

一种提高 Ad hoc 网络中多播可靠性的算法和模型

冯美玉 程胜 张勳 丁炜
(北京邮电大学 继续教育学院 北京 100876)

摘要 在 Ad hoc 网络中,低可靠性的无线链路使多播应用网络扩展性较差,然而有线网络多播应用中 FEC(Forward Error Correction)技术和 NACK(Negative ACKnowledgement)重传机制不适于 Ad hoc 网络。该文旨在提高 Ad hoc 网络中多播可靠性,提出由 PDA(Packet Dispersal Algorithm)算法和 ARM(Active Retransmission Model)重传模型相结合的可靠多播传输方案(RMTS)。理论分析和仿真结果都表明该方案具有效率高、复杂度低、资源占用少和开销小等优点。

关键词 Ad hoc, 多播, 可靠性, NACK, 可靠多播传输方案

中图分类号: TN915.04

文献标识码: A

文章编号: 1009-5896(2006)05-0780-05

An Algorithm and Model to Improve Reliability of Multicast in Ad hoc Networks

Feng Mei-yu Cheng Sheng Zhang Xu Ding Wei

(Continuing Education School, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract In Ad hoc networks, low reliability of wireless link makes the scalability of multicast group degrade drastically, and FEC(Forward Error Correction) technique and NACK(Negative ACKnowledgement)-based mechanism, which guarantees reliable transport of multicast in wired networks, is not adapt to be deployed in Ad hoc networks. In this paper, focusing on improving the scalability of multicast in Ad hoc networks, a Reliable Multicast Transport Scheme (RMTS) consist of PDA(Packet Dispersal Algorithm) and ARM(Active Retransmission Model) is proposed to guarantee reliable transports of multicast. Both theory analyses and simulation result show the merits of validity, simpleness, low resources-occupied and moderate overhead.

Key words Ad hoc, Multicast, Reliability, NACK, RMTS

1 引言

虽然 Ad hoc 网络不依赖于任何基础网络设施,能提供不受时间和空间限制的通信服务,但是多播应用仍然是有待研究的课题。在 Ad hoc 网络中,无线链路固有的高比特错误率使多播应用网络扩展性较差。到目前为止,已有多种多播协议适用于 Ad hoc 网络,如 MAODV^[1], AMRIS^[2], ODMRP^[3], MCEDAR^[4]和 AMRoute^[5],但它们都是基于尽力而为(Best effort)的多播协议,缺乏可靠性保障机制。在多播中,传输的不可靠性会导致较大的重传开销和多播树的重复配置,严重降低多播应用的网络扩展性能,尤其是当多播组规模较大时,性能下降尤为剧烈。虽然应用层、TCP 层和 MAC 层都能提供一定程度的可靠保障,但是都不适于 Ad hoc 网络中多播应用。应用层和 TCP 层支持端到端的可靠性,然而在多播应用中,多个多播接收节点造成的综合丢包效应严重,端到端的数据恢复会使点到多点通信的多播应用丧失节约带宽消耗的初衷。Ad hoc 网络多采用 IEEE 802.11^[6]作为 MAC 层协议,通过 ACK 和重传机制保障传输的可靠性。但在多播应用中,数据包以广播的形式发送,MAC 层不使用 ACK 和

重传,因此在多播应用中,MAC 层不能提供任何可靠性保障。在有线网络中,多采用 FEC(Forward Error Correction)^[7]技术中和部分比特错误,并以 NACK(Negative ACKnowledgement)^[8]重传机制相辅,共同提高多播传输的可靠性。但是 FEC 的计算复杂度高,不适于实施于计算能力有限的 Ad hoc 网络节点,有线网络中 NACK 机制也不适用于具有天然广播特性的无线链路^[9]。本文吸收 FEC 和 NACK 的基本思想,提出一种由 PDA 算法和 ARM 传输模型相结合的可靠多播传输方案,从理论上分析其实施的可行性,并通过仿真验证它的性能。

2 PDA 算法描述

Bestavros 提出对高效的 IDA(Information Dispersal Algorithm)^[10],将原始数据分割成 m 份,并编译成 n 份数据片段用于传输($n \geq m$),接收方只要接收到其中的任意 m 份即可恢复原始数据(其中的冗余思想和 FEC 类似)。这样, n 份数据中携带的冗余信息能中和部分比特错误。IDA 算法起初用硬件设计,适用于特殊的 MAC 芯片,通过更大的带宽消耗来提高链路级传输可靠性,对单播和多播没有选择性。一般来说,单播和多播对可靠性和带宽消耗有不同的需求,需要灵活的机制在可靠性和带宽消耗之间均衡。本文提出

PDA(Packet Dispersal Algorithm)算法, 依靠软件实现, 用于在网络层提高传输可靠性。不同的应用可以根据需求决定是否选择该算法, 实现可靠性和带宽消耗之间均衡。

在 PDA 中, 原始数据包可表示为

$$F = 0x \underbrace{b_1 b_2 \dots b_m}_{F_1} \underbrace{b_{m+1} b_{m+2} \dots b_{2m}}_{F_2} \dots \underbrace{b_{(k-1)m+1} b_{(k-1)m+2} \dots b_{km}}_{F_k} \quad (1)$$

其中 b_i 为一个 4bit 十六进制数, 原始数据包分为 k 个片段, 每个片段 m 个数据单元, 即 $F_i = 0x b_{(i-1)m+1} b_{(i-1)m+2} \dots b_{im}$, $1 \leq i \leq k$ 。用矩阵可将 F 表示为

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \dots & b_m \\ b_{m+1} & b_{m+2} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(k-1)m+1} & b_{(k-1)m+2} & \dots & b_{km} \end{bmatrix} \quad (2)$$

编译过程依靠一个 $n \times m$ 矩阵 A_{nm} , 定义如下:

$$A_{nm} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \quad (3)$$

A_{nm} 是一个有特殊要求的矩阵, 要求 n 个行矢量中的任意 m 个行矢量所构成的矩阵应当是可逆的。

原始数据包中第 i 个数据片段 F_i 中的 m 个数据单元可以通过矩阵 A_{nm} 编译成由 n 个编译数据单元构成的集合 Q_i , 编译过程如下:

$$Q_i = A_{nm} F_i' = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \begin{bmatrix} b_{(i-1)m+1} \\ b_{(i-1)m+2} \\ \vdots \\ b_{(i-1)m+m} \end{bmatrix} = \begin{bmatrix} C_{1,i} \\ C_{2,i} \\ \vdots \\ C_{n,i} \end{bmatrix} \quad (4)$$

那么 k 个数据片段可编译成以下结果:

$$C_{nk} = [Q_1, Q_2, \dots, Q_n] = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,k} \\ C_{2,1} & C_{2,2} & \dots & C_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n,1} & C_{n,2} & \dots & C_{n,k} \end{bmatrix} \quad (5)$$

将编译结果 C_{nk} 按行分成 n 份, 封装成 n 个编译数据包发送, 每个编译数据包为 $P_j = 0x C_{j,1} C_{j,2} \dots C_{j,k}$, $1 \leq j \leq n$ 。每当接收方接收到一个 P_j 时, 必须为其选择相应的 V_j 。当接收方接收到其中的 m 个数据包时, 相应的 C_{mk} 和 A_{mm} 即可获得。为了叙述方便, 我们以最前面 m 个编译数据包为例, 获取的 C_{mk} 和 A_{mm} 可表示如下:

$$C_{mk} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,k} \\ C_{2,1} & C_{2,2} & \dots & C_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m,1} & C_{m,2} & \dots & C_{m,k} \end{bmatrix} \quad (6)$$

$$A_{mm} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{bmatrix} \quad (7)$$

将 C_{mk} 按列分成 k 份, 构成 k 个编译数据矢量, 每个编译数据矢量为 $X_i = 0x C_{1,i} C_{2,i} \dots C_{m,i}$, $1 \leq i \leq k$ 。那么原始数据包中的第 i 个片段 F_i 可以通过如下方式恢复:

$$F_i' = A_{mm}^{-1} X_i' = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{bmatrix}^{-1} \begin{bmatrix} C_{1,i} \\ C_{2,i} \\ \vdots \\ C_{m,i} \end{bmatrix} = \begin{bmatrix} b_{(i-1)m+1} \\ b_{(i-1)m+2} \\ \vdots \\ b_{(i-1)m+m} \end{bmatrix} \quad (8)$$

待 k 个数据片段 F_i 都恢复后, 原始数据即可获得。

为了编译数据和反编译数据可以用十六进制的数据单元表示, 将要它们按 16 模运算, 下面是编译和反编译过程中模运算的实例。

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 8 \\ 11 \end{bmatrix} = \begin{bmatrix} 11 \\ 8 \\ 19 \end{bmatrix} \triangleq \begin{bmatrix} 11 \\ 8 \\ 3 \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 3 \end{bmatrix} = \begin{bmatrix} 8 \\ -5 \end{bmatrix} \triangleq \begin{bmatrix} 8 \\ 11 \end{bmatrix} \quad (10)$$

3 ARM 重传模型

虽然 PDA 算法能中和一定量的比特错误, 但不能提供绝对的可靠性, 所以必须考虑实施有效的重传机制。根据无线链路天然的广播特性, 我们设计适于 Ad hoc 网络的主动重传模型 ARM。在 ARM 中, 每个节点维护 3 个缓冲区(图 1)、1 个解码器和 1 个编码器; 接收缓冲区用于缓存接收到的编译数据, 解码器将编译数据解码并投入输入缓冲区中。输入缓冲区是一个带阻塞机制的缓冲区, 没有空闲区域时不接收新的数据; 解码后数据送至本地或由编码器编译并转发, 而且转发后的数据投入输出缓冲区缓存; 输出缓冲区是一个带溢出功能的缓冲区, 优先溢出最陈旧的数据。

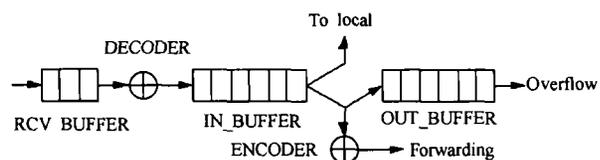


图 1 节点内缓冲区设置
Fig. 1 Buffers in a node

在 ARM 中, 多播组源节点以升序的方式发送数据包 PACKET(data_sn), 其中 data_sn 为原始数据的序列号。通过 ENCODER, PACKET(data_sn) 将被编译成 n 个编译数据包 PACKET(data_sn, compile_sn), compile_sn 为编译数据包序号。如果接收节点接收到 n 个编译数据包中的 m 个, 即 data_sn 相同、compile_sn 不同的 m 个编译数据包, DECODER 就可成功恢复原始数据包并投入 IN_BUFFER 中。同时, 节点监测 IN_BUFFER 中的数据包序号是否连续, 如果 data_sn 不连续, 说明有数据包丢失, 节点将主动发起重传请求过程, 通过发送 NACK 消息请求其他节点重传遗失的数据包, 多播组中每个节点都有重传能力, 而不是只有上游节点才能重传。每个节点都设有一个 NACK 消息列表, 管理最近一段时间接

收到的 NACK 消息，只有不重复的 NACK 有效。当节点接收到一个 NACK 消息时，如果本节点 IN_BUFFER 或 OUT_BUFFER 缓存有 NACK 请求的数据包，则发起重传过程，否则向上游节点转发 NACK。

在多播应用中，一个上游节点往往拥有多个下游节点，很可能有多个节点同时监测到丢失同一个数据包，如果它们都发送 NACK 或对每一个 NACK 进行相应的重传，必然会导致请求风暴和重传风暴，冗余的请求和重传会引发过量的开销。为了减少重传开销对带宽的浪费，我们利用无线链路天然的广播特性，通过定时器方式来抑制请求风暴和重传风暴。

在 ARM 中，当一个节点监测到丢包时，它不是立刻发送 NACK，而是启动一个定时器，初始定时值随机设定，当定时器超时时才发送 NACK，如果在定时器超时之前，接收到其它节点发送 NACK 请求相同数据包的重传，则中止定时器，取消 NACK 的发送。这样，虽然可能有多个节点需要重传相同的数据包并同时启动定时器，但定时器不会同时超时，所以只需发送一个 NACK，请求风暴得以抑制。同样，收到 NACK 的节点也不是立刻重传被请求的数据包，而是启动一个重传定时器，如果定时过程中接收到其他节点重传这个数据包，立刻取消定时器，否则，当定时器超时，重传被请求的数据包。与请求定时器类似，重传定时器能压制重传风暴的发生。请求定时器和重传定时器的状态转移关系如图 2 所示。定时器的初始值按式(11)设定，其中 α 为协议参数，可根据不同的多播协议进行调整，RTT 为节点间往返时间，通过周期性测量获得。

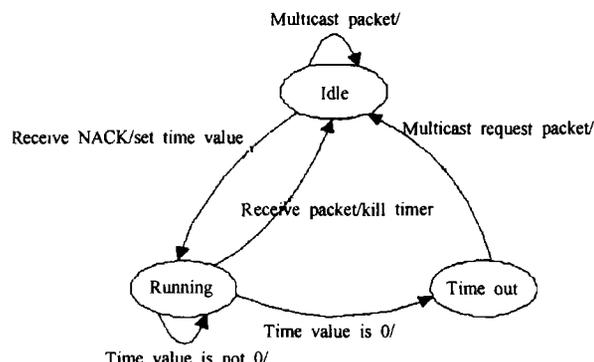
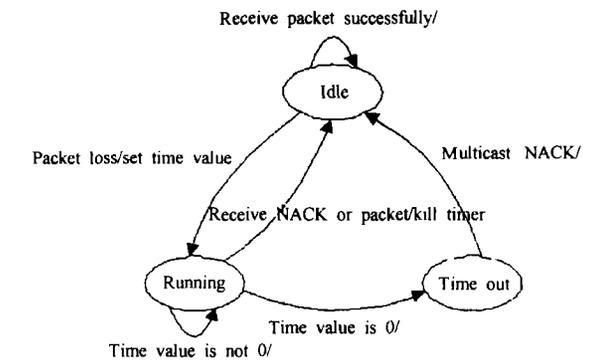


图 2 请求定时器和重传定时器状态变迁图

Fig. 2 State transition diagram of a timer

$$T_{\text{timer}} = \text{RAND}(0, \alpha \text{RTT}) \quad (11)$$

为了减缓 RTT 测量值的大幅度波动，我们用以下过滤函数为 RTT 进行整形。根据多次测试，整形参数 $\beta=0.2$ 时整形效果较好。

$$\text{RTT} = (1 - \beta)\text{RTT}(t-1) + \beta\text{RTT}(t) \quad (12)$$

4 性能分析

根据 PDA 特性，传输时，一个长度为 L 的原始数据包将被编译为 n 个长度为 L/m 的编译数据包。接收节点只要接收到其中任意 m 个编译数据包即可成功恢复原数据包，所以错误容忍度为 $C_n^m \times (n-m)/m$ ，可见 PDA 能中和一定的比特错误率。由于编译数据包携带一定的冗余信息， n 个编译数据包的总长度为 $n(L/m)$ ，比原始数据包大，给带宽带来的开销比例为 $(n/m-1) 100\%$ 。在实际部署 PDA 时，往往选择 n 略大于 m ，所以开销比例比较小。

PDA 的可行性直接取决于能否找到有效的 A_{nm} ，使其中任意 m 个行矢量构成的矩阵为可逆矩阵。下面以 $m=4$ 为例，论证 A_{nm} 的存在性。设 $A_{44} = [V_1', V_2', V_3', V_4']$ ，根据矩阵论知识可知，秩为 4 的 A_{44} 必然存在。设 $V_5 = a_1V_1 + a_2V_2 + a_3V_3 + a_4V_4$ ，取 A_{44} 中的任意 3 个矢量和 V_5 组成矩阵 4×4 矩阵 B_{44} 。如果 3 个矢量为 V_1, V_2 和 V_3 ，则 $B_{44} = [V_1', V_2', V_3', V_5']$ 那么 $r([V_1', V_2', V_3', V_5']) = r([V_1', V_2', V_3', a_4V_4'])$ ，只要 $a_4 \neq 0$ ， B_{44} 就是可逆的。根据推理可知，当 $a_1, a_2, a_3, a_4 \neq 0$ 时， B_{54} 存在。用类似的方法可证 B_{64} 也存在。不失一般性， A_{nm} 必然存在。式(13)式是一个 A_{64} 的实例。

$$A_{64} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \quad (13)$$

PDA 算法的计算复杂度主要取决于编解码过程中的矩阵运算。对于 $n \times m$ 矩阵和长度为 $k \times m$ 的数据包，计算复杂度为 $O(nmk)$ ，所以 PDA 的复杂度为 $O(nmk + mmk)$ 。一般来讲， n 和 m 的取值均不会太大，所以 PDA 的复杂度不高，而且与数据包大小呈线性递增关系，不会导致复杂度不可接受。

ARM 充分利用无线链路天然的广播特性，通过请求定时器和重传定时器，能将同一传输范围内的 NACK 和重传控制为一次，有效压制了请求风暴和重传风暴的发生，而且重传范围为一跳。重传能否控制在一跳范围之内，将取决于节点 OUT_BUFFER 的大小。如果 OUT_BUFFER 足够大，即一跳邻居节点总是缓存请求的数据，那么重传将只在本地发生。实际上节点的缓存能力总是有限的，因此需要研究多大的缓存能取得较好的性能。最佳的缓存大小设置与定时器的定时值密切相关，如果定时器初始值太小，则发挥不了压制

风暴的作用;如果太大,则要求更大的缓存能力才能将重传限制在本地。在本文方案中,定时器的初始值随机设定,最大值为 3RTT,相应的请求时间和重传时间为 6RTT,如果数据流以恒定速率、数据包等长的方式到达,那么单个数据包的传输时间为 0.5RTT。在一次成功重传过程中,将会有 12 个新数据包达到,所以要求节点具有 12+1 个数据包的缓存能力。13 个数据包的缓存大小不算过大,现有的微电子技术不难做到。

5 仿真分析

为了进一步说明 RMTS 的有效性,我们用 MatLab 6.5^[11] 建立仿真试验进行验证分析。仿真拓扑为 1 个源节点向 6 个多播接收节点传送数据,所有节点均在同一传输范围之内,有效带宽能力为 2Mbps,每个节点的缓存能力为 20×512Byte。数据包到达时间服从泊松分布,数据包大小服从均匀分布,最大长度为 512Byte。PDA 中的 A_{nm} 如式(13)所示, $m=4, n=6$ 。ARM 中时钟参数 $\alpha=3$, RTT 通过测量得到,整形参数 $\beta=0.2$ 。

我们先测试了不同长度的数据包在不同误码率(Bit Error Ratio, BER)下的成功传输的概率。包长选择为 64-512bit, BER 选择为 0.0001 和 0.0002,共进行 16 组试验,每组试验包长和 BER 都相等并重复进行 20 次,每次测试数据包个数为 10000。图 3 数据包成功传输概率的测试结果,从图上可以看出,包长越大,成功率越低,而且 BER 越大,成功率也越低。成功率走势总体上符合一定规律,重复测试结果不会波动太大。在 PDA 算法中,编译数据包长度为 L/m ,比原始数据包长度 L 小,这一点就能提高数据包传输的成功率。

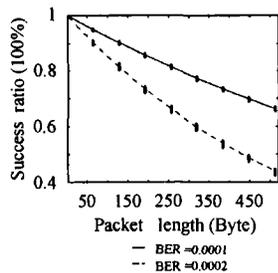


图 3 数据包传输成功率测试

Fig.3 The successful receiving probability of packet

图 4 为 PDA 对端到端传输速率的影响试验结果。采用 PDA 方案时传输速率有明显提高,而且 n 越大,提高的程度越大。这说明 PDA 有效的降低了数据包传输错误的概率,接收节点成功接收数据包的比例增大,所以传输速率得到提高。当增大 PDA 中的 n 参数时,传输速率能进一步提高,这是因为 n 越大,冗余信息越多,成功接收的可能性就越大。从图上还可以看出,接收速率不能达到原始速率,说明尽管 PDA 能提高接收成功率,但不能保证绝对的可靠性。虽然增大 n 能提高成功率,但也会招致更多的传输开销,因此在实际应用中不一定可取。

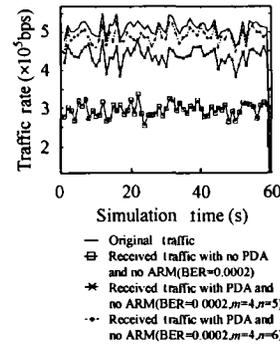


图 4 PDA 对端到端传输速率的影响

Fig.4 The end-to-end transport rate with PDA

图 5 展示的是 ARM 对端到端传输速率的提高,实验中最大重传次数为 1。不难看出,在 PDA 基础上,ARM 能将传输速率进一步提高,而且几乎能达到原始速率的程度,说明 ARM 能提供较高的一跳内数据恢复性能。尽管也不能提供绝对的可靠性,但可靠程度已接近 100%,而且还可以通过增大最大重传次数来提高可靠程度,因此 ARM 能有力地保障传输的可靠性。

RMTS 的宗旨是在不引发过大开销的前提下提高传输的可靠性,因此传输开销是一个重要的性能参数。为此,我们测试了 RMTS 在传输带宽上的消耗,从图 6 上可以看出,PDA 造成的带宽消耗比例大致为 1.3, PDA 和 ARM 共同消耗比例约为 1.44,可见 RMTS 不会造成对带宽的过量占用。

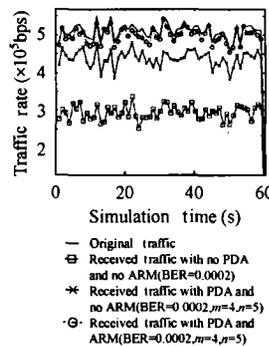


图 5 ARM 对端到端传输的影响

Fig.5 The end-to-end transport rate with PDA and ARM

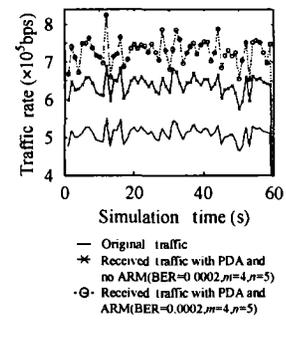


图 6 PDA 和 ARM 对带宽的消耗

Fig.6 The bandwidth consumption by PDA and ARM

6 结束语

本文针对 Ad hoc 网络多播应用中的传输可靠性问题,提出旨在提高传输可靠性的 RMTS 方案,通过 PDA 算法和 ARM 模型,可以大幅度提高传输可靠性。理论分析证明, PDA 算法的数学模型是正确的,而且可以根据实际需要调整算法的参数来获取对比特错误不同程度的抵抗力,算法简单,易于软件实现。ARM 重传模型利用无线链路天然的广播特性来抑制风暴的发生,仅需要有限存储空间的支持,在现有技术条件下切实可行。仿真结果表明, RMTS 方案能将端到端的接收速率提高到接近原始速率的程度,而且带宽消

耗适中。

参考文献

- [1] Royer E M, Perkins C E. Multicast operation of the Ad hoc on-demand distance vector routing protocol [C]. MOBICOM99, Seattle, Washington, Aug. 1999: 207-218.
- [2] Wu C W, Tay Y C. AMRIS: A multicast protocol for Ad hoc wireless networks [C]. Proceeding of MILCOMM99, Atlantic City, New Jersey, 1999: 25-29.
- [3] Gerla M, Pei G, Lee S J. On-Demand Multicast Routing Protocol (ODMRP) for Ad-hoc networks [S]. IETF, Internet-Draft, draft-ietf-manet-odmrp-04.txt, Nov. 2002.
- [4] Sinha P, Sivakumar R, Bhargavan V. MCEDAR: Multicast core-extraction distributed Ad hoc routing [C]. Proceeding of IEEE Wireless Communication and Networking Conference, Orlando, Florida, Sept. 1999: 172-181.
- [5] Bommaiah E, Liu M, Talpade R. AMRoute: Ad hoc multicast routing protocol [S]. Internet draft, draft-talpade-manet-amroute-00.txt, Aug. 1998.
- [6] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications [S], IEEE standard 802.11, 1997.
- [7] Lin S, Costello D J. Error Control Coding: Fundamentals and Applications [M]. New Jersey, Prentice-Hall, 1983, chapter 2-4.
- [8] Paul S, Sabnani K K. Reliable Multicast Transport Protocol (RMTP) [C]. INFOCOM96, San Francisco, 1996: 414-424.
- [9] Sobeih A, Baraka H. ReMHoc: A reliable multicast protocol for wireless mobile multihop Ad hoc networks [C]. IEEE CCNC04, 2004: 146-151.
- [10] Bestavros A. SETH: A VLSI chip for real-time information dispersal and retrieval for security and fault-tolerance [C]. Proceeding of international Conference on Parallel Processing, 1990: 457-464.
- [11] <http://www.mathworks.com/> [EB/OL], 2004.
- 冯美玉: 男, 1974年生, 博士生, 研究方向为 Ad hoc 网络、宽带通信、移动通信和网络安全。
- 程胜: 男, 1976年生, 讲师, 博士, 研究方向为软交换、Ad hoc 网络、WiMax 与移动通信。
- 张勛: 女, 1973年生, 讲师, 博士, 研究方向为 Ad hoc 网络、移动 IP 和宽带通信。
- 丁炜: 男, 1935年生, 教授, 博士生导师, 研究方向为宽带通信、网络安全、移动通信和 NGN 技术。