MPTCP与CMT-SCTP多路径传输协议性能分析

符 发[1],周 星[1],杨 雄[1],Hakim Adhari[2],Erwin P.Rathgeb[2]
FU Fa[1], ZHOU Xing[1], YANG Xiong[1], Hakim Adhari[2], Erwin P.Rathgeb[2]

1.海南大学 信息科学技术学院,海口 570228
2.杜伊斯堡-埃森大学 实验数学学院,德国 埃森 45326
1.College of Information Science and Technology, Hainan University, Haikou 570228, China
2.Institute for Experimental Mathematics, University of Duisburg-Essen, Essen 45326, Germany

FU Fa, ZHOU Xing, YANG Xiong, et al. Performance analysis of MPTCP and CMT-SCTP multi-path transport protocols. Computer Engineering and Applications, 2013, 49(21):79-82.

Abstract: Achieving multipath transport by terminal's multi-homed characteristics is a hot topic of Internet protocol application. MPTCP and CMT-SCTP are both more mature and key protocols for multipath transmission, but there is still a lot of performance analysis needed before large-scale application. This paper presents the behaviors of MPTCP and CMT-SCTP multipath transmission and compares the difference between both protocols on the local test-bed environment. Results show that both MPTCP and CMT-SCTP protocols can significantly increase throughput. Meanwhile the scheduling strategy for transmission is still not perfect and needs further improvement.

Key words: multipath transmission; load-sharing; path management; congestion control; buffer; performance analysis

摘 要: 利用终端的多宿主特性实现多路径传输是因特网协议的一个热点研究问题,MPTCP和CMT-SCTP是目前比较成熟而关键的两个多路径传输协议,但在大规模应用前还需要作大量的性能测试分析研究。在本地测试床环境中,对MPTCP和CMT-SCTP多路径传输实际吞吐量的性能进行对比测试与分析。结果表明MPTCP和CMT-SCTP都能获得吞吐量的提高,但传输调度算法仍然不完善,需要进一步改进。

关键词: 多路径传输;负载共享;路径管理;拥塞控制;缓存;性能分析
文献标志码: A **中图分类号:** TP393 doi:10.3778/j.issn.1002-8331.1306-0218

1 概述

随着多种接入技术的发展,近年来智能移动终端越来越普及,它们往往都拥有多个网络接口(如,WLAN和3G/4G等),甚至连个人电脑也都配备有多个网卡以通过多个接口同时接入到不同的网络供应商。这种具有多个网络接口的多宿主特性使设备具有了更好的可移动性、快速恢复能力、安全性和负载共享功能。负载共享是其中一个很重要的特性,它能够聚合不同链路的带宽,使得设备能获得更大的网络吞吐量。负载共享功能可以在网络协议栈中的多个层上实现[1],但本文研究其在传输层上的实现方法,这也是当前IETF国际组织关注的热点问题。目前已经有众多的研究者提出了在传输层实现的多种方法,如有:R-MTP[1]、Parallel TCP[2]、mTCP[3]、MPTCP(Multipath TCP)[4-5]和CMT-SCTP(Concurrent Multipath Transfer for SCTP[6])[7-8],其中MPTCP和CMT-SCTP是讨论最多的、且比较成熟的两种方法,目前二者已被IETF组织采纳,并成为热点研究的课题。

MPTCP和CMT-SCTP都是端到端实现负载共享的多路径传输协议。A.Ford和C.Raiciu等人在文献[5]说明了将TCP协议扩展为MPTCP以实现多路径传输的概念和设计思想。P.D.Amer和M.Becke等人在文献[8]中描述了利用SCTP的多宿主特性实现多路径传输CMT-SCTP的设计思想和方法。目前对这两种多路径传输协议的研究比较成熟,但在大规模部署之前仍然有不少关键问题需要解决,比如拥塞控制机制、缓存空间大小的配置和路径管理等问题。本文对这两种多路径传输协议吞吐量的性能进行测试和分析。

本文首先介绍了多路径传输在互联网应用中的需求,概述了MPTCP和CMT-SCTP多路径传输协议的概念及相关研究工作;然后说明测试环境的构建与配置;接着对测试

基金项目: 国家自然科学基金(No.61163014);海南省创新引进集成专项科技合作项目(No.KJHZ2013-20)。
作者简介: 符发(1978—),男,实验师,研究方向:计算机网络与互联网协议;周星,通讯作者,教授;杨雄,教授;Hakim Adhari,博士研究生;Erwin P.Rathgeb,博士,教授。E-mail:zhouxing@hainu.edu.cn
收稿日期: 2013-06-20 **修回日期:** 2013-08-10 **文章编号:** 1002-8331(2013)21-0079-04

结果进行分析与讨论,最后总结并提出下一步工作的设想。

2　MPTCP和CMT-SCTP多路径传输协议

MPTCP(Multipath TCP)是传统TCP协议的扩展,即在传输层上使用多条路径实现同时进行数据传输的协议。MPTCP使用TCP作为子流传输,多个路径就意味着有多个TCP子流,其协议系统模型如图1所示[4]。而CMT-SCTP(Concurrent Multipath Transfer for SCTP)则是利用SCTP所特有的多宿主特性(如图2所示),从而使用不同的路径同时进行数据传输[9]。这两个协议从理论上都能充分利用终端的多条路径同时进行数据传输,以实现网络吞吐量的最大化。下面分别介绍这两个多路径传输协议的路径管理、拥塞控制和缓存空间管理的工作机制。

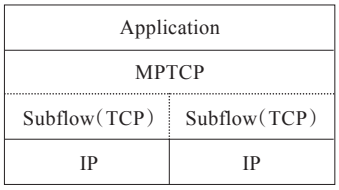

图1　MPTCP协议子流模型

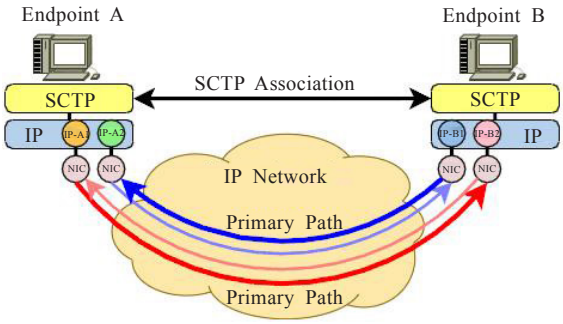

图2　SCTP的Multi-Homing特征

2.1　路径管理机制

路径管理是影响多路径传输性能的一个关键因素,它决定了通信双方如何去探测和利用两个节点之间所存在的路径去传输数据,以获得更高的吞吐量性能。

(1)CMT-SCTP路径管理:CMT-SCTP是基于SCTP[6]实现的多路径传输协议。SCTP分组由首部和多个数据块(Chunks)组成。两个主机的SCTP协议通过4次握手完成连接,称为偶联。在CMT-SCTP中,主机 H_A 首先会通知对方主机 H_B 它自己所使用的IP地址列表,当主机 P_B 收到建立偶联的INIT(Initiation Chunk)块后,就会给 P_A 响应INIT-ACK(Initiation Acknowledgment)消息进行应答。对于图3,主机 H_A 与 H_B 都有两个接口,如 H_A 拥有 IP_{A1} 到 IP_{A2} 两个IP地址,H_B 拥有 IP_{B1} 和 IP_{B2} 两个IP地址。当 H_A 和 H_B 开始建立初始连接时,H_A 首先通过 IP_{A1} 发送INIT消息到 H_B 主机的 IP_{B1} 地址建立偶联,在SCTP中这条先建立起来的偶联会被选做主路径,即 P_{A1--B1} 用来传输数据,其他的路径作为备份路径,而在CMT-SCTP中则会同时使用所有的不相交路径来传输数据,H_A 和 H_B 完成建立连接后,将拥有如图3所示的 P_{A1--B1} 和 P_{A2--B2} 两条路径,并在以后的数据传输中将由 P_{A1--B1} 和 P_{A2--B2} 这两条传输子流(sub-flows)路径共同承载 H_A 和 H_B 两主机之间的通信数据流负载。

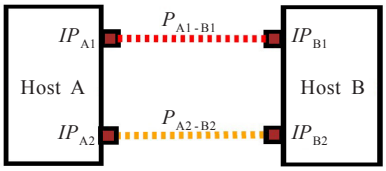

图3　CMT-SCTP连接子路径

(2)MPTCP路径管理:与CMT-SCTP协议不同,MPTCP协议会使用所有可用的路径共同进行数据传输,从而达到使网络吞吐量最大化的目的。每个MPTCP终端都维护着由各接口IP地址组成的一个IP地址列表,MPTCP终端利用源主机IP地址列表和目的主机IP地址列表中的IP地址,分别组合多个源 IP_S 到目的 IP_D 的地址对,构建出相应的TCP子流,将应用程序数据进行分段并将各个分段通过不同的TCP子流同时进行传输。如图4所示,两个MPTCP终端 H_A 和 H_B 以正常的TCP建立连接,在建立连接过程中通过使用MP_CAPABLE TCP选项通知对方启用多路径传输,当该连接建立完成后,通过使用MP_JOIN选项可将其他连接子流加入到该连接中来共同承载数据的传输。最终 H_A 和 H_B 将拥有 P_{A1--B1},P_{A1--B2},P_{A2--B1} 和 P_{A2--B2} 四条路径同时进行数据传输。当IP地址出现变化时,通过使用ADD_ADDR和REMOVE_ADDR消息来通知对方对相应的连接子流进行调整。

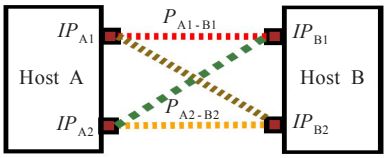

图4　MPTCP连接子路径

从图3和图4可以看出,相同结构的两个双接入主机系统使用MPTCP协议可以有四条路径被建立并进行数据传输,而CMT-SCTP则只会使用两条路径传输数据。

2.2　拥塞控制机制

在多路径传输中,要找到一个合适的拥塞控制是比较困难的,使用现有的TCP拥塞控制机制不能满足在多路径传输中网络资源分配的公平合理[10]。理论上多路径传输拥塞控制要求能够将数据从拥塞的路径上转移到非拥塞的路径上进行传输,以减轻拥塞路径上的负担,降低数据丢失率从而提高整个网络的稳定性。一个良好的拥塞控制机制应能遵循以下三个原则[11]:

(1)提高吞吐量:一个多路径传输的实际吞吐量不能低于单个最优路径传输的最大吞吐量。

(2)不能影响其他用户:多路径传输中任何一个子流

不能占用超过使用该路径进行单路径传输时所占用的带宽,保证其他用户也能够公平合理地获得网络资源。

(3)平衡拥塞:尽可能地将数据从拥塞的路径上转移到其他空闲的路径。

为了解决这个问题在文献[12]中提出了在CMT-SCTP中使用RP(Resource Pooling)[13]机制的拥塞控制方法。新的MPTCP多路径传输拥塞控制在文献[14]提出每个子流使用TCP标准拥塞控制机制。在MPTCP连接上使用加法递增的拥塞避免算法及TCP的退避方法(如慢开始,快重传,快恢复和乘法减少等)。该策略对子流i的拥塞控制窗口$cwnd_i$增加值由如下的公式(1)确定[14]。

$$\min\left(\frac{\alpha\times bytes\_acked\times mss_i}{\sum_i cwnd_i},\frac{bytes\_acked\times mss_i}{cwnd_i}\right) \tag{1}$$

其中$cwnd_i$和mss_i分别是第i条路径子流的拥塞窗口和最大分段长度;$\sum_i cwnd_i$是所有子流的拥塞窗口的和;$bytes_acked$是子流i确认字节数;α参数由公式(2)进行计算。其中rtt_i是子流i的往返时间。

$$\alpha=\left(\sum_i cwnd_i\right)\times\frac{\max\limits_i\dfrac{cwnd_i}{rtt_i^2}}{\left(\sum\limits_i\dfrac{cwnd_i}{rtt_i}\right)^2} \tag{2}$$

2.3　缓存管理机制

在多路径传输中,需要通过MPTCP与CMT-SCTP连接级缓存将来自不同的子流的数据分段进行按序重组,然后提交给应用层进行处理。因此,必须设置足够大的缓存来存放所收到的数据,以等待失序或者丢失的分段到达。所以,缓存的大小是影响多路径传输性能重要的系统参数。在多路径传输中$P=\{P_1,P_2,\cdots,P_n\}$,各子路径的带宽和往返时间分别为$Band_i$和rtt_i,则合理的最小的Buffer可用如下公式表达:

$$B_{\min}=2\times\left[\max_{1\leqslant i\leqslant n}\{rtt_i\}\times\sum_{i=1}^{n}Band_i\right]$$

当出现拥塞或重传时,最差情况下则需要三倍的最大RTT(第一次传输,快速重传,定时重传)加上最大RTO的缓存时间,因此需要的最小缓存空间为:

$$B_{\min}^{T}=(3\times\max_{1\leqslant i\leqslant n}\{RTT_i\}+\max_{1\leqslant i\leqslant n}\{RTO_i\})\times\sum_{i=1}^{n}Band_i$$

对于多路径传输,使用越多的缓存性能就会越好,但由于系统存储空间资源的有限性,应尽可能地保持发送和接收双方缓存平衡。一种最基本的解决方法就是根据带宽和延时的乘积来配置缓存的大小,其他的缓存管理方法,像PSC自动TCP缓存调节[15]与DRS(Dynamic Right-Sizing)[16]则通过收集链路延时信息来调整缓存空间。另一个常用的方法是在linux内核中使用的autoruning,它是简单地根据socket缓存空间和系统内存来调整缓存大小的方法,当数据填满时就会自动增加缓存。

3　测试床构建与参数配置

为了测试和分析多路径传输协议的性能,构建了本地局域网的测试床,其测试床的架构如图5所示。在图5中配置了两台安装Ubuntu和FreeBSD的双网卡主机,在Ubuntu系统中安装最新的MPTCP内核[17],在FreeBSD中安装CMT-SCTP内核。配置Dummynet对路由控制,以模拟不同的链路带宽,构建相似与非相似链路组成的多路径传输测试环境。

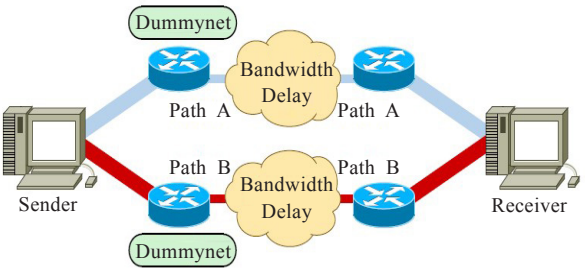

图5　本地局域网测试床

CMT-SCTP和MPTCP都使用了文献[11]所提出的拥塞控制机制,测试时使用所有的路径传输数据,发送方尽可能多地发送数据;所有的数据都按序传输。本地测试床中,路由器的RED排队按照文献[18]的建议进行配置,两条路径的带宽都设置为70 Mb/s,发送和接收缓存大小都在实验中进行调整。单次测试时间为300 s,间隔时间为20 s。每个测试都至少重复10次,取测试的平均值作为最终测试结果以获取足够的统计精度。

4　性能测试分析

在本次测试中,在测试床首先进行了由相似带宽链路组成的多路径传输的吞吐量性能测试,然后测试由非相似带宽链路组成的多路径传输的吞吐量性能,观察和分析了发送与接收端缓存大小与吞吐量性能的关系,并对MPTCP和CMT-SCTP与单路径TCP和SCTP传输吞吐量进行了比较。

4.1　相似带宽链路的多路径传输

本项目测试了在相似链路组成的并行多路径传输环境中的吞吐量,并与单路径TCP和SCTP传输的网络吞吐量进行了比较。各链路带宽都设置为70 Mb/s,延迟为0 ms,执行了10轮次测试,使用文献[11]所提出的拥塞控制机制,测试结果如图6所示。从图6可见,曲线#1与曲线#3分别是使用MPTCP与CMT-SCTP协议多路径传输时主机所获得的实际吞吐量;曲线#2与曲线#4则分别是对应单一路径的TCP与SCTP协议传输所获得的实际吞吐量。在这种网络环境下,在两端buffer量增加的初期,多路径与单路径传输的吞吐量在快速放大,当buffer达到35 KB时两者的吞吐量开始趋于平稳,达到饱和。其中曲线#2、#4达到了67 Mb/s;而曲线#1、#3获得了两倍于曲线#2、#4的吞吐量。毫无疑问,使用MPTCP和CMT-SCTP多路径传输主机的实际网

络吞吐量获得巨大的提高。

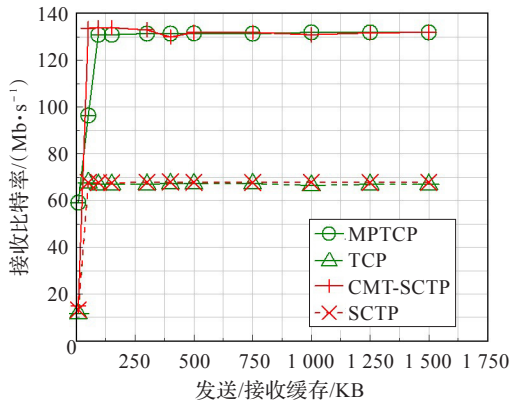

图6　相似链路组成的多路径传输吞吐量

4.2　非相似带宽链路的多路径传输

在这个测试当中,模拟了常见的由非相似带宽链路组成的多种接入主机的场景,类似移动计算机具有有线连接的以太网,同时又有WLAN或者3G/4G接口的多种接入标准的网络。为了模拟这种场景,在图5中配置Dummynet[19]机制以进行带宽控制,路由器RED队列根据文献[18]配置,将上面的链路带宽设置为100 Mb/s,下面的链路带宽设置为10 Mb/s,延迟都为0 ms,其测试结果如图7所示。

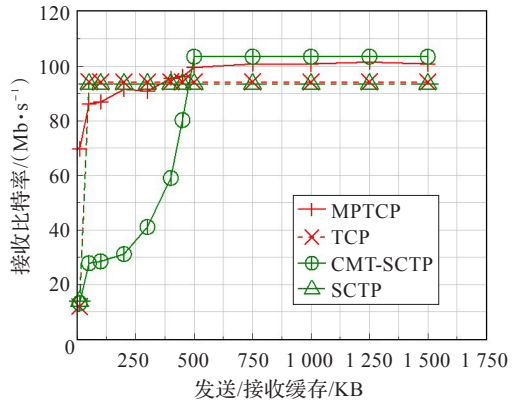

图7　非相似链路组成的多路径传输吞吐量

从图中可见曲线#2与曲线#4分别是使用SCTP与TCP在100 Mb/s链路上进行传输获得的实际吞吐量;而曲线#1和曲线#3分别为对应CMT-SCTP和MPTCP多路径传输时所获得的吞吐量数据。结果显示在缓存小于500 KB的情况下,使用MPTCP多路径传输所获得的吞吐量比使用最优单路径的TCP传输所获得的吞吐量还要小,这违背了文献[12]所提出的第一个原则(一个多路径传输的吞吐量不能低于单个最优路径传输的最大吞吐量),而CMT-SCTP曲线#1表现出来的性能则更差。随着缓存大于500 KB后,可以看到MPTCP和CMT-SCTP能够获得比单路径传输更高的吞吐量,这也证明了在使用多路径进行传输时需要更多的缓存空间才能获得好的性能表现,同时也说明其数据传输调度算法仍需改进。

5　结束语

MPTCP和CMT-SCTP是多路径传输协议中两个比较成熟,而且关键的解决方案。从测试场景中可以看到,使用这两个多路径传输协议都可以获得比单路径传输(TCP和SCTP)更大的网络吞吐量。但缓存空间的占用仍然是问题,比如,相对于单路径传输,多路径传输需要使用巨大的缓存空间,这对于终端主机性能是一个严峻的挑战。从测试结果来看,在非相似链路中多路径传输的性能受到缓存空间的影响较大,还需要对其数据传输调度算法进行更深入的分析研究,改进其工作机制。下一步将在基于Internet的测试床上开展更广泛的测试,并对路径管理和拥塞控制机制进行分析研究。

参考文献:

[1] Magalhaes L,Kravets R.Transport level mechanisms for bandwidth aggregation on mobile hosts[C]//The 9th IEEE International Conference on Network Protocols(ICNP).Riverside,California,U.S.A.:IEEE,2001.

[2] Hsieh H Y,Sivakumar R.pTCP:an end-to-end transport layer protocol for striped connections[C]//The 10th IEEE International Conference on Network Protocols(ICNP).Paris,France:IEEE,2002.

[3] Zhang M,Lai J,Krishnamurthy A,et al.A transport layer approach for improving end-to-end performance and robustness using redundant paths[C]//The USENIX Annual Technical Conference.Boston,Massachusetts,U.S.A.:IEEE,2004.

[4] Ford A,Raiciu C,Handley M,et al.RFC 6182 Architectural guidelines for multipath TCP development[S].IETF International,2011.

[5] Ford A,Raiciu C,Handley M,et al.RFC6824 TCP extensions for multipath operation with multiple addresses[S].IETF,2012.

[6] Stewart R R.RFC 4960 Stream control transmission protocol[S].IETF,Standards Track,2007.

[7] Iyengar J R,Amer P D,Stewart R.Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths[J].IEEE/ACM Transactions on Networking,2006,14(5).

[8] Amer P D,Becke M,Dreibholz T,et al.Load sharing for the Stream Control Transmission Protocol(SCTP)[S].2012.

[9] Dreibholz T.Evaluation and optimisation of multi-path transport using the stream control transmission protocol[D].Essen,Germany:University of Duisburg-Essen,2012.

[10] Dreibholz T,Becke M,Adhari H,et al.On the impact of congestion control for concurrent multipath transfer on the transport layer[C]//The 11th IEEE International Conference on Telecommunications(ConTEL).Graz,Steiermark,Austria:IEEE,2011.

[11] Raiciu C,Wischik D,Handley M.Practical congestion control for multipath transport protocols[R].University College London,London,United Kingdom,2009.

(下转105页)

随机选择的两个随机数,进而获得密文中所包含的明文比特。最后通过计算实验攻击,验证了本文格攻击方法的正确性和有效性。因此,本文破解了文献[7]中较快速的全同态加密方案。

参考文献:

[1] Rivest R,Adleman L,Dertouzos M.On data banks and privacy homomorphisms[C]//Foundations of Secure Computation,1978:169-180.

[2] Gentry C.Fully homomorphic encryption using ideal lattices[C]//STOC 2009,2009:169-178.

[3] Smart N P,Vercauteren F.Fully homomorphic encryption with relatively small key and ciphertext sizes[C]//LNCS 6056:Public Key Cryptography-PKC 2010,2010:420-443.

[4] van Dijk M,Gentry C,Halevi S,et al.Fully homomorphic encryption over the integers[C]//LNCS 6110:Eurocrypt 2010,2010:24-43.

[5] Stehle D,Steinfeld R.Faster fully homomorphic encryption[C]//LNCS 6477:Asiacrypt 2010,2010:377-394.

[6] Gentry C,Halevi S.Implementing Gentry's fully-homomorphic encryption scheme[C]//LNCS 6632:Eurocrypt 2011,2011:129-148.

[7] 汤殿华,祝世雄,曹云飞.一个较快速的整数上的全同态加密方案[J].计算机工程与应用,2012,48(28):117-122.

[8] Lenstra H W,Lenstra A K,Lovasz L.Factoring polynomials with rational coefficients[J].Mathematische Annalen,1982,261:515-534.

[9] Nguyen P Q,Vallée B.The LLL algorithm:survey and applications[M]//Information Security and Cryptography.[S.l.]:Springer,2009:33-35.

[10] Shoup V.NTL:a library for doing number theory[EB/OL].[2011-11-20].http://shoup.net/ntl/.

(上接82页)

[12] Dreibholz T,Becke M,Pulinthanath J,et al.Applying TCP-friendly congestion control to concurrent multipath transfer[C]//The 24th IEEE International Conference on Advanced Information Networking and Applications(AINA).Perth,Western Australia,Australia:IEEE,2010.

[13] Wischik D,Handley M,Braun M B.The resource pooling principle[J].ACM SIGCOMM Computer Communication Review,2008,38(5).

[14] Raiciu C,Handley M,Wischik D.RFC6356 Coupled congestion control for multipath transport protocols[S].IETF,Internet Draft,2012.

[15] Semke J,Mahdavi J,Mathis M.Automatic TCP buffer tuning[J].Computer Communication Review,1998,28.

[16] Fisk M,Feng Chun.Dynamic adjustment of TCP window sizes,Los Alamos Unclassified Report(LAUR)00-3221[R].Los Alamos National Laboratory,2000.

[17] Institute of Information and Electronics Communication Technologies and Applied Mathematics(ICTEAM).MultiPath TCP-Linux kernel implementation[EB/OL].[2013-04-10].http://multipath-tcp.org/.

[18] Floyd S.RED:discussions of setting parameters[EB/OL].[2013-04-10].http://www.icir.org/floyd/REDparameters.txt.

[19] Carbone M,Rizzo L.Dummynet revisited[R].Dipartimento di Ingegneria dell'Informazione,Universit`a di Pisa,Pisa,Italy,2009.

(上接89页)

卡出厂时固定的唯一PIN。因此改进受到了硬件的极大限制,很难实现每发起一次连接认证就使用一个不同的PIN。因此WPS应慎重使用。

5 结论

随着WLAN在我国的日趋普及,采用Wi-Fi WPS PIN模式进行认证,其安全性备受关注。针对WPS通过CPN对其分析指出其存在的缺陷并给出了相应的改进策略,分析表明WPS应慎重使用。

参考文献:

[1] Wi-Fi Alliance.Wi-Fi protected setup specification version 1.0[S].2007.

[2] Aboba B,Blunk L,Vollbrecht J,et al.RFC 3748 Extensible Authentication Protocol(EAP)[S/OL].[2011-11-10].http://tools.ietf.org/html/rfc3748.

[3] Nyberg K.Connect now to MitM[EB/OL].[2011-11-10].http://www.tcs.hut.fi/Publications/knyberg/crypto06_rump.pdf.

[4] Jensen K.Colored Petri nets:basic concepts,analysis methods and practical use,volume 1[M]//Monographs in Theoretical Computer Science.[S.l.]:Springer-Verlag,1992.

[5] 王文化,沈庆国,韩春永,等.基于染色Petri网的BGP连接过程模型[J].计算机工程,2011,37(6):82-84.

[6] 孙涛,叶新铭,刘靖,等.一种基于CPN的协议测试序列生成方法[J].解放军理工大学学报:自然科学版,2012,13(2):165-170.

[7] 朱俊,郭长国,吴泉源.一种基于CPN的运行时监控服务交互行为的方法[J].计算机研究与发展,2011,48(12):2277-2289.

[8] Aly S,Mustafa K.Protocol verification and analysis using colored Petri nets[EB/OL].[2011-11-10].http://facweb.cs.depaul.edu/research/techreports/tr04-003.pdf.