

Internet网 多播通信 主动网络 数据恢复

(12)

主动结点启动的可靠多播通信^{*}

Active-Node-Initiated Reliable Multicast

50-53

蔡洪斌 周明天 杨国纬

(电子科技大学信息中心 成都 610054)

TP393.4

Abstract Sender-initiated and receiver-initiated reliable multicast protocols can suffer performance degradation as increasing the number of receivers. New technology and new service of active network would resolve the problem of the scalable and reliable multicast. The paper presents an Active-Node-Initiated Reliable Multicast (ANIRM) protocol. The protocol, which guarantees the data packet to transport correctly from active node to active node, shifts the burden of providing reliable data transfer to active nodes and receivers. Theoretical analysis shows that ANIRM proposed here is, comparing with the traditional network protocol, better for bandwidth and data recovery delay.

Keywords Active-node-initiated, Active network, Reliable multicast

一、概述

在 Internet 网上实现多播通信还存在没有完全解决的技术问题,如反馈信息爆炸、丢失数据局部恢复功能等,针对这些问题目前有两个主要研究方向:定时器方法(如 SRM^[1])和等级结构化方法(如 RMTP^[2]),但它们都只能部分地解决这些问题。

随着主动网络^[3]的发展,开始利用主动网络或其思想来解决多播通信中的问题。文[4]利用了主动网络的思想,扩充了网络结点功能以支持可靠多播通信的局部数据恢复,其主要思路是:在 NACK 上传过程中,到达返回点(turning point)后,再向下传,查找有请求数据的成员,从而恢复数据,但确定返回点很困难。文[5]利用网络模型分析了集中式错误恢复(CER)和分布式错误恢复(DER)的性能,得到了 DER 性能更优。DER 仅使用了主动网络的思想,没有基于主动网络深入研究 DER。文[6]利用主动网络技术来解决多播通信的问题,但没有充分利用主动结点的功能,数据的检测仍然依赖于接收者,使错误数据仍在网络中传送,从而延长了数据恢复时间。本文提出主动结点启动的可靠多播通信 ANIRM(Active-Node-Initiated Reliable Multicast),即主动结点参与对数据包丢失和位错误的检测,一旦发现数据错误及时请求重传。在 Internet 中,由发送者负责处理 ACK、NACK 和数据重传,发送者的负担极重,因此,多播通信广泛使用接收者启动

(receiver-initiated)方式,从而减轻发送者的负担。本文提出的 ANIRM,将进一步减轻发送者负担,保证数据在主动结点间可靠传递,它有如下的优点:反馈信息量很少;具有局部恢复错误数据的能力;降低错误数据恢复时延;不必在网络中传送错误数据,有效地节约带宽。文中分析了没有主动结点和主动结点启动二种方案占用带宽和数据恢复时延的情况,通过比较可知:ANIRM 占用带宽少,数据恢复时延短,是一种更好的方案。

二、网络模型

使用 ANIRM 的主动网络仍是一个不可靠网络,数据在传送过程中会发生错误,如丢包、包位错误等。规划主动结点的基本功能等已超出了本文的讨论范围,可参见文[7~9]。设多播通信树如图 1 所示,与文[5]中的树相似,但引入了主动结点和修改了网络错误假设。引入的主动结点位于树的中间,是骨干网络的边缘结点。树中第一级包含一条链路,称为发送链路,连接多播通信的发送者到骨干网络,第二级包含 G 条骨干链路,分别连接主动结点;第三级由主动结点连接接收者,称为接收链路,设每个主动结点连接 Z 个接收者,称为一个接收组,共有 G 个接收组。每个主动结点负责一个接收组中的所有接收者正确收到数据。多播通信树中共有 $R=G*Z$ 个接收者。设丢包、包错误发生在传送过程中,与接收者的缓存等无关,发送链路错

^{*} 本文得到四川省重点基金项目支持,蔡洪斌 博士生,研究方向为计算机网络、主动网络及远程教学,杨国纬 教授,研究方向为计算机网络、操作系统,周明天 教授,博士生导师,研究方向为网络计算、开放分布式系统、并行处理与系统集成。

误的概率为 p_s , 传输时延为 T_s ; 骨干链路错误的概率为 p_a , 传输时延为 T_a ; 接收链路错误的概率为 p_r , 传输时延为 T_r 。

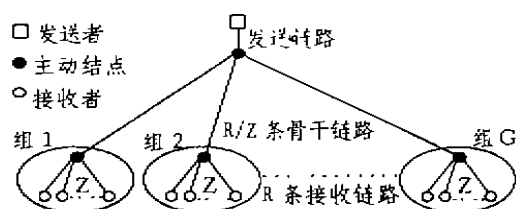


图1 多址通信树

图中的主动结点负责处理与可靠性有关的工作, 如通过执行程序检测数据错误或处理数据重传等。由于主动结点增加了检测和重传工作, 将增加处理数据的时间, 因此, 并不是主动结点越多越能改善协议性能。通过下面的分析可知, 如图1所示的多播通信树是合理的。为了清楚地阐明 ANIRM 的优点, 基于图1的多播通信树定义二种网络协议。

1) 协议 N 是接收者启动的通用协议, 没有主动结点, 接收者负责检测错误, 发送者负责处理 NACK, 重传数据包。协议 N 的数据传送按如下方式进行:

发送者: ①发送原始数据包, 等待 NACK; ②收到 NACK, 则使用多播通信方式, 在全局范围重传原始数据包。

接收者: ①负责检测错误, 向发送者反馈 NACK; ②正确收到包, 则不反馈 ACK, 否则反馈 NACK。

2) 协议 ANIRM 是主动结点启动协议, 由主动结点和接收者负责检测错误, 主动结点和发送者负责处理 NACK。ANIRM 的数据传送按如下方式进行:

发送者: ①发送原始数据包, 等待其直接子主动结点的 NACK; ②收到 NACK, 则重传原始数据包。

主动结点: ①检测错误。当主动结点收到数据包时, 执行错误检测程序, 如果发现丢失包或包位错误, 则不再传送下一个数据包或该数据包, 而是立即发 NACK, 请求父结点重传数据包, 正确收到数据包后, 再下传到子结点; ②数据缓存。每个主动结点都负责其所有直接子结点正确收到数据, 因此, 必须能缓存数据用于丢失数据的局部恢复; ③局部多播。对所有直接子结点, 使用多播方式响应 NACK。

接收者: ①负责检测错误, 向主动结点反馈 NACK; ②正确收到包, 则不反馈 ACK, 否则反馈 NACK。

三、性能分析

协议 N、ANIRM 最主要的区别在于对数据包的检测上, N 由接收者检测错误, ANIRM 由主动结点和

接收者检测错误。为叙述方便, 把检测错误的主动结点或接收者称为检测者。

设通信组 i 有 D_i 个检测者, 每个检测者检测到错误的概率为 p 。用 L_{D_i} 表示任一个检测者为正确收到一个包而请求重传的次数, 则 L_{D_i} 的概率分布为:

$$F_{L_{D_i}}(l) = 1 - p^l \quad (1)$$

设 L 指所有检测者正确收到数据包而请求重传的最大次数, 则最大概率分布为:

$$F_L(l) = (1 - F_{L_{D_i}}(l))^{D_i} \quad (2)$$

设所有检测者正确收到任一个包, 要求该包重传 M_i 次, 则 M_i 的数学期望为:

$$E[M_i] = \sum_{l=1}^{\infty} (1 - F_L(l)) = \sum_{l=1}^{\infty} (1 - (1 - p^l)^{D_i}) \quad (3)$$

一个数据包占用带宽的定义为: 在多播通信组 i 中, 数据包的传送次数 $1 + M_i$ 与每次传送遍历的链路数 H_i 之积, 全网总链路数为 $H = R + R/Z + 1$, 所以, 每个包在每条链路上平均占用带宽为:

$$E[B_i] = \frac{1}{H} \sum_{i=1}^N (1 + E[M_i]) \cdot H_i \quad (4)$$

错误数据恢复时延的定义为: 从检测者检测到错误, 发出 NACK, 到收到正确数据的时间, 包括发送、反馈、传输等时延^[5]。本文忽略与发送者处理速度、队列等待、抑制 NACK 等因素, 仅考虑因请求重传数据而增加的传输时延。设通信组 i 中, 重传一次数据增加的传输时延为 T_i , 则数据恢复时延 T 的数学期望为:

$$E[T] = \sum_{i=1}^N E[M_i] \cdot T_i \quad (5)$$

1) 协议 N 协议 N 中, 无论是发送链路、骨干链路或接收链路发生错误, 都是由接收者检测错误, 因此, 重传数据包都将遍历所有链路, 即协议 N 只有一个多播通信组, 组内的检测者就是接收者, 共有 R 个, 组内有 H 条链路, 检测者发现错误的概率 p 为:

$$p = 1 - (1 - p_s)(1 - p_a)(1 - p_r) \quad (6)$$

所以, 根据式(4), 协议 N 占用带宽为:

$$E[B_N] = 1 \sum_{l=1}^{\infty} (1 - (1 - p^l)^R) \quad (7)$$

重传一个包增加的传输时延 $T_i = 2 * (T_s + T_a + T_r)$, 所以, 根据式(5), N 的数据恢复时延为:

$$E[T_N] = 2 * \left(\sum_{l=1}^{\infty} (1 - (1 - p^l)^R) \right) (T_s + T_a + T_r) \quad (8)$$

2) 协议 ANIRM ANIRM 中, 有三种不同性质的多播通信组, 第一种多播通信组有一个, 组内有一个检测者和一条发送链路, 且 $p = p_s$, $T_1 = 2 * T_s$; 第二种多播通信组有一个, 组内有 R/Z 个检测者, R/Z 条骨干链路, 且 $p = (1 - p_s)p_a$, $T_2 = 2 * T_a$; 第三种多播通信组有 R/Z 个局部接收组, 每个接收组中有 Z 个检测者和 Z 条接收链路, 且 $p = (1 - p_s)(1 - p_r)p_r$, $T_3 = 2 * (T_r + T_s + T_a)$ 。

T_s 。所以,根据式(4)、(5),ANIRM 占用带宽和数据恢复时延分别为:

$$E[M_1] = \frac{p_s}{1-p_s} \quad (9)$$

$$E[M_2] = \sum_{k=1}^{\infty} [1 - (1 - ((1-p_s)p_b)^k)^{p_r}] \quad (10)$$

$$E[M_3] = \sum_{k=1}^{\infty} [1 - (1 - ((1-p_s)(1-p_b)p_r)^k)^{p_r}] \quad (11)$$

$$E[B_A] = 1 + \frac{1}{H} E[M_1] + \frac{R}{HZ} E[M_2] - \frac{R}{H} E[M_3] \quad (12)$$

$$E[T_A] = E[M_1] \cdot 2T_s + E[M_2] \cdot 2T_b + E[M_3] \cdot 2T_r \quad (13)$$

据式(7)、(8)、(12)、(13),下面通过计算机计算并绘图来比较两种协议的性能。

图2表明ANIRM占用的带宽比协议N少,并且图中ANIRM占用的带宽近似为一条直线,即几乎不随R变化,说明使用ANIRM能很好地解决多播通信的可伸缩性问题。ANIRM不随R变化的原因是:(12)式中只有 $E[M_2]$ 随R变化,据文[10]骨干链路的 p_b 非常低,使 $E[M_2]$ 几乎不发生改变。由此可知:在主动网络中,ANIRM实现逐段可靠传输能有效地降低带宽占用量,有利于解决多播通信的可伸缩性问题。

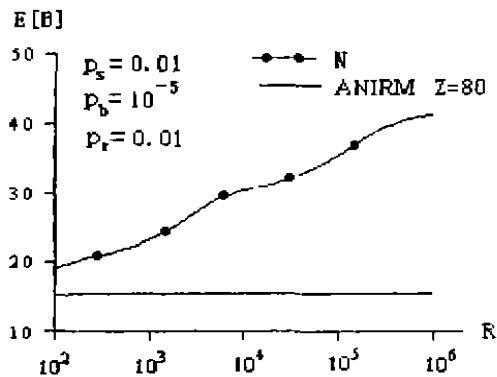


图2 协议N与ANIRM占用带宽的比较

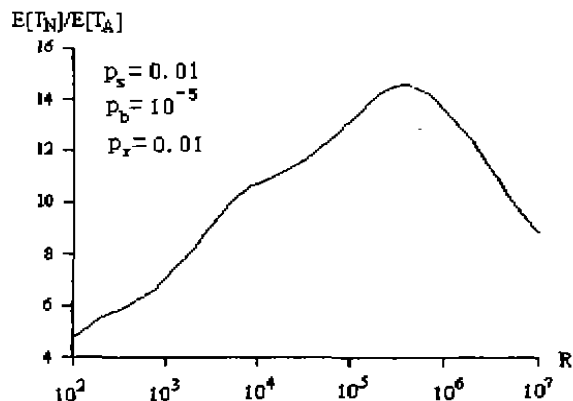


图3 协议N与ANIRM数据恢复时延的相对变化

在计算 $E[T]$ 时,令 $T_s = T_b = T_r = c$ 。图3表明协议N数据恢复时延大于ANIRM。在ANIRM中,主动结点能及时发现错误,因此,其数据恢复时延短。图3中,当 $R = 400000$ 时,比值为14.63达到最大值,说明随着R的增加,ANIRM数据恢复时延低的优势将弱化。

四、ANIRM分析

上面对带宽与数据恢复时延的比较表明ANIRM的性能优于协议N,下面将研究 p_s 、 p_b 、 p_r 和Z对 $E[B_A]$ 和 $E[T_A]$ 的影响。

图4、图5表明当Z增加时将使 $E[B_A]$ 、 $E[T_A]$ 性能变坏。但当Z保持较小时,ANIRM还是能获得较好的性能,如 $Z = 30$ 时, $E[B_A] = 1.26$, $E[T_A] = 1.11c$ 。所以,在生成多播通信树时应该让接收组中接收者的数量较小,如 $Z < 100$,才能发挥ANIRM的优点。

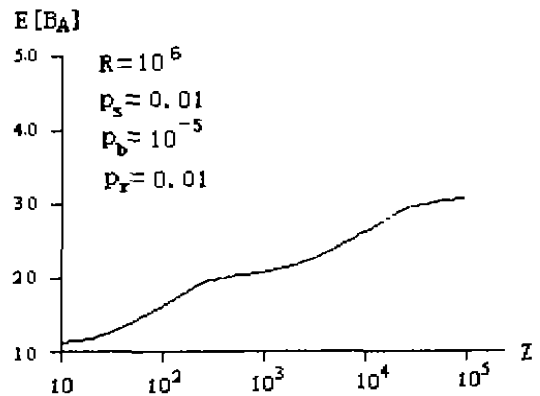


图4 Z对 $E[B_A]$ 的影响

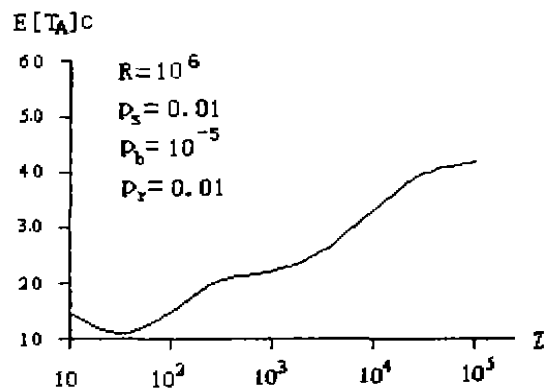


图5 Z对 $E[T_A]$ 的影响

由于只有一条发送链路,大大小于总链路数H,因此, p_s 几乎不影响 $E[B_A]$ 、 $E[T_A]$ 。

为了清楚地表明每种链路对 $E[B_A]$ 、 $E[T_A]$ 的影响情况,在考虑某种链路的影响时,假设另外两种链路

不发生错误。图 6 表明 p_b 、 p_r 都将使 $E[B_A]$ 增加。文 [10] 指出 p_b 非常低, 一些偶然因素将导致瞬时 p_b 增加, 而使用 ANIRM 后, 即使 $p_b = 0.1$ 时, $E[B_A] = 1.05$, 性能仍然较好。 p_r 的影响大于 p_b , 但当 p_r 较小时, 仍能获得较好的性能, 如 $p_r = 0.001$, $E[B_A] = 1.08$ 。

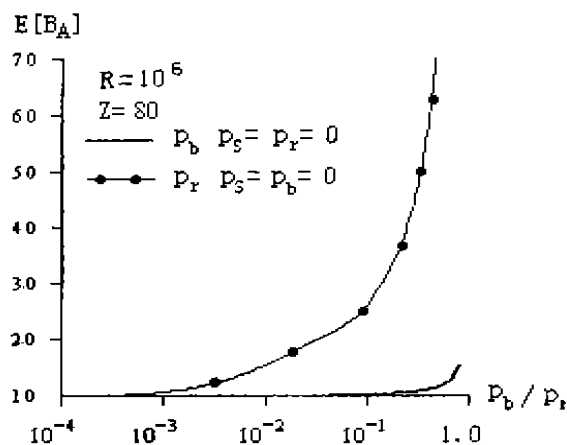


图 6 p_b/p_r 对 $E[B_A]$ 的影响

图 7 表明 p_b 、 p_r 都将使 $E[T_A]$ 增加, 因为 p_b 、 p_r 增加, 重传次数增加, 使数据恢复时延增加, 并且骨干链路 p_b 影响更大, 说明在传输过程中, 发生错误的位置越靠近发送者, 对数据恢复时延的影响越大, 因此, 及时发现错误是减少数据恢复时延的有效手段之一, 而 ANIRM 的一个特点就是能及时发现并解决错误。

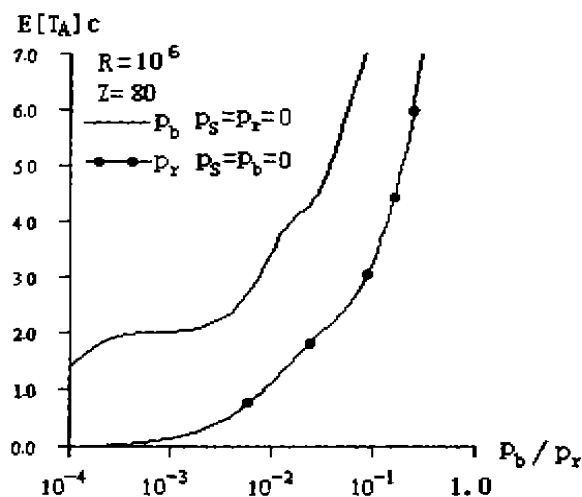


图 7 p_b/p_r 对 $E[T_A]$ 的影响

从上面分析可知: 骨干链路 p_b 很小, 则对协议性

能的影响较小, 即使在骨干链路上增加主动结点也不能进一步改善协议性能; 接收链路 p_r 较大, 但由于 Z 较小, 对协议性能影响也较小, 也没有必要在接收链路上增加主动结点, 所以, 图 1 抽象的网络模型是合理的。

结束语 ANIRM 是主动结点启动的协议, 即主动结点检测错误, 及时发现丢包或包位错误, 立即请求父结点重传数据包。ANIRM 把保证数据可靠性的工作从发送者转移到了主动结点和接收者, 实现数据在主动结点间可靠传输, 从而能解决多播通信中可伸缩性和可靠性问题。本文比较了没有主动结点和 ANIRM 两种协议, 证明 ANIRM 能有效地减少带宽, 降低数据恢复时延。通过文中分析还可以得出以下结论: 接收组中接收者的数量 Z 增加, 将导致协议性能恶化, 应该保持 $Z < 100$; 不同链路对协议性能的影响不同, 在骨干网络“边缘”设置主动结点是合理的。

参考文献

- 1 Floyd S, et al. A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing. IEEE/ACM Trans. on Networking, 1997, 5(6): 784~803
- 2 Paul S, et al. Reliable Multicast Transport Protocol (RMTP). IEEE, JSAC, 1997, 15(3): 407~421
- 3 Tennenhouse D L, et al. A Survey of Active Network Research. IEEE Commun. Mag., 1997, 35(1): 80~85
- 4 Papadopoulos C, Parulkar G, Varghese G. An Error Control Scheme for Large-Scale Multicast Application. Available at: <http://www.cse.ucsc.edu/research/ccrg/publications.html>
- 5 Nonnenmacher J, et al. How Bad is Reliable Multicast without Local Recovery? In: Proc. of IEEE INFOCOM, San Francisco, CA, USA, March 1998
- 6 Lehman Li-wen, Garland S, Tennenhouse D. Active Reliable Multicast. IEEE INFOCOM'98
- 7 Decasper D S, Plattner B. A Scalable High-Performance Active Network Node. IEEE Network, 1999 (Jan./Feb.): 8~19
- 8 Legedza U, Wetherall D, Gutttag J. Improving The Performance of Distributed Applications Using Active Networks. IEEE INFOCOM'98
- 9 Decasper D, Plattner B. Distributed Code Caching for Active Networks. In: Proc. of Infocom'98, San Francisco, April 1998
- 10 Yajnik M, Kurose J, Towsley D. Packet Loss Correlation in the Mbone Multicast Network. IEEE Global Internet mini-conference, GLOBECOM'96, 1996. 94~99