“From Internet to Active Net”

John Guttag
MIT Laboratory for
Computer Science
Cambridge, MA 02139-4307
guttag@lcs.mit.edu

What Makes a Net Active ?

No clean separation between

- Network, transport, and application layers

Environment for running distributed programs

- Many problems best addressed by distributed program
- Routers replaced by nodes that participate fully

Is This a Good Idea ?

Not obvious

Offers considerable flexibility

- What can be accomplished using this flexibility?
  - New functionality
  - Better performance
- At what cost?
  - Security
  - Performance when flexibility not needed

Our goal is to find out

- Start with willing suspension of disbelief

Overview of Activities at MIT

Making it possible to build active nets

- Active IP option
- *Capsule-based active nets*

Understanding performance issues

Search for useful applications

- Low level, e.g., reliable multicast, filtering, ...
- High level, e.g., simulation, video conferencing,...

The Players

David

Murphy

Tennenhouse

Wetherall

Steve Garland

John Guttag

Frans Kaashoek

Ulana Legedza

Li Wei Lehman

Liuba Shrira

Building Active Nets

Active IP option v1, complete

Maintain packet abstraction

Invoke set of pre-defined primitives

Active IP option v2, in progress

Maintain packet abstraction (IPv6)

More flexible program distribution model

***Encapsulated programs**

Routers replaced by capsule execution engines

(Some capsules contain data in the usual sense)

No out-of-band communication needed

Solving the security problems, especially denial of service

Distributed solutions

The Capsule Approach

Conventional nets rely on agreement about protocols

- Standardization a long and wrenching process
- Stakes are high

Active nets rely on agreement about model of computation

- Encoding of programs and data
- Access to node's resources
- Addressing mechanisms
  - Finding programs
  - Finding data

The Ants Architecture

Capsules programs (and data too)

- Programs run to completion (or self-suspension) at a node

Node primitives

- Soft state, cannot be relied upon
- Querying environment
- Capsule creation and manipulation
- Capsule control (e.g., forwarding)
- Capsule scheduling

Code distribution

- Some guaranteed present at each node
- Some transferred using code distribution protocol
  - Demand loading, prefetching, caching all used
  - Well-suited to flows

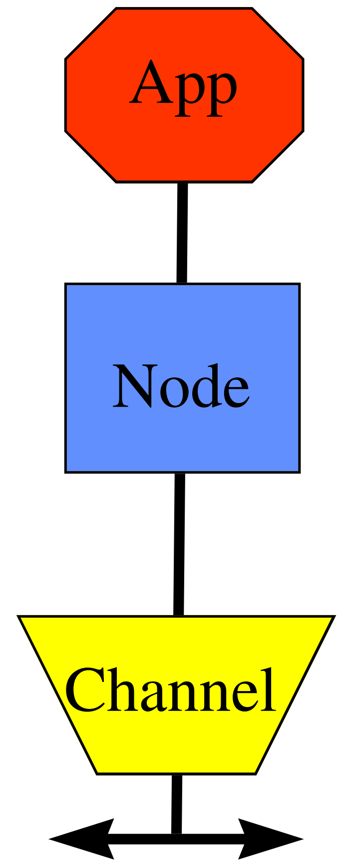
The Ants Toolkit

Components (Java classes)

Connect nodes and channels across machines to form a virtual network on top of existing physical network

Programming Model

Specialize capsule for network-based processing, byte codes are transferred



Status of Ants

Code

All Java, user-level prototype, targeted at Linux
Compact (~5000 lines) and flexible
Limited performance (>3ms to forward capsule)

Users

A few early adopters, hope to release mid-year
Mostly simple applications so far
Real applications (e.g, reliable multicast) underway

Understanding Performance

Key questions

- What does performance mean?
  - What comes in does not necessarily come out
  - Value throughput
- Impact of application-specific processing on performance?
  - On network and applications

What we have done

- Defined some metrics
- Defined models of active nodes
- Simulated one plausible application of fusion
- Simulated different reliable multicast protocols
- Instrumented prototype implementation

An Experiment

Application: audio mixing

Sending twelve 64Kb/s sources to one destination

Compare

Doing mixing at receiver

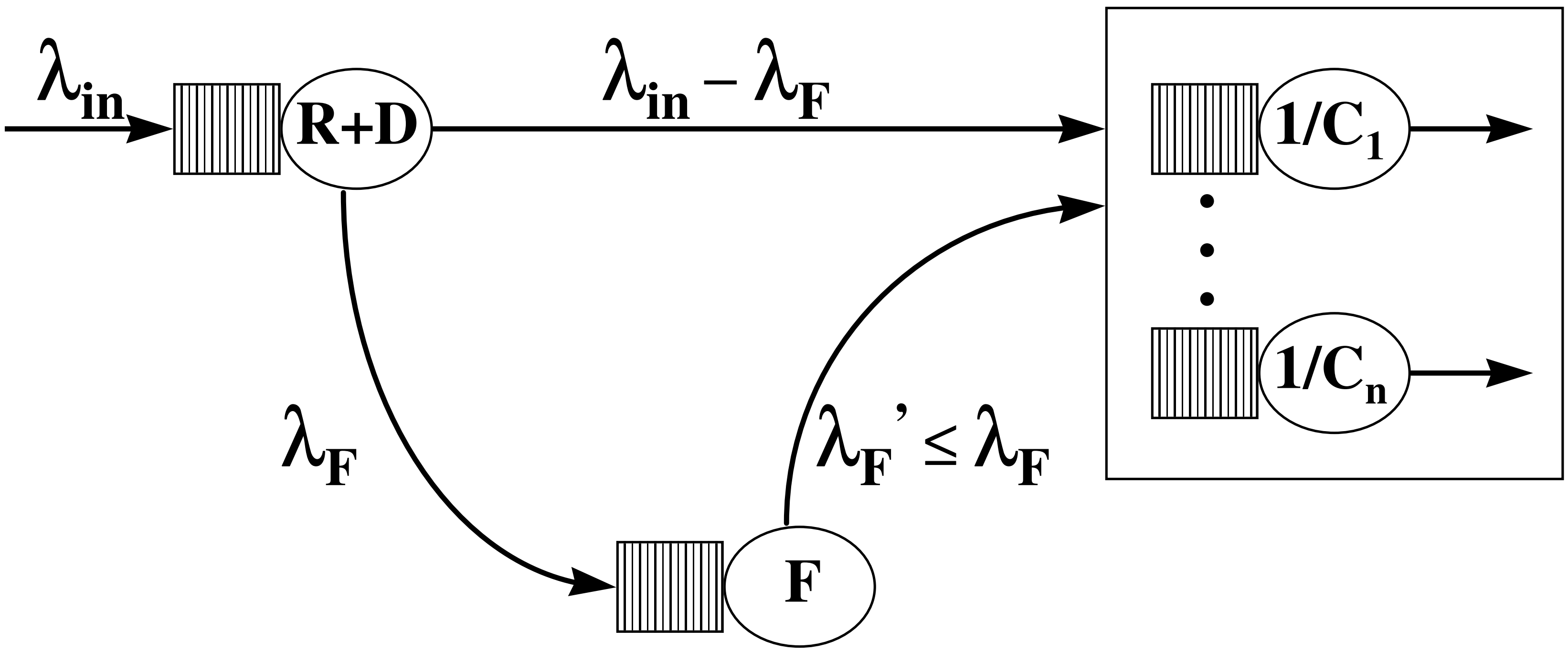
Doing mixing within network

Measure

Impact on mixing application

Impact on other network traffic

Node Model



Fusion server reduces traffic

- Combines up to 5 packets
- 500 μs timeout

Results of Experiment

Packet throughput went down almost 60%

Value throughput went up almost 23%

- Conservative number since queues unbounded

Utilization at last link reduced 66%

Latency a more complicated story

- Greatly reduced because of bottlenecks (>90%)
- If bottlenecks discounted
  - Latency of cross traffic slightly improved (17%)
  - Latency of fusion traffic dramatically increased (490%)
    - But absolute value is small
    - Why? Waiting, not processing

Conclusion

A good start

- Convinced that capsule approach can work
- Plenty of opportunities for performance enhancement

Utility still an open question

- Demand for protocol innovation
- Difficulty of protocol innovation

Next steps

- MIT: more small controlled experiments
- Community
  - Deploy a large and useful active net
    - Leverage existing infrastructure
  - Find and implement a driving application
    - Distributed simulation is my nomination