节点进行丢失恢复, RMA 算法通过发送端发送 ACK 来进行数据包确认。Chandra 等人提出的 Anonymous Gossip(AG)算法^[4]则是在进行丢失恢复时随机选择一个节点作为恢复节点,但是在 AG 中随机选择的丢失恢复节点能否完成丢失恢复具有未知性。单纯的依靠源节点,或者单纯的依靠接收节点进行丢失恢复的设计,或者选择某几个节点作为恢复节点,都是将整个网络的丢失恢复集中在这几个节点上进行处理,这样就会增加这几个节点的处理负担,并且无法避免大量重传时恢复节点无法负荷而成为网络瓶颈的情况发生。选择划分本地组的方式^[5]虽然在 NAK 抑制和避免产生瓶颈方面有较好的表现,但是面对 Ad hoc 中节点频繁的流动性,为了维护各个本地组的成员关系会增加很多的网络开销。

基于以上考虑,本文针对 Ad hoc 网络的特点和已有算法的优缺点,将协同的设计思想^[7,8]引入到丢失恢复设计中提出了新的基于协同的可靠组播丢失恢复算法(CoreRM)。CoreRM 算法是基于树状的路由结构的,例如可以采用 Ad hoc 按需距离向量组播路由协议(MAODV)^[9],目前还有基于分层树状结构的丢失恢复算法^[10]。CoreRM 选择将一跳邻居节点、组播路径上游节点和发送节点协同进行恢复丢失。算法设计目标在于:各级协同,分散网络中的丢失恢复,将网络的丢失在最短的距离内得到恢复,最终增加 Ad hoc 网络中组播传输的可靠性。对于无线网络中的节点缓存控制和缓存管理我们已经有了-定的研究^[11],因此在 CoreRM 算法的缓存策略设计上提出了新的想法:节点上缓存的不仅仅是接收到的数据,还缓存了能够帮助节点进行丢失恢复的路径信息。应对重传恢复中可能出现的 NAK 风暴,CoreRM 算法使用随机等待一定时间以后再发送 NAK 来避免竞争信道。针对 Ad hoc 网络中节点的频繁移动会造成拓扑结构的多变性,CoreRM 算法通过周期性发送源路由信息(SPM),来及时获知组播树结构中的上一跳节点信息的变动。

2 协同可靠组播丢失恢复(CoreRM)

2.1 CoreRM 算法概述

协同丢失恢复算法(CoreRM)是基于树型路由(如 MAODV)、NAK 确认丢失的可靠组播协同丢失恢复算法。为了避免在某个节点负担过多的丢失恢复处理,算法通过整个网络中的所有节点间相互协同提供丢失恢复,使得丢失在最小的范围内得到恢复,同时也使恢复负担分布化。所有的组播树成员

节点(包括路由器节点和接收节点)在缓存中不仅存储了传输的数据包,还存储数据丢失恢复路径向量(Loss Recovery Distance Vector, LRDV),LRDV 主要是记录进行丢失恢复的最短路径信息。通过接收到的 NAK 或者 NCF 中携带的发送节点的数据恢复路径信息 DV 向量(DV 是 LRDV 的一部分)而得到此信息。由于 LRDV 中仅仅存储了数据恢复的路径信息,因此不会占用很大的空间。多播树上每一个节点即使不能直接对丢失提供恢复,也可以通过自己的 LRDV 里存储的恢复路径信息来帮助其他节点获得恢复路径来进行丢失恢复。

总的来讲,CoreRM 算法的协同丢失恢复过程可概括为:

(1)单播恢复阶段:节点的 LRDV 中存在丢失数据的恢复路径信息,直接单播发送 NAK 请求到该节点请求进行相应数据包的重传。

(2)本地恢复阶段:对于无法通过 LRDV 进行单播恢复的丢失数据包,进入本地恢复阶段进行丢失恢复。广播 NAK, DV 附加在 NAK 包中进行传输。广播的距离是一跳,目的在于发现周围能够提供协同恢复帮助的最近邻居节点,接收到 NAK 请求的节点广播 NCF 并且检查缓存中是否存储了该丢失的数据包,如果能够提供丢失恢复就单播发送恢复数据包。否则超时后结束本地恢复,进入下一个丢失恢复阶段。

(3)单播恢复阶段:再次检索节点的 LRDV 中是否存储了丢失数据包的恢复路径,如果存储则直接单播恢复。因为在进行本地恢复的过程中,节点可能因为接收到的 NAK 数据包或者 NCF 数据包所携带的 DV 向量信息而获知丢失数据包的恢复路径,这样就可以再次通过邻居节点之间的协同帮助来进行丢失恢复处理。

(4)全局恢复阶段:对于仍然无法恢复的数据包,进入全局恢复阶段进行丢失恢复。沿着组播路径向上游节点发送 NAK,缓存中存储了丢失数据包的节点广播就发送恢复数据包,接收到恢复数据包的所有下游节点都能够完成丢失恢复。如果所有的上游节点都无法提供协同恢复帮助,NAK 数据包将最终被传递到发送端,发送端就针对丢失的数据包进行最终的丢失恢复重传。

2.2 数据包缓存策略

CoreRM 算法选择在组播树的每一个成员节点上都缓存接收到的数据包和恢复路径信息。由于 Ad hoc 网络是无结构的,每个节点同时具有转发功能和接收功能。因此 CoreRM 算法充分发挥转发节点的作用,设计转发节点处也进行数据包的缓存。转

发节点往往接近于发送节点, 相对于远距离的接收节点出现丢失的可能性较小, 当接收节点发生丢失请求恢复时, 就可能在转发节点处得到丢失恢复, 不再需要汇聚到源节点处进行丢失恢复。并且由于转发节点和接收节点的协同帮助, 就将整个网络的丢失恢复进行分布式处理, 分布式处理虽然需要每个节点都进行数据包缓存操作, 但是可以避免整个网络由于某个节点处理能力有限而出现瓶颈。

接收节点者通过设立数据包接收序列来发现数据包的丢失。由于源端发送数据包是按序发送的, 在规定的时间内没有正确接收到相应的数据包或者出现了接收空隙, 都意味着传输中发生了丢失。接收空隙产生的原因是在规定的时间内数据包没有按序达到, 在接收端建立最大接收序号累加算法, 每当接收到按序达到的数据包时就递加接收端接收到的最大序号。如果接收到的数据包序号小于最大的存储序号则丢弃该数据包; 如果接收到的数据包序号超过了接收端存储的最大接收序号加 1 时, 表示出现空隙有数据包发生了丢失。

2.3 恢复路径信息缓存策略

在 CoreRM 中, 每一个节点对于每一个组播组维护一个 LRDV 路由表用来存储数据包的恢复路径信息。每一个 LRDV 表具有表 1 的形式。

表 1 LRDV 数据包格式

| PacketID=(<i>S</i> , seqNo) | | Dst1 | Ts | hops |
|------------------------------|--------|--------|--------|--------|
| 32 bit | 32 bit | 32 bit | 32 bit | 32 bit |

Packet ID: 数据包标识(数据包发送端的地址和序号), 用于标识不同数据流中的不同数据包; Dst1: 数据包的恢复节点地址; hops 表示到达恢复节点的跳数; Ts 是时间戳, 在超过 T_{max} 时间后 LRDV 中的内容被重设。同其他的距离向量算法一样, 在 LRDV 表中每一个向量只保留了最优的恢复节点地址, 最优的节点地址就是指距离丢失节点跳数最少, 即最近的节点。如果是正确接收的数据包, 那么该数据包的向量中 Dst1 是本节点地址, 其余参数都是初始值。

在 LRDV 中的距离信息的传递是通过 NAK 或者是 NCF 的广播发送方式来传递的。每个 NAK 或者是 NCF 包的格式如表 2 所示。

TYPE 是数据包的类型, NAK 或者 NCF; *G* 是组播组地址; *S* 是发送节点地址; RCV 是接收节点地址, 例如: NAK 广播时是广播地址, NAK 单播时可能是上游节点地址或者是恢复节点的地址; seqNo: 请求重传/回复请求的丢失数据包的序号;

表 2 NAK/NCF 数据包格式

| TYPE | <i>G</i> | <i>S</i> | RCV | seqNo | [DV] |
|-------|----------|----------|--------|--------|----------------|
| 4 bit | 32 bit | 32 bit | 32 bit | 32 bit | <i>N</i> -unit |

[DV]是发送 NAK/NCF 节点的数据包接收情况组成的向量, 存储了从丢失数据包序号 seqNo 开始向后的 *N* 个数据包的恢复路径信息, 因为丢失可能是由于数据包的乱序导致的, 这也意味着序号后面可能存在正确接收到的数据包, 相对于序号前面的数据这些数据更有可能被其它节点请求重传。在 [DV] 中, [DV(*i*)] 是数据包 $\langle S, seqNo + i \rangle$ 的向量。

LRDV 中的信息在节点接收到 DATA, NAK, NCF 和 RDATA 时进行更新。

接收到数据包 seqNo, 节点 *A* 存储数据包, 建立或者更新 LRDV 信息。

$$\langle (S, seqNo), A, 0, current\ time, 0 \rangle$$

当接收到到 NAK 或者是 NCF 时, 节点 *A* 就根据其携带的 [DV] 中的信息更新 LRDV 表。注意到 [DV(*i*)] 影响到了 *A* 的 LRDV 中的 $\langle S, seqNo + i \rangle$ 。如果接收到的 NAK/NCF 中携带的第 *i* 个向量的序号和 *A* 节点的 LRDV 中存储的序号相同, 那么比较 DV 中的向量的 hops 和 LRDV 中的 hops 大小, 如果 DV 小于 LRDV, 则将 LRDV 中的信息替换成 DV [*i*] 中的信息。如果接收到的同序号的 DV 信息在 LRDV 中仍没有建立向量, 则按照 DV 中携带的信息建立相关向量。

2.4 SPM 数据包帧格式设计

SPM 数据包的设计是为了适应 Ad hoc 网络中节点频繁移动造成的组播树不断重建的问题。SPM 包是源路径信息数据包。在发送数据包之前被广播发送, 用来帮助各个节点建立组播树结构的上游节点信息; SPM 在数据包发送间隔被广播发送, 用来应对网络拓扑结构变动产生的上游节点变动。由于 SPM 仅仅用来监督网络拓扑的变化, 数据包长度很小, 因此在网络通信中也不会产生很大的网络开销。

SPM 数据包的帧格式中包括: 常用头部信息、IP 报头信息和 SPM 报头信息 3 部分。常用头部信息包括 4 部分信息: Tsi, seqno, size 和 type。其中 Tsi 是数据包标识, 因为网络中可能存在很多的数据流, 因此根据 Tsi 就可以区分不同的数据流。不同的 Type 值区分不同的数据包类型。例如: SPM 数据包的 type 值是: PT_SPM。IP 头部信息 (dst_addr) 是存放数据包发送的目的节点地址。SPM 头部信息主要是存储在 spm_path 中的源路由路径信息, 也就是在组播树建立以后, 节点的上游节点信息, 这里通过节点地址来进行标识。源节点

发送 SPM 数据包时, 将 `spm_path` 地址设置成自己的地址, 这样接收到 SPM 数据包的节点就获知上游节点是源节点。节点接收到 SPM 数据包以后改变 `spm_path` 为本节点的地址之后转发 SPM。这样就通过发送 SPM 数据包, 可以及时获知上游节点是否发生变化。

2.5 NAK 抑制策略设计

CoreRM 中 NAK 抑制算法设计主要是为了避免生成 NAK 风暴设计的。NAK 风暴产生的原因很多, 主要是: 不同的节点丢失同时请求相同序号的数据包, 所有节点同时发送 NAK, 等待 RDATA 的过程中不断发送重复的 NAK。NAK 风暴的出现很有可能导致整个网络的瓶颈链路断裂或者某个节点由于处理能力有限而瘫痪整个网络, 因此 NAK 抑制算法的设计是 CoreRM 中设计的重点。如图 1 所示是 CoreRM 算法的状态机转换图。当节点发现丢失数据包以后, 进入到 `NAK_STATE` 状态, 随机等待一段时间以后发送 NAK, 并且进入到 `WAIT_NCF_STATE` 状态中等待接收 NCF, 在等待 NCF 的过程中节点停止重复发送 NAK, 只有当等待 NCF 超时以后再次进入 `NAK_STATE` 阶段时, 才再次发送 NAK; 如果接收到 NCF 以后进入到 `WAIT_DATA_STATE` 阶段等待接收重传的数据包。否则当等待重传 NCF 的次数超过了 `MAX_NCF_RETIRES` 以后, 或者等待 RDATA 的次数超过了 `MAX_DATA_RETIRES` 的次数以后结束本丢失恢复阶段对数据包的丢失恢复。

3 CoreRM 性能分析

本节使用一个简化的网络模型对比分析 CoreRM 和 PGM 可靠组播丢失恢复的性能。考虑一个发送端多个接收端的组播传输模型, 发送端为

S 。组播通信是基于树状的路由结构, 网络中节点的密度为 ρ , 无线节点的传输半径为 r , 组播树成员节点到发送节点 S 之间的跳数为 H_i 。组播树成员节点 MH_i 存储了数据包 d_j 的概率为 P_{ij} 。当组播通信需要进行丢失恢复时, 如果丢失重传的请求无法被组播组成员节点满足就会一直传递到发送端。减少请求节点到提供丢失恢复节点之间的跳数 L_j 就能够减少重传的时延 T_j , 节约带宽和能量损耗, 因为这样就会引入很少的成员节点到丢失重传的处理中。丢失恢复时延 T_j : 是指从发现丢失后发出丢失重传请求开始到接收到重传的数据包之间的延时。 T_{avg} 是整个组播丢失恢复过程中的平均丢失恢复延时。 L_{avg} 是整个组播丢失恢复过程中的平均跳数。由于 CoreRM 进行协同的丢失恢复时, 经历了本地恢复阶段、全局恢复阶段和源节点恢复阶段 3 个阶段。每个丢失恢复阶段完成丢失恢复因为距离请求节点不同的跳数距离导致要经历不同的延时, 本地恢复阶段的平均延时为 T_z , 全局恢复的平均延时为 T_g , 源端恢复的延时为 T_s 。CoreRM 设计每次丢失恢复等待重传的最大延时为 T_r 。因此 CoreRM 算法的丢失恢复跳数和丢失恢复时延为

$$L_j = \left[1 - (1 - P_{ij})^{\rho\pi r^2 - 1} \right] \cdot 1 + \sum_{k=1}^{H_i-1} (1 - P_{ij})^{\rho\pi r^2} \cdot (1 - P_{ij})^{k-1} \cdot P_{ij} \cdot k + (1 - P_{ij})^{\rho\pi r^2} \cdot (1 - P_{ij})^{H_i-1} \cdot P_{ij} \cdot H_i \quad (1)$$

$$T_j = \left[1 - (1 - P_{ij})^{\rho\pi r^2 - 1} \right] \cdot T_z + \sum_{k=1}^{H_i-1} (1 - P_{ij})^{\rho\pi r^2} \cdot (1 - P_{ij})^{k-1} \cdot P_{ij} \cdot T_g + (1 - P_{ij})^{\rho\pi r^2} \cdot (1 - P_{ij})^{H_i-1} \cdot P_{ij} \cdot T_s \quad (2)$$

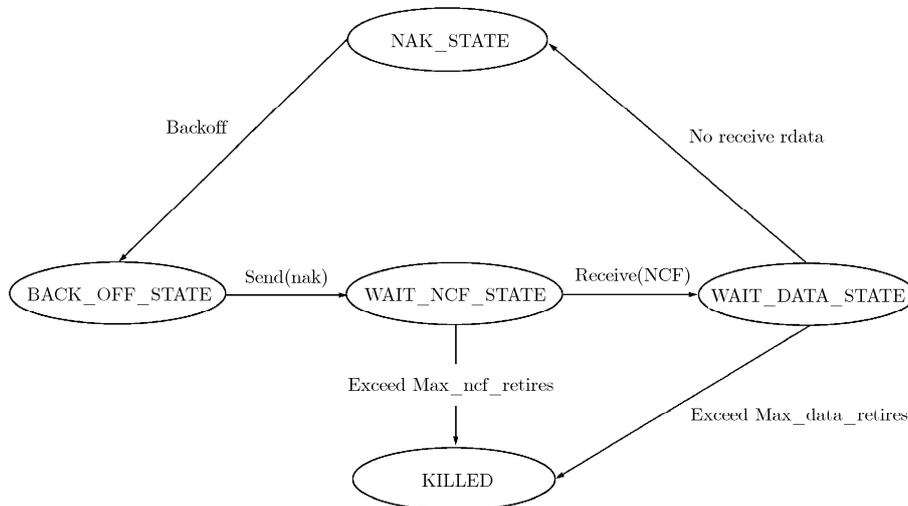


图 1 CoreRM 算法的状态机转换图

为了分析 CoreRM 算法和 PGM 算法的丢失恢复跳数和丢失恢复延时的性能, 本文通过一个简单的例子来进行说明, 例子的拓扑结构如图 2 所示。为了方便计算, 假设每一条链路上的丢包概率 p 相同, 每个节点上的存储概率 P 都可以根据该节点和发送节点之间的组播路径的链路丢包率 p 来确定。组播树上的每一个节点都对传输的数据包进行存储, 并且节点的缓存空间为无限大。发送节点 S 一定能够提供请求数据包的重传恢复。请求重传的 NAK 和 NCF 数据包不会发生丢失。

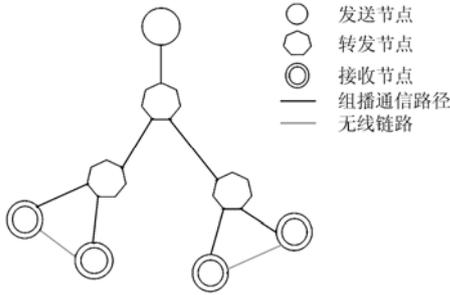


图 2 树形结构的组播树拓扑

使用 PGM 作为可靠组播通信协议:

$$L_{\text{avg}} = \sum_{n=1}^{\infty} 3 \cdot (1-p)^3 \cdot [p \cdot 1 + (1-p) \cdot p \cdot 2 + (1-p)^2 \cdot p \cdot 3]^{n-1} \quad (3)$$

$$T_{\text{avg}} = \sum_{n=1}^{\infty} T_s \cdot (1-p)^3 \cdot (n-1) \cdot T_r \cdot [p + (1-p) \cdot p + (1-p)^2 \cdot p]^{n-1} \quad (4)$$

使用 CoreRM 作为可靠组播通信协议:

$$L_{\text{avg}} = \sum_{n=1}^{\infty} [(1-p^5) \cdot 1 + (1-p) \cdot p^5 \cdot 2 + 3 \cdot p^6] \cdot \{p^2 \cdot [(1-p) \cdot p \cdot 2 + p \cdot 1] \cdot [p \cdot 1 + (1-p) \cdot p \cdot 2 + (1-p)^2 \cdot p \cdot 3]\}^{n-1} \quad (5)$$

$$T_{\text{avg}} = \sum_{n=1}^{\infty} [(1-p^5) \cdot T_z + (1-p) \cdot p^5 \cdot T_g + p^6 \cdot T_s] \cdot (n-1) \cdot T_r \cdot \{p^2 \cdot [(1-p) \cdot p + p] \cdot [p + (1-p) \cdot p + (1-p)^2 \cdot p]\} \quad (6)$$

图 3 所示的是在图 2 的树形拓扑结构下的丢失恢复跳数。虚线为 CoreRM 算法的平均丢失恢复跳数, 实线为 PGM 的平均丢失恢复跳数曲线。图 4 所示的是在图 3 的树形拓扑结构下平均恢复延时曲线。由于 PGM 使用的是发送端重传, 每次重传都需要发送端重传才能完成, 并且根据传输链路上的丢包率, 重传的数据包有可能再次发生丢失。从图

3 中可以看出, 使用了 CoreRM 算法以后图 2 的拓扑结构中发生了数据包丢失的接收节点, 理论上可以在两跳范围内完成丢失恢复, 而使用 PGM 算法时重传跳数就已经超出了 3 跳。使用了 CoreRM 算法以后的丢失恢复延时曲线增长幅度随着链路丢包率的增长也要小于 PGM, 因此可以看出 CoreRM 算法的优势。

4 计算机仿真与性能评估

本文使用 NS2 作为 Ad hoc 网络的仿真平台来估计 CoreRM 的性能。使用 MAODV 组播路由协议产生组播树拓扑。仿真中, 分布 16 个节点在 $670 \text{ m} \times 670 \text{ m}$ 的范围内, 选取一个节点作为源节点, 通过 CBR 流量生成器产生整个网络仿真的数据, 数据包长为 210 Byte。仿真中使用 MAODV 组播路由协议作为路由层协议, 802.11 作为 MAC 层协议。已有的 PGM 是针对有线网络设计的基于 NAK 的可靠组播协议, 根据本文的仿真需要我们对 PGM 做了一定的改进, 以适应于 Ad hoc 网络的传输环境。PGM 协议具有丢失恢复功能; 当组播组成员发生分组丢失时, 向源节点发送 NAK 请求进行丢失恢复。

本文采用性能指标主要是网络的吞吐量和平均丢失恢复延时。

平均吞吐量: 每个节点在单位时间内接收到的数据量。

平均丢失恢复时延: 数据包丢失发现到数据包丢失恢复的时延。

图 5 和图 6 是存在信道损耗的静态网络中的性能仿真结果。图中横坐标表示的是传输丢包概率, 也就是不同的信道损耗。图 5 所示的是网络的吞吐量曲线, 图 6 所示的是丢失恢复延时曲线。由于 CoreRM 和 PGM 对于数据包重传次数做了限制, 有些数据包可能因为超过了重传的最大次数而被放弃恢复, 这些数据包不会被计入到丢失恢复重传延时中。由于 UDP 是面向无连接的传输协议, 对于发生丢失的数据包不采取任何措施进行恢复, 因此 UDP 的性能下降的最快。PGM 协议的性能优于 UDP 协议, 然而由于使用的是发送节点进行丢失恢复, PGM 丢失恢复延时和吞吐量性能要差于 CoreRM 算法。CoreRM 算法在不同的信道损耗情况下始终能够保持较高性能, 因为 CoreRM 算法使用了最短距离的丢失恢复, 将整个网络的丢失恢复进行了分布式处理, 因此在信道传输情况变差时, 仍然能够通过较短时间完成丢失恢复, 保持较高的吞吐量。图 7 和图 8 所示的是在移动网络中的性能仿真结果。图 7 中显示的是网络的平均吞吐量曲线。

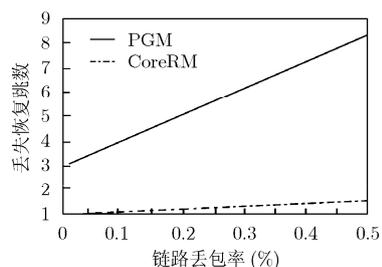


图 3 链路丢包率影响下的丢失恢复跳数

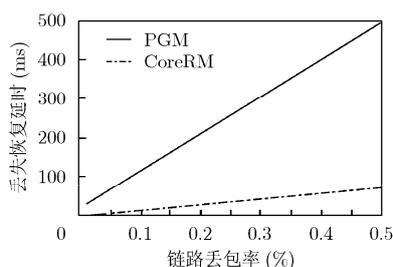


图 4 链路丢包率影响下的平均恢复时延

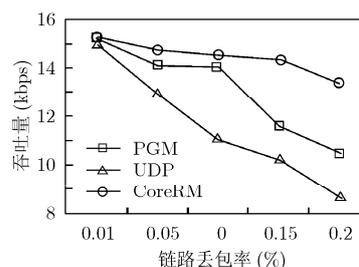


图 5 信道损耗影响下的吞吐量

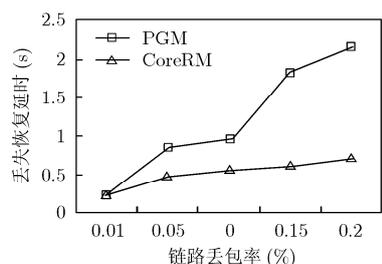


图 6 信道损耗下的丢失恢复延时

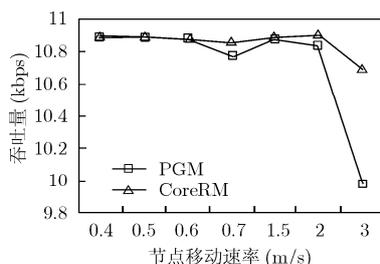


图 7 移动速率影响的平均网络吞吐量

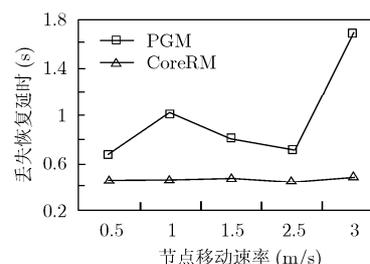


图 8 移动速率影响下的丢失恢复延时

图 8 中显示的是丢失恢复时延曲线。随着节点移动速度的加快,网络的拓扑变换更加的频繁,这样就增加的数据包丢失的概率,所以随着节点移动速度的加快,数据包请求重传的次数增加,整个网络的正确投递率也随之下降。由于 PGM 使用的是源端重传而延长了丢失恢复的时延,并且基于源节点的重传恢复增加了网络传输的控制开销,影响到整个网络数据包的正确投递和吞吐量性能,导致性能差于 CoreRM。CoreRM 算法使用了最短距离丢失恢复设计,丢失节点周围的邻居节点协同帮助恢复,使得丢失重传的数据包的传输范围缩小,避免了随着节点移动速度加快而产生的丢失恢复时延急剧增长。

5 结束语

在本文中提出了一个适用于 Ad hoc 网络的可靠组播协同丢失恢复算法 CoreRM。CoreRM 是基于 NAK 确认的协同丢失恢复算法,邻居节点通过提供丢失数据重传或者是数据恢复路径信息,尽力而为地协同帮助丢失数据的恢复;组播路径上的上游节点和源节点在本地恢复无法完成丢失恢复时,提供丢失恢复帮助。CoreRM 算法使用周期性发送 SPM 包来应对 Ad hoc 网络中拓扑的频繁变动,使用 NAK 抑制策略来避免 NAK 风暴。算法的设计主要目的是将 Ad hoc 网络中产生的丢失恢复在整个网络中分散处理,尽量使用最短的距离和最短的时间完成丢失恢复。通过仿真各种 Ad hoc 网络环境,对比其它两种非协同式的组播传输协议算法性能,

发现使用了协同恢复的 CoreRM 的性能在各种仿真环境中可靠性要高于非协同恢复算法。我们计划在未来的设计中,将基于窗口或者基于速度的拥塞控制机制引入到 CoreRM 算法中,进一步的提高算法的可靠性。

参考文献

- [1] Gemmell J, Montgomery T, Speakman T, Bhaskar N, and Crowcroft J. The PGM Reliable Multicast Protocol. IEEE Digital Object Identifier 10.1109/MNET.2003.Volume 17, Issue1, Jan-Feb. 2003: 16-22.
- [2] Gopalsamy T, Singhal M, Panda D, and Sadayappan P. A reliable multicast algorithm for mobile ad hoc networks. 22nd International Conference on Distributed Computing Systems, Vienna, Austria, July 2-5, 2002: 563-570.
- [3] Ouyang B and Hong Xiao-yan, and Yi Yun-jung. A comparison of reliable multicast protocols for mobile ad hoc networks. Proceeding IEEE, SoutheastCon, 2005, April 8-10, 2005: 339-344.
- [4] Chandra R, Ramasubramanian V, and Birman K P. Anonymous gossip: improving multicast reliability in mobile ad-hoc networks. Distributed Computing Systems, Arizona, USA, 2001, April 16-19, 2001: 275-283.
- [5] Fung Alex and Sasase Iwao. Hybrid local recovery scheme for reliable multicast using group-aided multicast scheme. WCNC2007, Hong Kong China, March 11-15, 2007: 3478-3482.
- [6] Koutsonikolas D, Hu Y C, and Wang Chih-Chun. Pacifier: high-throughput, reliable multicast without "crying babies" in wireless mesh networks. IFOCOM 2009. The 28th

- Conference on Computer Communications. Rio de Janeiro, Brazil. April 19-25, 2009: 2473-2481.
- [7] Fen Hou, Cai L X, Ho Pin-han, Shen Xuemin, and Zhang Junshan. A cooperative multicast scheduling scheme for multimedia services in IEEE 802.16 networks. *Wireless Communications*, ISSN: 1536-1284, March 2009: 1508-1519.
- [8] Alay O, Pan Xin, Erkip E, and Wang Yao. Layered randomized cooperative multicast for lossy data: a superposition approach.. CISS 2009.43rd Annual Conference, Baltimore, MD, March 18-20, 2009: 330-335.
- [9] Zhu Yufang and Kunz T. MAODV implementation for NS-2.26. systems and computing engineering, Carleton University, Technical Report SCE-04-01, January 2004.
- [10] Saikia L P and Hemachandran K. Reliable multicast protocol adopting hierarchical tree-based repair mechanism. ICACT 2009.11th International Conference, Gangwon-Do, South Korea, Feb. 15-18, 2009: 1969-1973.
- [11] Feng Gang, Zhang Jing-yu, and Xie Feng. Efficient buffer management scheme for loss recovery for reliable multicast[C]. *Proceedings of IEEE Globecom 2004*, Dallas, USA, 2004: 1152-1156.
- 黄梦洁: 女, 1985年生, 硕士, 研究方向为无线通信与网络技术.
- 冯 钢: 男, 1964年生, 博士, 教授, 博士生导师, 研究方向为无线通信与网络技术.
- 张翼德: 男, 1977年生, 博士, 副教授, 研究方向为无线网络组播技术.