How Healthy are Todayś Enterprise Networks?

Saikat Guha*, Jaideep Chandrashekar†, Nina Taft†, Konstantina Papagiannaki†

*Cornell University, and †Intel Research

IMC 2008

Enterprise networks are noisy

- Wide range of applications and infrastructure services
- Heavy policy from IT
  - Anti-virus, Software patches, App blacklist
- Traffic should be well behaved. But it is not.
- High levels of noise
  - Spurious flows, unknown destinations, mysterious failures
- IT: "If it ain't broke, don't fix it."

Enterprise networks are noisy

Embracing noise

- Complicates analysis
  - Anomaly detection harder
  - Increased security concerns
- Increases costs
  - Processing and memory overheads
  - Power consumption, transmission costs

Enterprise Networks

Embracing high levels of noise is short-sighted. We attempt to quantify this noise and associate causes with it.

Network Health: A Metric for Noise

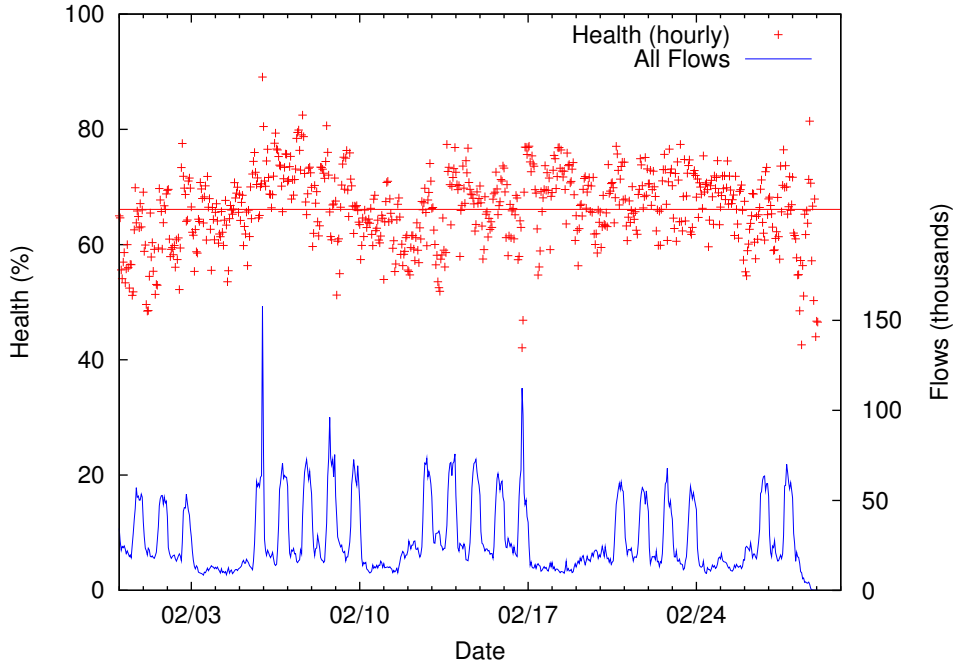
- Fraction of flows that are useful
- Useful flow
  - One that successfully contacts the intended destination
- Non-useful flow
  - e.g. timeouts, unreachable destinations, flows explicitly refused

Key Findings

- Endhost perspective is crucial for eliminating noise
  - Network-crossing effects are significant
- Lack of environment awareness primary cause of noise
- Manifested as:
  - Persistent application-level retries
  - Ad hoc self-(re)configuration
- Not too hard to fix
  - Few bad (but popular) applications
  - Short-term: exponential backoff for retries
  - Long-term: network level environment awareness service

Methodology

- Captured all network traffic at the endhost
  - Enterprise-internal, VPN, home or foreign networks
  - Traffic in response to environment change
- Flows summarized by BRO
- 357 users (95% mobile), Feb '07, 31M flows
- Overall Health: 66%



Lack of Environmental Awareness

- Many means and points of connections
  - Enterprise LAN, Wi-Fi, VPN
  - Cellular, Starbucks, Home network
- Different IP address and reachability
  - 77% failures within 1 minute of acquiring new IP
  - usually to hosts successfully contacted 8min earlier
- Many anomaly detectors treat failed flows as suspicious
  - Recommendation: Ignore failures for first few minutes after node joins network
- Blind probing going from enterprise to outside
  - A security hazard (see paper)

Failure Taxonomy

1. Persistent Retries (>54%)
   - App keeps hammering server with new flows while server is down/unreachable
   - Fix: App-level exponential backoff for retry flows
2. Service Discovery (>48%)
   - Apps individually probe to self-(re)configure
   - Fix: Amortize effort
3. Vulnerability Testing (4.8%)
   - Designated enterprise host scans endhosts
   - Lesson: Accept as "useful failures"

Persistent Retries

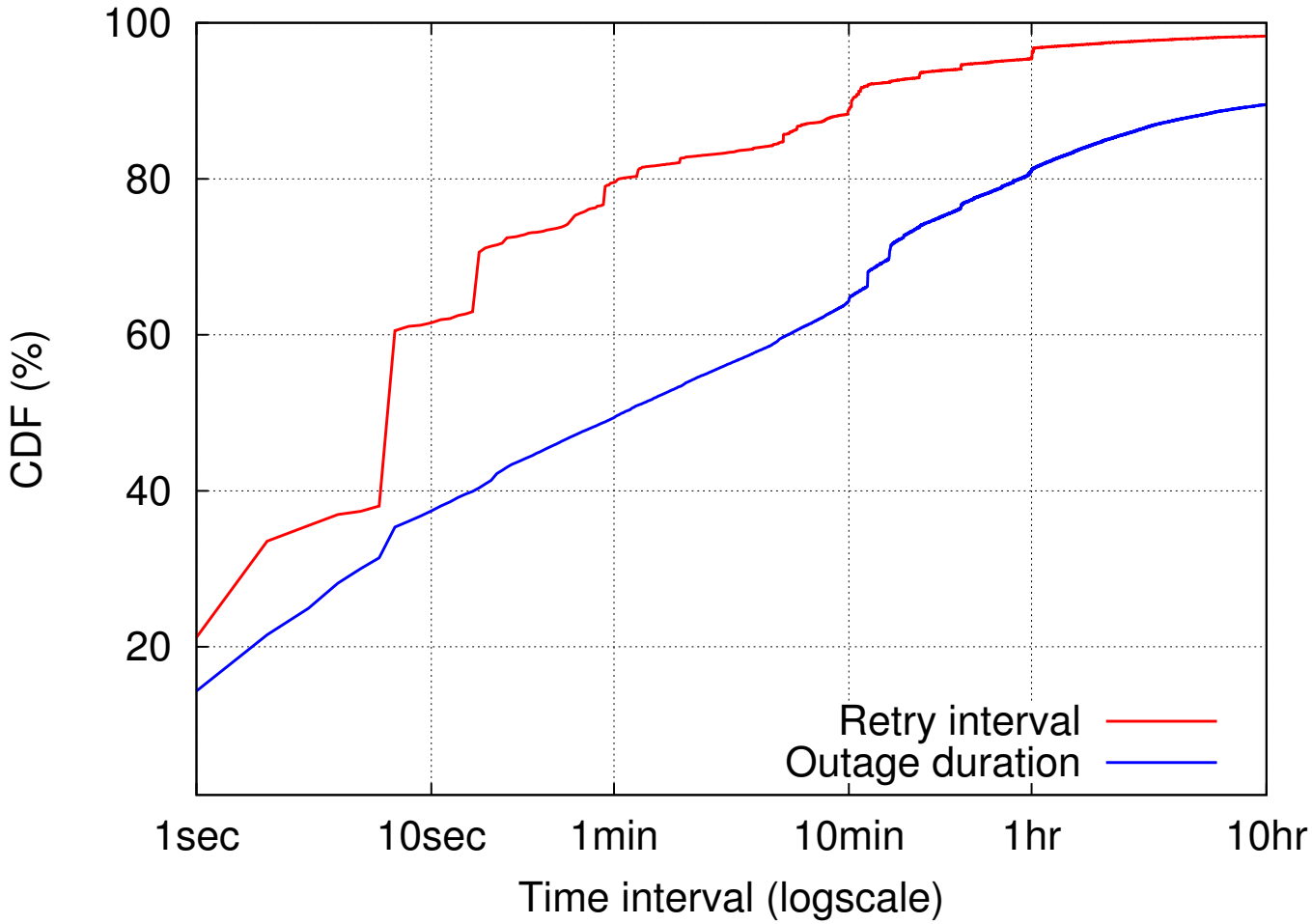


Figure: Applications retry far more frequently than necessary

Service Discovery

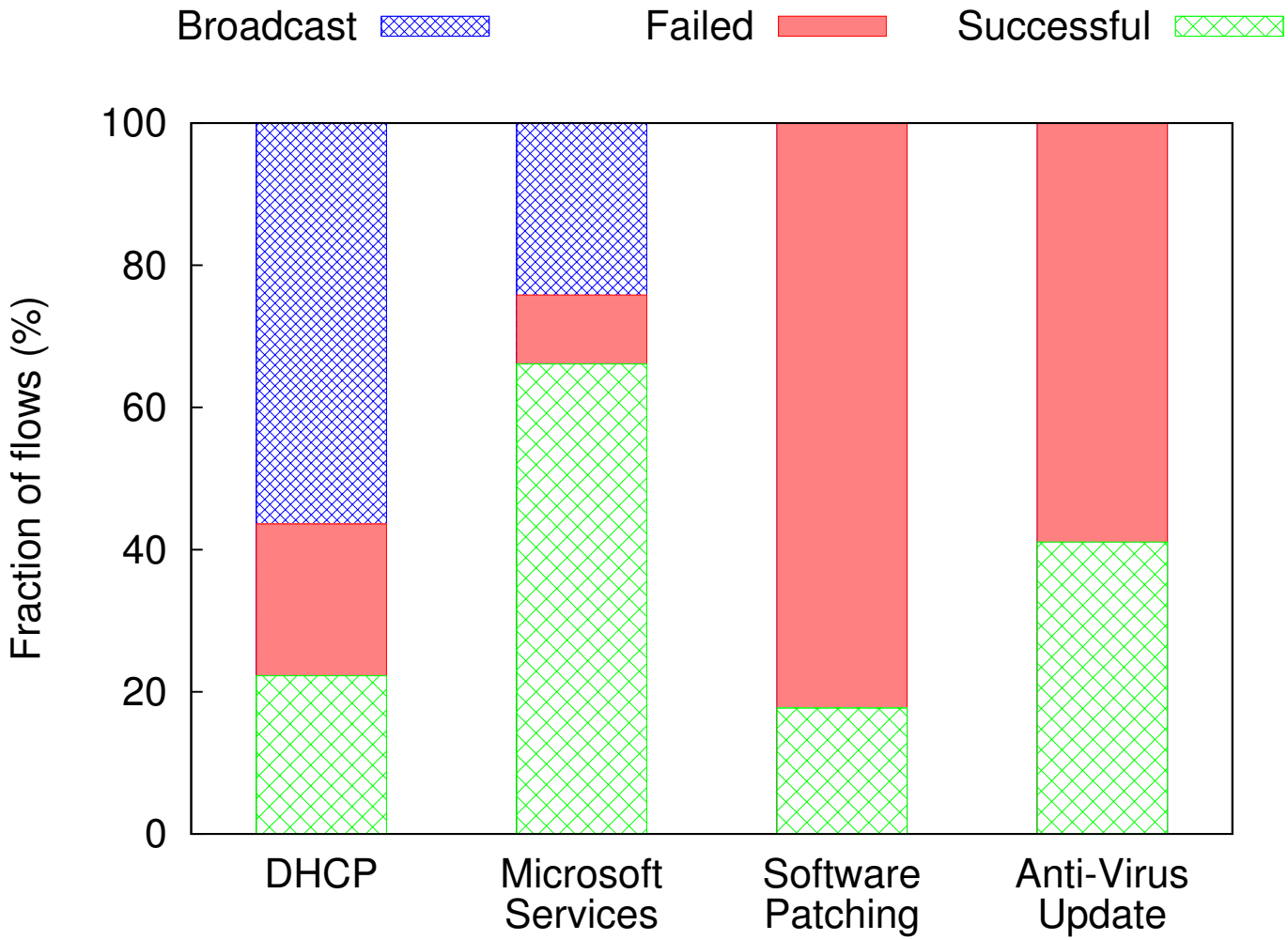
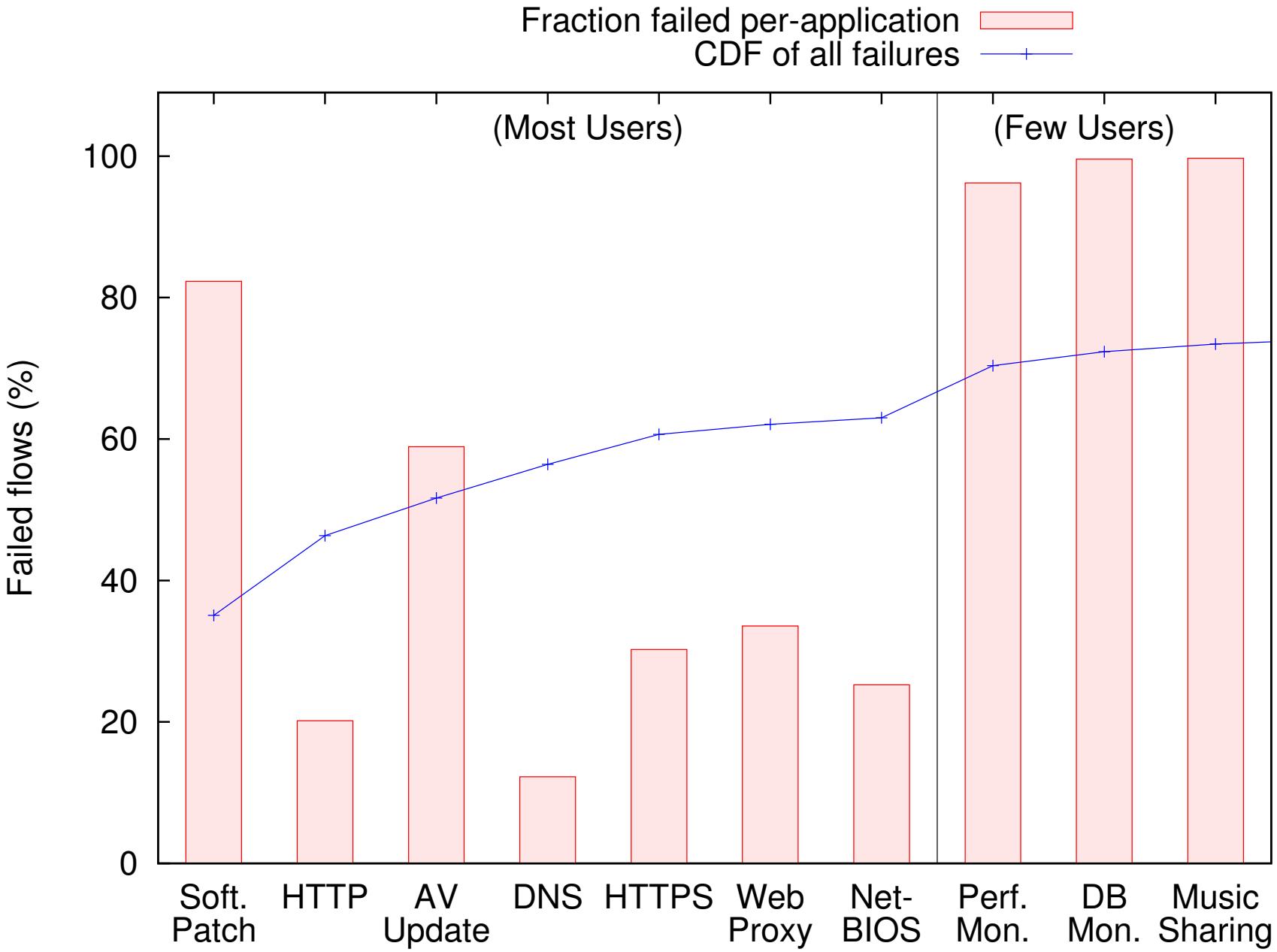


Figure: Applications duplicate discovery and self-configuration in ad hoc ways

Culprits


Fraction failed per-application
CDF of all failures
(Most Users)
(Few Users)
Failed flows (%)
0
20
40
60
80
100
Soft. Patch
HTTP
AV Update
DNS
HTTPS
Web Proxy
Net-BIOS
Perf. Mon.
DB Mon.
Music Sharing

Summary

- Health as fraction of useful flows
- Understanding requires endhost perspective
- Lack of environmental awareness is a problem
  - Grave security implications
- Simple short term fix to few apps
- Need architectural support for environmental awareness in the long term

Security Leak

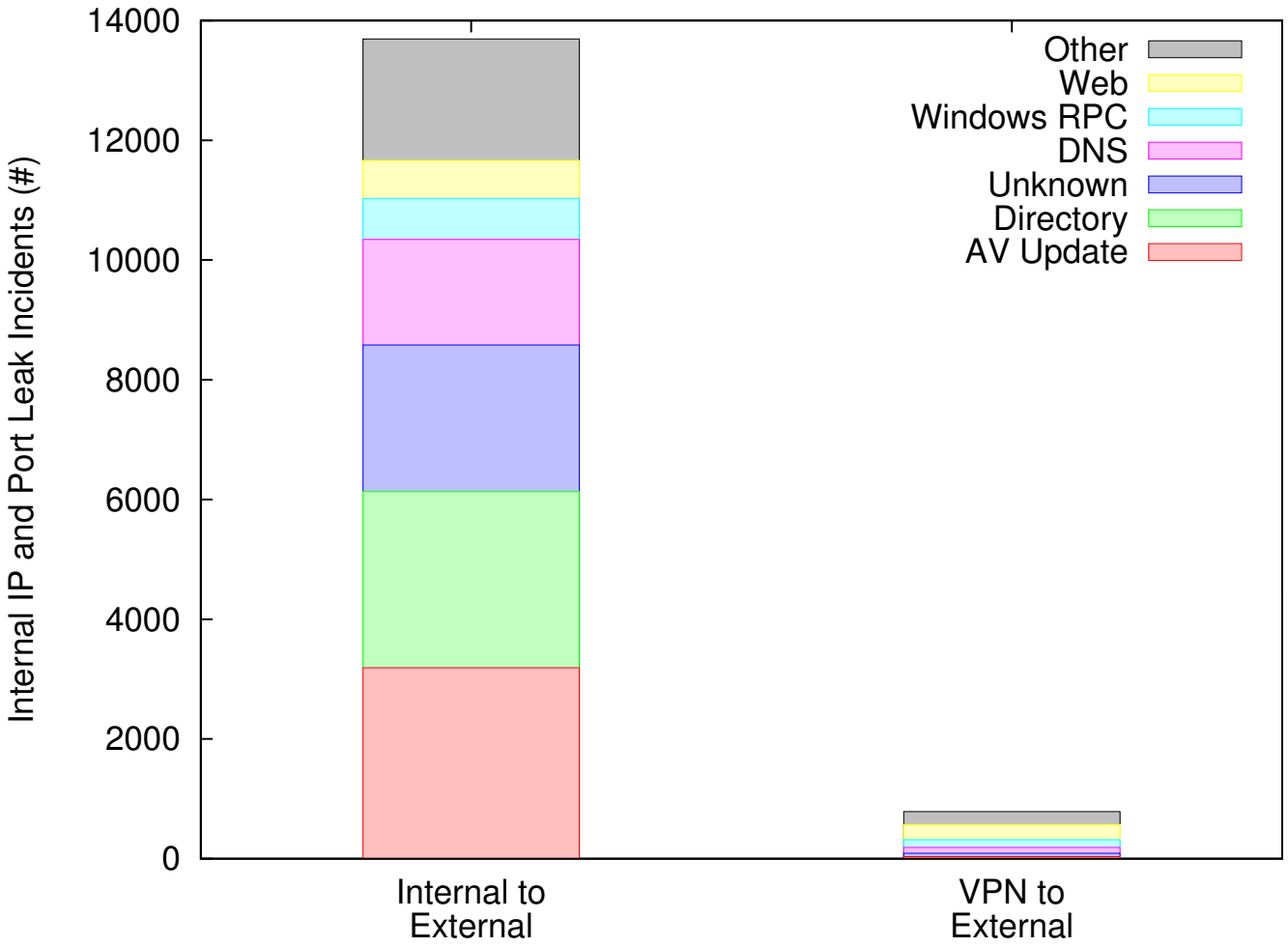


Figure: Applications leak sensitive information when transitioning from enterprise networks to foreign networks