

Active Networking

ECE/CS598HPN

Instructor: Radhika Mittal

Active Network Definition

- Network nodes (routers) can perform computation on, and modify, packet contents.
- This processing can be customized on a per-user per-customer basis.

Motivation

- Internet took off in mid 1990's
 - New applications, larger scale.
- Innovations in protocol design (e.g. multicast, IPv6, mobile IP)
 - Slow and tedious: 10 years from lab prototype to deployment.

“Allow applications to customize message processing to suit their purpose.”

Two approaches

- **Capsule**

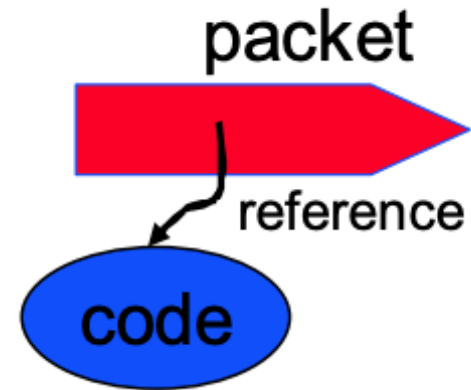
- **Packets carry code.**
- Program is embedded in the packet itself.
 - Instructions that perform basic computation on capsule contents.
 - Outcome: state change at node, transmission of zero or more capsules.
- In-network node identifies capsule boundaries and “safely” executes it.
- *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, MIT*



Two approaches

- **Programmable switch**

- Packet carry reference to code.
- Network nodes injected with desirable packet programs.
- They examine packet header and execute appropriate program.
- *An architecture for active networks, Georgia Tech*
- *SwitchWare: Accelerating network evolution, UPenn*



Key enablers of the idea

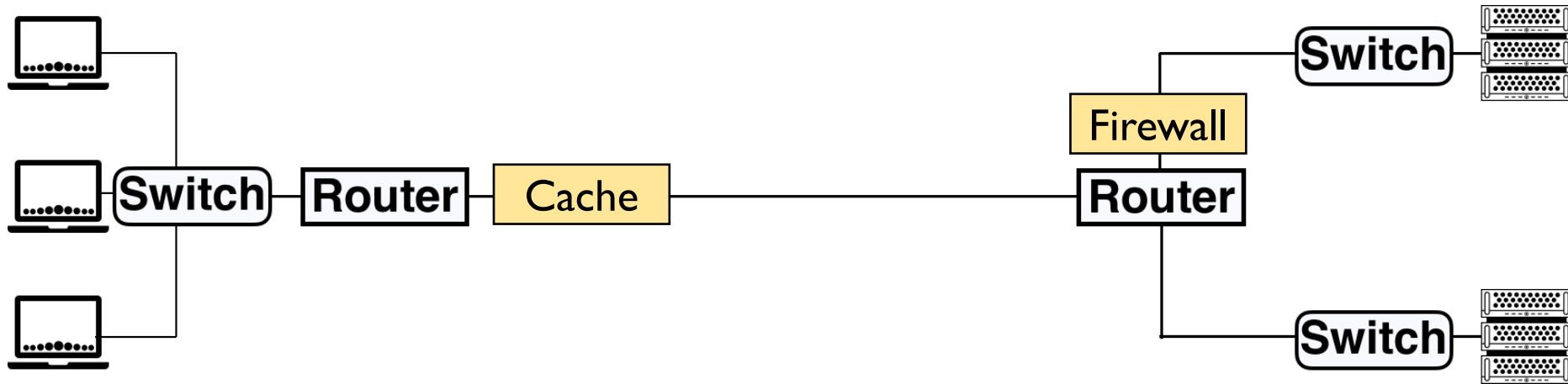
- Reduction in cost of computing
- “New” programming languages such as Java
 - Code portability
 - Some safety in execution
- Virtual machine technology
- Push from DARPA.

Proposed usecases

- Update firewalls more easily.
- A router that “adapts” to the system it is connected to (perform compression, encryption, etc).
- TCP proxies, web proxies, and other application-specific proxies.

Are we just talking about middleboxes?

Middleboxes



Can NFV (software middleboxes) be thought of as an example of active networks?

Do we really need switches and routers to be “active”?

Other proposed usecases

- Support for in-network multicast
 - Deployment barrier wasn't the only hurdle.
 - How to bill customers for multicast traffic?
- Security and authentication in the network
 - Why not end-to-end?
- Sophisticated approaches to network monitoring.

Long-term impact

- Active networking shares common roots with today's efforts for programmable networks:
 - Software-defined networking
 - Programmable dataplane
 - Network Function Virtualization
- *Often unacknowledged....*

What went wrong...

- Lack of compelling usecases.
- Extreme design point that was difficult to deploy.
- Performance not given enough attention.

Discussion in SOSp'99

Capsule-based design presented by David Wetherall

Jeff Mogul from Compaq WRL observed that there appears to be a contradiction in the design of Dave's system. The performance is best at the edges of the network, where there a lot of cycles per packet, but the applications described are better suited for the center of the network. Dave answered that there's no contradiction: the applications also work at the edges. Some things are also possible in the core, such as Random Early Discard and Explicit Congestion Notification. It's a continuum.

Discussion in SOSp'99

Capsule-based design presented by David Wetherall

Ken Birman from Cornell stated that it's time to call the question of whether active networks are ever going to come to anything or not. He has yet to see one good reason for using an active network. Dave's response: you don't need novel applications; the old problems like multicast still aren't solved: that should be reason enough. Ken asked: but do we really need active networks for those old problems? The answer was that for fast deployment, we absolutely do need them. Dave believes that we're going to succeed, but in the end it won't be called "active networks" any more.

Discussion in SOSp'99

Capsule-based design presented by David Wetherall

Timothy Roscoe from Sprint was only allowed to ask another question on the condition he *not* agree with Ken. Roscoe replied that he only slightly agreed and hence proceeded to declare that multicast was indeed in every router but not turned on, and that ISPs offer it only under special circumstances....he added that billing was the real key which had been overlooked. **How do we charge for chewing up resources?** Wetherall replied that he had not considered billing, but that he will look at it once the other problems have been solved!

“We realize that suggestions for software-intensive approaches to networking surface every ten years or so.....”
-- Authors of active networking, mid 1990's

mid 2000's: Software-defined Networking

mid 2010's: Programmable Dataplanes (switches)

What differs in later approaches

- **Right technology**

- OpenFlow (SDN), programmable switching hardware, P4, ...
- Realize that full-fledged programmability is at odds with performance.

- We have a suitable deployment environment (datacenters).

- But do we have compelling usecases?

Relationship with E2E

- Application autonomy:
 - Higher layers should be free to organize network level resources to achieve specific goals
 - In-network programmability enhances autonomy.
- Network transparency
 - Allow multiple independent applications to effectively share network resources.
 - One must be able to predict network behavior.
 - In-network programmability makes network behavior harder to predict.
- *Simple, yet flexible, network programs.*

Project Guidelines

- Groups of up to 3 students
 - Can use Campuswire to find teammates.
 - Not necessary to find a partner; ok to work on a project on your own.
- Start thinking of an idea.
- Schedule a meeting with me to discuss it
 - Watch out for an email with instructions to schedule the meeting.
 - Prepare a 5-10mins long presentation on the idea
 - What you plan on doing, why is it interesting/useful, how you will go about it.

Properties of a good project idea

- Well-motivated
- New
- Tractable
- Relevant to the topics we will discuss in class

In-network programmability

- New use-cases
 - e.g. improved routing, load-balancing, congestion control, etc.
- Or pick an existing use-case and show that an end-host based design could work as well

Endhost Network Stack

- Characterizing the trade-off between implementing a network stack primitive in hardware vs software.
- New use-cases of programmable/smart NICs.

Networking for AI

- Rethinking network architecture and protocols for speeding up communication for distributed AI training or inference.
 - Redesigning RDMA / hardware based transport.
 - Creative use of new technologies (e.g. CXL).
 - SmartNIC offloads.

Beyond Datacenters: Edge and IoT

- Rethinking network stack design and protocols for emerging applications such as robotics, augmented/virtual reality, IoT sensors, etc.
- Using SDN principles (global knowledge/centralized control) to manage distributed IoT devices.
- New usecases for programmable networking hardware (e.g. smartNICs) in this context.

5G/6G Cellular Networks

- Leveraging software-based RAN to rethink cellular protocols and mechanisms.
 - load-balancing in multi-cell deployments, RAN resource scheduling, admission control, etc.
- Mobile edge computing: cellular infrastructure provides edge servers that can be used for various offloads.

How to get ideas?

- Pick a topic that fascinates you the most.
- Look at some papers on that topic
 - E.g. those in the reading list
 - But more in programs of top conferences and workshops
 - Networking: SIGCOMM, NSDI, NINeS, **HotNets**
 - Systems: SOSP, OSDI, EuroSys, ASPLOS, MLSys, **HotOS**
 - Architecture: Micro, ISCA, ...
 - Mobile/Sensing/IoT: Mobicom, Mobisys, Sensys
 -
- Think about what's missing, question the assumptions, can you extend the idea to some other domain/scenario....
- Combining two disciplines enhances novelty and excitement

Resources

- Simulation:
 - ns2: <https://www.isi.edu/websites/nsnam/ns/>
 - ns3: <https://www.nsnam.org/>
 - OMNET/INET: <https://inet.omnetpp.org/>
- Emulation: Mininet
 - Can also be used for experiments with OpenFlow and P4.
 - Helpful tutorials:
 - <https://github.com/mininet/openflow-tutorial/wiki/Installing-Required-Software>
 - <https://github.com/p4lang/tutorials>
- Free Xilinx packages for FPGA synthesis

Resources

- CloudLab
 - Project created for this class. Students can be added upon request.
- AWS free tier
- Linux workstations on campus
- I can try to arrange for specific requirements
 - Other academic clusters (e.g. Delta cluster)

Deliverables

- Sept 14 - Sept 24: project proposal meeting
 - 5-10mins long presentation on what you plan on doing, why is it interesting/useful, how you will go about it.
- Oct 7th: First progress report due
 - Motivation
 - Related work survey
 - Key challenges
 - Rough design sketch
 - Implementation and evaluation plan
 - Progress made so far
 - Tentative timeline
 - For teams with multiple people: who is working on what.
 - AI attribution and acknowledgement (if applicable)

Deliverables

- Nov 6th: Second progress report due
 - What changed since the first report?
 - Implementation progress
 - Preliminary results
- Dec 4th: Final project report due (6 pages, 10pt font)
 - Introduction: motivates the problem
 - Related Work
 - Design
 - Implementation
 - Evaluation
 - AI attribution and acknowledgement (if applicable)
- ~Dec 9th: class presentation (10-15mins long)