NSF DRAFT PROPOSAL

Protecting Networks with COPS:
Making Networks More Robust by Checking, Observing, and Protecting Services

Randy H. Katz, Scott Shenker, Ion Stoica

Computer Science Division

Electrical Engineering and Computer Science Department

University of California, Berkeley

Berkeley, CA 94720-1776

Session Goals and Objectives

- We need your thoughtful feedback!
- Present basic concepts and general work plan of our Draft NSF Proposal
- Will distribute the draft to you; due @ NSF 21 January!
  - Treat as "Berkeley Confidential" and "For Your Eyes Only"—do not distribute the draft to your colleagues without asking us first!
  - These slides are ok to share
- Please read draft over next two days: special proposal feedback session W AM 0830-1000
  - If leaving early, please email us with comments AS SOON AS POSSIBLE

Observations and Motivations

- Internet reasonably robust to point problems like link and router failures ("fail stop")
- Successfully operates under a wide range of loading conditions and over diverse technologies
- During 9/11/01, Internet worked reasonable well, under heavy traffic conditions and with some major facilities failures in Lower Manhattan

Observations and Motivations

- But …
  - Single misconfigured border router able to bring the Internet to its knees (1997)
  - Worm outbreaks (e.g., Code Red, Nimda, Slammer) cause wide-spread havoc, generating BGP session resets mostly affecting the lower levels of the AS hierarchy
  - Campus IS&T tells us latest worms & file sharing apps cause traffic surges rendering campus network unmanageable due to control plane starvation (Spring/Summer 2004)
  - Berkeley EECS network loses ability to mount file systems and render other network services under suspected DNS DoS attack (December 2004)
  - No way to distinguished between "semantically" malformed traffic and that which is syntactically correct
  - Extremely hard to understand why network services fail, poor tools for post mortem analysis

Why and How Networks Fail

- Existing work focuses on loss of reachability due to routing anomalies & dynamics (e.g., convergence)
- Recent work investigates effect on *wide-area* routing infrastructure of surges caused by worm-induced and DoS traffic
  - BGP session resets a bigger problem for edge networks than peered ISPs
  - "Background radiation": random port scans/malformed traffic rapidly becoming the dominant traffic reaching end networks!
- Left unaddressed: effect of surges on critical network services, e.g., DNS, DHCP, FS mounts, network storage services, web services, etc.

Why and How Networks Fail (continued)

- Complex phenomenology of failure
- Recent Berkeley experience suggests that traffic surges also render enterprise networks unusable
- Indirect effects of DoS traffic on network infrastructure: role of unexpected traffic patterns
  - Cisco Express Forwarding: random IP addresses flood route cache forcing all traffic to go through router slow path—high CPU utilization yields inability to manage router table updates
  - Route Summarization: powerful misconfigured peer overwhelms weaker peer with too many router table entries
  - SNMP DoS attack: overwhelm SNMP ports on routers
  - DNS attack: response-response loops in DNS queries generate traffic overload

Network Trends

- Tightly managed enterprises
  - "Lock down" network with highly restricted access rules from the outside
  - Strong policies about the kind of machines that can be connected within the network
  - We are not focused on such networks
- Open enterprises
  - Require a degree of access from outside the enterprise
  - Universities, Research Laboratories, Grid computing communities, …
  - "Virtual Corporations" collaborating on products and services
  - Balancing need for protection with openness is an essential motivation for our proposal

Technology Trends

- PNEs (aka Middleboxes)
  - Love them or hate them, they are proliferating
    - » NATs, firewalls, server load balancers, IDS, …
  - New generation emerging that will be more programmable
    - » E.g., Bivio Networks
  - New "Data Center in a Box" architectures: processing, storage, networking in blade centers
  - Issue:
    - » Aggressively use these for deep packet inspection and actions including rewriting packet actions

    OR

    - » Explore approaches which do not radically disturb protocol layering

COPS

Checking
Observing
Protecting
Services

Conceptual Architecture Component 1: "Check"

- Checkable Protocols: "Fix" Internet with new protocols that maintain invariants and techniques for checking/enforcing them
  - This is hard, but we have some experience:
    - » Listen & Whisper: well-formed BGP behavior
    - » Traffic Rate Control: Self-Verifiable Core Stateless Fair Queuing (SV-CSFQ)
    - » Other examples in the proposal
  - Existing work requires changes to protocol end points or routers on the path
    - » Way forward for new protocols, but difficult to retrofit checkability to existing protocols
    - » Leveraged Building Blocks:
      - Observable protocol behavior
      - Cryptographic techniques
      - Statistical methods

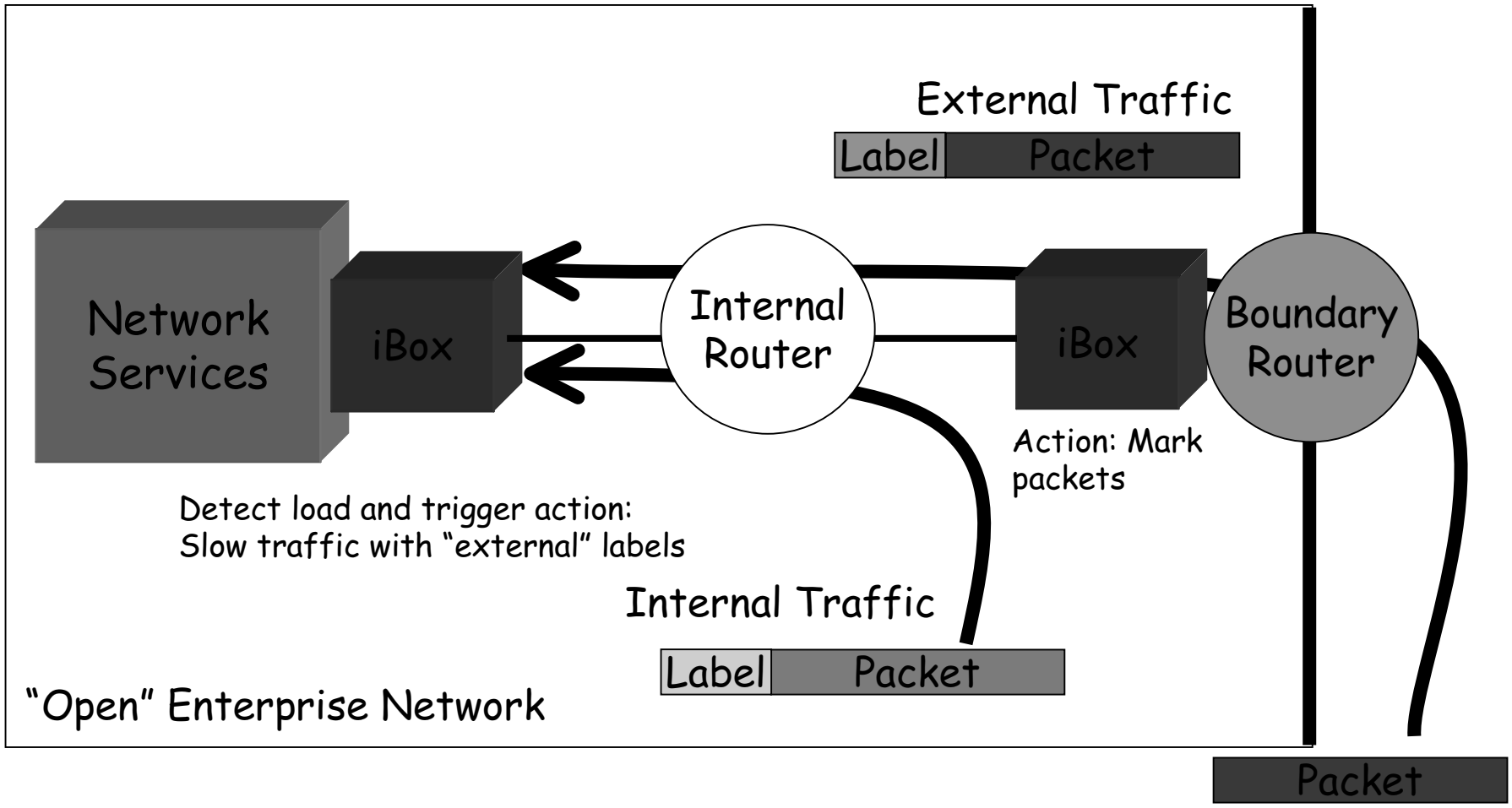
Conceptual Architecture Component 2: "Protect"

- Protect Crucial Services
  - Pragmatic Goal: minimize & mitigate effects of attacks & traffic surges
  - Distinguish between good, bad, ugly (suspicious) traffic
    - » Bad evolves much faster than good, and is harder characterize
    - » Good determined by long-standing patterns and operator-tunable policies
  - Filter the bad, slow the suspicious, maintain resources for the good (e.g., control traffic)
    - » Sufficient to reduce false positives
    - » Some suspicious-looking good traffic may be slowed down, but won't be blocked

Conceptual Architecture Component 3: "Observe"

- Observation (and Action) Points
  - Points within the network where control is exercised
    - » Traffic classified
    - » Resource allocation enforced
  - Extend Internet Architecture
    - » Routers + End Hosts + *Inspection-and-Action Boxes* (aka *iBoxes*)
    - » iBoxes prototyped on commercial PNEs
    - » Placed at Internet and Server edges of enterprise net
      - Single administrative environment
      - Not a core network technology
    - » Transparently cascaded with existing routers to extend their functionality
      - Place to retrofit checkability with already deployed services and routers

iBox Placement and Functionality



Check

- How far can you go with Whisper-like techniques?
- Can checkability be applied in protocol domains other than congestion control and routing?
- How can we exploit iBoxes to incrementally deploy checkable protocols?
- How far can you go with locally observable invariants? How to check for global properties?

Network Crash Recorder

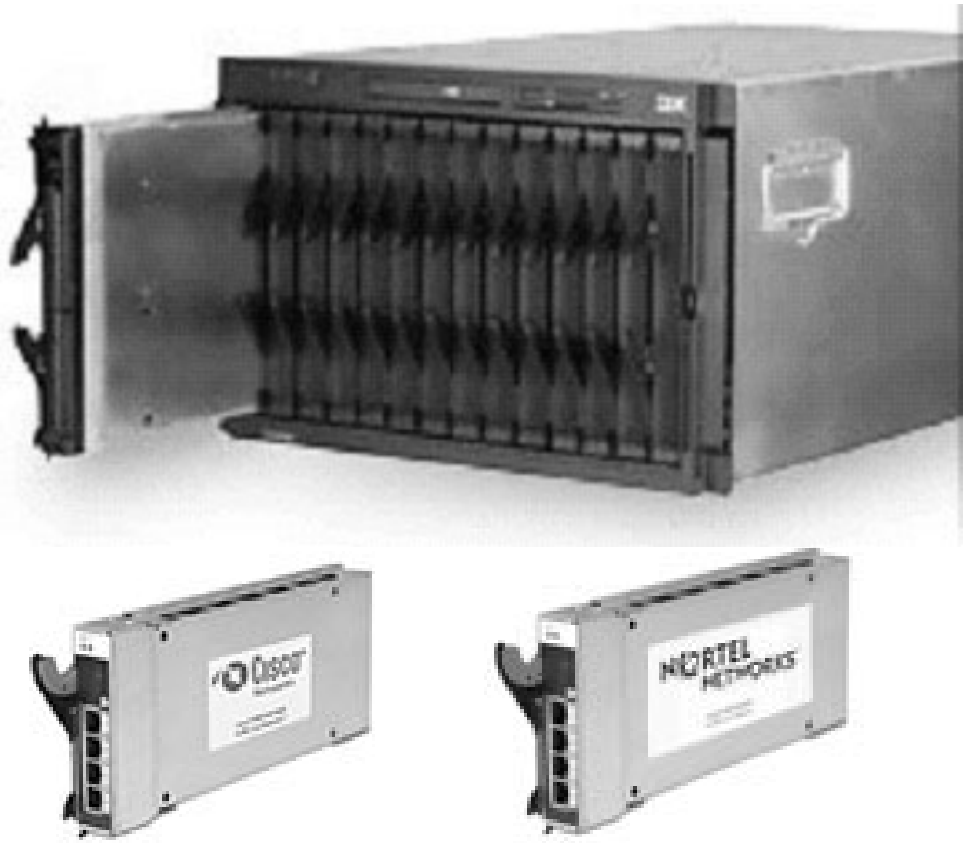
- Record and save network activity just before a crash for later analysis
- Many issues:
  - Just how do you detect a crash?
    - » Fail stop variety are easy (e.g., router crash)
    - » What about cascaded failures induced by certain kinds of traffic patterns?
  - How do you correlate logged activity from multiple observation points across the network?
    - » Focusing on enterprise networks makes this more tractable than the full-scale Internet
    - » Some experience in terms of of tools for DHT debugging (talk tomorrow)
  - Great challenge application for iBoxes!

Constructive Approach

- Network reliability benchmarks to better understand how networks fail plus signature of impending failure
  - Network Crash Recorder based on cooperating iBoxes to snapshot recent network state preceding a network service failure
- Architectural elements for raising the semantic level of the Internet
  - Design of checkable protocols
    - » Building blocks for enabling invariant checking
  - Design of iBoxes
    - » Observation and action operations to implement protection of network services

Observe and Protect

IBM @server BladeCenter



bivio networks

iBoxes implemented on commercial PNEs

- Don't: route or implement (full) protocol stacks
- Do: protect routers and shield network services
  - » Classify packets
  - » Extract flows
  - » Redirect traffic
  - » Log, count, collect stats
  - » Filter/shape traffic

Observe and Protect

- Other NEs do some of these things (e.g., Packeteer), but …
  - iBoxes are fully programmable by us
    - » Essential element of our agenda is understanding how to structure the programming environment for PNEs to ease implementation of iBox functionality
  - Don't require 100% successful classification: degree of freedom in distinguishing between good vs. bad vs. ugly
  - Learning algorithms: potentially discover new good traffic over time
  - Directly support newly designed checkable protocols
  - Focus on *protecting* network services, not performance per se
    - » Problems we are interested in cannot be solved simply by managing bandwidth better
    - » Integrate iBoxes with rest of the COPS approach

Expected Contributions

- Design, implementation, assessment of checkable protocols
- COPS framework: Check-Observe-Protect to simultaneously enable open enterprises while also protecting their critical network resources
- Evaluation-Design-Prototyping Methodology

- If successful, Internet protocols evolve to become increasingly more checkable plus iBox functionality migrates into future generations of routers

What We are Not Doing

- Building new PNE hardware
  - Though classification boosting algorithms may be of interest to hardware designers
- Making the wide-area network more reliability
  - Though checkable protocol technology may help
- General problem of containing worms and other malware
  - Though detecting traffic surges and protecting network services against them may help