

RMTP 协议的时延分析*

林宇, 王重钢, 王文东, 程时端

(北京邮电大学 程控交换和通信网国家重点实验室, 北京 100876)

E-mail: {b0073056,wdwang,chsds}@bupt.edu.cn

http://www.bupt.edu.cn

摘要: RMTP(reliable multicast transport protocol)是一种通过修复服务器(repair server)本地恢复来提供可靠性的组播协议.对改进的 RMTP 协议的时延性能进行建模分析.推导了数据包自发送主机发出到成功地被一个随机选择的接收主机接收之间的平均时延公式.分析表明,随着丢失率的增长,RMTP 的时延性能恶化点将迅速下降;随着每个修复服务器下连接尾链路数量的增加,RMTP 的时延性能恶化点也将下降.但是,随着尾链路总数量的增加而保持每个修复服务器连接的尾链路数量不变,RMTP 的时延性能的变化很小.仿真结果较好地验证了分析的有效性.

关键词: 可靠组播传输协议;本地恢复;修复服务器;可靠组播;时延分析

中图法分类号: TP393 文献标识码: A

随着 Internet 的迅猛发展,许多应用如多点文件传送、共享白板、分布计算等,都需要可靠组播的支持.著名的可靠组播协议包括 RMTP^[1],SRM^[2]等,改进后的 RMTP 和 SRM 协议都通过基于本地恢复的策略来提高性能.

关于可靠组播的性能分析已有一些研究成果^[3-6].文献[3]对本地恢复组播协议的吞吐量和消耗带宽进行了分析,但是没有涉及 RMTP 的时延分析.文献[4]将可靠组播协议划分为 sender-Initiated,receiver-Initiated 两类,并进行了时延分析,但是文献[4]没有分析基于本地恢复的 RMTP 协议.本文对基于本地恢复的 RMTP 组播协议进行了时延分析,分析表明,随着丢失率的增长,RMTP 的时延性能恶化点将迅速下降;随着每个修复服务器下连接尾链路数量的增加,RMTP 的时延性能恶化点也将下降.但是,随着尾链路总数量的增加(假定每个修复服务器连接的尾链路数量 k 不变),RMTP 的时延性能变化很小.仿真结果较好地验证了分析的有效性.

本文第 1 节简单说明 RMTP 协议和分析模型.第 2 节进行时延分析,第 3 节给出仿真和数值结果.

1 协议和分析模型

RMTP 协议最早在文献[4]中提出,S.Paul 等人又对其进行改进.RMTP 协议基于恢复树进行重传,这个恢复树就是基于多播路由建立的多播路由树,它以发送主机为根,以接收主机为叶子,修复服务器(repair server,简称 RS)位于树的中间层次.RS 可以与路由器进行功能集成(co-located with routers),改进的 RMTP 协议进行以下工作:(1) 发送主机采用多播将一个数据包发送到一个多播组.(2) 当多播组内的一个接收主机探测到数据包丢失时,它将等待一个随机时长,然后采用多播发送一个 NAK(文献[4]中采用 ACK,使用 NAK 可以减少发送数据包的数量,避免 ACK 汇聚的现象)给本地的 RS 和接收主机(发送 NAK 给本地接收主机的目的是为了抑制多个

* 收稿日期: 2001-11-19; 修改日期: 2002-03-06

基金项目: 国家高技术研究发展计划资助项目(2001AA121052);高等学校博士学科点专项科研基金资助项目(20010013003)

作者简介: 林宇(1976 -),男,福建浦城人,博士生,主要研究领域为 IP 组播,移动组播,TCP 建模分析,无线网络;王重钢(1974 -),男,四川仁寿人,博士生,主要研究领域为宽带网络资源管理,业务控制;王文东(1963 -)男,北京人,教授,主要研究领域为宽带网服务质量和信令;程时端(1940 -),女,上海人,教授,博士生导师,主要研究领域为 TCP/IP,协议工程,流量工程,宽带网性能和服务质量.

丢包的接收主机都发送相同的 NAK,避免 NAK 集中),然后启动一个 NAK 重传时钟.如果接收主机收到一个来自于本地的其他接受主机的 NAK,它将不发送 NAK,而是直接启动 NAK 重传定时器,就好像它已经发送了一个 NAK.(3) 当一个 RS 探测到丢失后,它将等待一个随机时间,然后采用多播发送一个 NAK 给发送主机(或者位于其上游的 RS),与它处于同层的 RS 将不再发送 NAK,而是直接启动一个 NAK 重传定时器,就好像它已经发送过 NAK.(4) 当 RS 收到来自它所负责的接收主机的 NAK 时,如果它有 NAK 对应序列号的数据包,它将采用多播向位于其层次之下的接收主机重传该数据包.如果 RS 没有 NAK 对应序列号的数据包,则它将向发送主机(或位于其上流的 RS)请求重传.(5) 发送主机在收到来自 RS 的 NAK 后,发送主机将采用多播方式重传该数据包给所有组成员(包括接收主机和 RS).(6) 如果在 RS(或者接收主机)处的 NAK 重传定时器超时,RS(或者接收主机)将重新发送一个 NAK.

更细致的信息可以参见文献[4]和文献[3]中基于服务器的本地恢复协议.本文采用文献[3]中引入的系统模型,如图 1 所示.文献[7]指出,在 Internet 中,大部分的数据丢失发生在从发送源到骨干网络的链路和从骨干网络到接收主机本地站点的链路上.文献[3]将它们分别称为“源链路”(source link)和“尾链路”(tail links).文献[8]也指出“尾链路”将会成为未来网络的瓶颈.骨干网络和本地站点几乎不发生数据丢失.因此,文献[3]假定丢失仅仅发生在源链路和尾链路.在源链路上发生的丢失将会影响多播组内所有的接收主机,而尾链路上发生的丢失仅仅会影响同尾连接相连的本地站点中的接收主机.假定在尾连接中发生的丢失是相互独立的,并假定丢失事件在时间上是独立的(temporally independent).本文假定在源链路上和尾链路上的丢失率为 p_l .

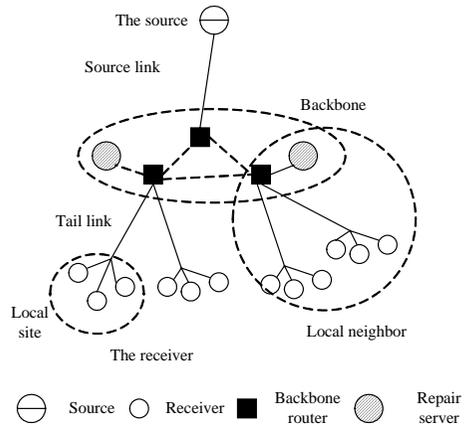


Fig.1 System model
图 1 系统模型

分析中假定 NAK 不会丢失,由于 NAK 远远小于数据包,因此这个假设条件对分析影响较小.另外,这个假设条件可以使用文献[5]中的分析放宽.假定有某种机制定义了良好的本地邻居关系(local neighborhoods),数据包能够仅多播给本地邻居的接收主机.假设有一个源链路和 T 个尾链路,每一个尾链路连接一个接收主机.假定每一个骨干网的边界路由器(RS)连接 k 个尾链路,这样,多播组内的骨干网边界路由器数量 W 为 T/k .

2 时延分析

现在开始推导协议 RMTP 时延的数学期望值.本文使用的符号如下(这些符号在文献[5]中定义,文献[3,4,9]对其进行了补充):

- X ——发送或者接收数据包的处理时间(假定发送和接收时间相同).
- Y ——发送或者接收 NAK 包的处理时间(假定发送和接收时间相同).
- p_l ——源链路和尾链路上的丢包率.
- T ——尾链路的数量.
- K ——每一个 RS 对应的尾链路数.
- M_s ——如果让所有的 RS 都成功地收到一个数据包,发送主机需发送该数据包的次数(首次发送加重传).
- M_r ——如果要让一个接收主机成功地收到一个数据包,RS 需要发送该数据包的次数(首次发送加重传).
- M_R ——如果要让一个 RS 下所有的接收主机都成功地收到一个数据包,RS 需发送该数据包的次数.
- $E[W^\theta]$ ——一个数据包在各个位置需等待的时长, $\theta \in \{S, P, R : \text{Sender, rePairserver, Receiver}\}$.
- T_{R1}, T_{R2} ——分别为从 RS 到发送主机之间重传超时定时器和从接收主机到 RS 间重传超时定时器的超时

时间.

R_{d1}, R_{d2} ——分别为 RS 发送 NAK 前随机等待时间、接收主机发送 NAK 前随机等待时间.

ρ^ω ——在 ω 处的工作负荷, $\omega \in \{S, P, R\}$, 分别对应发送主机、RS、接收主机.

W^ω ——在 ω 处的等待时延, $\omega \in \{S, P, R\}$, 分别对应发送主机、RS、接收主机.

$P(\beta)$ ——各种丢失情况的概率, $\beta \in \{a, b, c, d\}$.

T_α ——数据包在各种丢失情况下的端到端(从发送主机到接收主机)时延, $\alpha \in \{a, b, c, d\}$.

T ——从发送主机发送数据包,到一随机选定的接收主机成功接收该数据包的时延.

图 2 给出了 RMTP 协议的数据流情况.首先考虑 RMTP 的发送主机,发送主机的工作流包含 3 部分:发送主机首次发送的数据包流,标记为 λ_i^s ,设其到达率为 λ ;发送主机由于收到 NAK 包而重传形成的数据包流,标记为 λ_r^s ;来自于 RS 的 NAK 流,标记为 λ_{pn}^s .各数据流的强度如下:

$$\lambda_i^s = \lambda. \tag{1}$$

由于数据包的平均重传次数为 $E[M_s] - 1$,故 λ_r^s 为

$$\lambda_{pn}^s = \lambda(E[M_s] - 1). \tag{2}$$

由于假定存在 NAK 抑止机制,针对一个丢失数据包,一次只会有一 RS 发送 NAK 包,则 λ_r^s 为

$$\lambda_r^s = \lambda(E[M_s] - 1). \tag{3}$$

流 λ_i^s 和 λ_r^s 的处理时间为 X ,流 λ_{pn}^s 的处理时间为 Y .因此在接收主机处的负载为

$$\rho^s = \lambda E[M_s] E[X] + \lambda(E[M_s] - 1) E[Y]. \tag{4}$$

这里,为了简化问题,假定 Y 包含了设置和重置定时器的时间.设置和重置定时器的次数比发送和重传 NAK 包的次数少 1,由于设置定时器的时间很小,因此对分析结果影响很小.若假定所有流为 Poisson 流,并且数据包的服务时间为独立的随机变量,则接收主机处可以近似为 M/G/1 系统.根据文献 [10] 中的 Pollaczek-Khinchine 公式,可以得到一个数据包在发送主机处的等待时延 $E[W^s]$ 为

$$E[W^s] = \frac{(\lambda_i^s + \lambda_r^s) E[X^2] + \lambda_{pn}^s E[Y^2]}{2(1 - \rho^s)}. \tag{5}$$

再来看 RS,在每一个 RS 处,存在 4 个数据流:(1) RS 接收到的来自发送主机处的数据包流 λ^p ;(2) RS 向它所负责的接收主机进行重传的数据包流 λ_{pr}^p ;(3) RS 收到接收主机发送的 NAK 流 λ_{rn}^p ;(4) 从 RS 发送给发送主机的 NAK 流 λ_{pn}^p .各数据流的强度如下:由于 RS 和发送主机间的源链路丢失率为 p_l ,因此 λ^p 为

$$\lambda^p = \lambda E[M_s] (1 - p_l).$$

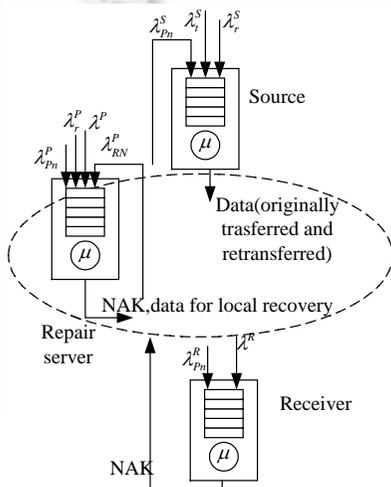


Fig.2 The data streams in RMTP

图 2 RMTP 协议中的数据流

要让一个 RS 下所有的接收主机都成功地收到一个数据包,RS 需发送该数据包的次数为 M_R ,则 RS 重传数据包流 λ_{pr}^p 为(扣除首次发送)

$$\lambda_{pr}^p = \lambda(E[M_R] - 1). \tag{7}$$

要让一个接收主机成功地收到一个数据包,RS 需要发送该数据包的次数(首次发送加重传)为 M_r ,则 RS 收到接收主机发送的 NAK 流 λ_{rn}^p 为

$$\lambda_{rn}^p = \lambda k(E[M_r] - 1). \tag{8}$$

要让一个 RS 成功地收到一个数据包,发送主机需要发送该数据包的次数(首次发送加重传)为 M_s ,则 RS 发送给发送主机的 NAK 流 λ_{pn}^p 为

$$\lambda_{Pn}^p = \lambda(E[M_s] - 1). \tag{9}$$

在 RS 处的负荷为

$$\rho^p = \lambda(E[M_s](1 - p_l) + E[M_R] - 1)E[X] + \lambda(E[M_s] + kE[M_r] - k - 2)E[Y]. \tag{10}$$

数据包在 RS 处的平均等待时延为

$$E[W^p] = \frac{(\lambda^p + \lambda_{pr}^p)E[X^2] + (\lambda_{pn}^p + \lambda_{rn}^p)E[Y^2]}{2(1 - \rho^p)}. \tag{11}$$

对于随机选择的接收主机,有两个数据流:接收主机收到的来自发送主机或者 RS 的数据流 λ^R ;接收主机发送给 RS 或者接收到来自本地其他接收主机的 NAK 流 λ_{Rn}^R .

由于 RS 和接收主机的尾链路丢失率为 p_l ,因此 λ^R 为

$$\lambda^R = \lambda E[M_R](1 - p_l). \tag{12}$$

要让一个接收主机成功收到一数据包,RS 需要发送该数据包的次数(首次发送加重传)为 M_r ,则 λ_{Rn}^R 为

$$\lambda_{Rn}^R = \lambda(E[M_r] - 1). \tag{13}$$

接收主机的工作负荷为

$$\rho^R = \lambda E[M_R](1 - p_l)E[X] + \lambda(E[M_r] - 1)E[Y]. \tag{14}$$

接收主机处的平均等待时间为

$$E[W^R] = \frac{\lambda^p E[X^2] + \lambda_{Rn}^R E[Y^2]}{2(1 - \rho^R)}. \tag{15}$$

现在来考察协议 RMTP 从发送端到接收端的时延,对于一个数据包 a_i ,随机选定的一个接收主机可能处于以下 4 种情况:

(a) 接收主机直接收到了来自发送主机首次发送的数据包.

如图 3 所示,在情况(a)下端到端时延 $E[T_a]$ 为

$$E[T_a] = E[W^S] + E[W^p] + E[W^R] + 2E[X] + E[Y] + \tau_1 + \tau_2, \tag{16}$$

其中 $E[W^S]$ 为数据包在发送端的平均等待时延, $E[W^R]$ 为数据包在 RS 的平均等待时延, τ_1 为从发送主机到 RS 的传递时延, τ_2 为从 RS 到接收主机间的传递时延.

(b) RS 接受到数据包 a_i ,但是在尾链路上该数据包丢失.

图 4 给出了在情况(b)下的操作过程,它可以分成 3 步:RS 直接从发送主机接收到数据包 a_i ;接收主机探测到数据包 a_i 的丢失(这段时间对应着 T_{RLD} (receiver loss detection));接收主机从 RS 处恢复数据包(这段时间对应着 T_{RLR} (receiver loss recovery)),其中 $E[T_{RLD}]$ 为

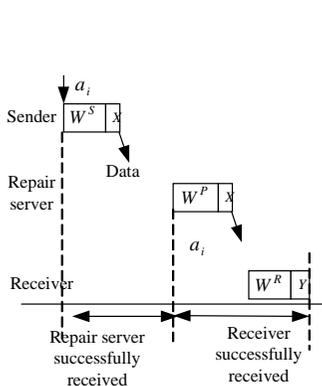


Fig.3 The operation process in case (a)
图 3 情况(a)的操作过程

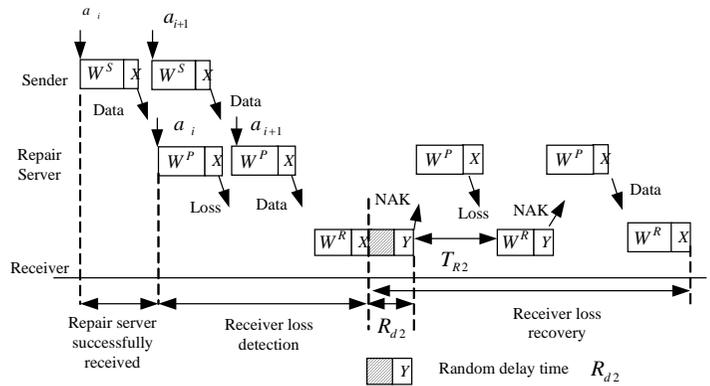


Fig.4 The operation process in case (b)
图 4 情况(b)的操作过程

$$E[T_{RLD}] = \sum_{j=0}^{k-1} \binom{k-1}{j} p_i^j (1-p_i)^{k-j-1} \frac{p_i^{j+1} / (\lambda^p + \lambda_{pr}^p)}{1-p_i^{j+1}} + E[W^p] + E[W^r] + 2E[X] + \tau_2. \quad (17)$$

式(17)中第 1 项为丢失的数据包 a_i 与下一个成功发送的数据包之间的间隔时间,其他项为下一个成功发送的数据包的传递和处理时间. $E[T_{RLD}]$ 的推导过程类似于文献[4]中的 $E[D_{N1}]$, 可以参考文献[4]中的推导过程.

从随机选定的接收主机的角度来看,它的丢失恢复过程(loss recovery phase)开始于一系列的定时器超时循环,每一定时器超时循环的时长为 $W^r + Y + T_{R2}$, 其中 T_{R2} 使本地 NAK(从接收主机到 RS)超时时长, W^r 是等待时间. 丢失恢复过程终止于 NAK 被发送给 RS 并且从 RS 接收到重传数据包. 因此有 $E[T_{RLR}]$ 为

$$E[T_{RLR}] = \sum_{j=1}^{\infty} p_i^{j-1} (1-p_i)(j-1)(E[W_{L1}^r] + E[Y] + T_{R2}) + 2\tau_2 + E[W^p] + E[W^r] + 2E[X] + E[R_{d1}]. \quad (18)$$

式(18)第 1 项为丢失恢复过程的时间,其他项为 NAK 被发送给 RS 并且从 RS 接收到重传数据包所需的时间,则在情况(b)下的平均时延 $E[T_b]$ 为

$$E[T_b] = E[T_{RLD}] + E[T_{RLR}] + E[W^s] + E[X] + \tau_1. \quad (19)$$

(c) 数据包 a_i 在源链路发生了丢失,但是在尾链路上没有发生丢失.

图 5 给出了情况(c)的操作过程,它可以分成 3 个步骤:RS 探测到数据包 a_i 丢失(这段时间对应于 T_{PLD} (repair server loss detection));RS 从发送主机恢复数据包 a_i (这段时间对应于 T_{PLR} (repair server loss recovery));接收主机直接从 RS 成功收到数据包 a_i . $E[T_{PLD}]$ 的推导类似于 $E[T_{RLD}]$, $E[T_{PLR}]$ 的推导类似于 $E[T_{RLR}]$. 推导结果如下:

$$E[T_{PLD}] = \sum_{j=0}^{N_p} \binom{N_p-1}{j} p_i^j (1-p_i)^{N_p-j-1} \frac{p_i^{j+1} / \lambda_t^s}{1-p_i^{j+1}} + E[W^s] + E[W^p] + 2E[X] + \tau_1, \quad (20)$$

$$E[T_{PLR}] = \sum_{j=1}^{\infty} p_i^{j-1} (1-p_i)(j-1)(E[W^p] + E[Y] + T_{R1}) + 2\tau_1 + E[W^s] + E[W^p] + 2E[X] + E[R_{d1}]. \quad (21)$$

式(19)中第 1 项为丢失的数据包 a_i 与下一个成功发送的数据包之间的间隔时间,其他项为下一个成功发送的数据包到达 RS 所需的传递和处理时间.

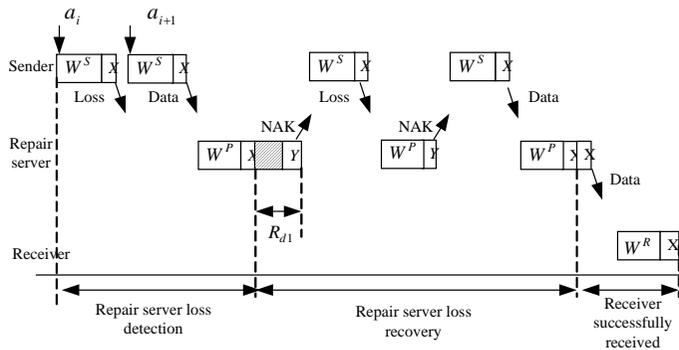


Fig.5 The operation process in case (c)

图 5 情况(c)的操作过程

情况(c)下的端到端时延 $E[T_c]$ 为

$$E[T_c] = E[T_{PLD}] + E[T_{PLR}] + E[W^r] + 2E[X] + \tau_2. \quad (22)$$

(d) 在源链路和尾链路上数据包 a_i 都发生了丢失.

图 6 显示了在情况(d)下的操作过程,它可以分成 4 个步骤:RS 探测到数据包 a_i 的丢失(对应于 T_{PLD});RS 从发送主机处成功恢复数据包 a_i (对应于 T_{PLR});接收主机探测到(对应于 T_{RLD});接收主机从 RS 成功恢复数据包 a_i (对应于 T_{RLR}).

情况(d)下的端到端时延 $E[T_d]$ 为

$$E[T_d] = E[T_{PLD}] + E[T_{PLR}] + E[T_{RLD}] + E[T_{RLR}]. \quad (23)$$

协议 RMTP 的平均端到端时延 $E[T]$ 为

$$E[T] = P(a)E[T_a] + P(b)E[T_b] + P(c)E[T_c] + P(d)E[T_d]. \tag{24}$$

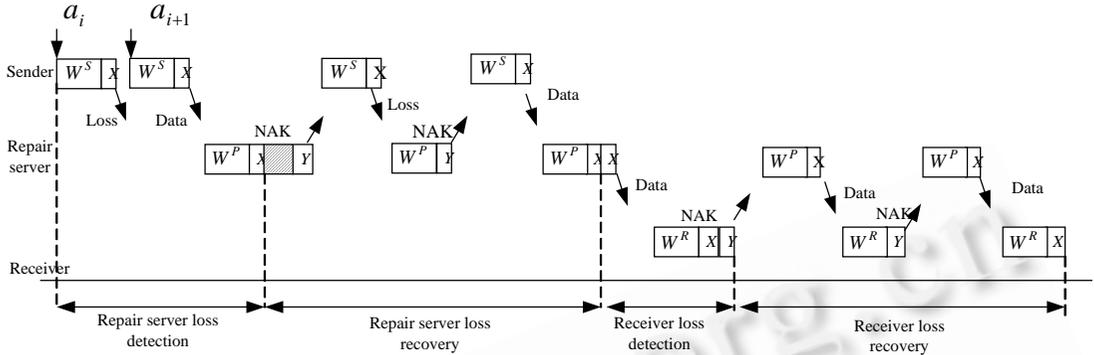


Fig.6 The operation process in case (d)

图 6 情况(d)的操作过程

由于假定源链路和尾链路的丢失率为 p_l , 因此各种情况的概率为

$$P(a) = (1 - p_l)^2, P(b) = p_l(1 - p_l), P(c) = p_l(1 - p_l), P(d) = p_l^2. \tag{25}$$

下面介绍几个参量的计算方法. $E[M_s]$ 可以如下计算, 显然有

$$P[M_s \leq m] = 1 - p_l^m, m = 1, \dots \tag{26}$$

对于一个整数型随机变量 X , 有

$$E[X] = 1 + \sum_{m=1}^{\infty} (1 - P(X \leq m)) = \sum_{m=1}^{\infty} P(X = m)m, \tag{27}$$

则 $E[M_s]$ 为

$$E[M_s] = 1/(1 - p_l). \tag{28}$$

由于 $P[M_R \leq m] = (1 - p_l^m)^k, m = 1, \dots$, 因此有

$$E[M_R] = 1 + \sum_{m=1}^{\infty} (1 - (1 - p_l^m)^k). \tag{29}$$

由于 $P[M_r \leq m] = 1 - p_l^m$, 因此有

$$E[M_r] = 1 + p_l/(1 - p_l). \tag{30}$$

3 仿真和数值结果

为了与以往文献保持一致, 本文采用文献[6]中数据包发送/接收、NAK 包发送/接收时长的测量结果. 取 $E[X] = 500\mu s, E[Y] = 100\mu s, \tau_1 = 15ms, \tau_2 = 5ms$. 为了便于计算, 假定每个数据包的处理时间为常数. 其他参数如下:

$$E[R_{d1}] = \tau_1 / 2 + E[Y], E[R_{d2}] = \tau_2 / 2 + E[Y],$$

$$T_{R1} = (2(\tau_2 + \tau_1) + E[X] + E[W^S]) * 1.5,$$

$$T_{R2} = (2\tau_2 + E[X] + E[W^P]) * 1.5.$$

为了验证分析的结果, 我们采用图 7 中的拓扑进行仿真验证. 图 7 中发送主机为多播源, Router1~Router6 为 RS, Router1~Router5 每个 RS 连接 16 个接收主机, Router6 连接 20 个接收主机, 接

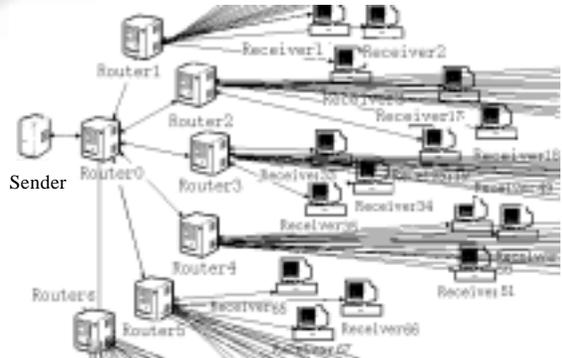


Fig.7 Simulation topology

图 7 仿真的拓扑图

收主机总数为 100 个.发送主机到 Router0 链路的时延为 10ms,其他链路时延为 5ms.发送主机到 Router0 之间链路(相当于“源链路”)以及 Router1~Router6 到接收主机间链路(相当于“尾链路”)的带宽都为 1Mb/s,Router0 和 Router1~Router6 带宽为 10Mb/s.我们在 Router0 路由器对数据包以概率 p_l 进行随机丢弃,以模拟“源链路”的丢包,在 Router1~Router6 路由器上对数据包以概率 p_l 随机丢弃,以模拟“尾链路”的丢包.

图 8 给出了 $T=100, k=8, p_l$ 从 0.0005, 0.005 到 0.051(对应的端到端丢失率 $p = 1 - (1 - p_l)^2$ 为 0.001, 0.01, 0.1) 的数值结果和在 $p_l=0.005$ 和 $p_l=0.051$ 下的仿真结果.从图中可以看出,随着到达率的增大,平均时延不断增大,当到达率增加到一个特定值时,时延性能将严重恶化.随着丢包率的增大,时延性能恶化点变小,这是因为较大的丢包率将增加重传的次数,重传的数据流加大了发送主机、RS 的负荷,增加了排队等待时间,导致端到端成功传递的时延迅速恶化.

图 9 给出了在 $p_l=0.051, T=100$ 条件下, k 在 8, 16, 32 时的平均时延数值结果和 k 在 8 和 16 下的仿真结果($k=32$ 的情况因节点太多无法进行仿真).从图中可以看出,随着每个 RS 连接的接收主机数量的增加,平均时延的性能恶化点不断下降.这是由于每个 RS 连接的接收主机增加,RS 的负荷将增大,导致平均时延的增大.

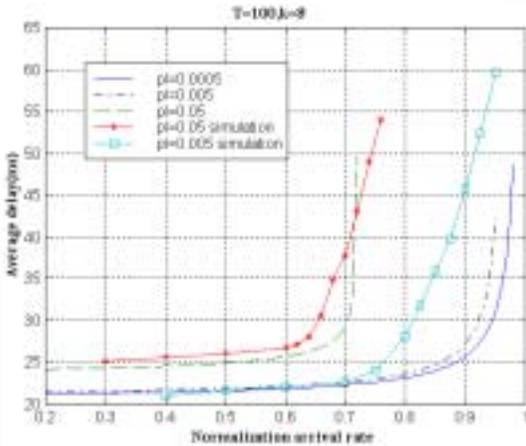


Fig.8 Average delay under different loss probability
图 8 不同丢包率下平均时延曲线

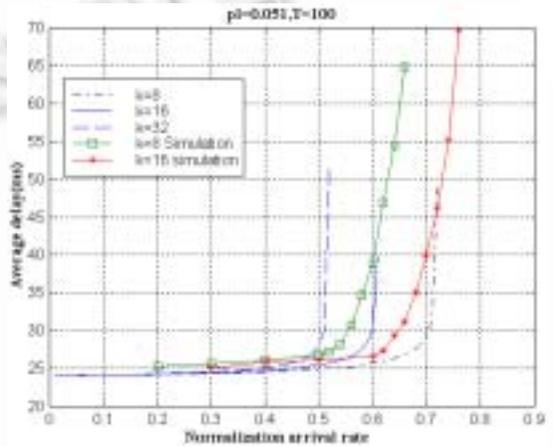


Fig.9 Average delay under different k
图 9 不同 k 值下平均时延曲线

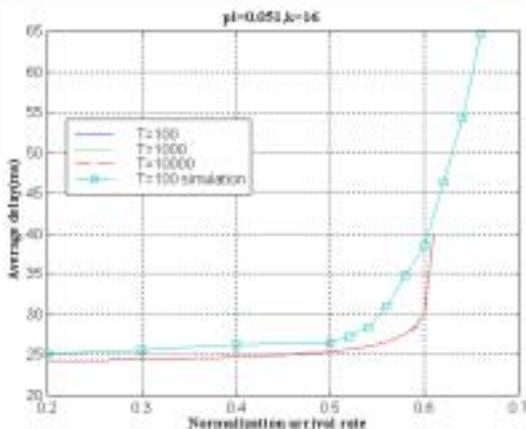


Fig.10 Average delay under different number of receivers
图 10 不同接收主机数量下平均时延曲线

图 10 给出了 $p_l=0.051, k=16$ 条件下, T 为 100, 1 000, 10 000 下的平均时延的理论曲线和 T 为 100 下的仿真曲线(T 为 1 000 和 10 000 的情况因节点太多无法进行仿真).从图中可以看出,随着接收主机数量的增长,时延性能恶化点几乎不变.在进行时延分析时,我们假定只有一个 RS 会发送 NAK 给发送主机,因此,保持 p_l 和 k 不变,接收主机的增加对发送主机的负荷影响不大,这一点与实际情况可能会有一些差别.

本文对 RMTP 协议的时延性能进行了建模分析,分析表明:(1) 随着丢失率的增大, RMTP 的时延性能恶化点将迅速下降,因此在丢包严重的网络环境下应用 RMTP 具有一定的困难.(2) 随着每个 RS 下连接尾链路数量的增加, RMTP 的时延性能恶化点也将下降,在实际应用中,应尽量将用户请求均匀地在多个 RS 间分担,避免少数 RS 负荷过重.(3) 随着尾链路总数量的增加而保持每个 RS 连接的尾链路数量 k 不变, RMTP 的时延性能变化很小.因此在应用中,可通过增加 RS 来提高系统的服务性能,特别是在用户密集的区域,可增加新的 RS 来提高服务性能.仿真结果较好地

验证了分析的有效性.

References:

- [1] Paul, S., Sabnani, K.K., Lin, J.C., *et al.* Reliable multicast transport protocol (RMTP). *IEEE Journal on Selected Areas in Communications*, 1997,15(3):407 ~ 421.
- [2] Floyd, S., Jacobson, V., Liu, C., *et al.* A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 1997,5(6):784~803.
- [3] Kasera, S., Kurose, J., Towsley, D. A comparison of server-based and receiver-based local recovery approaches for scalable reliable multicast. *Proceedings of the IEEE*, 1998,3:988~995.
- [4] Yamamoto, M., Kurose, J., Towsley, D., *et al.* A delay analysis of sender-initiated and receiver-initiated reliable multicast protocols. *Proceedings of the IEEE*, 1997,2:480 ~488.
- [5] Pingali, S., Towsley, D., Kurose, J.F. A comparison of sender-initiated and receiver-initiated reliable multicast protocols. *IEEE Journal on Selected Areas in Communications*, 1997,15(3):398~406.
- [6] Nonnenmacher, J., Biersack, E. Performance modeling of reliable multicast transmission. *Proceedings of the IEEE*, 1997,2: 471~479.
- [7] Yajnik, M., Kurose, J., Towsley, D. Packet loss correlation in the Mbone multicast network. In: *Proceedings of the Global Internet Conference*. 1996. 94~99.
- [8] Holbrook, H.W., Singhal, S.K., Cheriton, D.R. Log-Based receiver-reliable multicast for distributed interactive simulation. In: *Proceedings of the ACM SIGCOMM*. 1995. 328~341.
- [9] Kasera, S.K., Kurose, J., Towsley, D. Scalable reliable multicast using multiple multicast groups. *IEEE/ACM Transactions on Networking*, 2000,8(3):294~310.
- [10] Kleinrock, L. *Queueing Systems Theory*, vol.1. Wiley-Interscience, 1974.

A Delay Analysis of RMTP*

LIN Yu, WANG Chong-gang, WANG Wen-dong, CHENG Shi-duan

(Department of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

E-mail: { b0073056,wdwang,chsdl }@bupt.edu.cn

<http://www.bupt.edu.cn>

Abstract: RMTP is a reliable multicast protocol based on local loss recovery through repair servers. A delay modeling analysis of RMTP is presented, which derives the expression of average delay between the initial arrival of a packet at a sender and its correct reception at a randomly chosen receiver. The analysis results indicate that the saturated value of RMTP delay performance decreases rapidly under more loss environment. With the increasing of the number of tail links served by per repair server, the saturated value of delay also decreases. However, with the increasing of the number of total tail links while keeping the number of tail links served by per repair server constant, the saturated value of delay hardly changes. Finally, the analysis is confirmed by simulation results.

Key words: RMTP; local recovery; repair server; reliable multicast; delay analysis

* Received November 19, 2001; accepted March 6, 2002

Supported by the National High-Tech. Research and Development Plan of China under Grant No.2001AA121052; the Research Fund for the Doctoral Program of Higher Education (RFDP) of China under Grant No.20010013003