The Case for Network-Layer, Peer-to-Peer Anonymization

Michael J. Freedman

Emil Sit, Josh Cates, Robert Morris

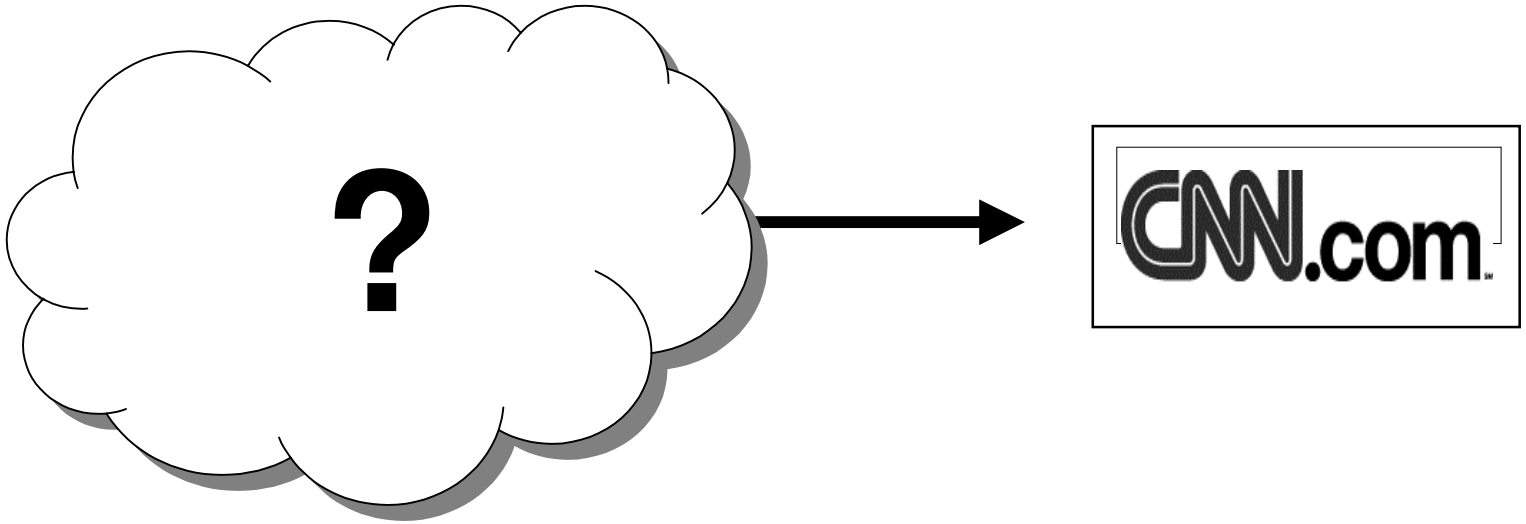
MIT Lab for Computer Science

IPTPS'02 March 7, 2002

`http://pdos.lcs.mit.edu/tarzan/`

The Grail of Anonymization

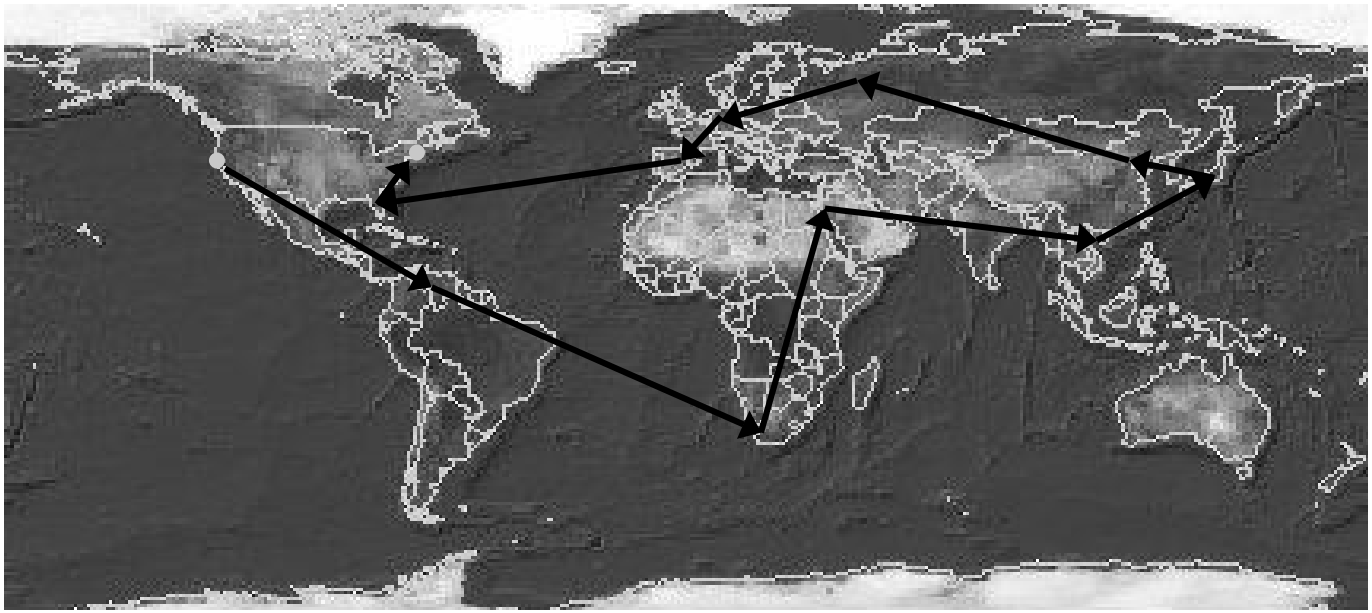
- Participant can communicate anonymously with non-participant



- User can talk to CNN.com
- Nobody knows who user is

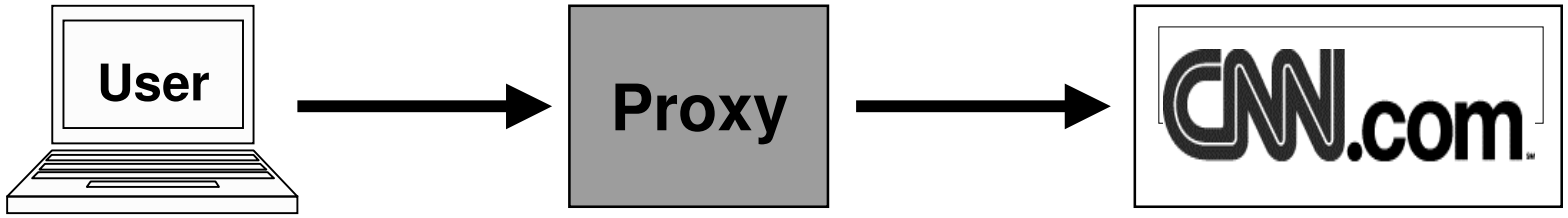
Our Vision for Anonymization

- Millions of nodes participate
- Bounce traffic off one another



- Mechanism to organize nodes: peer-to-peer
- All applications can use: IP layer

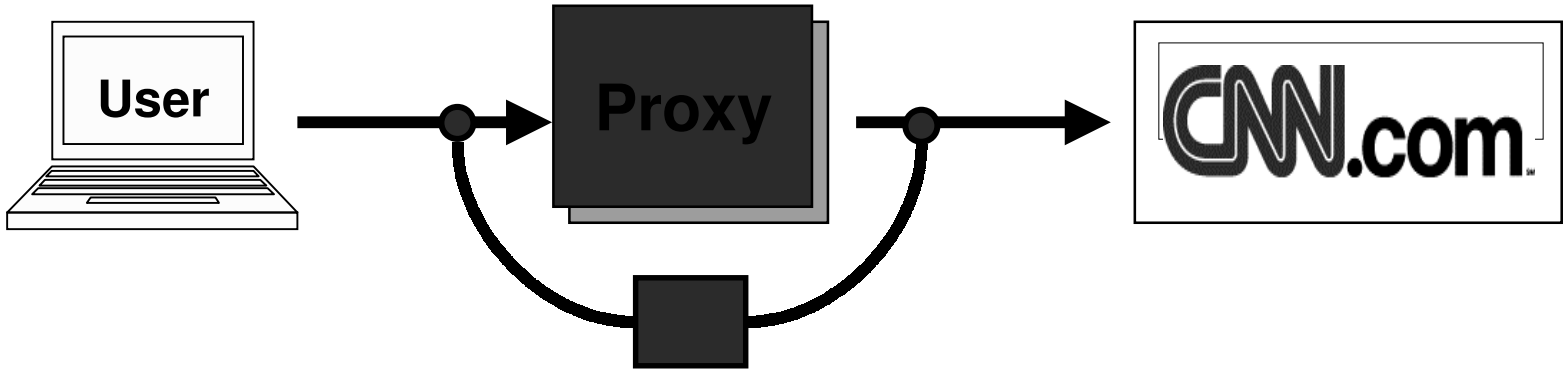
Alternative 1: Proxy Approach



- Intermediate node to proxy traffic
- Completely trust the proxy

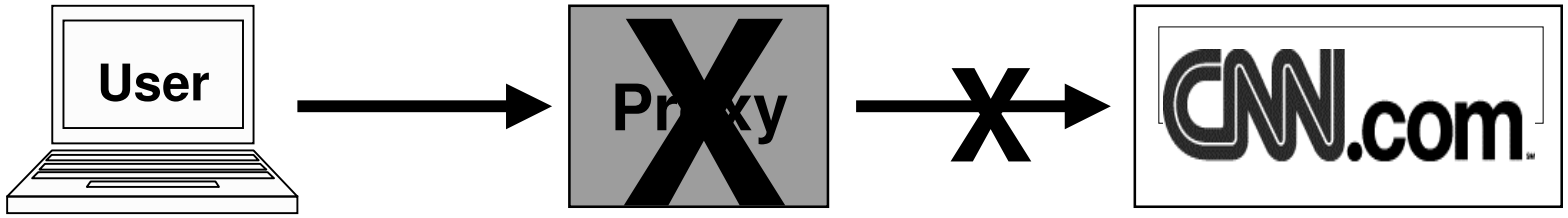
  Anonymizer.com

Failures of Proxy Approach



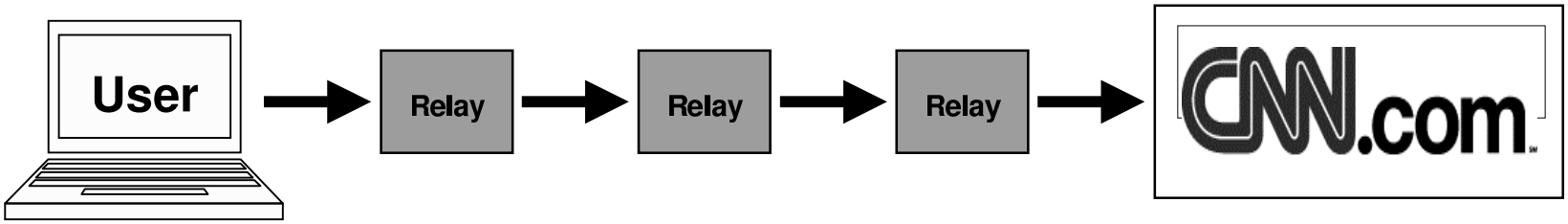
- Proxy reveals identity
- Traffic analysis is easy

Failures of Proxy Approach



- Proxy reveals identity
- Traffic analysis is easy
- CNN blocks connections from proxy
- Adversary blocks access to proxy (DoS)

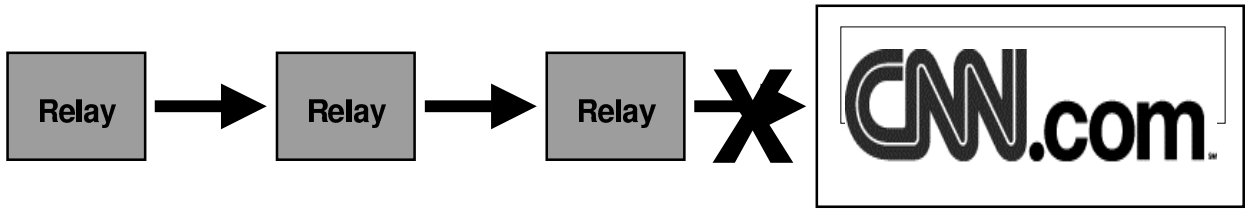
Alternative 2: Centralized Mixnet



- Removes single point of failure
- Packet forwarding through a tunnel of relays
- Small-scale, static network

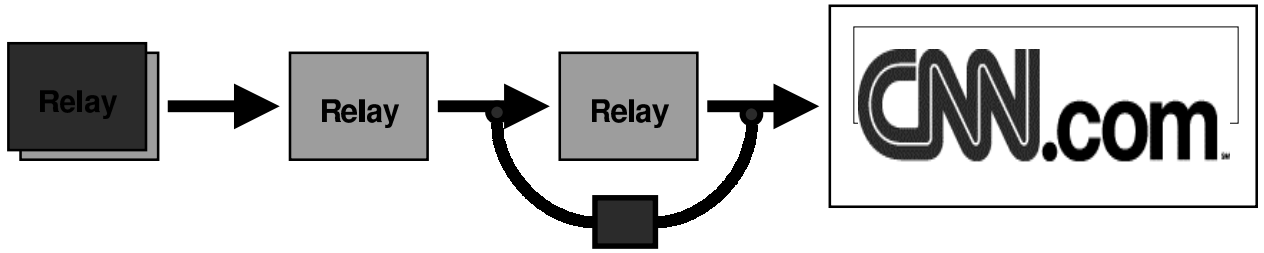
  Onion Routing, Freedom

Failures of Centralized Mixnet



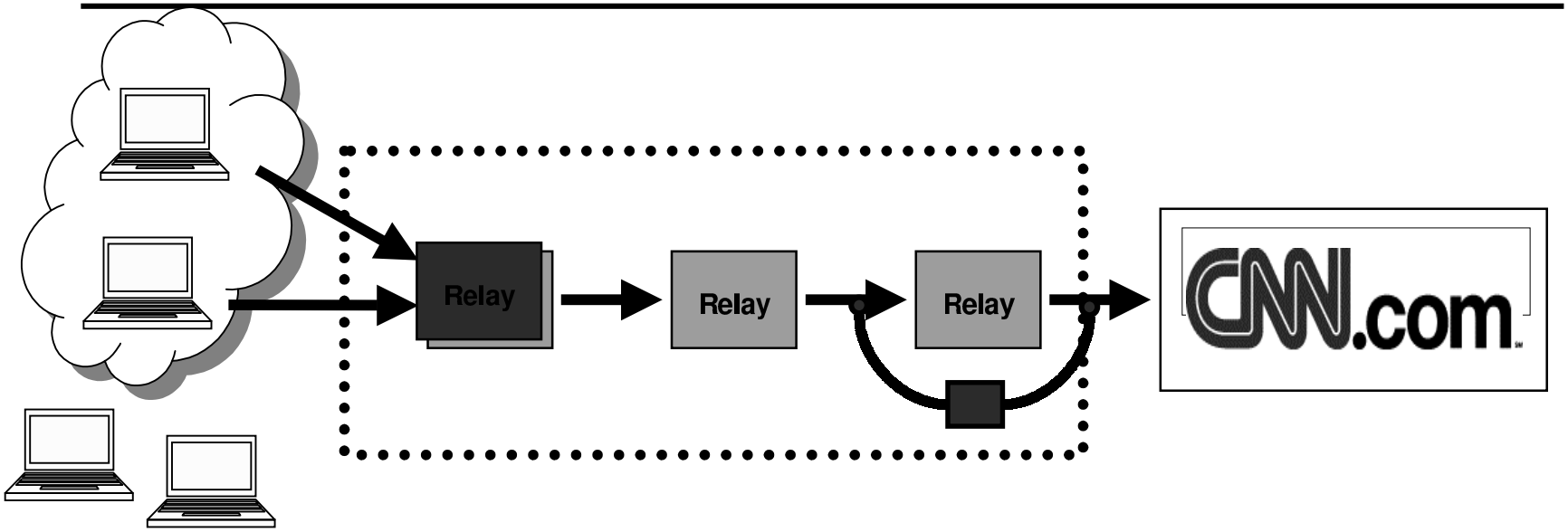
- CNN blocks core routers

Failures of Centralized Mixnet



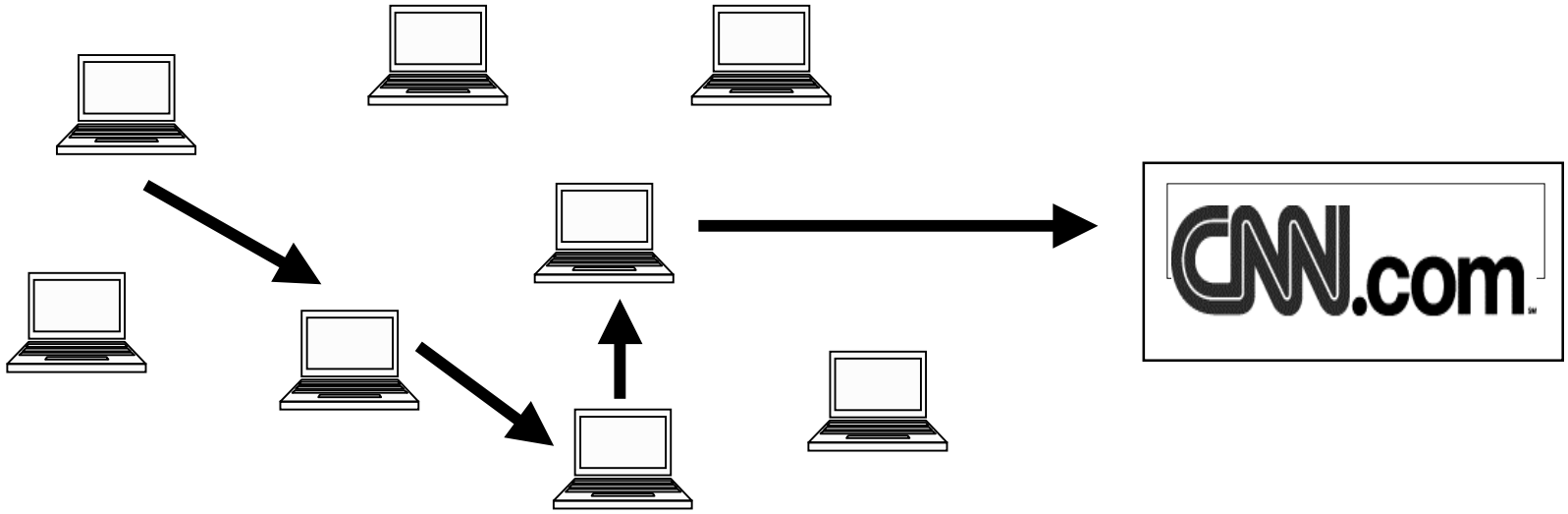
- CNN blocks core routers
- Adversary targets core routers

Failures of Centralized Mixnet



- CNN blocks core routers
- Adversary targets core routers
- Allows network-edge analysis

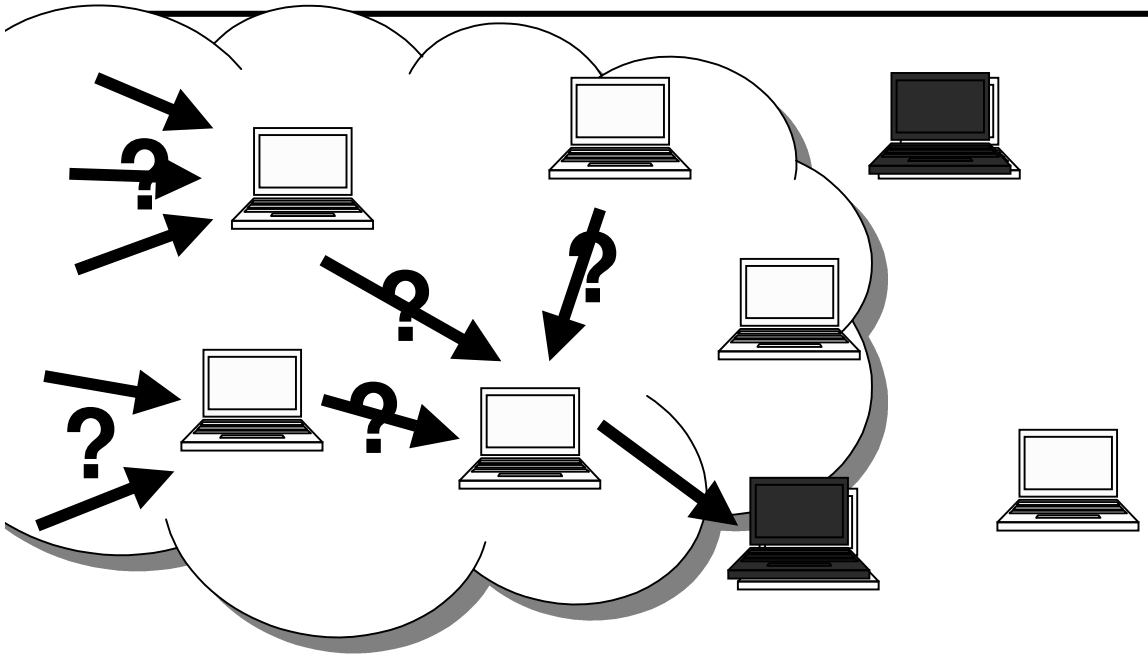
Tarzan: Me Relay, You Relay



- Millions of nodes participate
- Build tunnel over random set of nodes

(Crowds: not self-organizing, not a mixnet)

Benefits of Peer-to-Peer Design



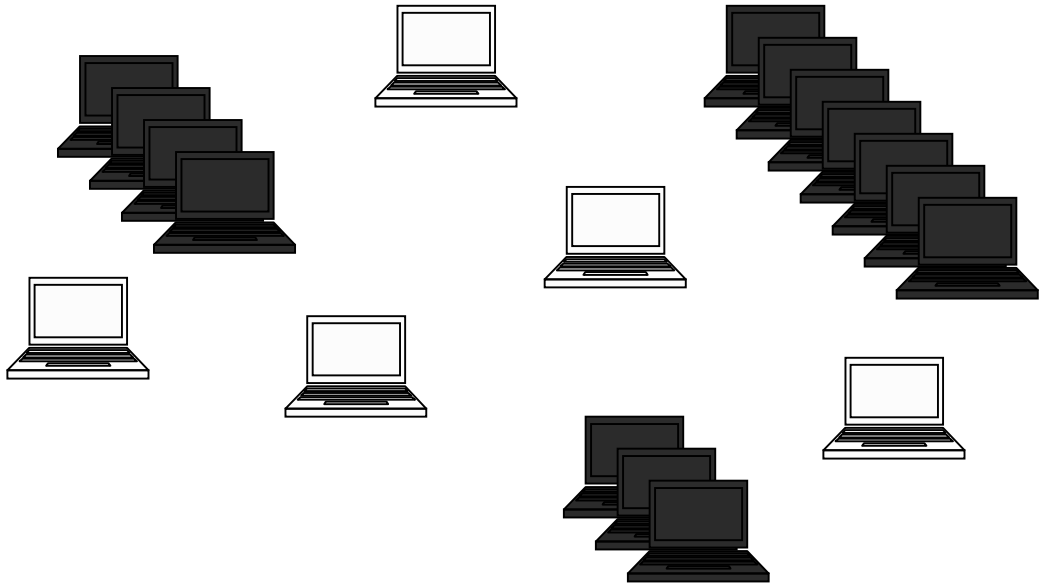
- CNN cannot block everybody
- Adversary cannot target everybody
- No network edge to analyze:

  First hop does not know he's first

Managing Peers

- Requires a mechanism that
  1. Discovers peers
  2. Scalable
  3. Robust against adversaries

Adversaries Can Join System

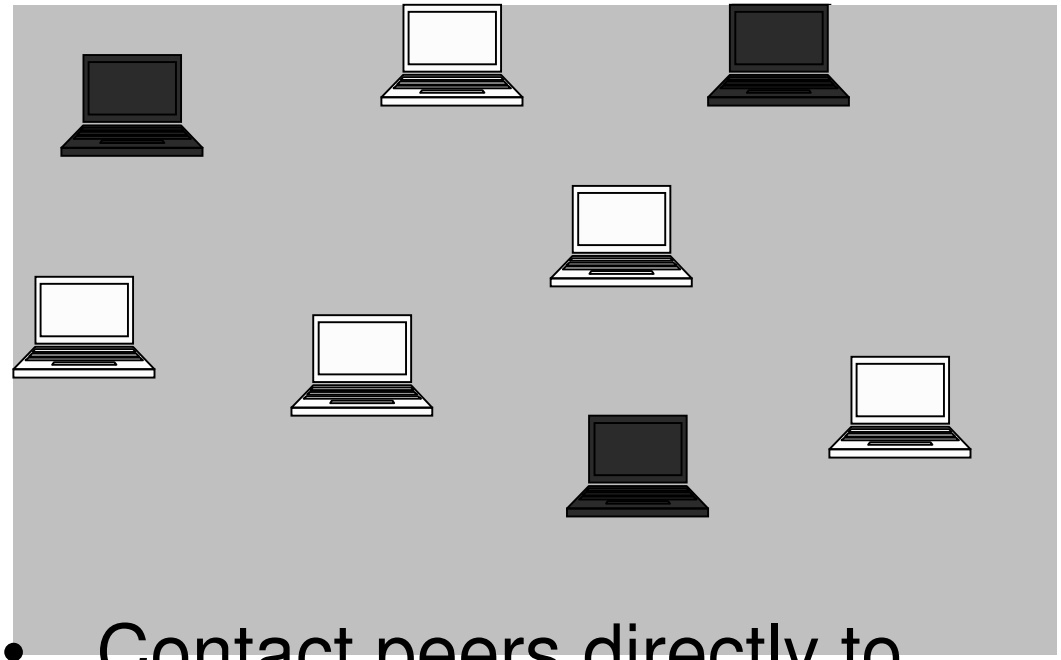


- Adversary can join more than once

  → Due to lack of central authentication

- Try to prevent adversary from impersonating large address space

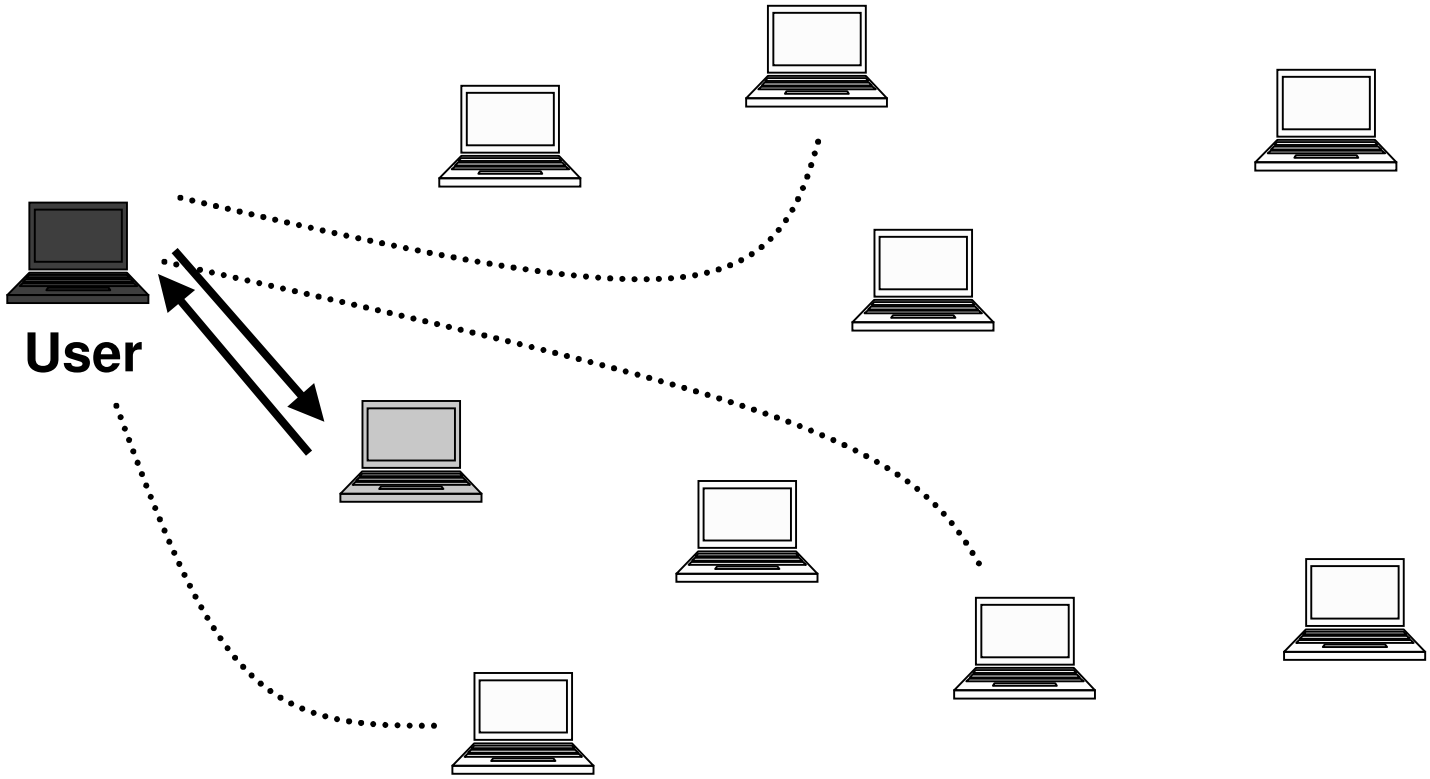
Stopping Evil Peers



- Contact peers directly to
  - Validate IP address
  - Learn public key

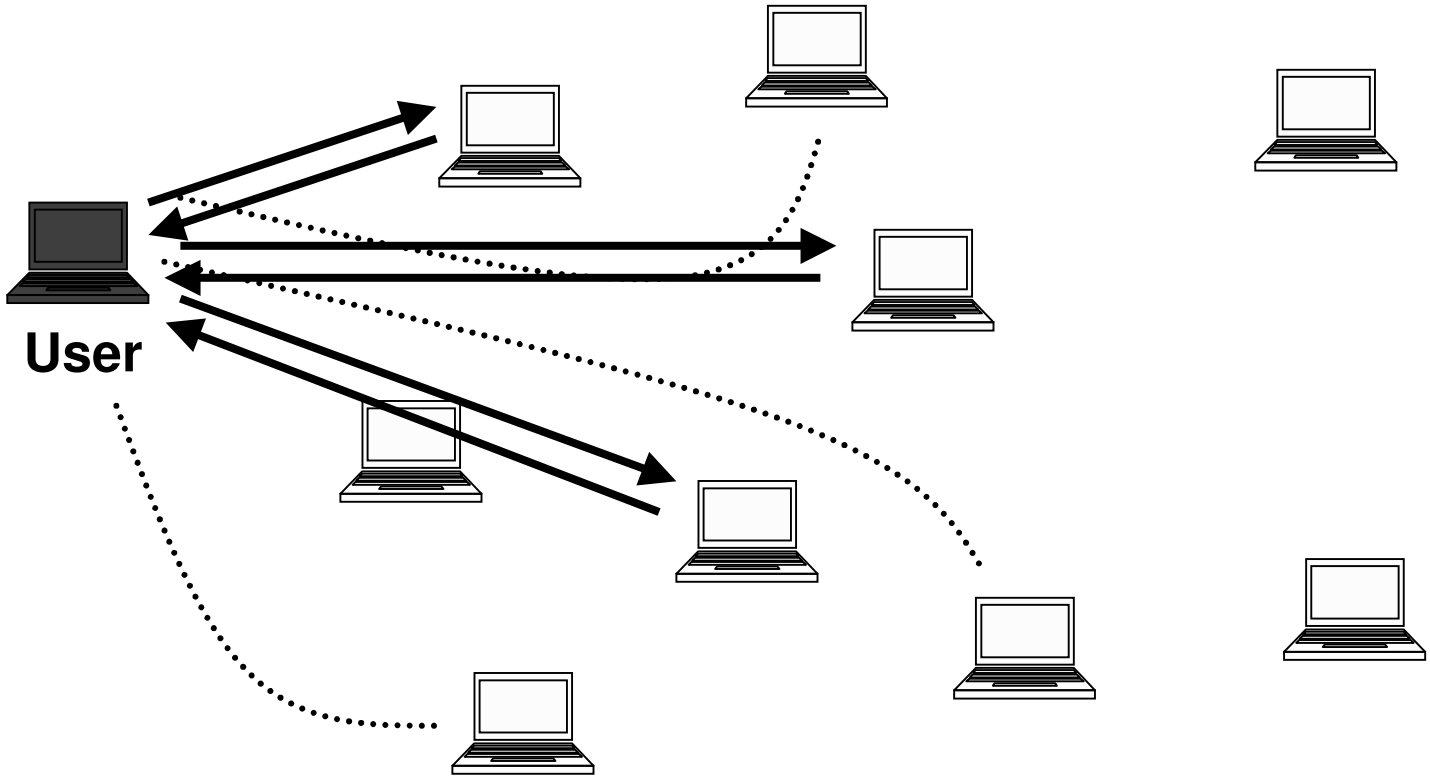
Adversary can only answer small address space

Tarzan: Joining the System



1. Contacts known peer in big (Chord) network
2. Learns of a few peers for routing queries

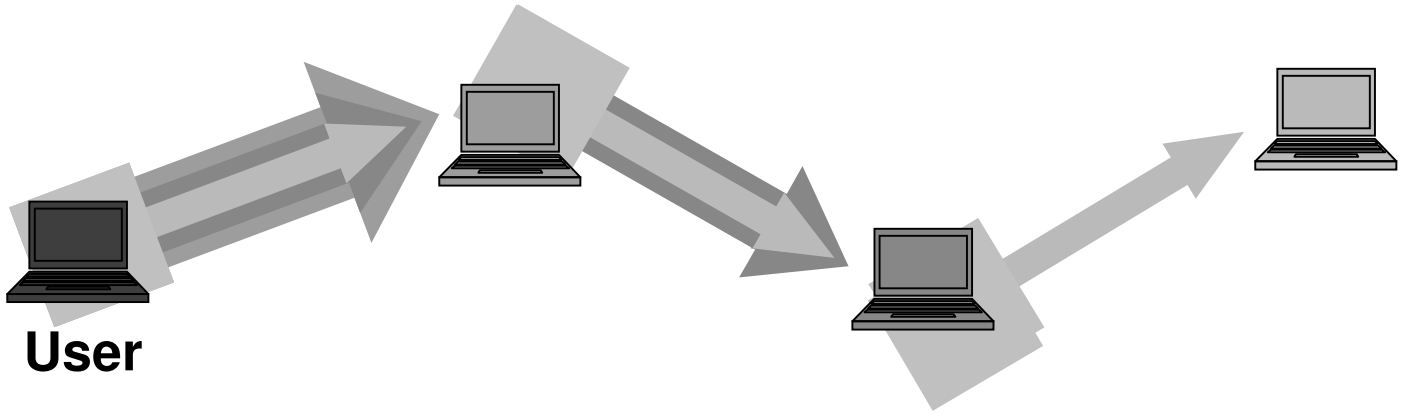
Tarzan: Discovering Peers



3. Contacts random peers to learn {IP addr, PK}

Performs Chord *lookup(random)*

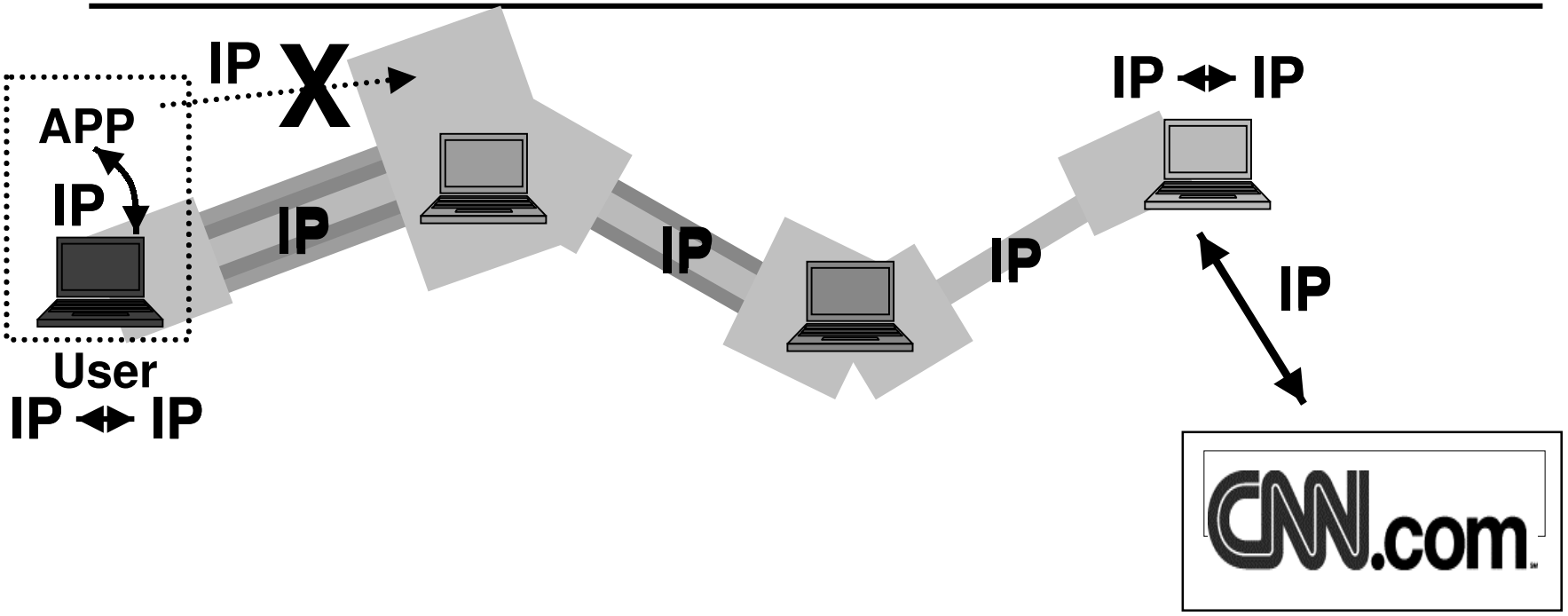
Tarzan: Building Tunnel



5. Iteratively selects peers and builds tunnel

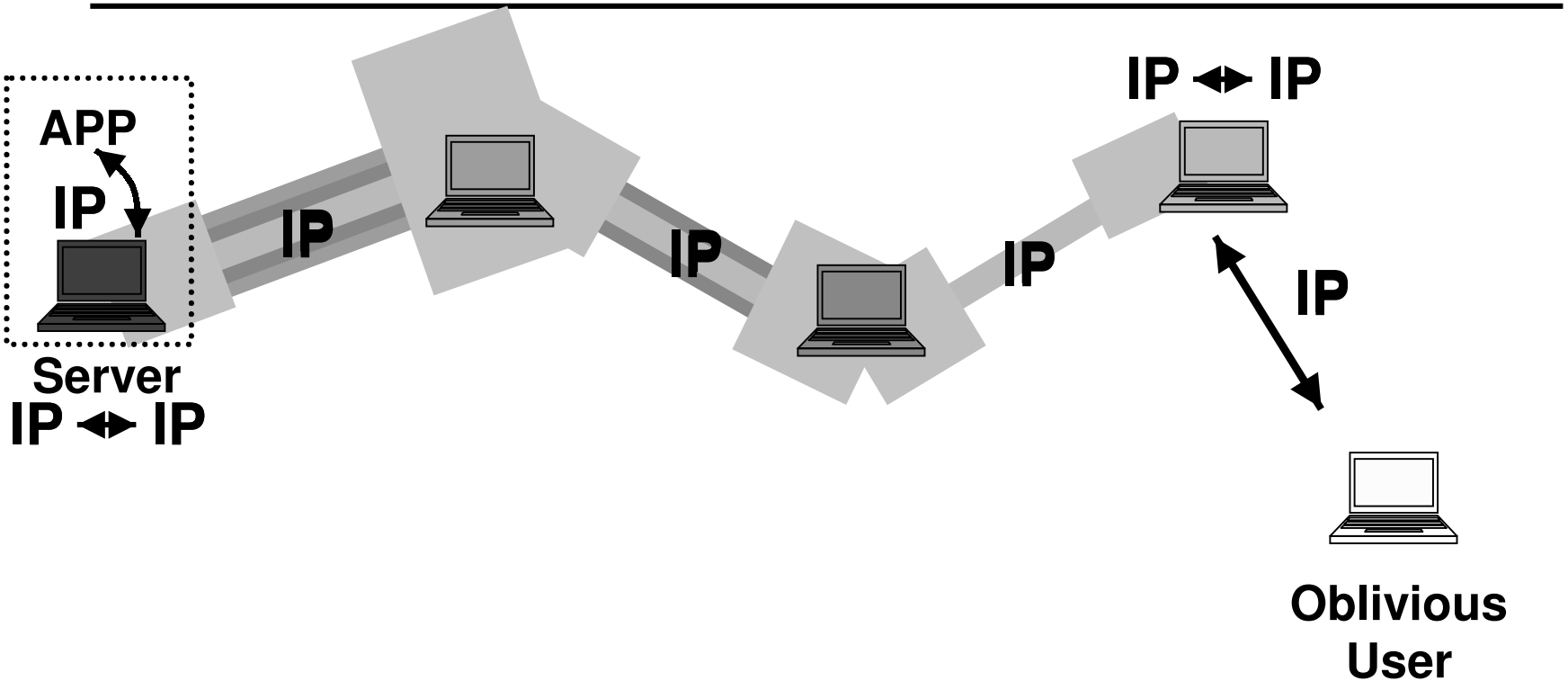
Maps flowid → session key, next hop IP addr

Tarzan: Tunneling Data Traffic



5. Reroutes packets over this tunnel

Tarzan: Tunneling Data Traffic



Transparently supports anonymous servers

Tarzan is Fast (Enough)

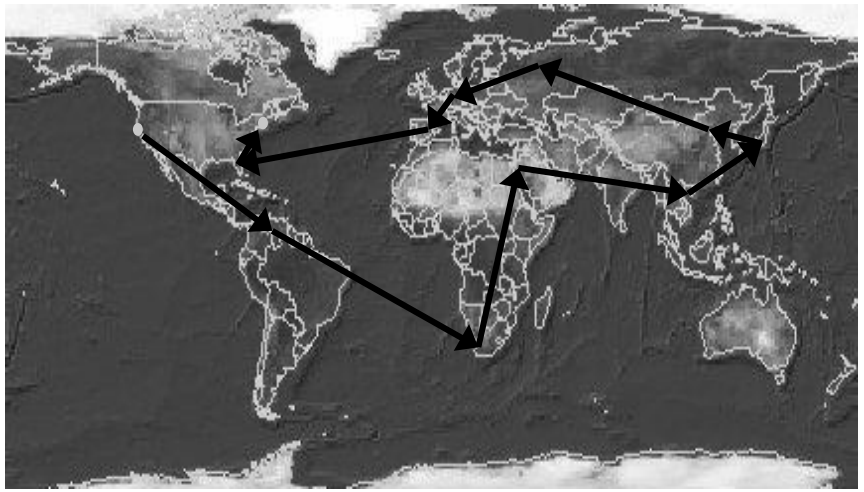
- Prototype implementation in C++
- Setup time per hop:

  ~20 ms + transmission time
- Packet forwarding per hop:

  < 1 ms + transmission time
- Network latency dominates performance

Summary

- Anonymity by rerouting traffic through MIX relays
- Manage practical pool of millions



- Random choice of relays gives robustness against adversary
- Transparent IP-layer anonymization
  - Handle existing protocols, simple to build systems
  - Easier to get critical mass of users