A Network Processor Based Passive Measurement Node

Ramaswamy Ramaswamy, Ning Weng and Tilman Wolf

Department of Electrical and Computer Engineering
University of Massachusetts Amherst, MA 01003
{rramaswa,nweng,wolf}@ecs.umass.edu

1 Introduction

The complexity of network systems and the heterogeneity of end systems will make networks increasingly difficult to manage. To understand the operational details of networks it is imperative that sufficient information on their behavior is available. This can be achieved through network measurement.

Passive network measurement systems typically collect packet traces that are then stored in trace databases. To extract information on the state of the network, the traces are searched and post-processed. In our work, we envision two extensions to this approach:

- **Distributed Measurement Nodes.** To provide a richer set of network management applications and traffic profiling capabilities, traffic is collected and correlated from multiple measurement nodes.
- **Preprocessing of Trace Data.** Scalability in distributed measurement is a key problem. The aggregate bandwidth of trace data from multiple measurement nodes can easily overwhelm a conventional database system. To alleviate this problem, we preprocess packet traces on the measurement node and perform simple statistics collection online.

The basic architecture of our measurement system is shown in Figure 1. In this paper, we discuss how to implement the packet capture and online preprocessing functions of this system on a network processor. Network processors are software programmable system-on-a-chip multiprocessors that are optimized for high bandwidth I/O and highly parallel processing of packets. We use the Intel IXP 2400 [1] network processor for our proposed measurement system. The IXP 2400 contains eight multi-threaded microengines for packet processing along with an XScale core processor to perform control plane related functions.

The measurement node performs three functions:

- **Packet Capture and Header Parsing:** Each packet is parsed to determine the sequence of headers that are present. This allows the consideration of nested protocol headers as well as different header sizes due to options.
- **Anonymization:** To ensure the privacy of network users, IP addresses are anonymized online on the network processor during trace collection.

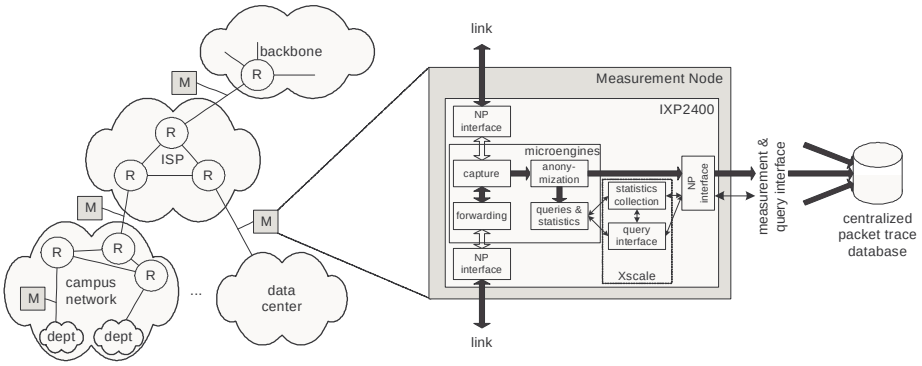

Fig. 1. Network Measurement Architecture.

– **Online Queries and Statistics Collection:** Packet traces can be pre-processed on the measurement node to reduce the load on the centralized collection system. If traffic statistics match a particular query, a response is pushed from the measurement node.

2 Related Work

Traditionally, two approaches have been taken towards network measurement: active and passive.In the active approach (e.g., NLANR's AMP and Surveyor), a sender and/or receiver measure and record the traffic that they send/receive, obtaining end-to-end (e.g., path) characteristics.

In the passive approach, measurements are taken at a given point in a network and are typically used to characterize local properties of the network and its traffic. Traces of packets passing through a passive measurement point can be analyzed for traffic mix (e.g., protocol or application), or flow size and burstiness. The passive measurement projects that are most closely related to our proposed efforts here are Sprint's IPMON project [2], AT&T's Gigascope project [3] and NLANR's passive measurement efforts [4].

None of these efforts, however, leverage the use of network processors, which allow customized online queries. In this context the use of network processors is particularly crucial as complex centralized post-processing and storage of traffic traces can be off-loaded into the measurement node.

3 IP Address Anonymization

To ensure that no private information is revealed in a network trace, the IP source and destination addresses need to be anonymized. The main constraint on the anonymization algorithm is that it should be "prefix-preserving." Thus, some information on network-level characteristics of the measured traffic can be preserved across the anonymization step.

It is desirable to perform anonymization as early in the collection process as possible. By anonymizing header fields on the measurement node itself instead

of external post-processing, it is less likely that unanonymized data is leaked. This requires the anonymization process to operate at a speed that can keep up with the link rates of the measurement node. This sort of *online* anonymization, however, cannot be achieved with current prefix-preserving anonymization algorithms ([5] and [6]), since they are computationally intensive.

We have developed a novel prefix-preserving anonymization algorithm, called TSA (top-hash subtree-replicated anonymization) [7], that addresses this problem by computing all necessary cryptographic functions offline. An IP address is anonymized by making a small number of accesses to a set of lookup tables in memory.

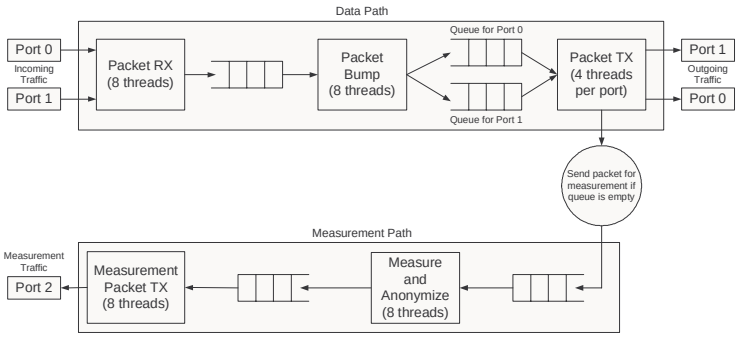

Fig. 2. Network Measurement Node on IXP2400.

4 Measurement Node Prototype

The prototype implementation of the proposed measurement system is based on the IXP2400 network processor platform. The data flow and allocation of tasks to the underlying NP components is shown in Figure 2. The data path bumps incoming traffic from Port 0 to Port 1 and vice versa. Once a packet has successfully proceeded through the data path, it is enqued to the measurement part of the system. If this queue is full, the packet is dropped and no measurement tasks are performed on it. Thus, the measurement path is designed to have a minimal impact on the performance of the network processor in the data path. In the measurement path, packet headers are collected, IP addresses are anonymized, statistics are updated, and a "measurement packet" (which contains the packet headers and some meta data) is generated for each packet observed, and transmitted from Port 2.

The measurement system was simulated on the simulator for the IXP 2400 network processor. Simulation traffic consisted of unidirectional 60 byte TCP/IP packets over Ethernet. We were able to sustain a transmit rate of up to 1120Mbps ($\sim$900,000 packets per second) on the measurement port (Port 2). The measurement node was also tested on the Radisys ENP-2611 board [8], on a network access link of the University of Massachusetts. The node was observed to be functional at data rates of up to 140,000 packets per second.

5 Future Work

We are exploring an extension to the current measurement prototype that allows collection of online statistics and the implementation of queries to the measurement node. The key research question is what traffic statistics to collect and how this information can be accessed through the query interface. This issue is closely related to the capabilities of the underlying hardware.

We propose to implement simple counting functions on the microengines and leave more complex processing to the Xscale control processor. In particular, we consider collection of packet counts, layer 3 and 4 protocol distributions, counts of packets with special significance (e.g., TCP SYN packets). These statistics can be further extended to gather more information through (1) per-flow statistics, (2) window-based statistics, and (3) multi-resolution counters. In all three cases there is a tradeoff between memory requirements and accuracy.

For the query interface we consider two possible types of queries. Queries that "pull" information from the measurement node are comparable to those done on conventional packet trace collections. The query is sent to the system and the appropriate information is retrieved and sent in response. With the online operation of our system another type of query is possible. "Push" queries are such that they continuously monitor the packet stream. When a particular condition is matched, a response is triggered.

Finally, it is necessary to obtain accurate time information for timestamping. We are currently in the process of integrating a GPS receiver with the IXP2400 to operate an NTP-style clock synchronization mechanism on the Xscale control processor.

References

1. Intel Corp.: Intel Second Generation Network Processor. (2002) http://www.intel.com/design/network/products/npfamily/ixp2400.htm.
2. Fraleigh, C., Diot, C., Lyles, B., Moon, S.B., Owezarski, P., Papagiannaki, D., Tobagi, F.A.: Design and deployment of a passive monitoring infrastructure. In: Passive and Active Measurement Workshop, Amsterdam, Netherlands (2001)
3. Cranor, C., Gao, Y., Johnson, T., Shkapenyuk, V., Spatscheck, O.: Gigascope: High performance network monitoring with an SQL interface. In: Proc. of the 2002 ACM SIGMOD, Madison, WI (2002) 623
4. McGregor, T., Braun, H.W., Brown, J.: The NLANR network analysis infrastructure. IEEE Communications Magazine **38** (2000) 122–128
5. Minshall, G.: TCPDPRIV, (http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html)
6. Xu, J., Fan, J., Ammar, M.H., Moon, S.B.: Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In: Proc. of 10th IEEE ICNP, Paris, France (2002) 280–289
7. Ramaswamy, R., Wolf, T.: High-speed prefix-preserving IP address anonymization for passive measurement systems. (under submission)
8. Radisys Corporation: ENP-2611 Product Data Sheet. (2004) http://www.radisys.com.