

MAPI kernel level implementation evaluation

Konstantinos Xinidis

June 5, 2004

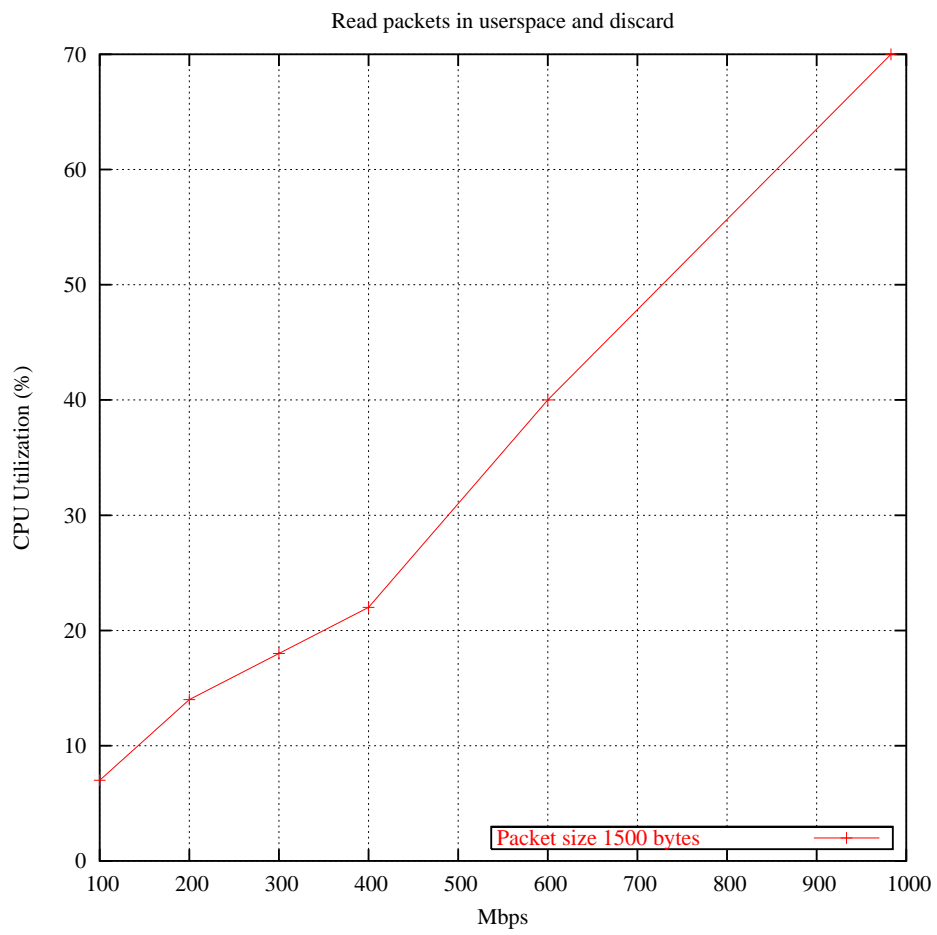
1 Testing infrastructure

For the experiments described in this document we used two machines. The sender was an IBM PC with a Pentium 4 Xeon 2GHz processor and 512Mbytes RAM. The receiver was a Dell PC with a Pentium 4 Xeon 2GHz processor and 512Mbytes RAM. The network card of the receiver was a Dual Intel Pro 1000 MP Server gigabit card. The two machines ran Redhat Linux 9. The driver of the gigabit card was tuned for network monitoring (not modified).

2 Evaluation

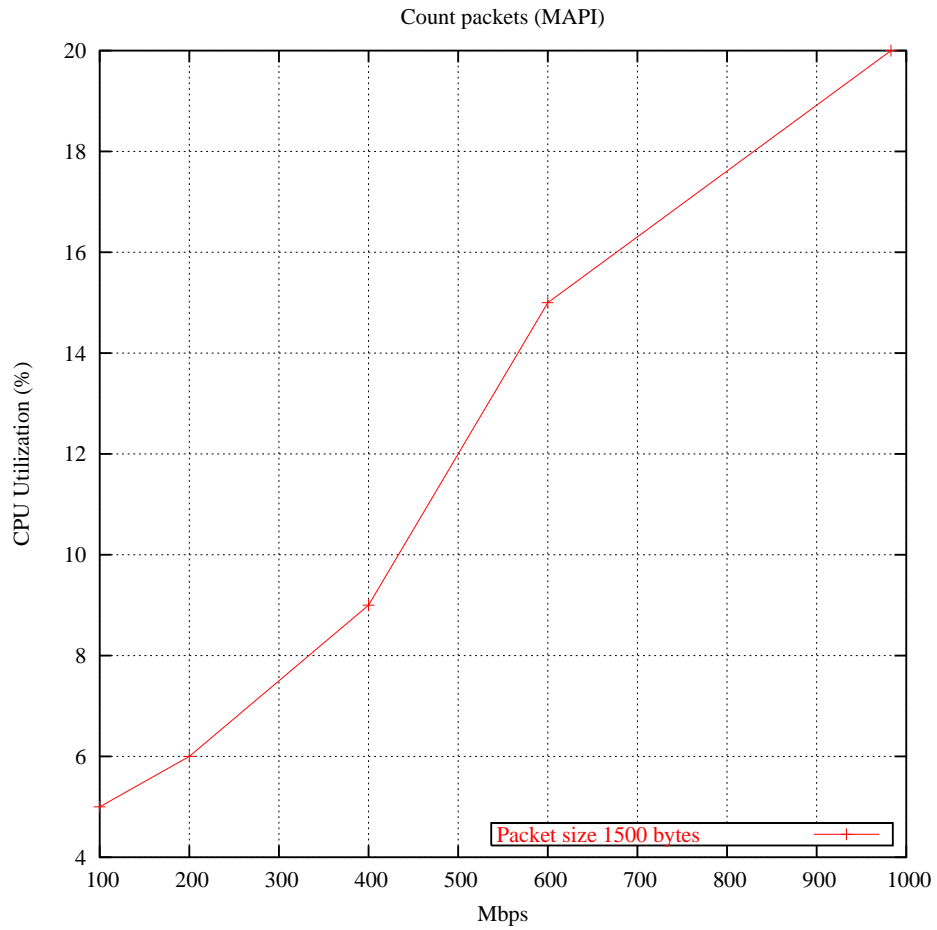
2.1 Read packets and then discard

In this experiment sender sent 1500 byte packets with increased rate. MAPI read the packets and send them on userspace. The application was reading the packets and then discard them. In the receiver side we measure the CPU percentage used. MAPI received all the packets.



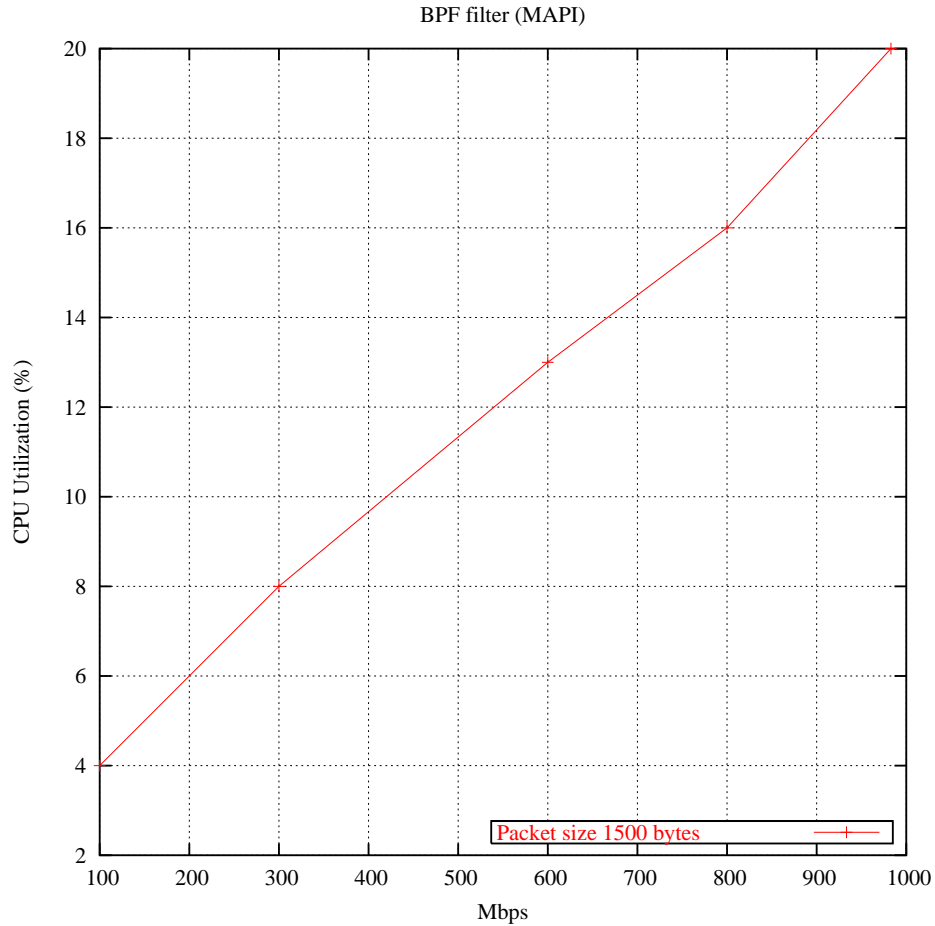
2.2 Count packets

In this experiment sender sent 1500 byte packets with increased rate. MAPI just read the packets and count them on the kernel (no packet reached userland). In the receiver side we measure the CPU percentage used. MAPI received all the packets.



2.3 Run BPF

In this experiment sender sent 1500 byte packets with increased rate. MAPI was reading the packets and running one BPF ¹. No packet matched the filter and so no packet reached userland. In the receiver side we measure the CPU percentage used. MAPI received all the packets.



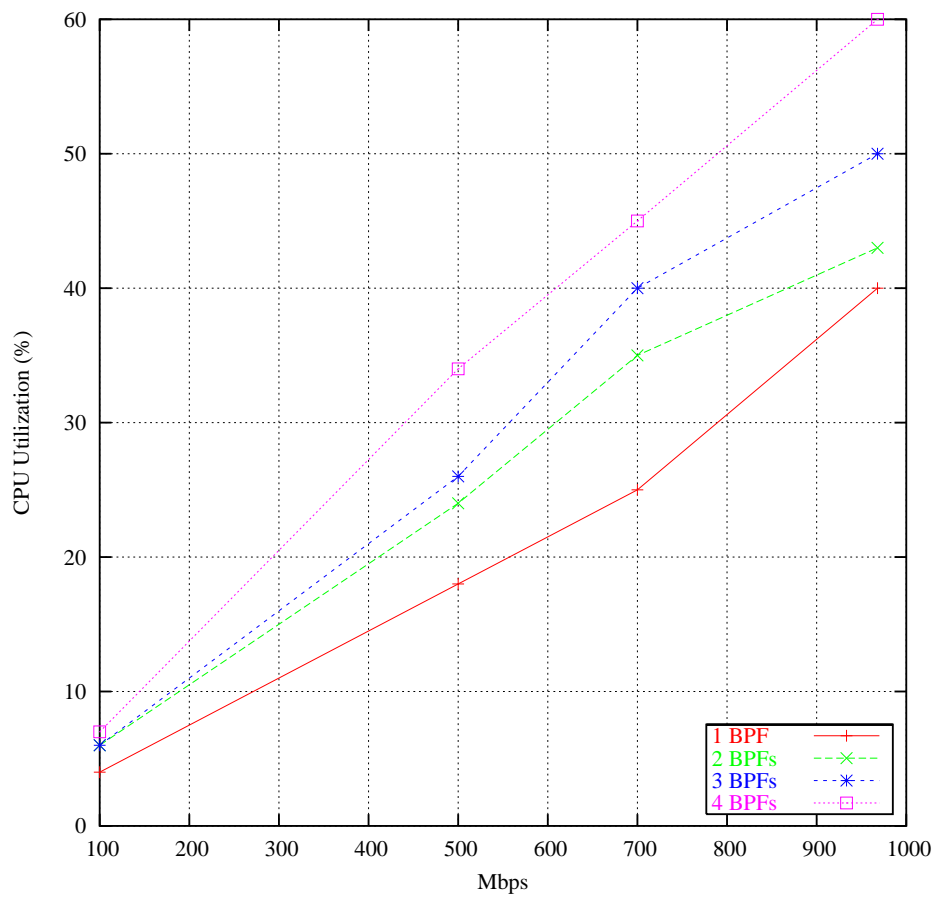
2.4 Run a couple of BPFs using MAPI

In this experiment sender sent variable size packets with increased rate. The average packet size was approximately 730 bytes. MAPI was reading the packets and running the BPF filters ². No packet matched the filter and so no packet reached userland. In the receiver side we measure the CPU percentage used. MAPI received all the packets.

¹port 8080

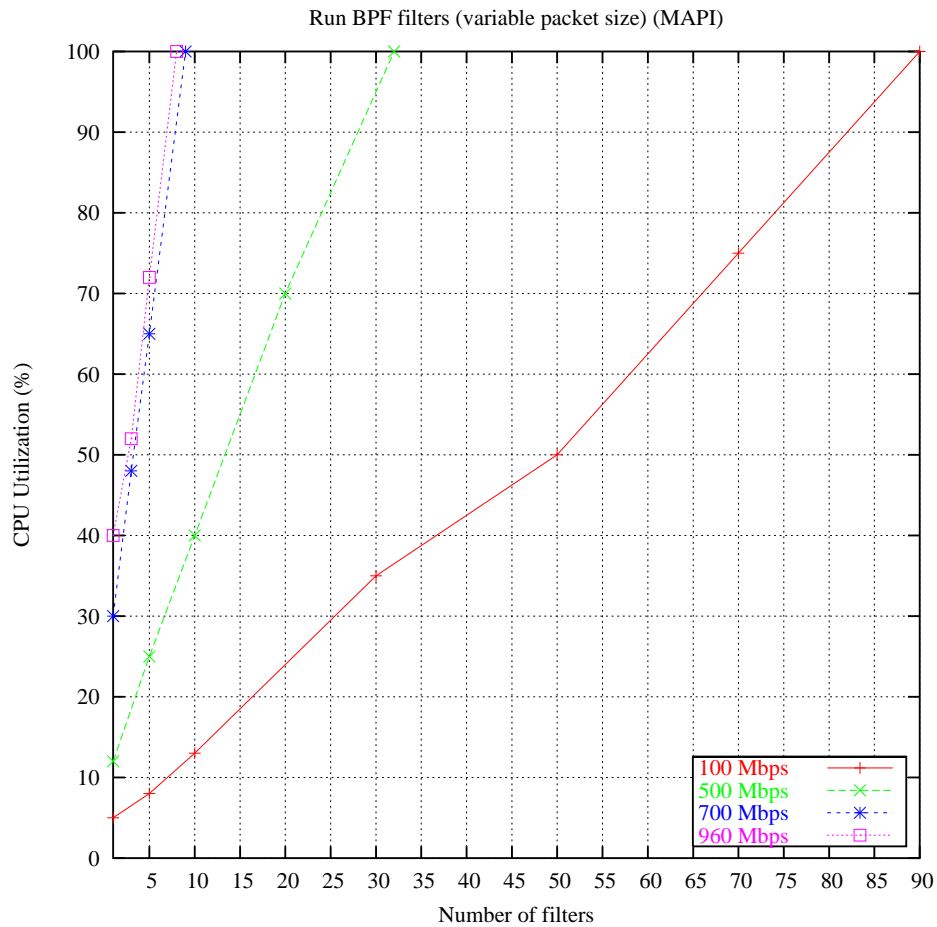
²port 8080

Run BPF filters (variable packet size) (MAPI)



2.5 Run many BPFs using MAPI

In this experiment sender sent variable size packets at constant rates. The average packet size was approximately 730 bytes. Receiver was increasing the BPFs. MAPI was reading the packets and running the BPFs³. No packet matched the filter and so no packet reached userland. In the receiver side we measure the CPU percentage used. MAPI received all the packets.

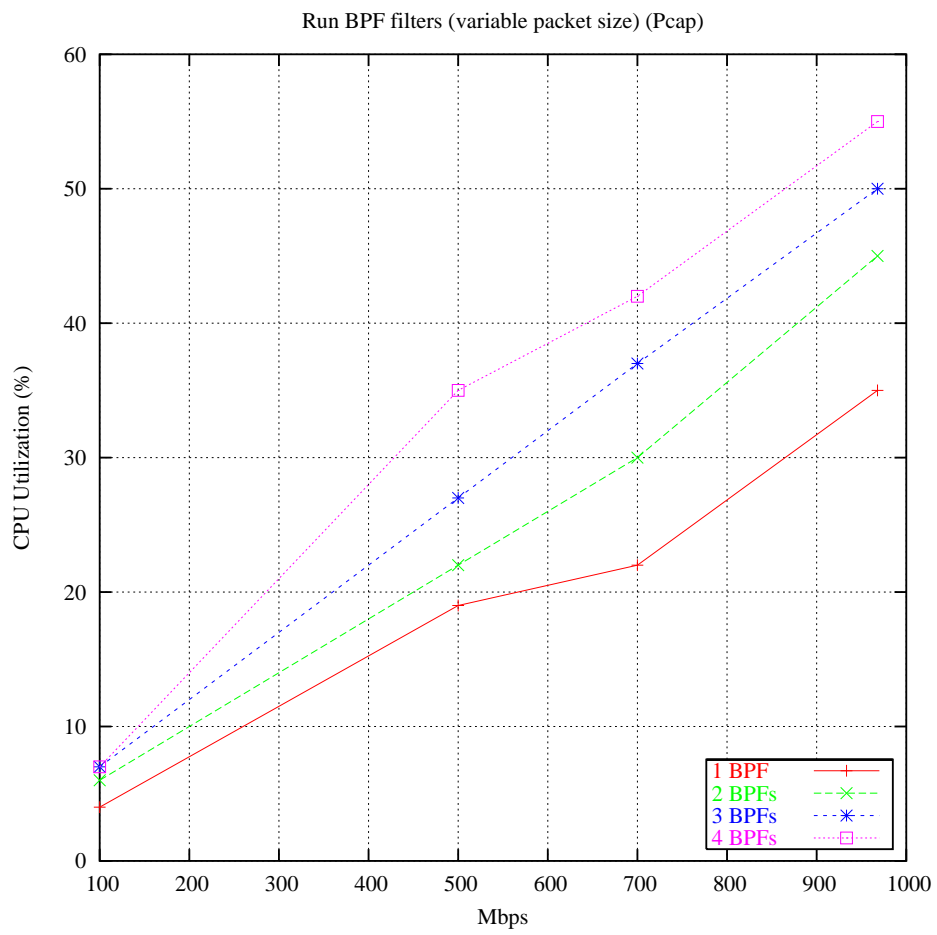


2.6 Run a couple of BPFs using Pcap

In this experiment sender sent variable size packets with increased rate. The average packet size was approximately 730 bytes. Pcap was reading the packets and running the BPF filters⁴. No packet matched the filter and so no packet reached userland. In the receiver side we measure the CPU percentage used. Pcap loose some packets.

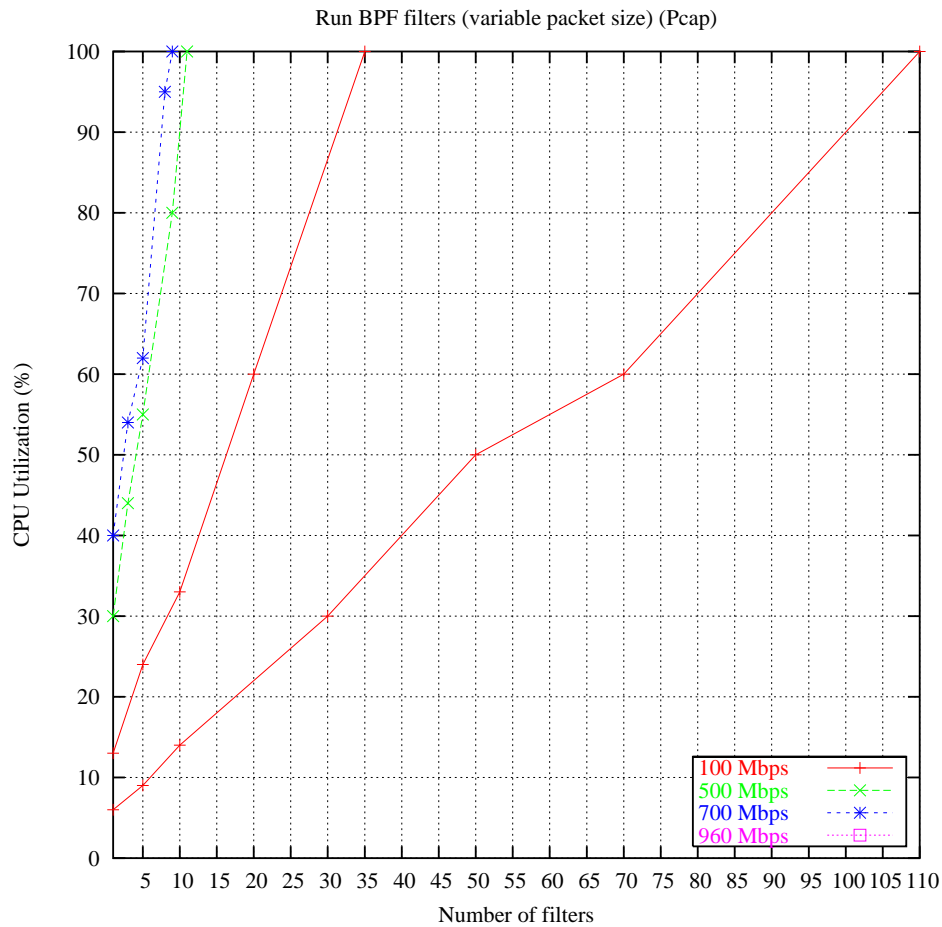
³port 8080

⁴port 8080



2.7 Run many BPFs using Pcap

In this experiment sender sent variable size packets at constant rates. The average packet size was approximately 730 bytes. Receiver was increasing the BPFs. Pcap was reading the packets and running the BPFs⁵. No packet matched the filter and so no packet reached userland. In the receiver side we measure the CPU percentage used. Pcap loose some packets.



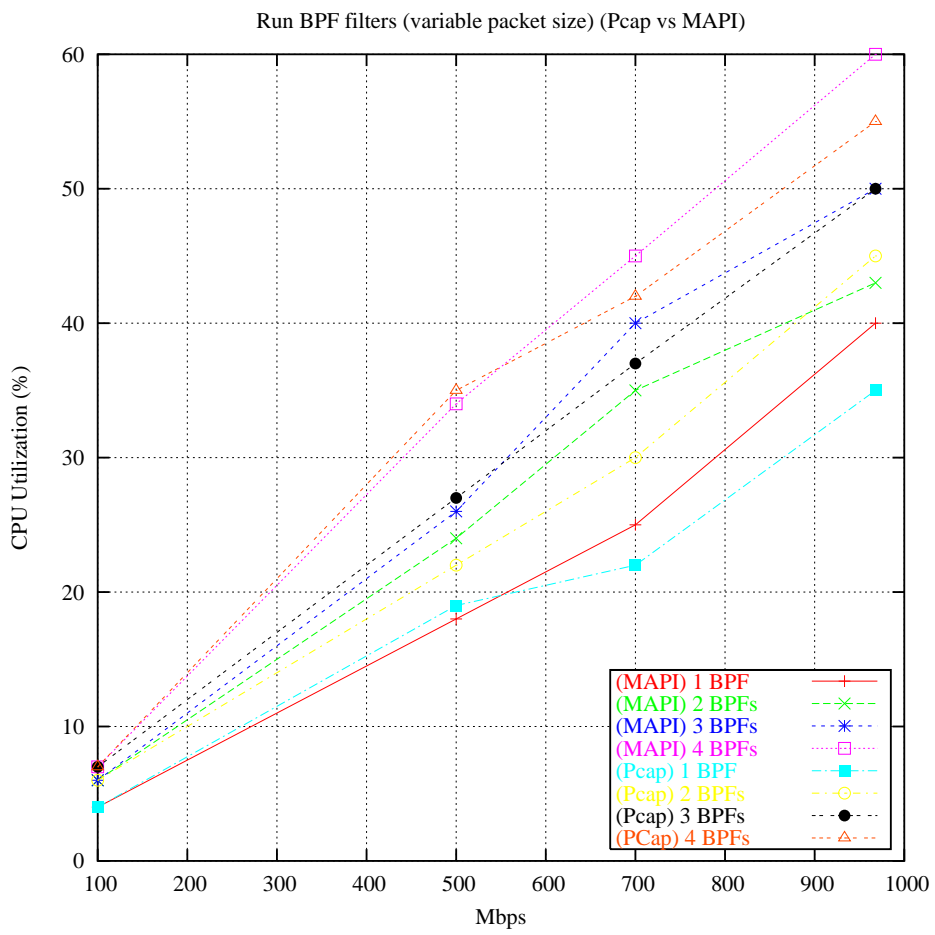
2.8 Pcap vs MAPI

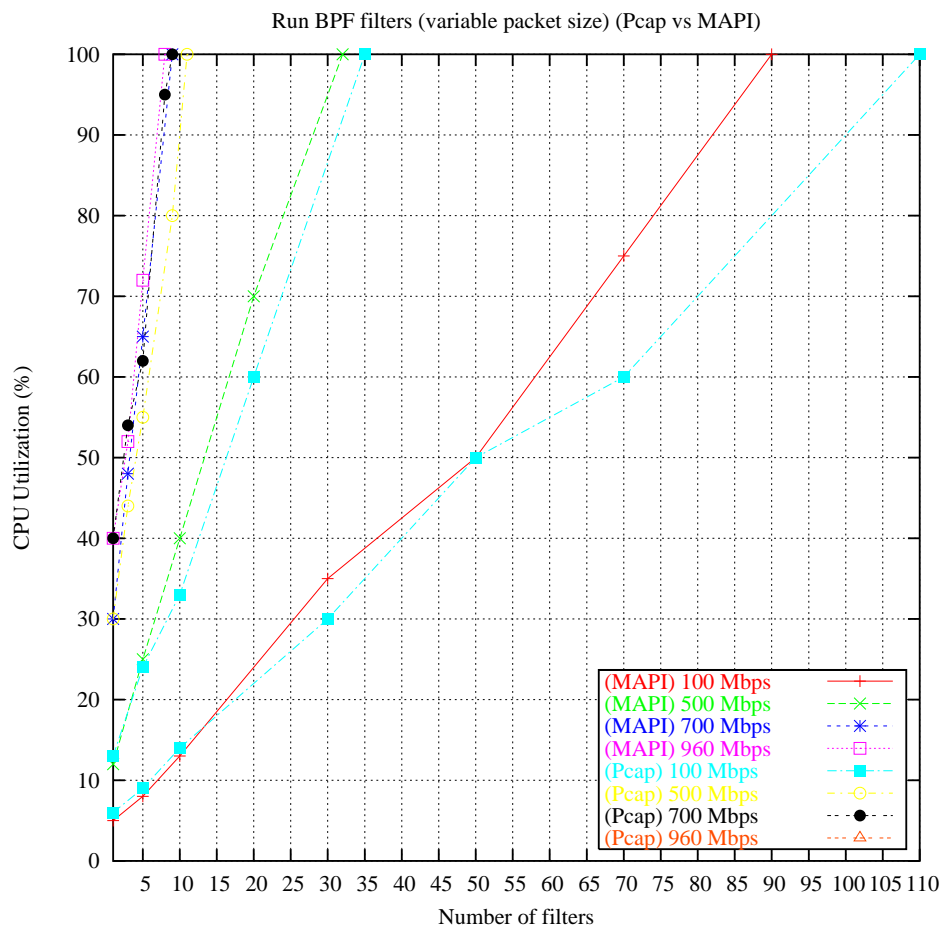
Here we plot the above figures together.

2.9 Substring search (MAPI)

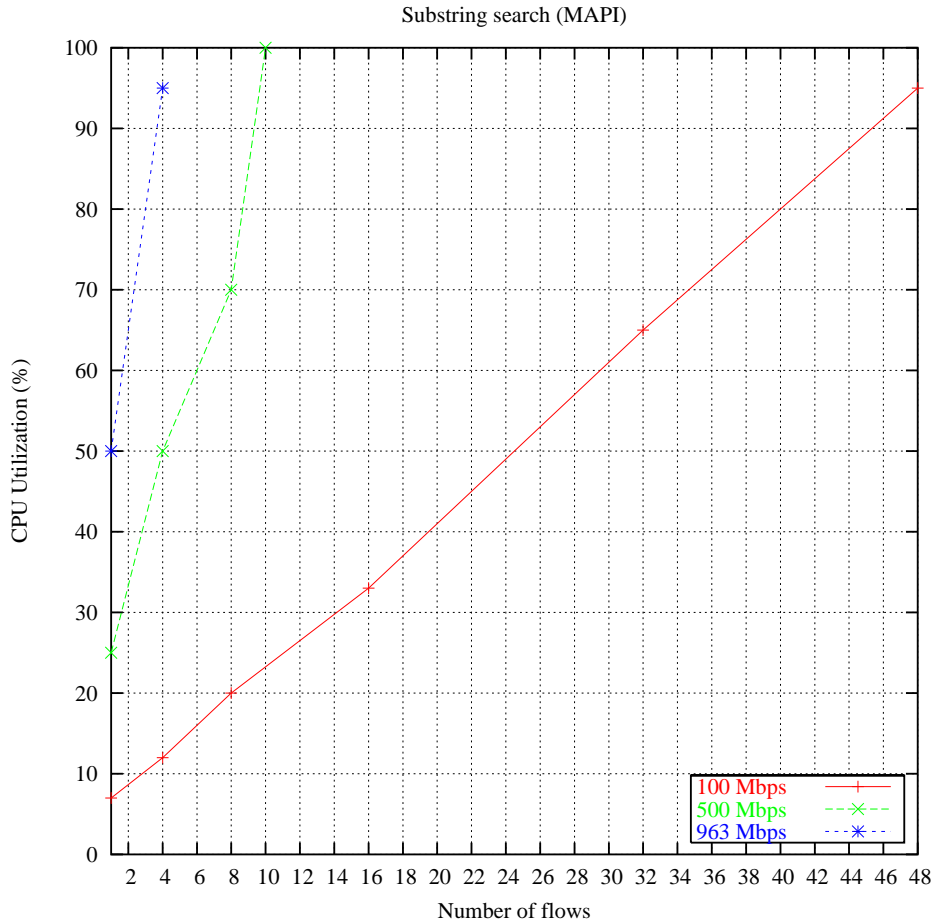
In this experiment sender sent variable size packets at constant rates. The average packet size was approximately 730 bytes. Receiver was increasing the signatures to search. MAPI was reading the packets and searching for signatures. No

⁵port 8080





packet matched the signatures and so no packet reached userland. In the receiver side we measure the CPU percentage used.



2.10 Substring search (Pcap - Mmap pcap)

In this experiment sender sent variable size packets at constant rates. The average packet size was approximately 730 bytes. Receiver was increasing the signatures to search. Pcap was reading the packets (in userspace) and searching for signatures. No packet matched the signatures. In the receiver side we measure the CPU percentage used.

