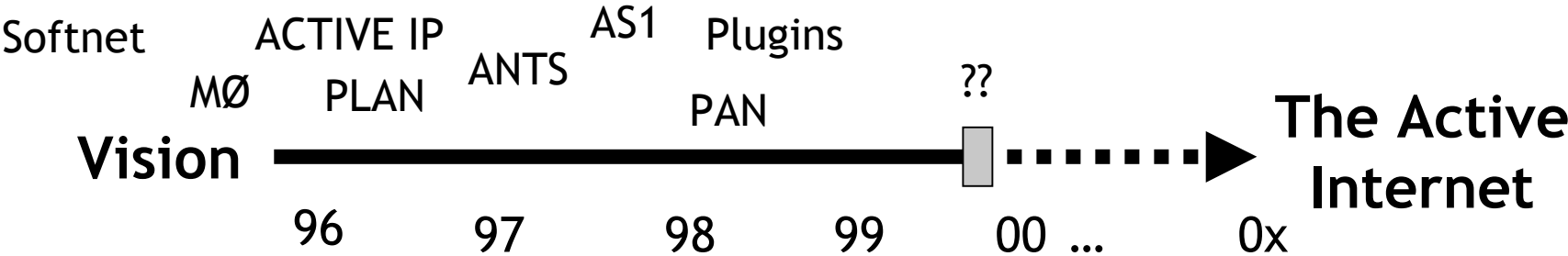


Towards an Active Network Architecture: An Epilogue

David Wetherall (djw@cs.washington.edu)
University of Washington
May 2002.

The State of Active Networks



Are we there yet?



Great! When can I use it?



This Talk

1. Vision
2. ANTS and lessons learned
3. Redux

1. Networks Need to be Flexible

- Applications are changing quickly
 - Internet increasingly diverse
- This affects the network
 - Web → caches, load balancing
- Yet network (IP) change is deathly slow
 - widespread consensus, manual deployment

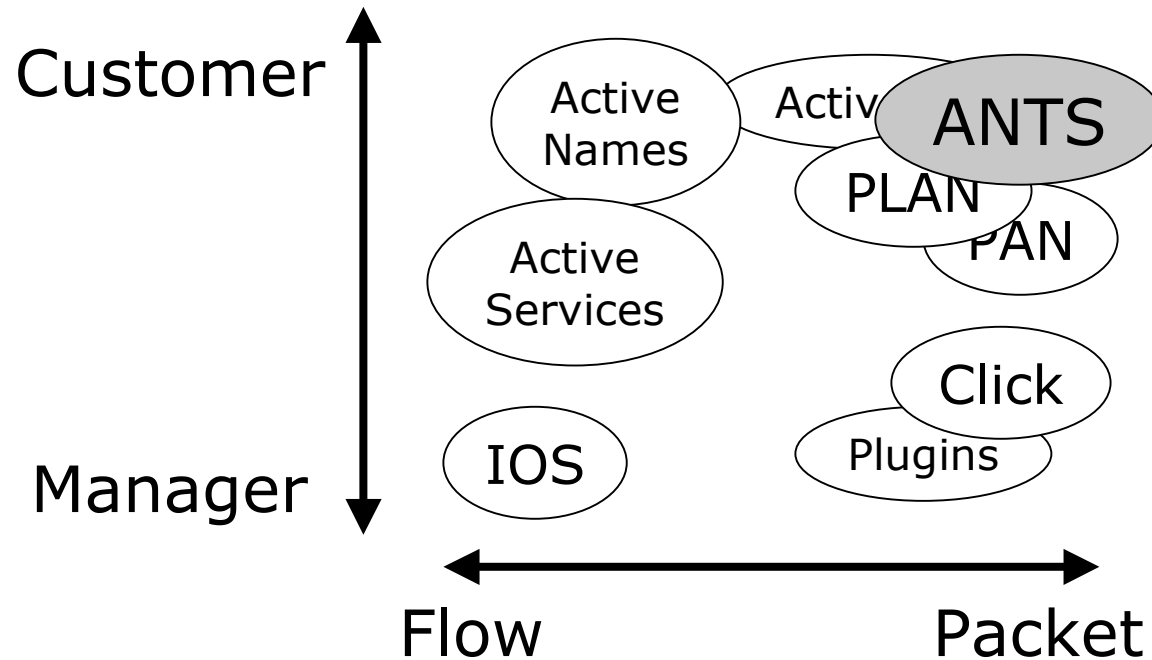


Active Networks

- Apply *mobile code* to:
 - Ease deployment; accelerate pace of innovation
- Network becomes extensible by its *users*
 - “Hollow shell” into which new services are loaded

“Towards an Active Network Architecture”,
– Tennenhouse and Wetherall, 1996.

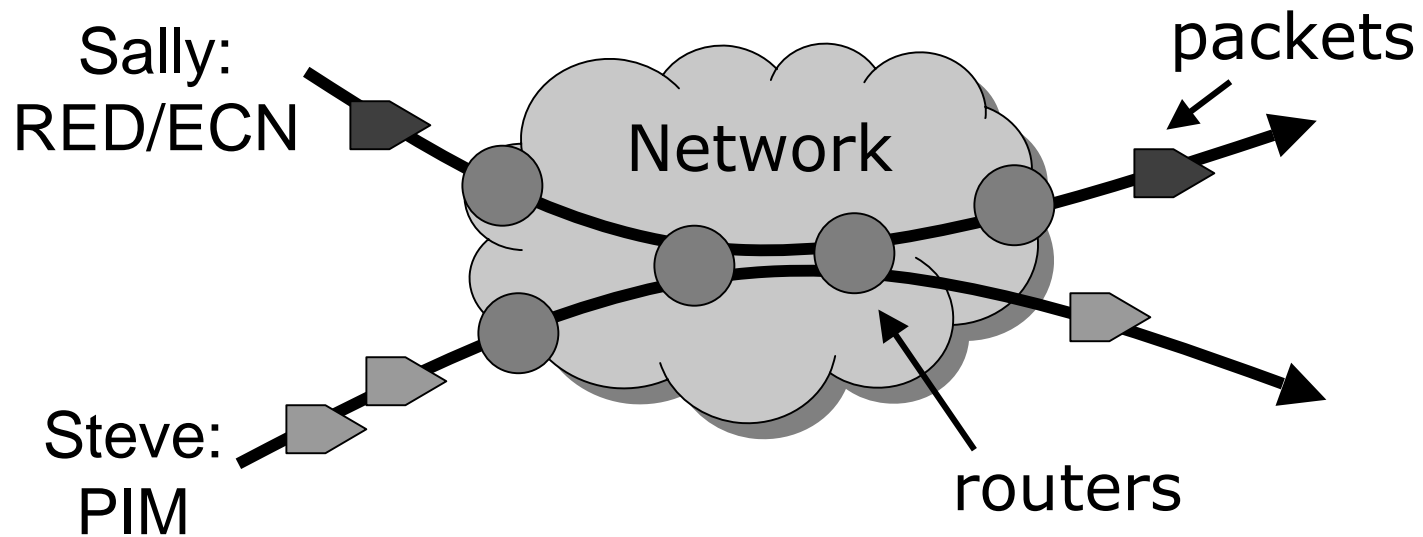
Kinds of Active Networks



ANTNS explores an aggressive vision

2. ANTS, Starting at 10,000 ft

Goal: Let each user control their packets



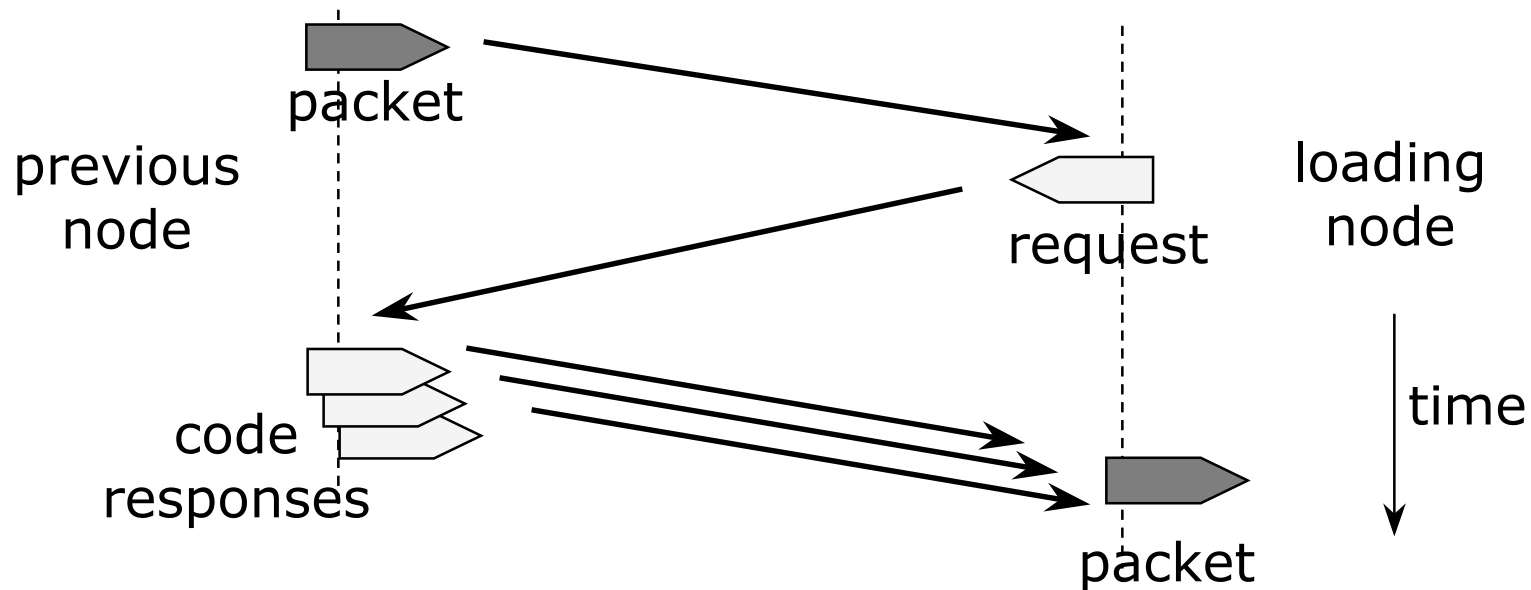
How do we build this?

Active Nodes Host New Services

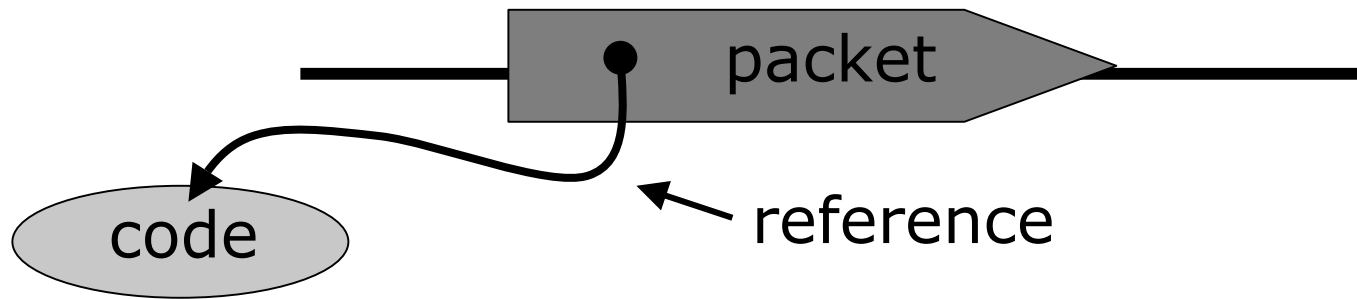
- Add active nodes to the infrastructure
 - Leave the rest as IP routers
- Nodes provide API for new service code
 - Soft-storage, routing, environment queries, packet manipulation
- Nodes run service code safely
 - Protect state at node; enforce packet invariants
- Nodes manage local resources
 - Bound code runtimes and other resources

Code Follows Packets As Needed

- Tie code distribution to packet forwarding
 - End-systems pre-load code, nodes load hop-by-hop as needed



Packets Refer To Code They Need



- Reference is based on fingerprint
 - Efficient: MD5 is 128 bits, quick to compute
 - Prevents code spoofing: verify without trust
 - No need for “standards body”: distributed naming

Implementation — ANTS Toolkit

- User-level reference platform
 - ~10,000 lines, 100% Java, UDP overlay
 - Nodes build on Java protection
- Publicly released since 1997
 - <http://www.cs.washington.edu/research/networking/ants/>
 - Used at MIT, Utah, TIS, TASC, SRI, UIUC, UCLA, ...
 - ANTS2.0 is the latest release, lead by Utah

Some ANTS Lessons Learned

Interested in three dimensions:

- a. Applications
- b. Performance
- c. Security and Resource Management

3a. Application Lessons

Q: Can we deploy important services?

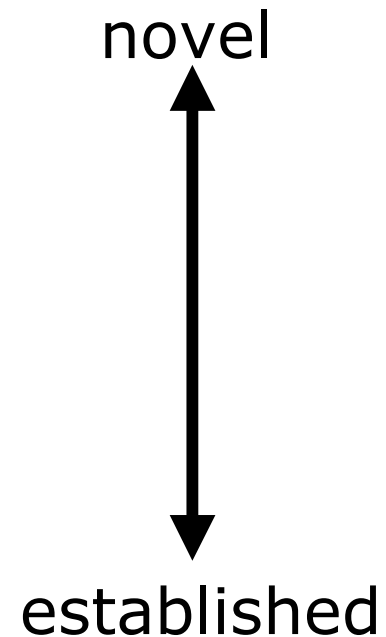
- Chosen approach lets users select independently
- Programs constrained by active nodes

A: Yes.

- Well-suited to experimenting w/ protocol variations (rather than computation pushed into network)
- Exceptions: enforcing policy at a point (firewalls) and resource control (guaranteed service)

Services We've Written

- Auctions
- Web cache diversion routing
- “TCP-SYN” filtering
- Reliable Multicast support
- Multicast (single source, PIM)
- Mobility
- Path MTU discovery



3b. Performance Lessons

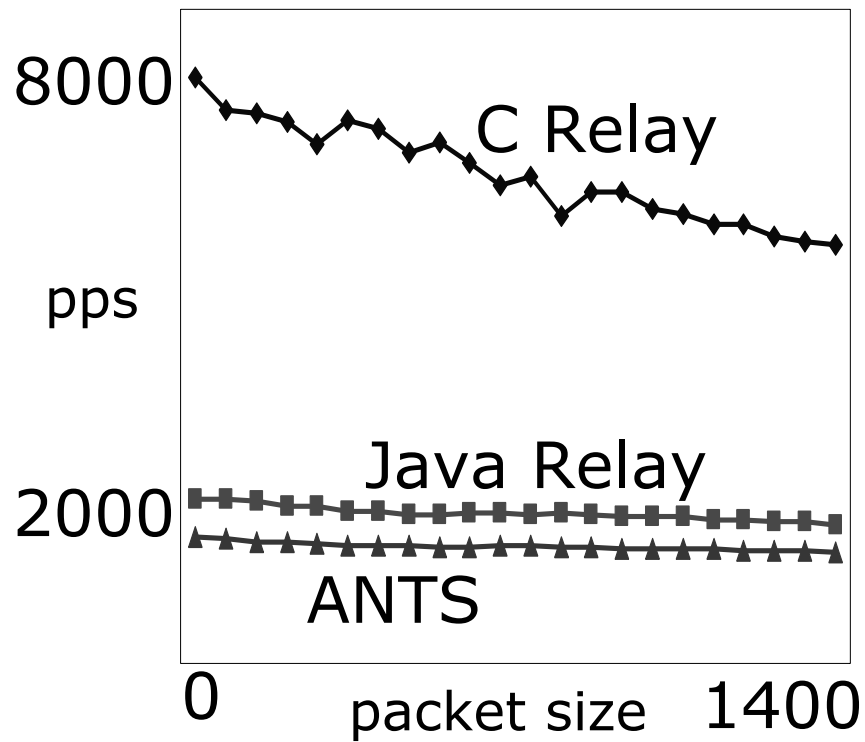
Q: Is performance a show-stopper?

- Model is more expensive than IP

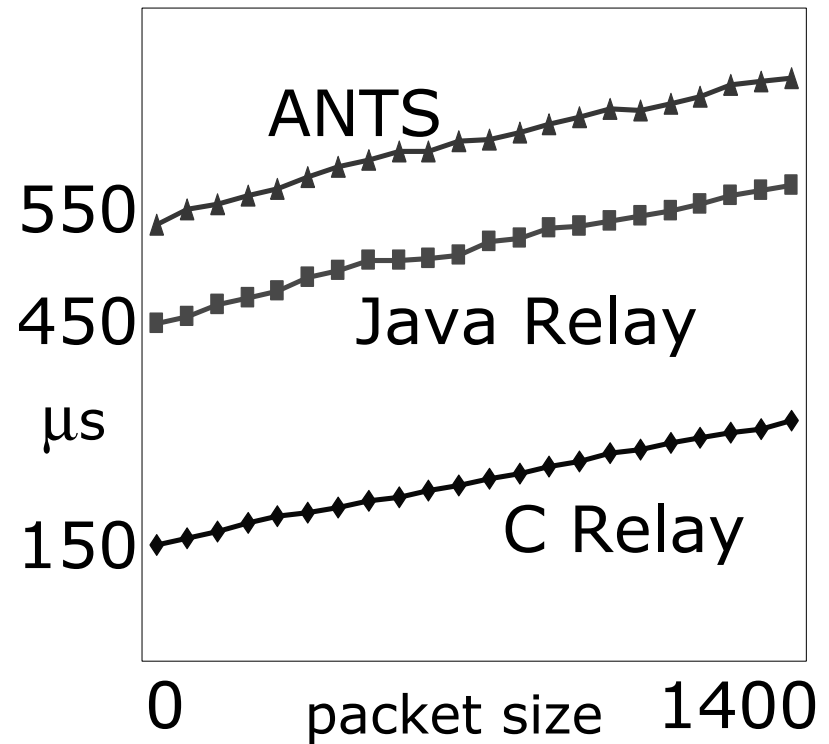
A: No.

- Very few extra steps over IP in common case and extra steps (demux, safe eval) known to run fast
- Balance available vs. required computation
 - e.g. 1GHz, 1Gbps, 1000b, 100% → 1000 cycles

Measured Performance



Throughput



Latency

3c. Security and Resource Mgmt.

Q: Can untrusted users program the network?

- Need to isolate services and protect network

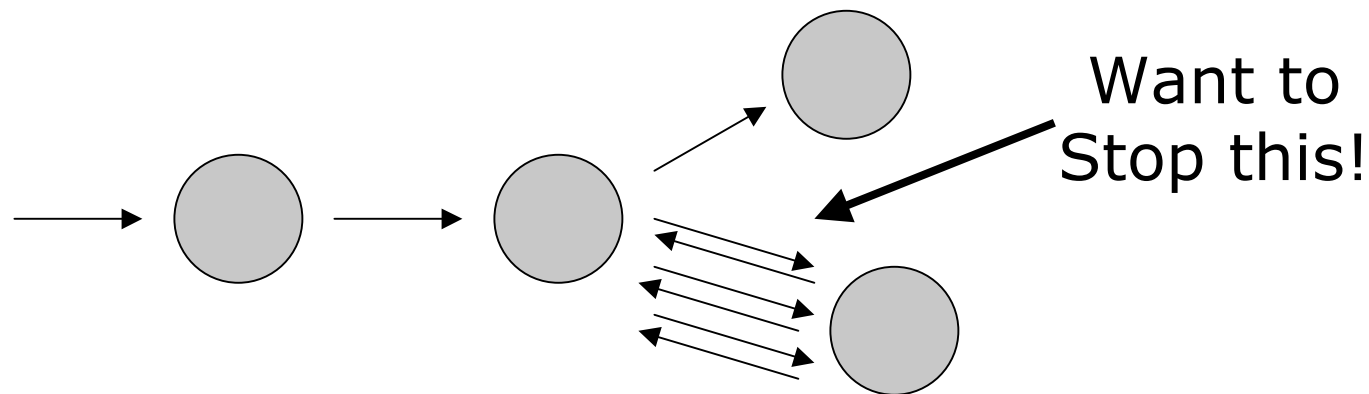
A: Partly. This is difficult!

- Requires Protection
 - My program can't corrupt your program → Solved
- Requires Resource Management
 - My program can't starve yours → Not solved

Handling of Resource Mgmt Tasks

	Internet	ANTS
Per Packet At A Router	Limited by design of IP	Limited for simple model
Per Packet Along Path	Limited by protocol design	Whoops ... hole here
Per User	Need FQ or penalty boxes	Same as Internet

Example of Buggy Multicast



- How should we prevent this?
 - TTLs are a weak solution; not related to topology
 - Fairness mechanisms mitigate, but not enough
 - ANTS falls back on certification of programs ...

Conclusions

- There are important applications
- Performance is not the limiting factor
- Protection is not the limiting factor
- Resource Management remains problematic

3. Active Networks Redux

- *If we were to ask the active network question again today, six years later, what would the answer be?*
- **Broad Observations**
 - The thrust of computation/services within the network is clearly a winning direction and still has legs ...
 - Active networks has become defined by mechanism rather than goals, and we need to change this!
 - There is much low-hanging fruit in-between the extremes that is of value in terms of gaining application experience

The emerging role of overlays

- Early MBONE/6BONE
- XBONE
- End-system multicast
- Chord/CAN/Pastry/Tapestry, for multicast too.
- Self-organizing overlays
- overQOS

Router building blocks

- I3
- Lightweight network services

Active end systems

- Extensible TCP
- Extensible network measurement

Being less Internet centric

- Wireless networks
- Ad hoc networks
- Sensor networks

The role of parties and incentives

- We are mostly oblivious of these factors
 - E.g., multicast
- Case studies

