Chapter 3.13
Extended Time Machine Design using Reconfigurable Computing for Efficient Recording and Retrieval of Gigabit Network Traffic

S. Sajan Kumar
Amrita Vishwa Vidyapeetham, India

M. Hari Krishna Prasad
Amrita Vishwa Vidyapeetham, India

Suresh Raju Pilli
Amrita Vishwa Vidyapeetham, India

ABSTRACT

Till date there are no systems which promise to efficiently store and retrieve high volume network traffic. Like Time Machine, this efficiently records and retrieves high volume network traffic. The bottleneck of such systems has been to capture packets at such a high speed without dropping and to write a large amount of data to a disk quicklt and sufficiently, without impact on the integrity of the captured data (Ref. Cooke.E., Myrick.A., Rusek.D., & Jahanian.F(2006)). Certain hardware and software parts of the operating system (like drivers, input/output interfaces) cannot cope with such a high volume of data from a network, which may cause loss of data. Based on such experiences the authors have come up with a redesigned implementation of the system which have specialized capture hardware with its own Application Programming Interface for overcoming loss of data and improving efficiency in recording mechanisms.

DOI: 10.4018/978-1-61350-456-7.ch3.13

1. INTRODUCTION

While investigating security incidents or trouble shooting performance problems, network packet traces especially those with full payload content can prove invaluable. Yet in many operational environments, wholesale recording and retention of entire data streams is infeasible. Even keeping small subsets for extended time periods has grown increasingly difficult due to ever-increasing traffic volumes. Passive monitoring (Deri, L. & Netikos S.P.A2003)) involves tapping the link on which data needs to be collected, and recording to disk either complete packets, or just packet headers and timestamps indicating their arrival time. Packet switching (Ref. Switching, Circuit switching vs. Packet switching) technology provides flexible and easy management in current Internet routing system comparing with circuit-switch (Ref. Switching, Circuit switching vs. Packet switching) technology. However, packet loss is still a major issue that hurdles high-speed network utilization and performance, and affects quality of real time network services. At transport layer, transmission protocols use packet loss as congestion signal to prevent further packet loss. Where in network layer, routers delay (queue) and drop packets to overcome congestion or to ensure high priority packets passing through the router as fast as possible. Dropping packets seems a necessary entity in current network infrastructure. GPU based snort implementation, Gnort and GPU based software Router (Ref. Han,S., Jang.K., Moon.S. & Park,K. (2010)) also having bottlenecks at packet capturing (Ref. Vasiliadis,G., Antonatos,S., & Polychronakis, M. (2008)). A better solution to control transmission rate and to ensure high-priority services on networks is to determine what is the available network bandwidth, and sends packets for applications at or below the available bandwidth (Ref. Aurrecoechea, C., Campbell, A.T., & Hauw,L. (1998)). The recording system consists of the following components (Figure 1). Generally the packet capture module receives network packets from the interface to the operating system instead in the system we designed we have a specialized capture hardware that delivers the packets to the classification module (Ref. Malomsoky. S., Molnar.S., Veeres. A. & Szabo G (2010)). Classification allows for different treatment of packets as they are associated with a certain class of packets according to defined properties. The storage containers handle the incoming packets and store them in memory, on disk and evict them later, according to the storage policy defined by their respective class. Indexing provides mechanisms for quick access to subsets of packets that are stored anywhere in the system. The query module accepts queries for stored data, retrieves it from the storage containers and returns it. Finally there is a module that communicates with its environment to offer the functionality of configuration and queries; this is the user interface to the system.

2. TRAFFIC CAPTURE

In general, a capture system has to run on a computer with a network interface to the network from which data has to be cached. Such a computer system is called a packet capture system or sniffer. The sniffer's network interface card (NIC) that is connected to the network to observe is put in a special operation mode by the operating system, such that all packets that are observed by that NIC are accessible by the packet capture application (i.e. recording system) (Dreger, H., Feldmann, A., Paxson, V., & Sommer, R. (2004), Kornexl,S. (2005)). The NIC's connection to the measurement network is called tap. The sniffer's operating system provides a high-level interface to the packet capture mechanisms in the operating system's kernel. On UNIX systems, on which we developed and tested implementations of this design, the libpcap library is such an application programming interface (API) for use by packet capture applications. Of course, there are other