一种基于包排队方式的网络路径可用带宽探测方法*)

朱尚明[1] 高大启[1] 庄新华[2]

(华东理工大学计算机科学与工程系 上海 200237)[1] (美国密苏里大学计算机科学系 美国 MO 65211)[2]

摘 要 对IP网络路径带宽的探测是目前网络研究领域的一个热点。本文提出了一种针对端到端的网络、基于包排队方式的双向双步长网络路径可用带宽的探测方法。该探测方法由时延监视和UDP发送两个进程组成,基于包的排队时延来获取路径的可用带宽,并通过采用双向双步长的方法来递增或递减UDP包的发送速率。所提出的探测方法可以明显减少探测次数和运行时间,从而降低探测带来的开销。实验结果显示,所设计的方法和技术是可行的和有效的。

关键词 瓶颈串路,包排队,双向双步长,可用带宽,双程时延

An Approach to the Available Bandwidth Measurement for Network Paths Based on Packet Queuing

ZHU Shang-Ming[1] GAO Da-Qi[1] ZHUANG Xin-Hua[2]

(Department of Computer Science, East China University of Science and Technology, Shanghai 200237)[1]

(Department of Computer Science, University of Missouri-Columbia, Columbia MO65211)[2]

Abstract The bandwidth measurement for IP network paths is a hotspot in network research area. A bi-direction bi-step approach to the available bandwidth measurement for an end-to-end network path based on packet queuing is proposed in this paper. This approach consisting of delay tracing and UDP sending processes computes the available bandwidth of a path by the delay of packet queuing, and increases or reduces the sending rate of UDP packets through bi-direction bi-step. The proposed approach can obviously shorten the measuring times and running time, so the overhead of measurement is reduced. The experiment results show that the proposed approach and implementation are valid and effective.

Keywords Bottleneck link, Packet queuing, Bi-direction bi-step, Available bandwidth, Round trip time

1 引言

对IP网络路径性能的探测一直是一个十分活跃的研究领域,尤其是路径时延和可用带宽的探测。IP网络路径性能的探测技术可广泛应用于优化网络性能、提高端到端的可靠性、进行多路径路由以及QoS(服务质量)控制等。

网络路径性能探测技术可以分为主动探测和被动探测两大类[1]。采用主动探测技术,源节点通过向目的节点发送探测数据包来获得路径的性能参数,例如类似于ping命令,主机向目的节点发送ICMP ECHO_REQUEST或者TCP SYN包,然后等待目的节点的回应;被动探测技术则通过跟踪源节点对目的节点的访问抽样信息来计算路径的性能,例如Web请求-应答数据包可以作为样本信息,来测量路径的性能参数。主动探测技术使用带外数据探测网络路径的性能,会带来一定的开销,但更有针对性和目的性,测量结果也较准确;被动探测技术使用带内数据来评估网络性能,开销较小,但获得的路径信息也有限。本文将采用主动探测技术对端到端网络中一条路径的可用带宽进行动态测量,并重点研究如何降低探测算法的运行开销。

针对Internet或IP网络,目前已开发出了不少网络路径的探测工具,例如用于探测Internet上瓶颈串路的探测工具BFind[2]和PathNeck[3]、测量一条端到端路径的可用带宽探测工具PathLoad[4]和TReno[5],此外还有基于包对机制来测量一条路径的瓶颈串路带宽容量的探测工具RateTracer[6]和NetTimer[7],以及通过发送不同大小的单包来测量IP网络的瓶颈串路的工具NCS[8]、PathChar[9]和PChar[10]等。但是,以上探测工具要么需要结合Traceroute工具进行测量,要么仅针对带宽容量进行探测,要么会带来较大的测量开销。因此,本文将在上述探测工具的研究基础上,提出一种针对端到端的网络、基于包排队时延的双向双步长网络路径可用带宽的探测方法。

2 可用带宽的度量

网络带宽反映了网络的数据传输能力,网络中一条路径或串路带宽的测量可以分为带宽容量和可用带宽两种。其中带宽容量指路径或串路理论上能够达到的最大带宽,一般是固定和预知的,不能反映网络负载的动态变化;可用带宽又称为剩余带宽,等于带宽容量减去当前已用带宽,反映了当前路径或串路的实际吞吐能力。因此,本文主要研究路径可用带宽的度量问题。

*)国家自然科学基金(No. 60373073)、美国NIH基金(DHHS1 R01 DC04340-01A2)和美国NSF基金(EIA 9911095)。朱尚明 副教授,博士研究生,研究方向为计算机网络和多媒体通信;高大启 教授,博士生导师,研究方向为计算机网络和智能理论。庄新华 教授,博士生导师,研究方向为多媒体通信和数字图像编码。

假设对于一个有向网络拓扑，源-目的节点间的每条路径 r 都由 k 个串路构成，即 $r=\{l_1,l_2,\cdots,l_k\}$。设 C_l 为一个串路的最大带宽容量，不失一般性，可假定 C_l 是固定的和已知的。再设 b_l 为串路 l 的可用带宽，b_r 为路径 r 的可用带宽。b_l 和 b_r 都是随着网络拓扑结构和流量负载而动态变化的，其度量方法如下：

若当前已有 m 条路径经由串路 l，那么串路 l 的可用带宽为

$$b_l=C_l-\sum_{i=1}^{m}B_i \tag{1}$$

其中 B_i 为经由串路 l 的第 i 条路径的当前已用带宽。

若路径 r 由 k 个串路构成，那么路径 r 的可用带宽为

$$b_r=\min\{b_1,b_2,\cdots,b_k\} \tag{2}$$

显然，一条路径的可用带宽即是其瓶颈串路上的带宽。

3 探测算法的设计

对网络带宽的测量是比较困难的，目前带宽的探测技术主要存在两个方面的问题：一是如何准确地测量可用带宽，二是如何降低测量开销。目前还没有一种很好的方法能对网络路径带宽进行精确的测量，当前可用带宽探测的主要技术有吞吐量方式、单包方式、包对方式和包排队时延方式等。吞吐量方式采用单位时间内传输的数据包量来测量可用带宽，比较简单，但是测试是不充分的，因为它没有考虑由串路层头、IP 头、TCP/UDP 头和重传所消耗的带宽；单包和包对方式都是基于包的传输时间跟包的大小成线性比例以及路由器工作在存储转发方式下进行探测，并假定传输路径上没有其他网络流量导致探测包的排队阻塞；包排队时延方式可以探测 IP 网络路径上瓶颈串路的带宽，它根据包的发送速率大于路径的可用带宽时将在瓶颈串路上引起包的排队，通过排队时延来获取该串路的带宽。包排队时延方式设计简单，且不需要路由器持续地处理 ICMP 包和不要及时的确认信息(ACK)，通过发送较少数量的包，而保持同样精确的测量效果。

本文提出了一种基于包排队时延的双向双步长算法对路径的可用带宽进行探测。该算法的设计思想类似于 BFind[2] 和 PathLoad[4]，基于包的排队时延来获取路径的可用带宽，它由时延监视和 UDP 发送两个进程组成，但通过采用双向双步长的方法来递增或递减 UDP 包的发送速率。所提出的算法采用双程时延来计算排队时延，仅需要在源节点运行，和 PathLoad 相比，实现更加简单；采用双向双步长法可以缩短探测次数和运行时间，和 BFind 相比，降低了探测带来的开销。

基于排队时延的探测原理分析如下：

假设源节点在时刻 0 发送了一个探测包，经过了 t_n 的时间到达了第 n 个节点，那么有

$$t_n=\sum_{k=0}^{n-1}\left(\frac{S}{b_k}+d_k+q_k\right) \tag{3}$$

其中 S 表示探测包的大小(是固定的)；b_k 表示第 k 个节点和第 $(k+1)$ 个节点之间的串路的可用带宽；d_k 表示传输时延，它等于探测包在第 k 个节点上的转发时延加上第 k 个节点和第 $(k+1)$ 个节点之间的串路的传播时延；q_k 表示在第 k 个节点上的排队时延，且有 $k=0$ 的节点表示发送探测包的源主机。

当探测包在所有节点上的排队时延等于 0 时，即是源节点到达第 n 个节点的最小时延，我们有

$$\min\{t_n\}=S\sum_{k=0}^{n-1}\frac{1}{b_k}+\sum_{k=0}^{n-1}d_k \tag{4}$$

随着探测包发送速率的增加，将首先在瓶颈串路上引起包的排队，考虑到上面的式(3)，探测包在瓶颈串路上的排队时延为

$$qbottleneck=t_n-\min\{t_n\} \tag{5}$$

根据探测包的排队时延增加的幅度和次数，便可以判别出探测包的发送速率是否达到或超过了路径的可用带宽。

基于以上原理，我们所设计的双向双步长可用带宽探测算法如下：

(1) 在源节点向目的节点发送若干个探测包，获得其最小时延，作为路径的传播时延；

(2) 在源节点启动一个 UDP 发送进程，以一个较低的初始速率向目的节点发送数据包；

(3) 在源节点同时启动一个时延监视进程对 UDP 发送过程进行跟踪监视，获得探测包在 UDP 发送过程中经由这条路径的传输时延；

(4) 根据监视进程获得的传输时延和初始计算的传播时延的差值，可以估算一条路径的队列长度，发送进程根据监视进程计算的排队时延的增加程度，采用双向双步长的方法来递增或递减发送进程的发送速率；

(5) 当排队时延的增加次数达到一定的阈值时，可认为此时 UDP 的发送速率即是路径(瓶颈串路)上的可用带宽；

(6) 对于没有识别到可用带宽的路径，需要适当增加探测的次数。

上述可用带宽的测量是基于时延进行测量的，因此必须首先解决时延的探测问题。网络中一条路径的时延可以分为单程时延和双程时延两种。单程时延的测量需要解决源-目的主机之间的时钟基准不一致的问题，精确测量起来困难很大，系统设计和部署都很复杂；双程时延的探测都在源节点进行计算，不存在时钟基准问题，而且容易部署和解析。由于上述可用带宽的探测算法采用传输时延的差值来估算路径的队列长度，简单起见，可以使用双程时延作为路径的传输时延。

双程时延的探测算法设计如下：

(1)在源节点通过填充随机比特位生成一个具有固定长度的探测包；

(2)目的节点准备接收探测包；

(3)在源节点给探测包打上一个发送时间戳，然后将探测包发往目的节点；

(4)探测包到达目的节点后，目的节点尽快回应一个应答包到源节点；

(5)如果应答包在一个合理的时间内到达，在接收应答包时尽快记下一个接收时间戳；接收时间戳和发送时间戳的差值就是双程时延；

(6)如果探测包不能在一个合理的时间内到达，则认为双程时延值是不确定的。

4 探测算法的实现

本文在.Net 环境下利用 MFC 的原始套接字对基于包排队方式的路径可用带宽探测算法进行实现。根据前面的分析，路径可用带宽的探测过程可分为时延监视(TraceRTT)和 UDP 发送(SendUDP)两个进程。

4.1 时延监视进程

对一条路径逐跳双程时延可以利用 ICMP 和 IP 数据包

头部中的 TTL(Time To Live)值进行探测。TTL 是一个 IP 数据包的生存时间,当每个 IP 数据包经过路由器的时候,路由器都会把该数据包的 TTL 值减去 1。这样,TTL 值就相当于一个路由器的计数器。当源主机向目的主机发送一个 TTL 大于 0 的 ICMP_ECHO(回显)报文时,网络中的路由器收到这个报文后,将根据 IP 报头中的 TTL 值对该报文进行处理,如果 TTL 减 1 后的值不为 0,将转发该数据包到下一网络节点;如果 TTL 减 1 后的值为 0,路由器将不再转发这个数据包,而是直接丢弃,并且发送一个 ICMP_TIMEOUT(超时)信息给源主机,且 ICMP_TIMEOUT 报文中 IP 报头的信源地址就是这个路由器的 IP 地址。如果一个 ICMP_ECHO 报文到达了目的主机,目的主机将回送一个 ICMP_ECHOREPLY(回显应答)报文给源主机,且在 ICMP_ECHOREPLY 报文中 IP 报头的信源地址就是这个目的主机的 IP 地址。

源节点在发送时延探测包时,将 ICMP_ECHO 回显报文的 IP 数据包的报头的 TTL 设置为当前跳数,同时获得当前时间并保存到 *SendTime* 中,然后将该报文发送给目的主机,同时启动接收进程。一旦接收到来自路由器的 ICMP_TIMEOUT 超时报文或来自目的主机回送的 ICMP_ECHOREPLY 回显应答报文,立即将当前时间保存到 *RecvTime* 中。然后将根据报文类型进行输出:如果报文类型为 ICMP_TIMEOUT 超时报文,说明来自路由器,输出路由器的 IP 地址和经历的双程时延,*RecvTime* 和 *SendTime* 之间的差值就是该数据包到达该路由器并返回的时间,同时将逐跳跳数加 1,启动下一个发送接收进程,将路径延伸到下一路由器;如果报文类型为 ICMP_ECHOREPLY 回显应答报文,说明数据包已到达目的主机,将输出目的主机的 IP 地址和经历的双程时延,*RecvTime* 和 *SendTime* 之间的差值就是该数据包到达该目的主机并返回的时间。如果逐跳跳数已超过最大跳数或者出现了目的主机不可达的情况以及发送和接收进程出现了超时等待,时延探测进程都将终止。

4.2 UDP 发送进程

UDP 发送进程实现的关键是 UDP 包发送速率的调节和控制,即如何根据监视进程计算的排队时延的增加程度来增加或降低发送速率,排队时延定义为在某一特定发送速率下监视进程获得的传输时延和初始计算的传播时延的差值。在某一发送速率下,如果探测到的排队时延的增加超过了设定的幅度,将视为探测到一次增幅,如果在探测时间段内累计增幅次数超过了设定的阈值,将视为此时的发送速率已经达到路径的可用带宽。本文采用双向双步长的方法来递增或递减 UDP 包的发送速率,算法实现框图如图 1 所示。

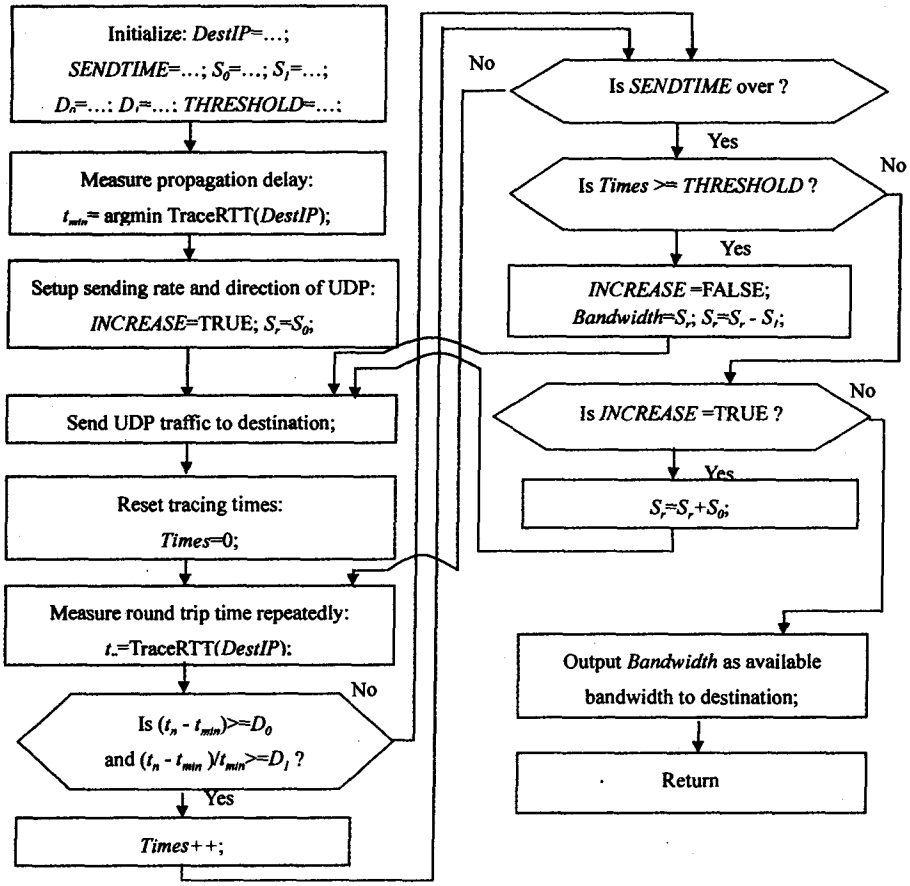

图 1 双向双步长法探测可用带宽

首先初始化算法的运行参数:目的节点的 IP 地址 *DestIP*,每个发送速率的持续时间 *SENDTIME*,递增步长 S_0,递减步长 S_1,排队时延的绝对增加幅度 D_0,相对增加幅度 D_1 以及排队时延的增幅次数阈值 *THRESHOLD*。接下来发送若干个探测包,通过时延监视进程(TraceRTT)获得从源节点到目的节点的路径最小时延 t_{min},作为路径的传播时延。

然后启动 UDP 发送进程。发送速率初始变化方向为递增(*INCREASE*=*TRUE*),UDP 发送进程从 S_0 的速率开始发送,根据监视进程排队时延的增加幅度和增幅次数进行调节:在某一发送速率下,如果探测到的排队时延的增加超过了

设定的绝对增加幅度 D_0 和相对增加幅度 D_1，将视为探测到一次增幅（*Times*++）；如果排队时延的增幅次数没有超过阈值 *THRESHOLD*，*UDP* 发送速率以 S_0 为步长持续递增，直至增大到一定速率时，使排队时延的增幅次数超过了阈值 *THRESHOLD*，此时则调整发送速率的变化方向为递减（*INCREASE*=*FALSE*），并以 S_1 为步长持续递减，直至递减到一定速率，使排队时延的增幅次数低于阈值 *THRESHOLD*，此时，上次的发送速率即可估算为源节点到目的节点之间路径的可用带宽。

整个发送和监视进程都在有限的步骤或时间内重复执行，如果探测到了路径的可用带宽或探测次数超过了一定的上限，探测程序将终止运行。探测程序的测量精度取决于递减步长 S_1，探测程序的时间开销取决于 S_0、S_1 以及每个发送速率的持续时间 *SENDTIME*。

5 探测结果及分析

为了对所设计和实现的探测算法进行验证和分析，我们使用的网络实验环境如图 2 所示，源主机和所有路由器均配置为 100Mbps 的网络接口，目的主机配置为 10Mbps 的网络接口。算法运行参数设置为：UDP 发送包的大小 *S* 为 4kByte，每个发送速率的持续时间 *SENDTIME* 为 180s，递增步长 S_0 为 2Mbps，递减步长 S_1 为 200kbps，排队时延的绝对增加幅度 D_0 为 5ms，相对增加幅度 D_1 为 20%，排队时延的增幅次数阈值 *THRESHOLD* 为 50。

首先在未启动发送进程的条件下，对双程时延进行探测，计算出源节点至目的节点之间路径的最小时延，然后启动发送和监视进程对可用带宽进行测量，运行结果如表 1 所示。表 1 中 *Times* 表示在某一发送速率下，探测到的排队时延的增加超过了设定的绝对增加幅度和相对增加幅度的次数。从运行结果中可以看出，经过 6 次探测即可获得源节点至目的节点之间路径的可用带宽，为 10Mbps，和实际网络环境下的设置是相符的，表明所提出的算法是可行的和有效的。同样条件下，如果采用 BFind 工具，则需要经过(10M－2M)/200k＋1＝41 次探测才能完成。

不同的发送速率下排队时延随时间的变化关系如图 3 所示。从图 3 可以看出，在发送速率低于路径的可用带宽 10Mbps 时，排队时延的变化相对较小，而一旦达到路径的可用带宽，其时延增加幅度和频度明显增大。

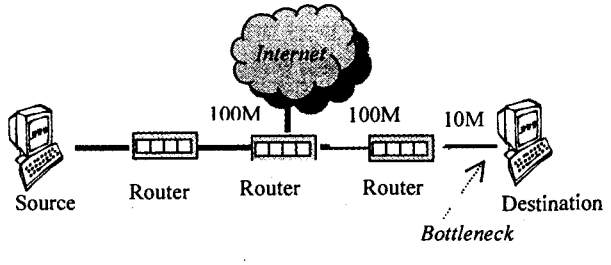

图 2 网络实验环境

表 1 可用带宽探测结果

Sending Rate(bps)	2M	4M	6M	8M	10M	9.8M
Times	24	22	20	15	69	28
Is *Times*>50	no	no	no	no	yes	no

排队时延超过设定阈值的个数随发送速率的变化关系如图 4 所示。从图 4 可以看出，在发送速率低于路径的可用带宽 10Mbps 时，双程时延超过了设定的绝对增加幅度和相对增加幅度的次数维持在 15～30 之间，而一旦达到路径的可用带宽，其时延超过增加幅度的次数急剧上升。

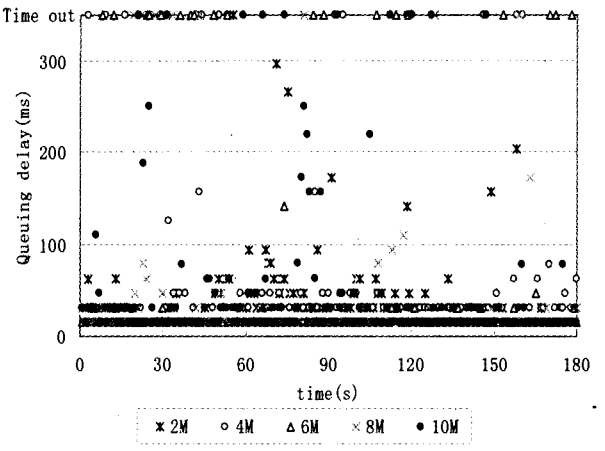

图 3 不同发送速率下的排队时延

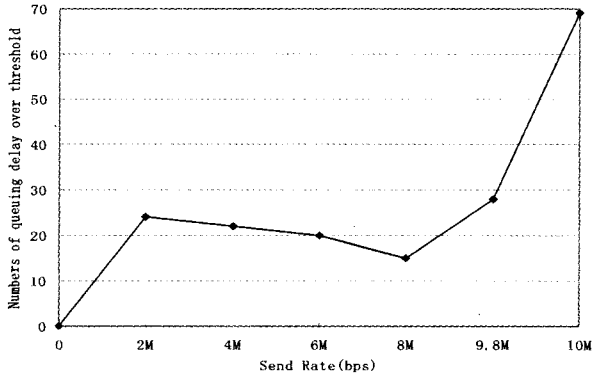

图 4 排队时延超过设定阈值的个数变化

结束语 网络路径可用带宽的探测技术对于优化网络性能、提高端到端的可靠性、进行多路径路由以及 QoS 控制等都十分重要。本文在已开发出的网络路径探测工具的研究基础上，针对端到端的网络，提出了一种基于包排队时延的双向双步长网络路径可用带宽的探测方法。该探测方法基于包的排队时延来获取路径的可用带宽，由时延监视和 UDP 发送两个进程组成，通过采用双向双步长的方法来递增或递减 UDP 包的发送速率，可以明显缩短探测次数和运行时间，从而降低探测带来的开销。实验结果显示，所设计的方法和技术是可行的和有效的。

参 考 文 献

1 Akella A. Endpoint-based routing strategies for improving Internet performance and sesilience:[Ph D Dissertation]. School of Computer Science,Carnegie Mellon University, Pittsburgh, Sep. 2005. 30～38

2 Akella A, Seshan S, Shaikh A. An empirical evaluation of wide-area internet bottlenecks. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, Miami Beach, FL, USA, October 2003. 101～114

3 Hu N, Li L, Mao M, et al. Locating Internet bottlenecks: algorithms, measurements, and implications. In: Proc. of ACM SIGCOMM '04, Portland, OR USA, August 2004. 41～54

（下转第 58 页）

系统总体数据流图有数据包发送过程和数据包接收两个过程。发送过程如图4所示，限于篇幅，本文对数据包接受不再赘述，原理近似。

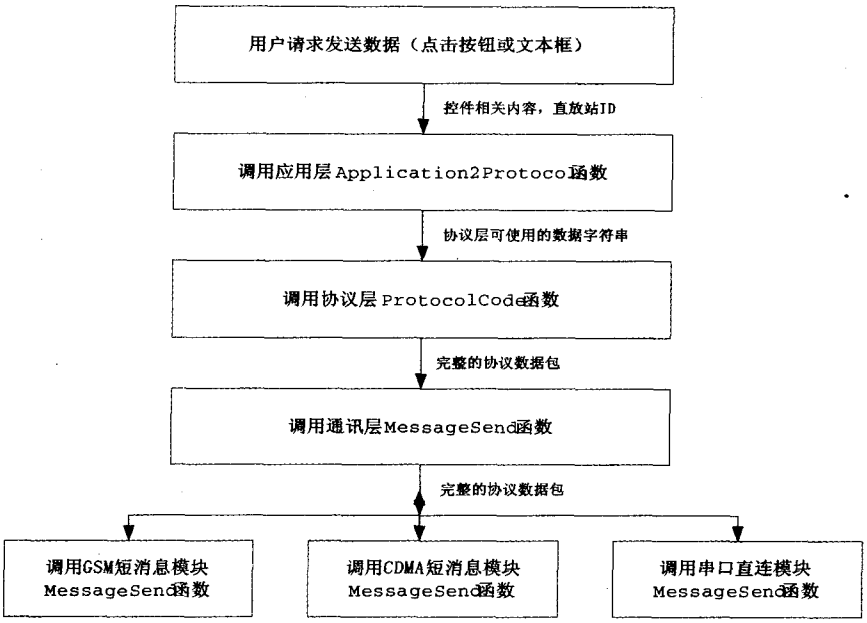

图4 iRepeat系统的数据包发送过程

3.2 实验过程与结论

为验证系统设计的有效性，本文对优化前后的系统性能进行了对比实验。实验中数据格式采用了《中国联通CDMA直放站综合监控管理协议规范1.0》的标准。首先采用原有系统结构，即一个监控中心通过串口连接20个短信模块来监控500个直放站，为简化起见，实验中并未使用大批直放站，而是采用了一台报警控制主机和500个短信模块，来模拟500台CDMA直放站。实验中，当报警主机驱动短信模块进行报警时，只要同时报警超过2个，监控中心就会出现延迟等待；同时报警超过10个，则系统发生拥塞，表现为死等或瘫痪。

然后对扩展优化后的系统进行实验，在实现了系统的软件功能后，搭建了用于测试的实验环境：监控中心端由一台扩展为10个串口的PC连接20个短信模块来实现；为简化起见，实验中并未使用大批直放站，而是采用了一台报警控制主机和500个短信模块，来模拟500台CDMA直放站。实验开始后，报警控制主机驱动短信模块模拟多个直放站同时报警的状况。当同时报警数为60以下，系统工作正常；60到180之间时，系统出现延迟现象；大于180时，则系统发生拥塞，表现为死等或瘫痪。由以上实验可见，优化扩展后的监控系统的性能得到了显著提高。

小结 本文基于排队论，对手机直放站监控系统的通信瓶颈和阻塞进行了理论分析和设计。通过对直放站监控计算机的PCI插槽进行扩展，设计并实现了一个高性能的iREPEAT监控系统，并通过实验对该系统的设计方法和性能进行了验证。结果证明，本文所采取的方法能有效地解决现有直放站技术中，管理规模与通信拥塞之间的矛盾。

参考文献

1 米永涛. CDMA直放站的原理及规划设计. 电信工程技术与标准化，2003(3)

2 孔让梨，王博. 现阶段CDMA网络中直放站的设计及应用. 电信技术，2004(6)

3 王亚丽. 直放站网管系统的现状与发展分析. 移动通信，2004，28(5)

4 邱相群. 移动通信直放站的应用分析. 移动通信，2003，27(2)

5 张柏生，任剑锋，孟相如. 基于排队论的网络通信系统的建模与分析. 空军工程大学学报(自然科学版)，2002，3(3)

(上接第51页)

4 Jain M, Dovrolis C. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. IEEE/ACM Transactions on Networking, 2003, 11(4): 537～549

5 Mathis M, Mahdavi J. Diagnosing Internet congestion with a transport layer performance tool. In: Proc. INET'96, Montreal, Canada, June 1996. 281～291

6 Shioda S, Yagi T, Mase K. A new approach to the bottleneck bandwidth measurement for an end-to-end network path. In: IEEE International Conference on Communications, Seoul, Korea, May 2005. 59～64

7 Lai K, Baker M. Nettimer: a tool for measuring bottleneck link bandwidth. In: Proc. of USENIX Symposium on Internet Technologies and Systems, March 2001. 1～12

8 Guojun J, Yang G, Crowley B R, et al. Network characterization service (NCS). In: Proc. of IEEE International Symposium on High Performance Distributed Computing (HPDC), San Francisco, CA, USA, August 2001. 289～299

9 Jacobson V. Pathchar - A tool to infer characteristics of Internet paths. http:// ftp. ee. lbl. gov/pathchar/

10 Mah B A. pchar: A tool for measuring internet path characteristics. http:// www. kitchenlab. org/www/bmah/software/pchar/