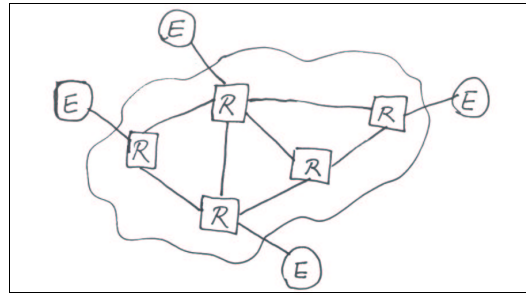


CS 518

Active Networks: Decoupling Infrastructure from Applications

Randy Wang
Fall 2002
Princeton University

AN and ANTS Goals



- Active Networking goal
 - Overcome slow vendor standardization
 - Decouple infrastructure from application intelligence
 - (there is still standardization, but at a lower level)
 - Foster innovation
- ANTS (extremes)
 - Allow **anyone** to inject code
 - Execute code **per packet**
 - Do it on the scale of the entire **Internet**

CS518

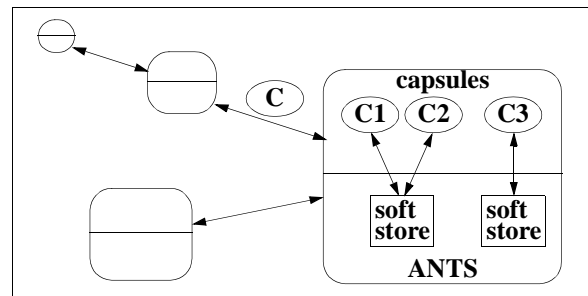
1

Randy Wang

Code Injection

- An alternative (“discrete”) approach:
 - Distribute code
 - Packets carry code pointers
 - Routers execute code
- Capsule approach:
 - Each packet carries its own code
- ANTS: a mixture of two
 - Separate code distribution mechanism
 - Packets carry checksums of codes as pointers
 - Codes cached on nodes
 - Packets dropped if codes not found
- Not clear if the extra flexibility is really necessary

Node Interface



- Node API controls all externally visible actions
- Manipulates a soft store
- Route the capsule to another node

CS518

2

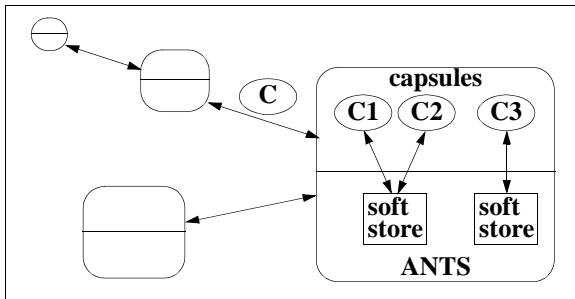
Randy Wang

CS518

3

Randy Wang

Protection



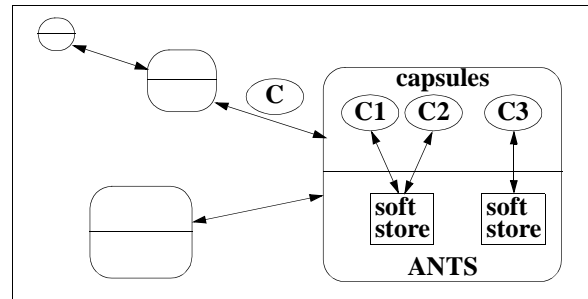
- Protect ANTS from capsules
 - Java, PCC, SFI
- Ensure the correct code that the capsules desire gets executed
 - Use code finger prints as names of codes
- Protect soft stores from being corrupted by “other” capsules
 - Soft stores tagged with finger prints of “services”

CS518

4

Randy Wang

Protection (cont.)



- What’s not included in ANTS:
 - Does not allow a “super-service” to manipulate capsules belonging to other (multiple) services
 - ANTS is self-contained: needs no external authentication mechanism
 - What prevents me from injecting an arbitrary capsule (of a certain finger print) and having it executed? (... IP analogy)
 - Application-specific protection

CS518

5

Randy Wang

Resource Management

	Internet	ANTS
Per-node	✓	✓
Network-wide	✓	✗
# of Packets	✗	✗

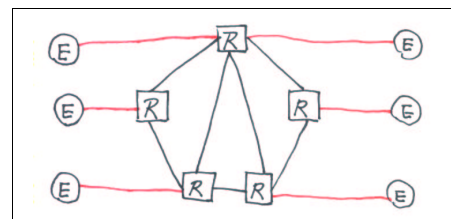
- Falls back to code-certification mechanism
- Falls short of the “anyone” goal

CS518

6

Randy Wang

Alternative Model



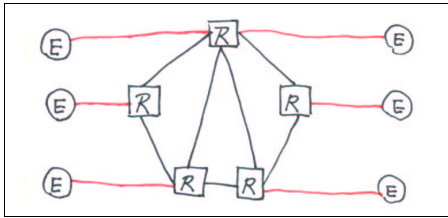
- ANTS: allow **anyone** to inject code, execute code **per packet**, do it on the scale of the entire **Internet**
- Alternative (more practical?) model
 - Distinction of service “providers” vs. service “users”
 - A conventional “user” interface
 - A Node API available to a smaller class of service “providers” who are allowed to customize the infrastructure
 - Discrete injection of intelligence by providers

CS518

7

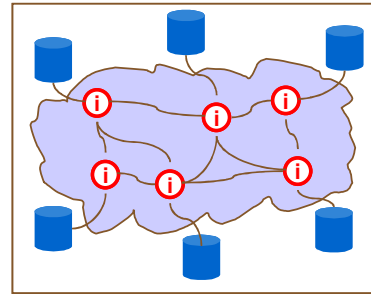
Randy Wang

Alternative Model (cont.)



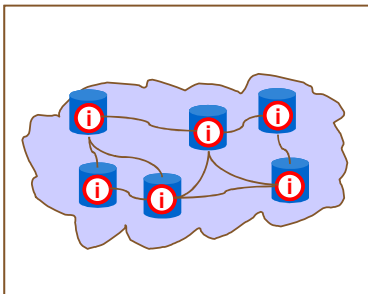
- Provider components authenticate each other
- Resource accounting based on provider identities
- Providers implement their own application-specific user authentication
- A secure “backplane” connecting the provider components (that is physically or virtually distinct from a user-accessible network)
- Even more practical in smaller-scale networks and applications
- Not even necessary to time-share

Applications



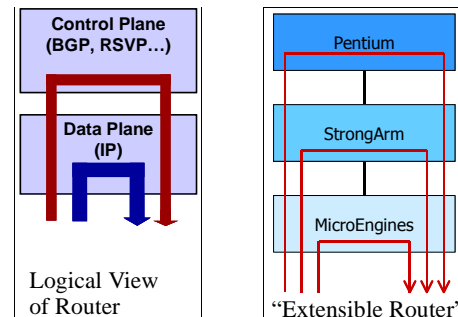
- Avoids tackling storage
- Intelligence restricted to forwarding decisions
- Wants to turn the network into the computer
- But the network can't be the computer without storage in it

Alternative Model



- Storage as “first-class” citizen of the node API
- Embedded storage can be the only storage
- Qualitatively different intelligence, applications, and research challenges

Performance Concern



- Careful separation of “fast paths” from “slow paths”
- Applies across different services
- Applies within each service