Introduction
A number of home-installed middleboxes, e.g., BISMark [1] and SamKnows [2], and web-based tools have been developed recently to enable residential broadband users to gauge their network service quality. One challenge to designing these systems is to provide stable network measurement. That is, the measurement results will not be fluctuated by sporadic overheads incurred inside the middlebox or web browser. In this poster, we propose a network measurement ware, OMware, to increase the stability of residential broadband measurement. The key feature is to implement the send and receive functions for measurement packets in the kernel. Our preliminary evaluation for an OpenWrt implementation shows that OMware provides very stable throughput and delay measurement, compared with typical socket-based measurement at the user level.

Implementation
Figure 1 shows the overall architecture of OMware. The OMware module is a loadable kernel module, which runs in the background and handles the required packet operations passed via the application interface. Through a set of APIs, OMware provides the services of dispatching measurement packets received from a measurement program according to the specified delay, and delivering to the measurement program measurement packets received from the network. These generic functions are often used in active measurement tools. The probe packet’s content and parameters are passed to the OMware module via a netlink socket.

Rule Management - It maintains a rule table to decide which application a packet belongs to.
Packet Sniffing - It captures the timestamp of the sending and receiving packet at the network driver.
Packet Transmission - It can buffer probe packets to be sent in the future and register a kernel task for sending the packets directly to the network card’s sending buffer.

Evaluations and Demo
Sending throughput measurement: The measurement application generates a packet train of 100 packets with zero time gap between two consecutive packets with packet sizes ranging from 54 bytes to 1454 bytes with a 10-byte increment. We compute the duration required for sending the entire packet train. Fig. 2 shows that both methods obtain the same throughput measurement in the absence of cross traffic for all packet sizes. However, in the presence of the cross traffic, the OMware-based measurement obtains a considerably higher throughput, especially when the packet size is greater than 800 bytes.

In this demo, we show the deployment of OneProbe running on top of the OMware in an OpenWrt router to support residential network measurement. We provide a side-by-side comparison of the measurement results to commodity PCs.

References

* Our Research Group - http://oneprobe.org