Intelligent Network Services through Active Flow Manipulation

T. Lavian, P. Wang, F. Travostino,
S. Subramanian, D. Hoang, V. Sethaput

Nortel Networks, UC Berkeley, Harvard U

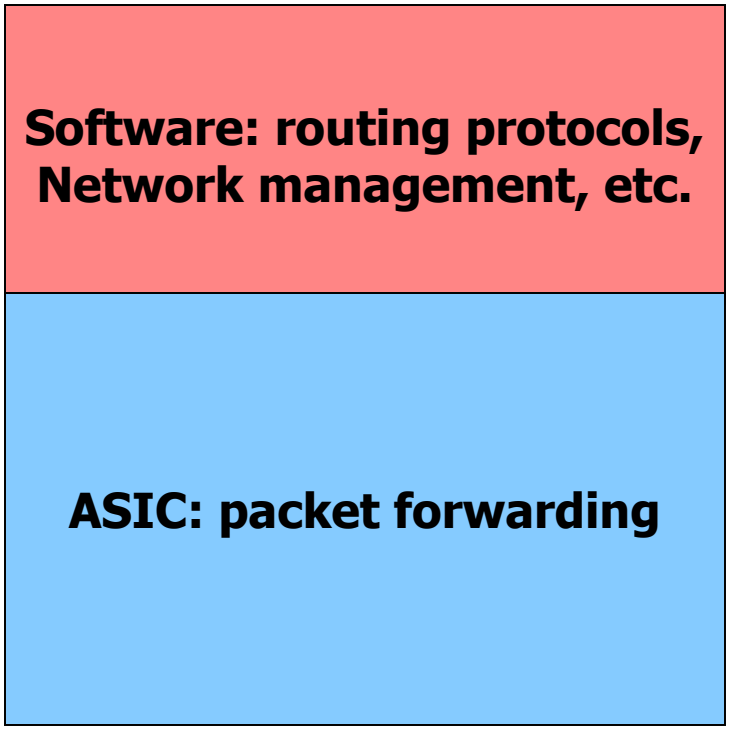
Outline

- Introduction
- Network Element – Control Plane/Forwarding Plane
- Active Flow Manipulation (AFM) abstractions
- OPENET
- Examples
- Conclusion

Programmability

- **A significant challenge in today's Internet is the ability to efficiently incorporate customizable network intelligence in commercial high performance network devices.**
  - **Framework for introducing services**
  - **API for programming network devices**

Network Element

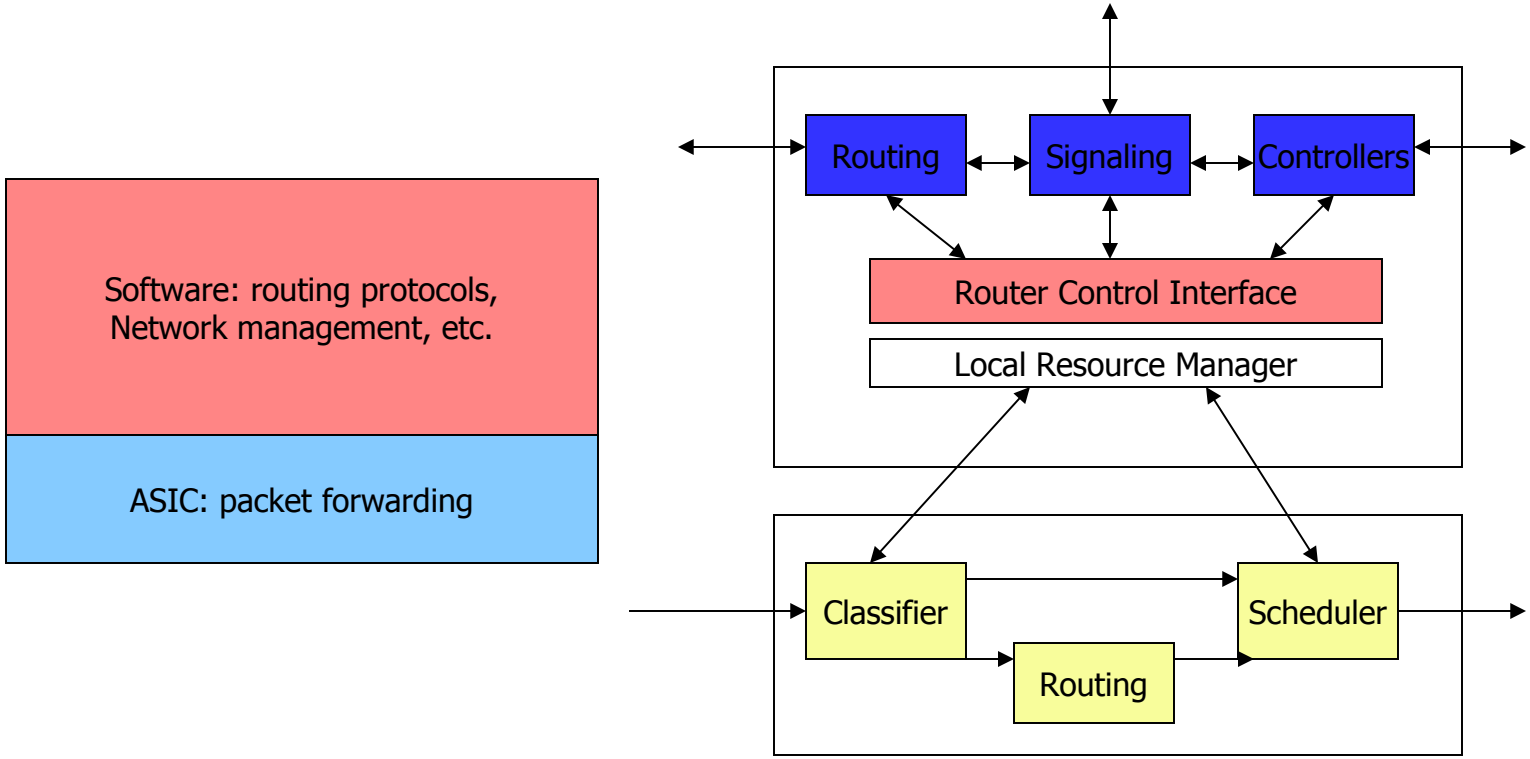


Limited control of the forwarding plane

Routers are not reprogrammable (except by vendors)

Users can only see IP/ICPM packets, but have no direct control over the internal handling of their data.

Programmable Network Element



Active Flow Manipulation Abstractions

- **Aggregate data into traffic flows**
  - **Flows whose characteristics can be identified in real-time**
  - **E.g., "all UDP packets to a particular service", "all TCP packets from a particular machine".**
- **Actions to be performed in the traffic flows**
  - **Actions that can be performed in real-time**
  - **E.g., "Change the priority of all traffic destined to a particular service on a particular machine", "Stop all traffic out of a particular link of a router".**

Identifiable Elements of Primitive Flows

Destination Address (DA)
Range of Destination Address (RDA)
Source Address (SA)
Range of Source Address (RSA)
Exact TCP protocol match (TCP)
Exact UDP protocol match (UDP)
Exact ICMP protocol match (ICMP)
Source Port number, for both TCP and UDP (SP)
Destination Port number for both TCP and UDP (DP)
TCP connection request (TCPReg)
ICMP request (ICMPReg)
DS field of a datagram (DS)
IP Frame fragment (FrameFrag)

Primitive Permissible actions

Drop
Forward
Mirror
Stop on Match (SOM)
Detect Out of Profile behaviour (Out)
Change DSCP value (DSCP)
Prevent TCP Connect Request
Modify IEEE 802.1p bit

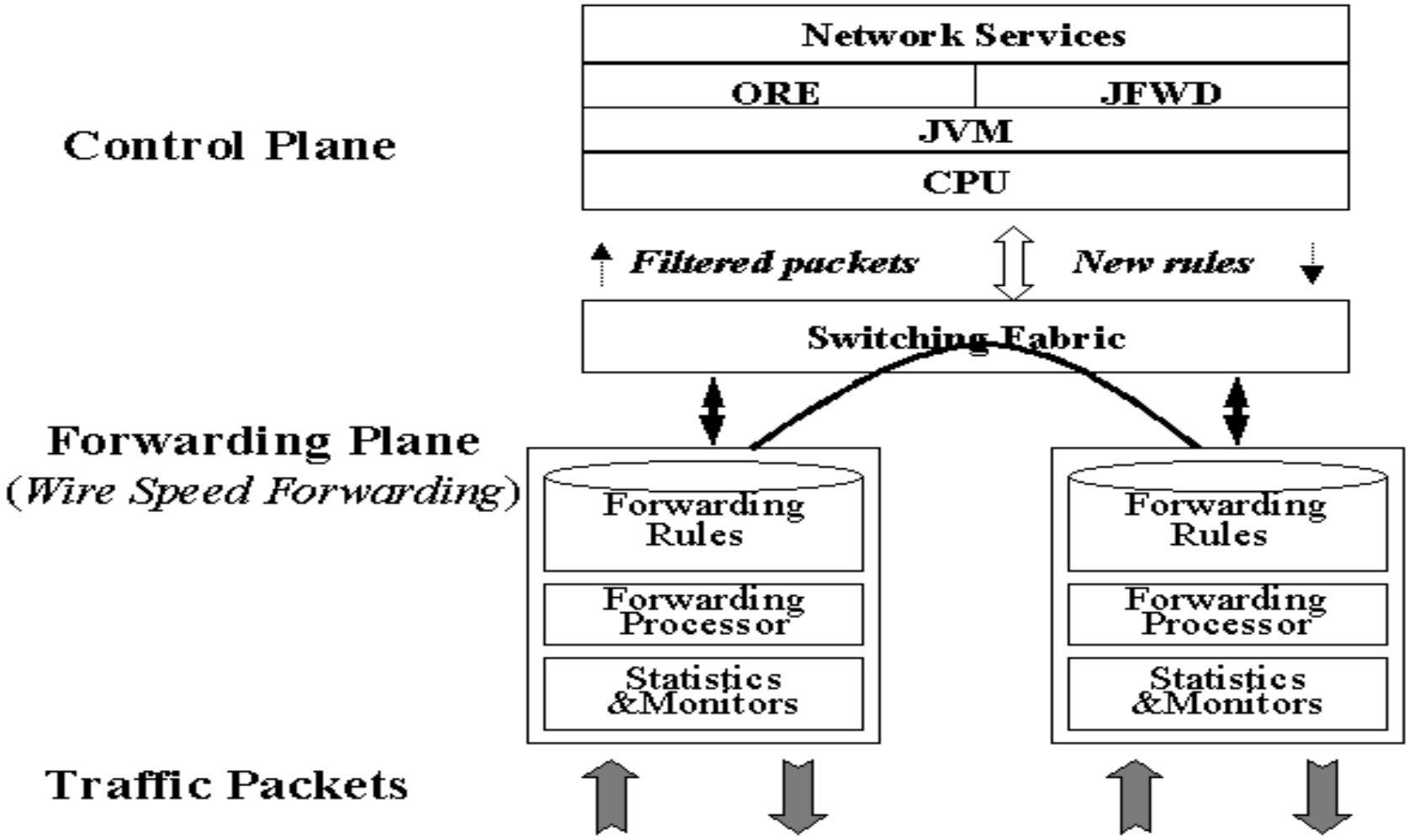
A subset of flows

	Destination Address (DA)
	All traffic to a particular destination machine
Range of DA	All traffic to a range of destination machines
Source Address (SA)	All traffic between 2 particular machines
Range of SAs	All traffic from many source machines to a particular destination
TCP	All TCP flows to a particular destination machine
UDP	All data gram packets to a particular destination machine
ICMP	All ICMP messages to a particular destination machine
ICMP Request	All ICMP requests to a particular destination machine
TCP ACK	All TCP acknowledgements to a particular destination machine
TCP RST	All TCP connection with the RST bit set
DP (TCP)	All TCP flows to a particular service in a particular server machine
DP (UDP)	All UDP datagram to a particular service in a particular machine
SA-SP (TCP)	All TCP flows from particular client of a source to a destination
SA-SP (UDP)	All UDP datagram from a client of a source to a destination
IP Fragments	All IP fragments to a particular destination machine
DS Field	All traffic of a particular QoS class to a particular destination
VLAN	All traffic from a particular VLAN to a particular destination
Switch-Port	All traffic on a particular switch port to a particular destination

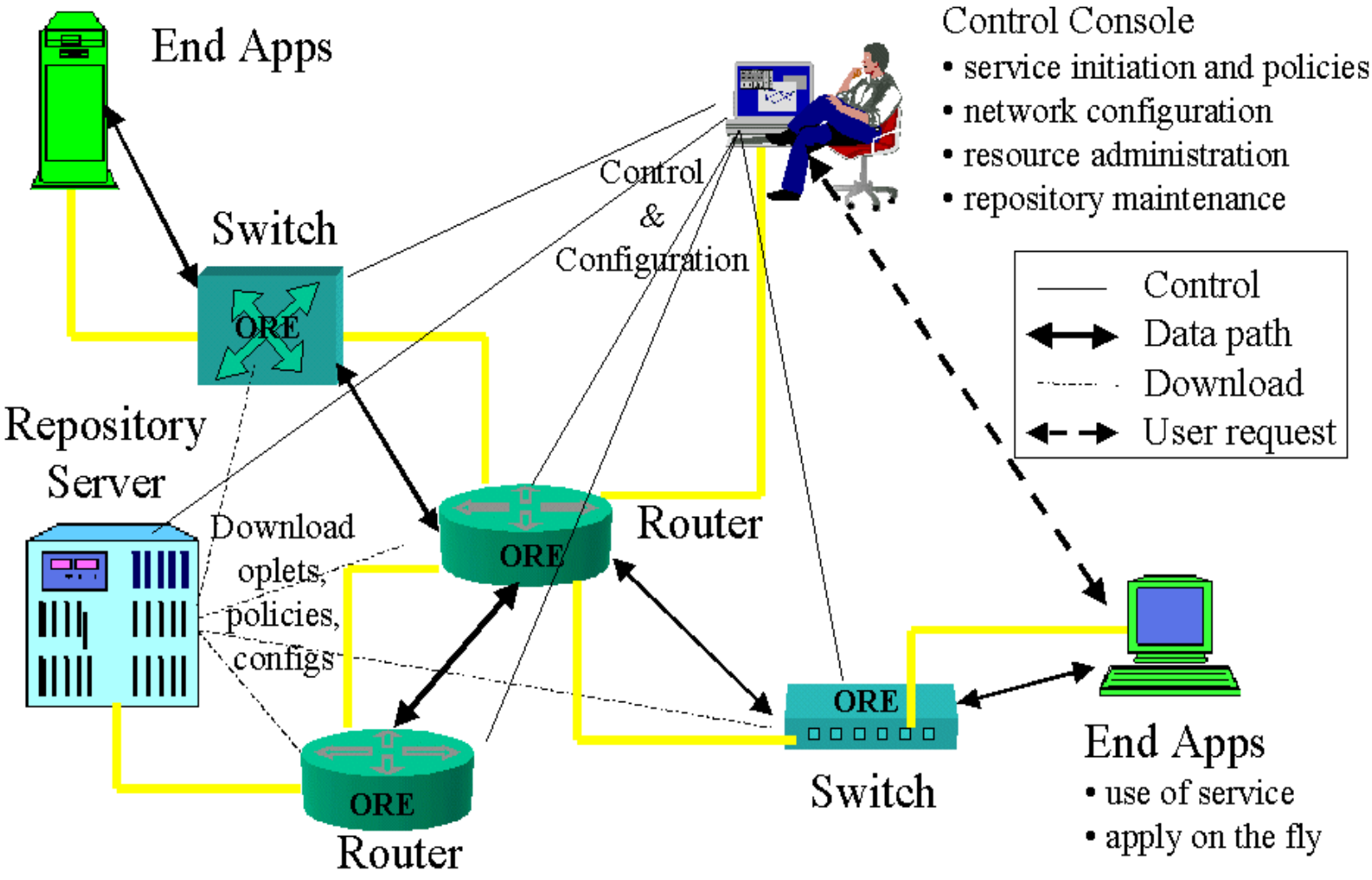
New Capabilities

- Allow introducing services and control on demands dynamically
  - Services can be any general network applications
  - Control on demands to manipulate flows and flow aggregates
- Allowing dynamic and mobile agents
- Respond quickly to changes in traffic conditions.
- Cope with unforeseen requirements
- Extending router functionality (optimization)
- Multiple control elements are installed at routers or hosts and they collaborate to achieve some overall objective.

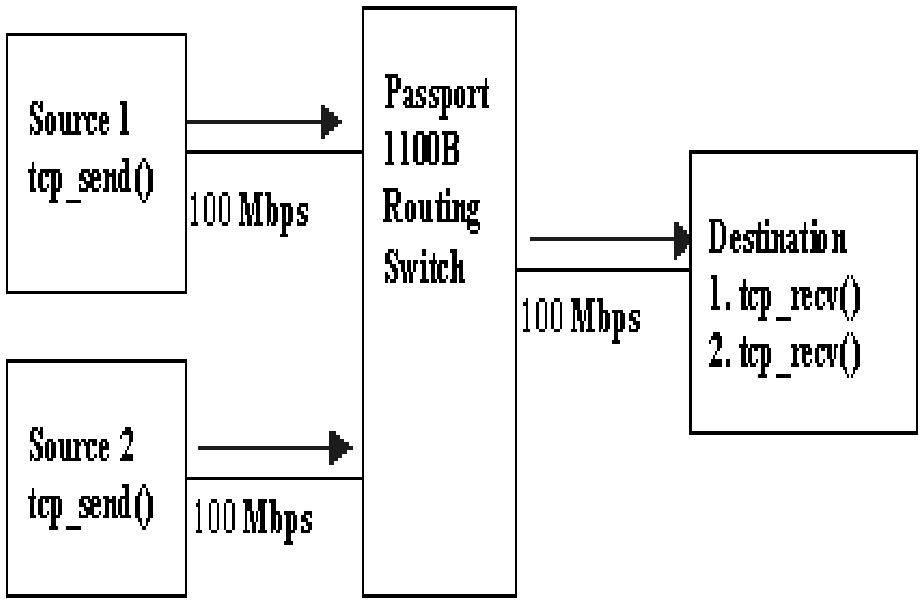
Openet Architecture

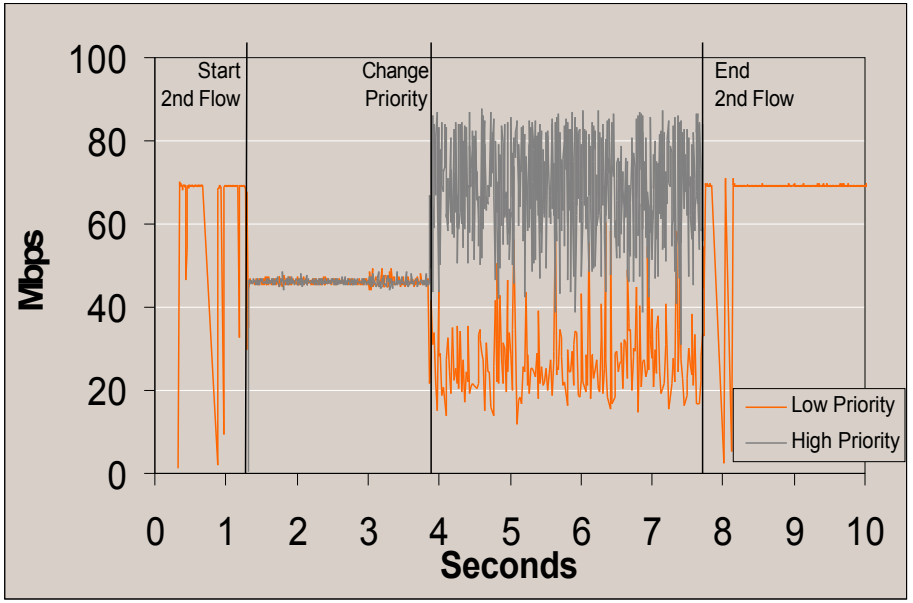


Openet: Passport Implementation

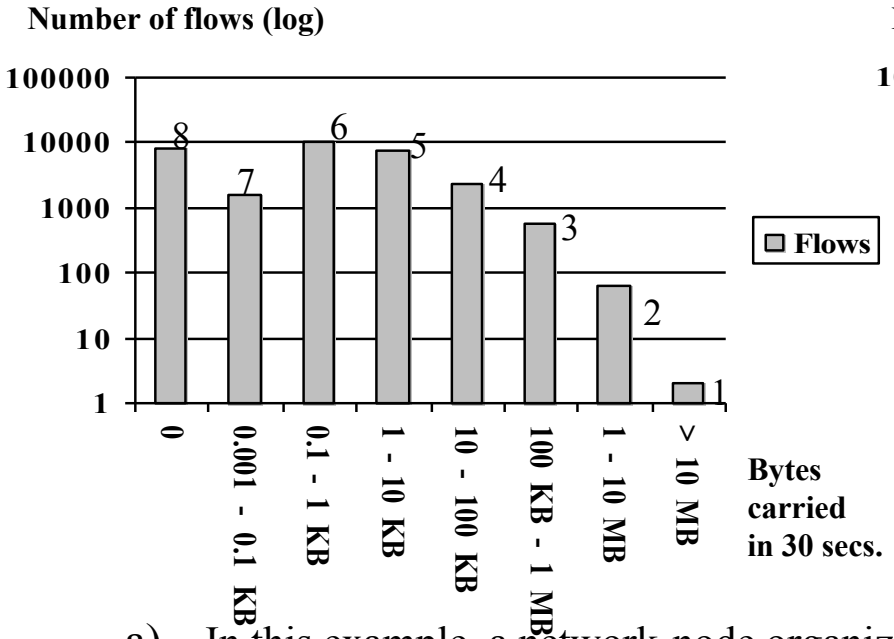


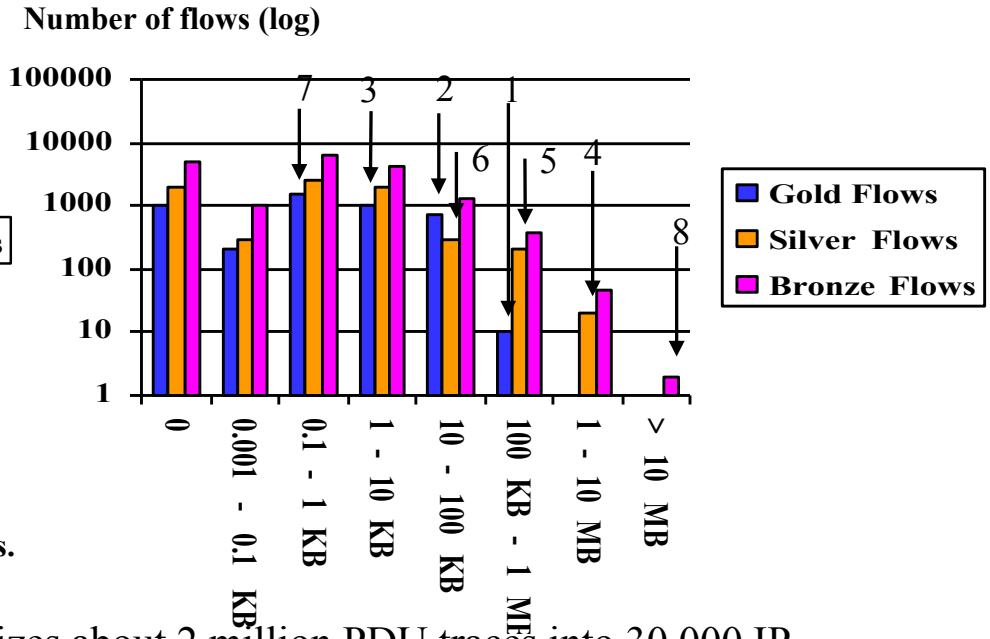
Active Flow Priority Change in Real-time





AIACE

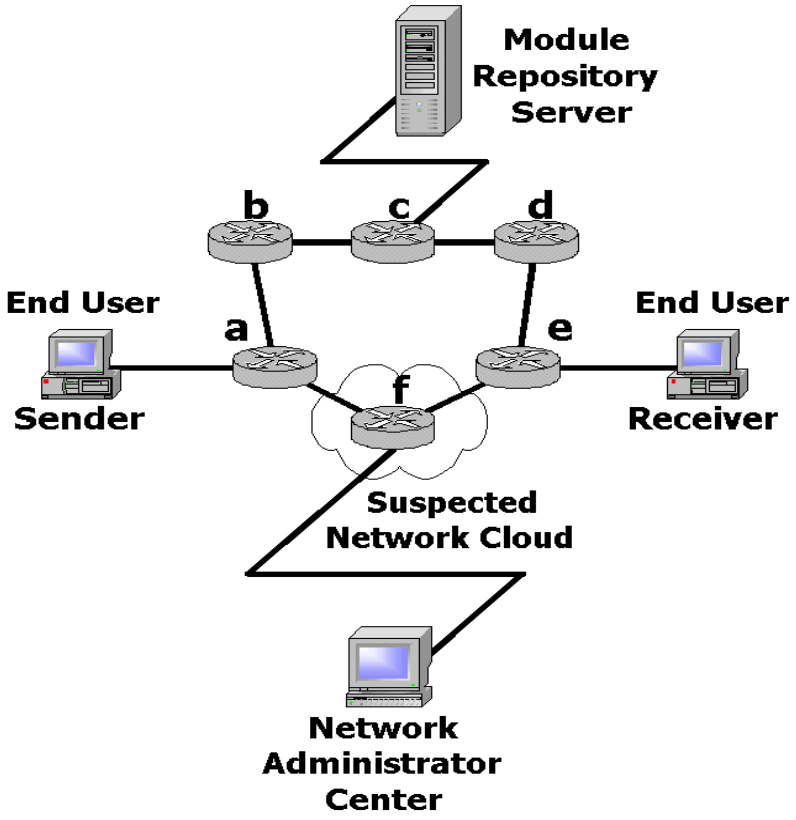


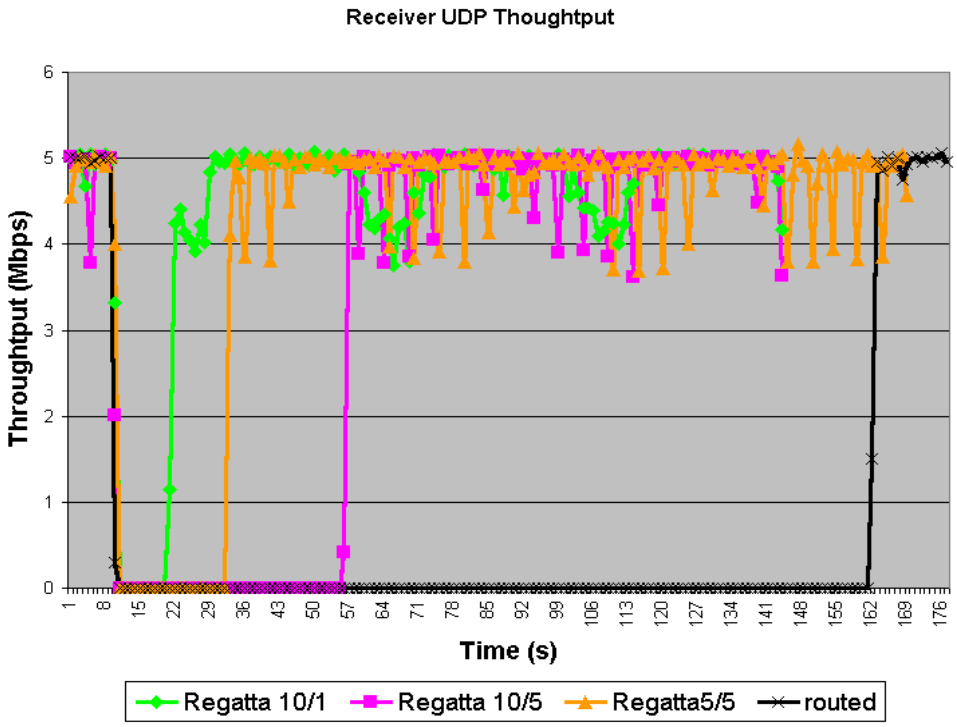


a) In this example, a network-node organizes about 2 million PDU traces into 30,000 IP flows. It classifies the resulting flows based on the bytes transferred on each flow. It then ranks flows (from 1 to 8). The higher the rank number, the higher the chance that the flow will not be transferred to the accounting server in case of data overload.

b) The node now structures the same accounting data into QoS-flavored flows (same X and Y axis as in a). After applying a QoS-specific weighting algorithm to the flows, the node ranks flows with different results than a). The weighting algorithm can be arbitrarily complex and take into account other considerations besides bytes transferred (e.g., hosts, number of packets, duration).

Regatta: Dynamic flow bypass





Regatta: Reactivity times

Flow Path	Reactivity Time (s)
Static route	Infinite
Routed	152
Regatta 10/1	10
Regatta 10/5	47
Regatta 5/5	24
Regatta M/HB	! M*HB

Conclusions

- AFM enables dynamic introduction of services
- AFM enables rapid network response to changing conditions
- AFM in a powerful control plane can lead to sophisticated control over forward plane
- AFM allows practical implementation of programmability in a real world network device