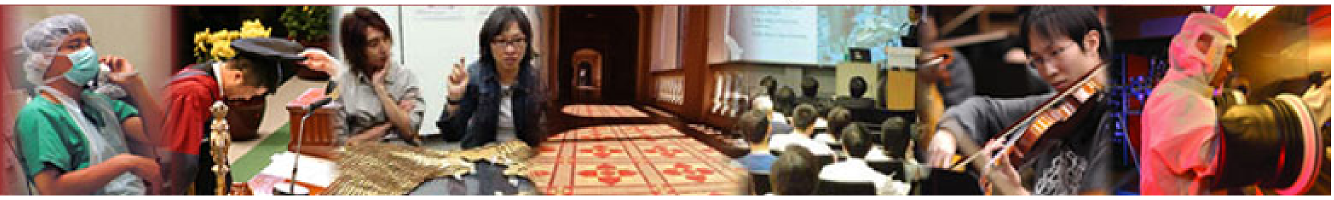


Title	A testbed for TCP in heterogeneous wired/wireless networks
Author(s)	Lai, C; Leung, KC; Li, VOK
Citation	The 29th IEEE International Conference on Computer Communications (IEEE INFOCOM 2010), San Diego, CA., 14-19 March 2010.
Issued Date	2010
URL	http://hdl.handle.net/10722/126162
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Demo Abstract: A Testbed for TCP in Heterogeneous Wired/Wireless Networks

Chengdi Lai, Ka-Cheong Leung, and Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong, China
E-mail: {laichengdi, kcleung, vli}@eee.hku.hk

***Abstract*— Transmission Control Protocol (TCP) design in heterogeneous wired/wireless networks is challenging due to packet reordering and error-prone channels. We have implemented a testbed for evaluating the performance of TCP variants under packet reordering, congestive loss, and non-congestive loss. Our demonstration shows the flexibility of our testbed to accommodate channels of different characteristics. This facilitates the experimental study of TCP over heterogeneous networks. It also demonstrates the feature of TCP-NCL as a unified solution for effective congestion control, sequencing control, and loss recovery in heterogeneous networks.**

I. INTRODUCTION

The next generation network is expected to be heterogeneous, including both wired and wireless components. Transmission Control Protocol (TCP) design is challenging due to packet reordering (say, introduced by link-layer retransmission) and error-prone channels. It is thus highly desirable to perform experimental study of TCP variants under packet reordering and non-congestive loss. However, existing experimental testbeds for TCP, such as WAN-in-Lab [2] and PLANETLAB [4], do not consider packet reordering, congestive loss, and non-congestive loss simultaneously.

We have set up a heterogeneous network testbed for studying the performance of TCP variants with the presence of congestive loss and either non-congestive loss or packet reordering or both. The intensities of packet reordering and random loss can be adjusted by varying some control parameters of the testbed, thereby mimicking various channel conditions.

The rest of the paper is organized as follows. Section II describes our testbed design. Section III presents the testing framework. Finally, we discuss our demonstration plan in Section IV.

II. TESTBED DESIGN

Our testbed consists of two major components, namely, the heterogeneous network and the supervision software, as illustrated in Fig. 1(a) [3]. The network consists of a number of EZ270 wireless nodes, desktop and laptop computers, servers, wired and wireless gateways, switches, and so on.

Each EZ270 node has multiple wireless interfaces and a wired Ethernet interface, as illustrated in Fig. 1(b) [6]. The G2M5477 module from G2 Microsystems is selected as the IEEE 802.11 radio interface. EZ270 is also equipped with XScale PXA270 as the micro-controller unit (MCU) with RAM, ROM, an SD card, and USB ports. The operating system of EZ270 is embedded Linux.

The supervision software consists of a socket service and a web server. The former is connected to the network through Ethernet to perform test data collection and network control operations. The latter serves as an interface to the end user. Users can access the web server to perform network configuration and test data analysis.

III. TESTING FRAMEWORK

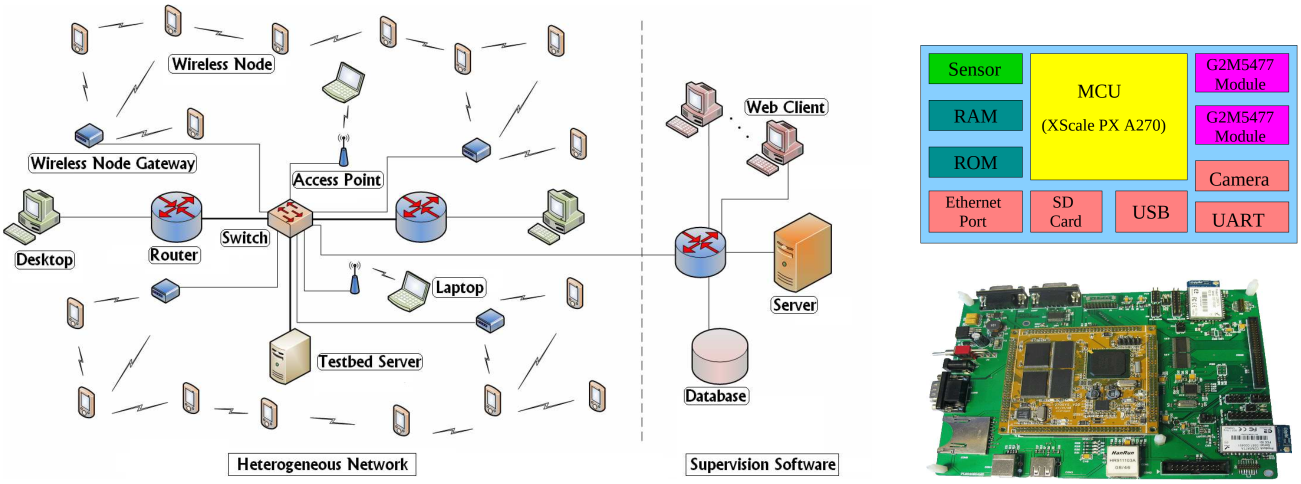
We design the test cases on the testbed by selecting various combinations of TCP variants and channel conditions. Section III-A presents a generic framework of deploying TCP variants. Section III-B describes the emulating software that helps to imitate various channel conditions.

A. TCP Variants

The difficulty in modifying the kernel hinders the implementation of most TCP variants proposed. To facilitate the implementation, we devise a generic framework for developing TCP variants in the Linux/UNIX kernel. We observe that a majority of TCP variants differ in their congestion control and loss recovery operations, which are realized cooperatively by three TCP major functions[1], namely, the TCP input function *tcp_input*, the TCP output function *tcp_output*, and the retransmission timer function *tcp_timer_rexmt*, as illustrated in Fig. 2. Thus, a TCP variant can be deployed by: 1) defining its data structures in the TCP control block, 2) maintaining the data structures in the three aforementioned functions, and 3) modifying/replacing the fast retransmit, fast recovery, and/or retransmission timeout algorithms.

In [1], we have devised TCP-NCL, a sender-side TCP variant as a unified solution for effective congestion control, sequencing control, and loss recovery over reordering, error-prone channels. High-precision timestamps are stored for packets transmitted in *tcp_output*, updated in *tcp_input* based on the incoming acknowledgements, and properly reset in *tcp_timer_rxmt*. The fast retransmit and fast recovery algorithms are replaced in *tcp_input* by our serialized timer structure.

[1] The architecture follows that of BSD UNIX [5]. Despite the possible difference in the naming conventions and function calling sequence, most Linux/UNIX kernel adopt similar structures in their TCP/IP stacks.



(a) System architecture of the testbed. (b) EZ270 wireless node.

Fig. 1. Our heterogeneous wired/wireless network testbed.

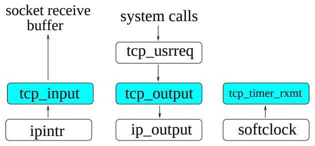


Fig. 2. Software architecture for TCP.

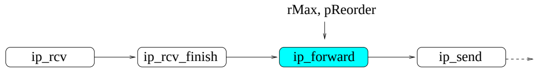


Fig. 3. IP processing path of the Linux kernel.

B. Emulating software

We modify the Internet Protocol (IP) implementation in the kernel of relays so that packet reordering can be introduced with a configurable intensity. Fig. 3 illustrates the processing of an IP packet in a Linux relay. *ip_rcv* takes an IP packet from a network interface and pass it to *ip_rcv_finish* after the validity check. The latter decides the next forwarding hop and invokes *ip_forward*, which updates the control information of the packet and calls *ip_send* for sending it. Packet reordering can be introduced by modifying *ip_forward* such that it postpones the invocation of *ip_send* based on the parameters controlling the intensity of packet reordering.

The packet loss rate over a wireless link can be tuned by adjusting the power level of a wireless interface (the G2M5477 module), or by varying the distance between two connected EZ270 nodes.

IV. DEMONSTRATION

The objective of our demonstration is two fold. First, we demonstrate the flexibility of the testbed in imitating various channel conditions. Second, we compare the performance of TCP variants, such as TCP-NewReno and TCP-NCL, under different channel conditions.

The demonstration setup is as shown in Fig. 4. Two EZ270 nodes and a laptop computer will serve as a source, a destination and a relay, respectively. The link between the source and the relay is connected via Ethernet. A File Transfer Protocol (FTP) connection will be set up between the source and the destination. The measured connection goodput will be uploaded from the destination to a monitoring terminal and plotted in real time. TCP-NewReno or TCP-NCL will run at the source by enabling different socket options. We compare the performance of TCP-NewReno and TCP-NCL under three sets of test cases, as shown below:

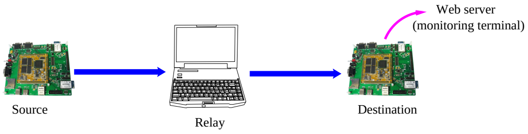


Fig. 4. Demonstration setup.

1. In-order, error-prone channel: The link between the relay and the destination is wireless. No packet is reordered in transit.
2. Reordering channel of negligible transmission error: The link between the relay and the destination is connected via Ethernet. We introduce packet reordering of various intensities at the relay.
3. Reordering, error-prone channel: The link between the relay and the destination is wireless. We introduce packet reordering of various intensities at the relay.

The demonstration will show how the performance of TCP-NewReno is affected by packet reordering, congestive loss, and non-congestive loss. The effectiveness of TCP-NCL as a unified solution for congestion control, sequencing control, and loss control will also be demonstrated.

REFERENCES

[1] C. Lai, K.-C. Leung, and V.O.K. Li. Enhancing Wireless TCP: A Serialized-Timer Approach. *Proc. of IEEE INFOCOM 2010*, Mar. 2010.
[2] G.S. Lee, L.L.H. Andrew, A. Tang, and S.H. Low. WAN-in-Lab: Motivation, Deployment and Experiments. *Proc. of PFLDnet*, pp. 85-90, Feb. 2007.
[3] V.O.K. Li, L. Cui, Q. Liu, G.-H. Yang, Z. Zhao, and K.-C. Leung. A Heterogeneous Peer-to-Peer Network Testbed. *Proc. of ICUFN*, pp. 46-51, Jun. 2009.
[4] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. *ACM SIGCOMM Comp. Comm. Rev.*, Vol. 33, No. 1, pp. 59-64, Jan. 2003.
[5] G. Wright and W. Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Weasley, 1995.
[6] Z. Zhao, Q. Liu, G.-H. Yang, V.O.K. Li, and L. Cui. A Multi-radio Wireless P2P Network Testbed. *Proc. of the ACM WiNTECH*, pp. 93-94, Sep. 2009.