维普资讯 http://www.cqvip.com

使用EHN+机制提高移动网络中TCP性能

孙向辉　王玲芳　侯自强

（中科院声学所网络与数字信号处理研究中心　北京100080）

摘　要　TCP协议是一种在因特网上广泛使用的端到端可靠传输协议，移动网络中无线链路的高误码率(BER)和越区切换导致了TCP性能严重降低。在本文中，我们分析了切换过程中TCP遇到的问题，着重考察了TCP从高可用带宽链路切换至低可用带宽链路时遇到的问题，并根据EHN机制提出了EHN+机制，解决了TCP从高可用带宽链路切换至低可用带宽链路时出现的超时重传导致TCP在切换过程中性能下降的问题，提高了TCP在切换时的可靠性和性能，并给出了模拟结果。

关键词　移动网络，无线TCP(传输控制协议)，切换

Using EHN+ Mechanism to Improve TCP Performance in Wireless Networks

SUN Xiang-Hui　WANG Ling-Fang　HOU Zi-Qiang

(DSP and Network Research Center, Institute of Acoustics, CAS, Beijing 100080)

Abstract　TCP is a popular Internet protocol for reliable end-to-end data delivery. The wireless high BER link and handover across cell degrade the TCP performance in mobile networks. In this paper, we explore the problem about TCP connections handover from higher available bandwidth link to lower available bandwidth link. We present EHN + mechanism to improve TCP reliability and performance when TCP connections switch from higher available bandwidth link to lower available bandwidth link. Additionally, we execute simulations using NS and give the results.

Keywords　TCP over wireless, Handover, Wirdess networking

1　简介

随着因特网和无线技术的发展，移动网络将会在未来通讯中扮演重要的角色。3G的R5在接入网部分通过引入IP技术实现端到端的全IP化，3G的R6版本提到了WLAN-UMTS互通。未来的接入网将是多种接入方式结合的全IP接入网[1]，移动终端可以在不同的接入方式之间切换和漫游。

TCP是因特网上使用最为广泛的可靠传输协议，它通过接收端发送对收到报文的确认(ACK报文)来保证数据报文的可靠传输。当报文丢失时，发送端通过判断接收到的ACK报文或超时机制来得知报文丢失情况，并重传丢失的报文。

最初的TCP协议是针对有线链路设计的。慢启动、快速重传和快速恢复算法适用于处理有线链路中的拥塞丢包[2]。在无线链路中，由于链路的高误码率和越区切换，从而产生两种新类型的丢包，通常称为随机丢包(random loss)和切换丢包(handover loss)。无线链路中的随机丢包、切换丢包与有线链路中的拥塞丢包有很大的不同，使用解决拥塞丢包的算法来解决随机丢包和切换丢包会产生很多问题，导致TCP性能的严重降低[3]。

针对无线网络中的TCP性能优化问题，近年来许多研究人员提出了很多改进方案。改进的方向是区分有线的拥塞丢包和无线的随机丢包及切换丢包，并做出相应的处理。到现在为止有大量改进的TCP方案提出，这些方案主要可分为三类[4]：

1)端到端方案，包括Freeze-TCP，Veno，TCP-Westwood等；

2)代理方案，包括snoop，WTCP，NS等；

3)分立TCP方案，包括I-TCP，M-TCP等。

这三类方案各有其特点，在不同的条件下，不同的方案会表现出各自的优势和不足。

本文着重考察越区切换对TCP性能的影响，并根据EHN机制[5]提出改进的EHN+机制，提高TCP在越区切换时的可靠性和性能。

2　切换时TCP性能分析

这一节讨论切换过程中TCP遇到的问题。

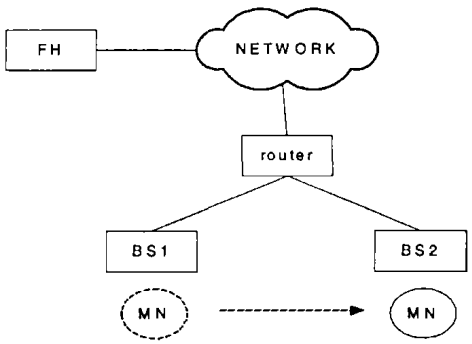

图1　IP接入网结构图

如图1是一个IP接入网络，MN通过无线链路与基站(BS)相连，BS1与BS2位于不同网段，FH通过BS和MN通信，移动管理协议包括MIP，Cellular IP，HAWIIA等。一般切换的时间为几百毫秒或者几秒，其中会丢失大量的报文。

我们假定FH为TCP发送端，MN为TCP接收端，采用MIP移动管理协议[6]。切换前，FH发向MN的报文通过BS1转发。MN从BS1向BS2移动，同时做运动检测，当MN检测

维普资讯 http://www.cqvip.com

来自 BS1的信号质量差到一定程度时，MN 与 BS1中断连接，这时如果来自 BS2的信号质量足够好，MN 开始向 BS2注册认证[7]，注册通过并由 MIP 完成切换功能后，FH 开始与 MN 通过 BS2继续传送报文。切换过程从 MN 与 BS1断开连接开始，到 FH 与 MN 通过 BS2传送报文为止。切换过程中有两个因素影响 TCP 的性能。

2.1 切换时延

切换的时间主要为二层切换时间和用户注册以及 MIP 机制时间，一般为1秒左右，有时甚至几秒。在 FH 端，当 TCP 发送的报文在一个 RTO(Retransmission TimeOut)之内未收到 ACK 时，TCP 进入超时重传阶段，根据 TCP 重传定时器的"指数退避"算法，下一个超时重传时间为两倍的 RTO，这样，即使在第一个重传定时器过期后，MN 立刻与 BS2建立连接，但由于这时 MN 并无 ACK 报文发向 FH(切换过程中 MN 不能收到来自 BS1的报文)，因此，TCP 发送端只能等待下一个重传定时器超时，导致建立好的 TCP 连接并无数据发送，严重降低 TCP 性能[8]。

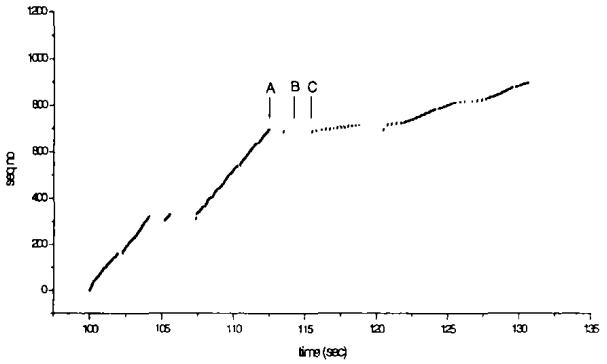

图2 传统 TCP 切换时序号-时间图

图2为使用 NS-2模拟结果图，具体模拟环境请见第5节。*A* 时刻表示 MN 与 BS1中断连接，这时 FH 和 MN 之间中断数据传输，*C* 时刻表示切换后 FH 和 MN 重新开始传输 TCP 报文，*B* 时刻是切换完成的时刻。*A* 时刻为112.464985s，*C* 时刻为115.464985s，*B* 时刻为114.396514s。当 TCP 连接在 *A* 时刻中断时，经过第一次超时，这时 MN 还没有和新 BS 建立连接，TCP 利用指数退避算法，下一次超时为2秒，即时刻 *C*，而在时刻 *B*，切换已经完成，MN 和 FH 之间已经可以传送 TCP 报文，但由于这时 MN 没有 ACK 报文发送至 FH，因此 FH 只有等到第二次超时重传定时器过时，即2秒后，才能重新发送报文。这样，时刻 *BC* 之间能够传送报文，却由于 TCP 的超时重传机制浪费掉了。而如果由于某种原因，切换时刻位于 *C* 以后，则要等待更长的时间。

2.2 切换时的发送报文序号

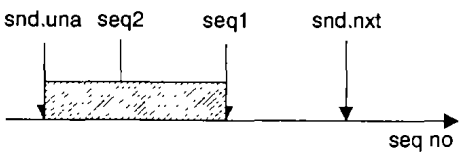

图3 切换过程中 TCP 发送端的一个状态

图3是切换过程中 TCP 发送端一个可能的状态。在切换前，FH 发送某个窗口的报文，snd.una 和 snd.nxt 的状态如图3。MN 收到窗口中部分报文，如图3中 snd.una 与 seq1之间的报文，这时 MN 与 BS1中断连接，随后所有报文都丢失，包括 MN 向 FH 发送的部分 ACK 报文，TCP 收到部分 ACK 报文，如确认 snd.una 与 seq2之间数据的报文，TCP 发送端的 snd.una 改变为 seq2，当切换完成后，TCP 发送端开始发送数据时，将从序号 seq2开始发送，seq2和 seq1之间的数据 MN 端已经存在，重传这些数据会浪费一定的网络资源，同时也可能会引起重传。

针对 TCP 在切换时遇到的问题，明确切换机制(EHN)提出了一个解决问题的方案。

3 明确切换机制[5]

明确切换通告机制(EHN)的原理是：移动终端保存最后发送的 ACK 报文的确认序号，当 TCP 的 *ssthresh* 和 *cwnd* 值改变时，TCP 发送端保存变化前的值。MN 切换至新的基站并通过注册后，立刻向 TCP 发送端发送一个带有 EHN 标志的 ACK 报文，这个 ACK 报文的确认序号为事先保存的确认序号。发送端通过收到这个 ACK 报文知道连接中的报文丢失或超时是由于移动终端在两个基站之间发生切换造成的，而不是出现了网络拥塞，因此 TCP 发送端不需要减小拥塞窗口，并立即以保存 cwnd 和 ssthresh 的值发送 TCP 报文。由此可见，EHN 通过使用 ACK 报文的明确切换标记，把由切换引起的丢包和其它原因的丢包区分开来，并进行相应处理。

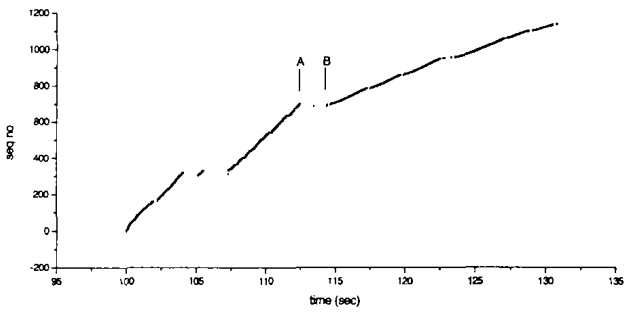

图4 使用 EHN 机制的 TCP 切换时序号-时间图

图4为采用了 EHN 机制的 TCP 的模拟图，它与图2的模拟条件完全相同。与图2中对比，*A* 时刻(112.464985s)MN 与 BS1中断连接，在 *B* 时刻(114.396514s)MN 与 BS2建立连接，采用 EHN 机制，可以看到在 *B* 时刻 TCP 立刻恢复传送数据，而无需如图2中等到定时器过时后的 *C* 时刻才传送数据，从而提高了 TCP 在切换时的性能。

EHN 机制属于前面提到的三种方案的第一种，它不需要在中间节点设置代理，保证了 TCP 的端到端的模型，但需要修改发送端和接收端的 TCP 代码。

全 IP 网络中，切换时延较长，容易引起 TCP 发送端定时器超时，出现第2节中讨论的情况。EHN 机制使得移动终端切换至新的 BS 下后，立刻通知 TCP 发送端发送报文，而无需等待重传定时器超时，并且保持原有的拥塞窗口和慢启动门限不变，从而避免了慢启动过程，提高了 TCP 在切换过程中的性能。

EHN 机制假设切换前后的链路带宽状态相同，因此切换后仍采用原来链路的拥塞窗口和慢启动门限，从而使 TCP 在切换后保持一定的报文发送速率。然而，实际情况中，切换前后的链路状态很少相同，而不同无线接入方式之间的链路状态更是有很大区别，比如 WLAN/UMTS 之间的垂直切换。从一个高可用带宽链路切换至低可用带宽链路时，如果仍然采用原来链路的拥塞窗口和慢启动门限，会导致 TCP 出现大量

维普资讯 http://www.cqvip.com

丢包，从而出现超时重传[11]，降低 TCP 的性能。针对这种情况，我们提出了 EHN＋机制，着重考虑了切换后 TCP 的拥塞窗口和慢启动门限的值对 TCP 性能的影响，并提出设定切换后的拥塞窗口和慢启动门限的方法。

4 EHN＋机制

EHN＋机制以 EHN 机制为基础，它与 EHN 机制不同的是考虑了移动终端切换至新链路后 TCP 的参数选择问题，并提出了切换至新链路后设置 TCP 参数的原则。由模拟结果可以看到，它避免了 EHN 机制切换至新链路时会出现的大量丢包而导致性能的降低，提高了 TCP 在切换过程中的可靠性及性能。

4.1 EHN＋的基本原理

当 MN 切换至新的链路后，由于 MN 完全不知道新链路的任何情况，我们把这样的阶段称作 MN 在新链路上的启动阶段。TCP 连接的慢启动门限、拥塞窗口及链路缓存大小对这个启动阶段的性能有着很大影响。以下先考察 EHN 机制在新链路启动阶段的情况。

根据切换前后的链路状态，切换可分为两种，一种是由高可用带宽链路切换至低可用带宽链路，另一种则为低可用带宽链路切换至高可用带宽链路，比如 WLAN/UMTS 之间的垂直切换。我们考虑第一种情况，由高可用带宽链路切换至低可用带宽链路，例如从高速的 WLAN 切换至低速的3G 网络的情况。在 EHN 机制中，当移动终端切换至新链路，TCP 发送端收到带有 EHN 标记的 ACK 报文后，将 snd. nxt 设为 snd. una，并以保存的变化前的 cwnd 和 ssthresh 的值在新链路上发送报文。这样，如果这个保存的 cwnd 和 ssthresh 的值超过了新链路的可用带宽，则很容易造成 TCP 在新链路启动阶段的大量报文丢失和重传定时器超时。文[8]中提到了另外一种切换完成后 MN 立刻通知 TCP 发送端的方法，它使用3个重复的 ACK 报文，根据拥塞控制算法，这导致 TCP 进入慢启动阶段(4. 3BSD-Tahoe)，然而这时的慢启动门限是由切换前链路状况决定的，而与切换后链路的状态无关。

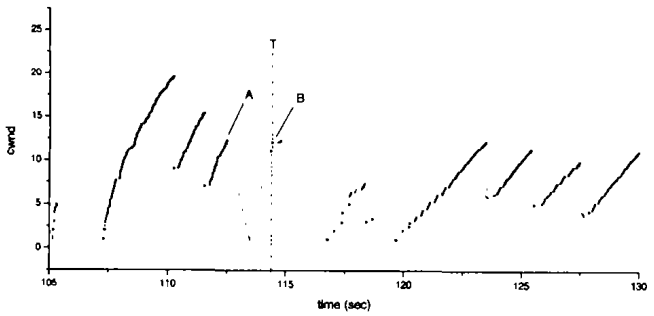

图5 切换前后链路可用带宽不同时使用 EHN 机制的 TCP 拥塞窗口变化图

图5为使用 EHN 机制从高可用带宽链路切换至低可用带宽链路过程的拥塞窗口图，模拟环境与图4中相同，图中 *T* (*B*)时刻 *TCP* 发送端收到 MN 发送的带有 EHN 标记的 ACK 报文。TCP 发送端保存的慢启动门限和拥塞窗口的值为 *A* 点处的值，cwnd 和 ssthresh 分别为12和7，则 TCP 发送端在 T 时刻，即在新链路启动阶段，使用参数(cwnd = 12, ssthresh = 7)发送 TCP 报文。而新链路的可用带宽小于切换前链路(通过调整 NS 中参数及加入背景流量实现)，由图5可看到，在 *B* 点使用切换前 *A* 点的参数发送 TCP 报文会导致 TCP 报文大量丢失，导致超时重传和随后的慢启动，降低了 TCP 在切换时的性能。

进一步可以通过改变切换至新链路后 TCP 使用的 cwnd 和 ssthresh 的值来考察不同的参数对 TCP 在新链路启动阶段性能的影响。

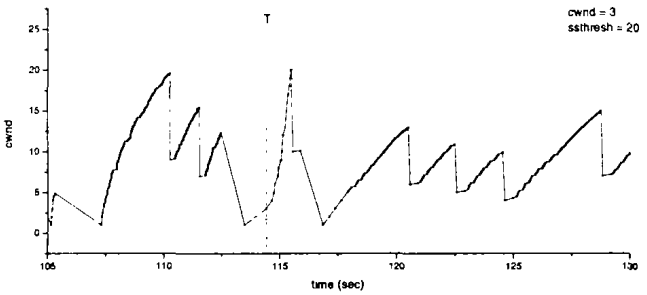

图6 (cwnd＝3，ssthresh＝20)切换过程拥塞窗口变化图

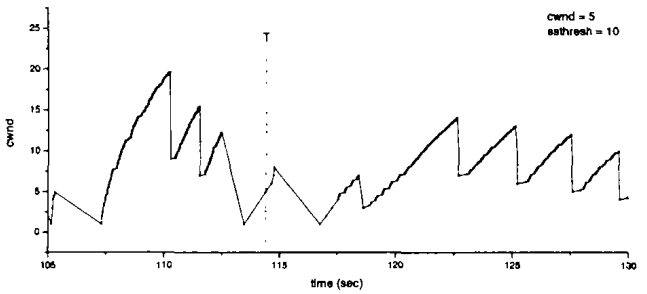

图7 (cwnd＝5，ssthresh＝10)切换过程拥塞窗口变化图

图6，图7为切换后采用不同参数时的拥塞窗口变化图，T 时刻表示切换完成，开始发送报文。图6为过大的起始 ssthresh 值引起报文大量丢失，而导致超时重传和慢启动的情况，这也可以在文[9]中看到；图7为起始 cwnd 设置过大而导致超时重传和慢启动的情况，这是由于链路带宽和缓存大小的限制，过快的向网络中注入流量，将导致缓存溢出，从而出现大量丢包[10]。由此可见，过高设置切换后的 cwnd 和 ssthresh 值，会引起报文的大量丢失，导致 TCP 进入超时状态，从而使 EHN 机制带来的优化性能降到最低，甚至不如传统未优化的 TCP。

因此，使 EHN 机制更好地发挥优化性能的关键是正确设置切换后 TCP 的拥塞窗口和慢启动门限的值，由此我们提出 EHN＋机制。EHN＋机制避免使用切换前链路 TCP 的慢启动门限和拥塞窗口的值，而是通过测量可用带宽机制，估计切换后链路的容量，并依据这个值设定切换后的 TCP 拥塞窗口和慢启动门限。

EHN＋机制建议切换后拥塞窗口和慢启动门限值设置如下：如果切换后 TCP 处于超时状态，则不改变 cwnd 大小，如果 TCP 切换后 TCP 没有进入超时状态，则将 cwnd 设为1；通过一定的机制快速测量新链路的可用带宽及链路延迟，由带宽和延迟计算链路的带宽延迟积，将 ssthresh 设为计算的带宽延迟积大小。这样，在切换后，从拥塞窗口为1开始慢启动阶段，到计算的链路容量开始进入拥塞避免阶段。

EHN＋设置拥塞窗口和慢启动门限的原理为：当移动终端切换至新链路时，TCP 发送端根本不知道新链路的状况，比如由 WLAN 切换至 UMTS 链路，或者在 WLAN 水平切换中，由一个相对空闲的链路切换至一个较为繁忙的链路。在这两种情况中，由于切换前链路可用带宽高，因此 TCP 的拥塞窗口较大，而以这个值在切换后的低可用带宽链路上发送报

维普资讯 http://www.cqvip.com

文，必然会导致大量报文丢失，而使 TCP 进入超时重传状态，由图5也可看到这个情况。因此，在不知道新链路状况的情况下，将起始拥塞窗口设为1，进入慢启动状态，逐步探测网络容量，是较为可靠的选择。慢启动门限是拥塞窗口从指数增长进入加性增长的阈值，过高的慢启动门限使拥塞窗口以指数增长的时间过长，特别当拥塞窗口很大时，指数增长的乘2关系导致下个拥塞窗口增加过快，这样容易超过链路的容量而导致报文的大量丢失，引起超时；而过低的慢启动门限使得 TCP 过早的进入拥塞窗口加性增长的拥塞避免阶段，这个阶段每个 RTT 内拥塞窗口增加1，尽管这样避免了报文丢失，但过慢的增加拥塞窗口使得 TCP 发送时间加长，影响了 TCP 的性能[9]。EHN＋机制选用测量到的新链路的带宽延迟积作为慢启动门限的大小。带宽延迟积（Bandwidth-Delay-Product，BDP）为链路的容量大小[12]，使用带宽延迟积作为切换后启动阶段的慢启动门限大小，使得 TCP 的拥塞窗口以指数增长快速到达链路的可用容量，然后以加性增长慢速地探测实际可用的容量，这样既可以避免使用较大慢启动门限值造成的大量丢包，也可以避免较小慢启动门限导致的发送时间加长，使切换更加平滑，这在后面的结果中可以看到。

一个链路上 TCP 窗口的最大值为瓶颈链路的容量（带宽延迟积）加上链路的缓存大小，在广域网（WAN）中，特别是在 RTT 很大时，链路的缓存远小于带宽延迟积[13]。我们使用带宽延迟积作为慢启动门限的大小。

EHN＋机制实现的关键是测量切换后链路的可用带宽和延时，下面介绍测量的方法。

4.2 切换后链路可用带宽和延时的测量

链路的延时可以使用 TCP 的时间戳选项获得，MN 向 TCP 发送端发送带有时间戳选项的 ACK 报文，这样，当 TCP 发送端收到 ACK 报文时，提取其中的时间戳，与接收时间相减，则可以获得链路的延时。

链路可用带宽的测量有很多种方法，EHN＋使用 Packet-Pair 方法[14]计算链路可用带宽。具体做法为：当 MN 切换至新的 BS 下时，立刻向 TCP 发送端发送2个完全相同的带有 EHN 标记的 ACK 报文，TCP 发送端根据收到的这2个连续 ACK 报文计算新链路的可用带宽。具体实现如下：

1）MN 连续向 TCP 发送端（FH）发送2个 ACK 报文，这些 ACK 报文中包含发送的时刻（时间戳选项），假设 TCP 发送端在 $t1$，$t2$两个时刻收到这2个报文；

2）发送端通过收到的报文中的时间戳选项计算链路的时延，假设为 $a1$，$a2$；

3）发送端在取得上述数据后丢弃前一个 ACK 报文，只有一个 ACK 报文进入以后的处理。

链路的可用带宽为：

$$B=\frac{s}{t2-t1}(\text{bytes/s})$$

链路的单向延迟为：

$$T=\frac{a1+a2}{2}(\text{s})$$

链路的带宽延迟积为：

$$BDP=2\times B\times T(\text{bytes})$$

其中 s 为 ACK 报文的大小，单位为字节。最后将 BDP 转换成 TCP 发送报文的大小，即 BDP 除以 TCP 发送的以字节为单位的最大报文段大小。

4.3 EHN＋机制模拟结果

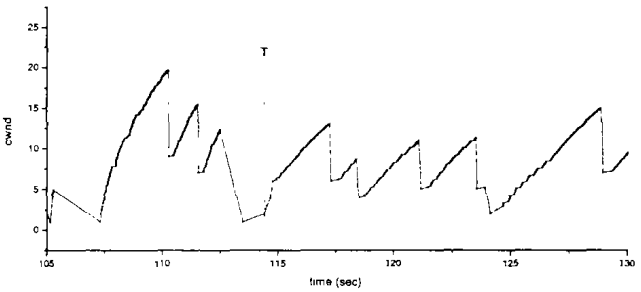

图8 TCP EHN＋切换过程拥塞窗口变化图

图8为采用 EHN＋机制的 TCP 切换时拥塞窗口的变化图。与图6和图7相比，当移动终端从高可用带宽链路切换至低可用带宽链路时，使用 EHN＋机制避免了切换后 TCP 出现的超时重传，保证了 TCP 的平滑切换。

EHN＋机制使 TCP 连接在切换到新链路后使用小的拥塞窗口并将慢启动门限设为链路可用容量大小，这样虽然在切换后的 TCP 的发送速率降低，但是它避免了切换后 TCP 使用较大窗口和慢启动门限发送数据时出现的大量报文丢失而导致的超时重传，提高了 TCP 在切换至新链路后的可靠性，从而提高了 TCP 在切换过程中的可靠性和性能。

5 网络模拟拓扑图

本文中使用的模拟工具为 NS-2网络模拟器[15]，拓扑图如图9。

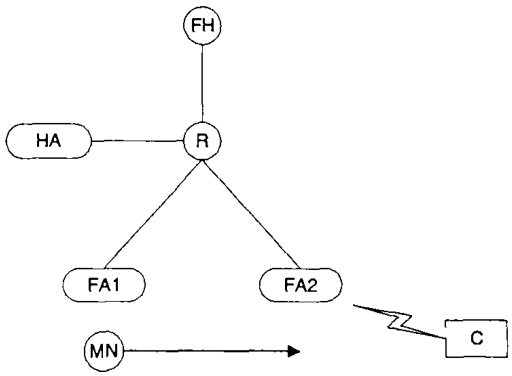

图9 NS 模拟拓扑图

拓扑图中的移动 IP 使用了 CMU 的 MIP 模块，无线路由采用 NOAH，FH 与 MN 之间使用 TCP EHN＋机制，MN 从 FA1向 FA2移动。为模拟新链路可用带宽情况，在 FA2下放置一固定节点 C，C 通过无线链路与 FA2相连，FH 与 C 之间建立 TCP 流，模拟背景流量。HA 为节点 MN 和 C 的家乡代理。

结论 本文在开始提到了 TCP 在无线链路上遇到的困难及已经提出的几种解决方案。然后详细分析了切换过程中影响 TCP 性能的两个因素。随后分析了 EHN 机制及其会产生的问题，最后提出了 EHN＋机制。

EHN＋机制在 EHN 基础上考虑了切换后 TCP 参数的选择问题，由于拥塞窗口和慢启动门限对 TCP 影响很大，因此采用合适的参数能避免切换后由于参数选择不当造成的超时重传，从而进一步提高 TCP 在切换时的性能。

参 考 文 献

1 侯自强，冯登国. 网络技术. 化学工业出版社，2002

2 Stevens W. TCP Slow Start，Congestion Avoidance，Fast Retransmit，and Fast Recovery Algorithms. IETF Jan. 1997，RFC2001

（下转第42页）

维普资讯 http://www.cqvip.com

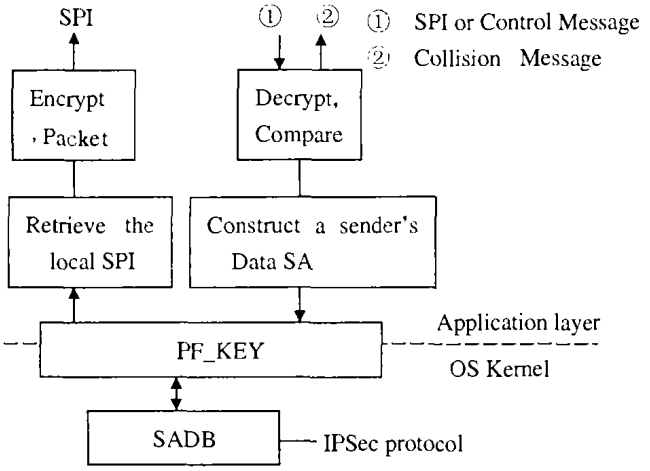

图3 用户 IPSec 主机上的 GDOI 扩展模块示意图

限于已有的条件，我们仅仅在用户方基于 Linux 主机利用手动方式对 SPI 的传送、Data SA 的构造、与 IPSec 内核的 SADB 之间的通信(包括 SPI 的检索以及 Data SA 的添加、更新、删除等)进行了模拟实验。PF_KEY[8]是一个新的 Socket 协议簇，用于实现应用层的密钥管理进程 IKE 与 IPSec 内核之间的通信。让 GDOI 的扩展模块利用这个 Socket 来发送和接收消息，实现了与 IPSec 内核的 SADB 之间的交互(如 SPI 的检索及 Data SA 的添加、更新、删除等)，(过程略)，从而验证了 Data SA 的可管理性。

通过在 SADB 中动态地存储不同 SPI 值的 Data SA，实现了对来自不同数据源的包含不同 SPI 值的数据包的处理。由于接收者的 SADB 中存储着每个当前发送者的 Data SA，所以可以基于 SPI 值及用户标识实现源认证和基于 IPSec 数据报头中的序列号实现重放保护。

(2) GCKS 执行的操作

1)一个新加入的成员在注册时，GCKS 也必须将当前发送成员的 SPI 值一道发送给该成员，使其 SADB 中能够存储当前发送者的 Data SA。

2)在 GDOI 的扩展协议中，GCKS 也同样作为一个接收者，但不接收和处理应用数据，只负责对发送者 SPI 值的安全验证和在发生异常情况时发出控制信息。在收到一个发送成员的 SPI 时，GCKS 查阅成员列表进行验证，若一个不良的组成员发送了一个伪造的 SPI，GCKS 将发出一个放弃接收这个 SPI 值的通知。由于 GCKS 只在发生 SPI 的伪造等异常情况时，才发出控制信息，因而不会增加通信的开销。

4.2 方案的安全性

(1)源认证：由于每个成员拥有独立的 SPI 及用户标识，因而接收者可以根据接收数据包的 SPI 值识别组内的明确发送者。

(2)应用数据的重放保护：由于接收者的 SADB 中存储着每个当前发送者的 Data SA，每个 Data SA 包含独立的 SPI 值，接收到的不同 SPI 值的数据包将会根据相应的 Data SA 来进行处理，因而保证了接收端的每个发送者数据包序列号的单调性和连续性，使系统可以基于 IPSec 数据报头中的序列号实现重放保护。

(3)SPI 的重放保护：在实现上保证不重复接收相同的 SPI 值。

(4)防伪造、截获和冒充：非多播组成员由于不知道组的通信密钥(TEK)，因而无法截获、伪造或冒充一个发送成员的 SPI 及应用数据；通过在 GCKS 方增加对发送者 SPI 值的验证，可防止组内不良成员的伪造；通过在用户主机增加对接收的 SPI 值的比较，可防止组内不良成员的冒充。

结束语 IPSec 是一个具有高度安全特性的数据安全协议，通过应用不同的密钥管理协议(如 IKE 或 GDOI)，可使已有的 IPSec 协议既适用于单播又可用于多播。为了使基于 GDOI 协议的 IPSec 多播进一步适用于多源的情况，我们对 GDOI 协议进行了扩展，增加了其对 Data SA 的管理功能，通过在用户方的 IPSec 主机上的模拟实验，验证了在多源情况下的 Data SA 的可管理性。分析和实验结果说明，通过对现有的 GDOI 组播密钥管理协议的扩展，可解决多源多播情况下的源认证和重放保护等问题。

本文的研究仅仅是解决多源多播安全问题的一个初步探索，实际上，多源多播是一个十分复杂的研究课题，要真正实现基于 IPSec 的多源多播，还有许多问题有待于进一步的研究和探索。

参 考 文 献

1 Kent S, Atkinsen R. Security Architecture for the Internet Protocol. RFC2401, 1998. 11
2 Harkins D, Carrel D. The Internet Key Exchange (IKE). RFC2409, 1998. 11
3 Baugher M, Hardjono T, Harney H. The Group Domain of Interpretation. Internet draft draft-ietf-msec-gdoi-07. txt, Dec. 2002
4 Baugher M, Canetti R, Dondeti L. Group Key Management Architecture. Internet draft draft-ietf-msec-gkmarch-03. txt, Fe-b. 2002
5 Hardjono T, Harney H. Group Security Association (GSA) Management in IP Multicast. NETWORKING 2002. 1123～1128
6 Wallner D M, Harder E J, Agee R C. Key management for multicast : Issues and architectures. RFC 2627, June 1999
7 Doraswamy D, Harkins N D. IPsec --新一代因特网安全标准. 北京：机械工业出版社，2000
8 McDonald D, Metz C, Phan B. PF_KEY Key Management API, Version 2. RFC2367, 1998. 7

(上接第25页)

3 Xylomenos G, Polyzos G C. TCP performance issues over wireless links. IEEE Communications Magazine, April, 2001. 52～58
4 Shah A. TCP Performance over Wireless Links. EE359-Wireless Communications, Autumn, 2001
5 Izumikawa H, et al. An Efficient TCP with Explicit Handover Notification in Wireless Nerworks. The Institute of Elexctronics, Technical Report of IEICE, Informantion and Communication Engineers, 2003
6 Perkins C, Ed. IP Mobility Support. IETF, RFC 2002, Oct. 1996
7 Xie Tiebing, Zhang Yu, Gao Jiming, Hou Ziqiang. A generic way for wireline and wireless access authentication. In: Intl. Conf. on communication Technology (ICCT 2003), IEEE Conf. 2003
8 Caceres R, Iftode L. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. IEEE Journal on Selected Areas in Communications, 1995, 13(5)
9 Hoe J C. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. in ACM SIGCOMM, Aug. 1996
10 Barakat C, Altman E. Impact of Buffer Size on TCP Start-Up. In: Proc. of IEEE Middle East Workshop on Networking, Beirut, Lebanon, Nov. 1999
11 Fall K, Floyd S. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. ACM Computer Communications Review, 1996, 26(3): 5～21
12 范建华，等，译. TCP/IP 详解 卷一：协议. 机械工业出版社，2000
13 Lakshman T V, Madhow U. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. IEEE/ACM Transactions on Networking, June 1997
14 Keshav S. A control-theoretic approach to flow control. In: Proc. of the ACM SIGCOMM '91, Sep. 1991. 3～15
15 The Network Simulator-ns2. http://www.isi.edu/nsnam/ns