

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

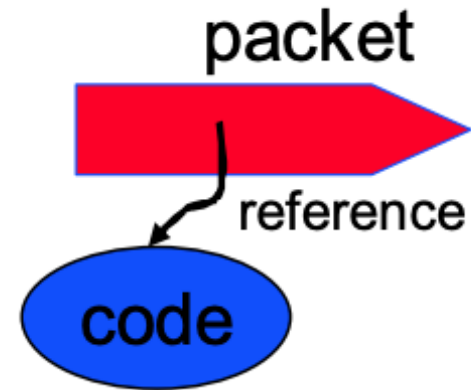
- Internet took off in mid 1990's
 - New applications, larger scale.
- Innovations in protocol design (e.g. multicast, IPv6, mobile IP)
 - Slow and tedious: 10 years from lab prototype to deployment.

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

Two approaches

- **Programmable switch**

- Packet carry reference to code.
- Download desirable packet programs.
- Examine packet header and execute appropriate program.
- *An architecture for active networks, Georgia Tech*
- *SwitchWare: Accelerating network evolution, UPenn*



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 on zero or more capsules.
- Node identifies capsule boundaries and “safely” executes it.
- *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, MIT*



Key enablers of the idea

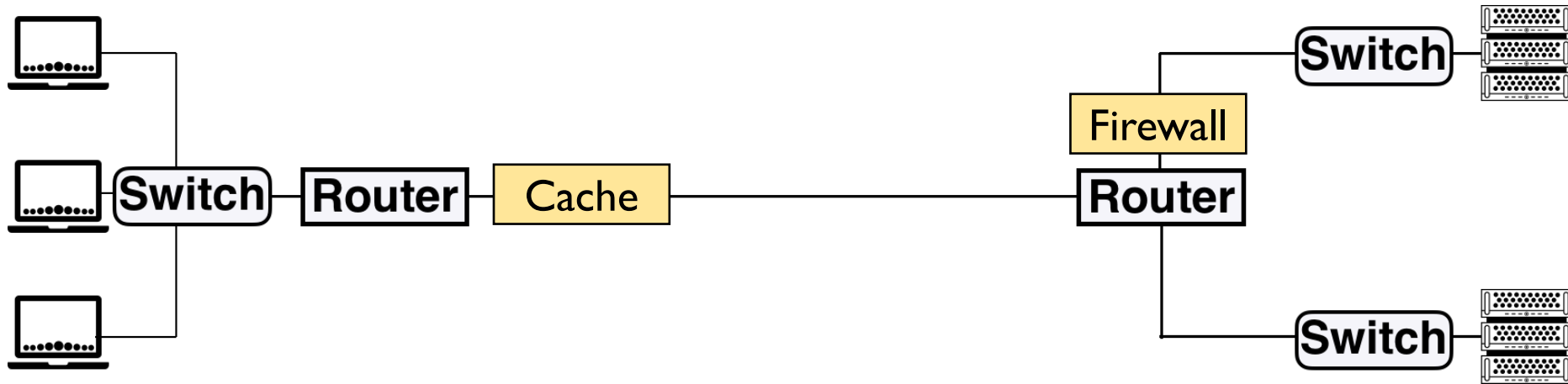
- Reduction in cost of computing
- “New” programming languages such as Java
 - Code portability
 - Some safety in execution
- Virtual machine technology
- Push from DARPA.

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

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!

Discussion in SOSPP'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.**

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

Software-defined Networking: mid 2000's

Programmable Dataplanes: mid 2010's

Are today's programmable networks here to stay?

What's different today....

- **We have the 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.*

Your Opinions

- Pros:
 - Innovative idea
 - Comprehensive overview
 - Talked about current research directions
 - Good motivation
 - Interesting usecases
 - Well-written

Your Opinions

- Cons:

- What are the trade-offs and limitations?
- Issues around safety and security.
- Alternate design points for the usecases.
- Lack of more details about the two approaches.

(please avoid comments on structure of the paper, presence/lack of figures etc).

Your Opinions

- Ideas:
 - Using active networks for enhanced routing, congestion control, scheduling etc.
 - Using active networks to monitor crypto-currency health.
 - How to multiplex switch-level resources across multiple applications.
 - Tool to catch bugs in / debug network programs