CE 817 - Advanced Network Security
Anonymity II

Lecture 19

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

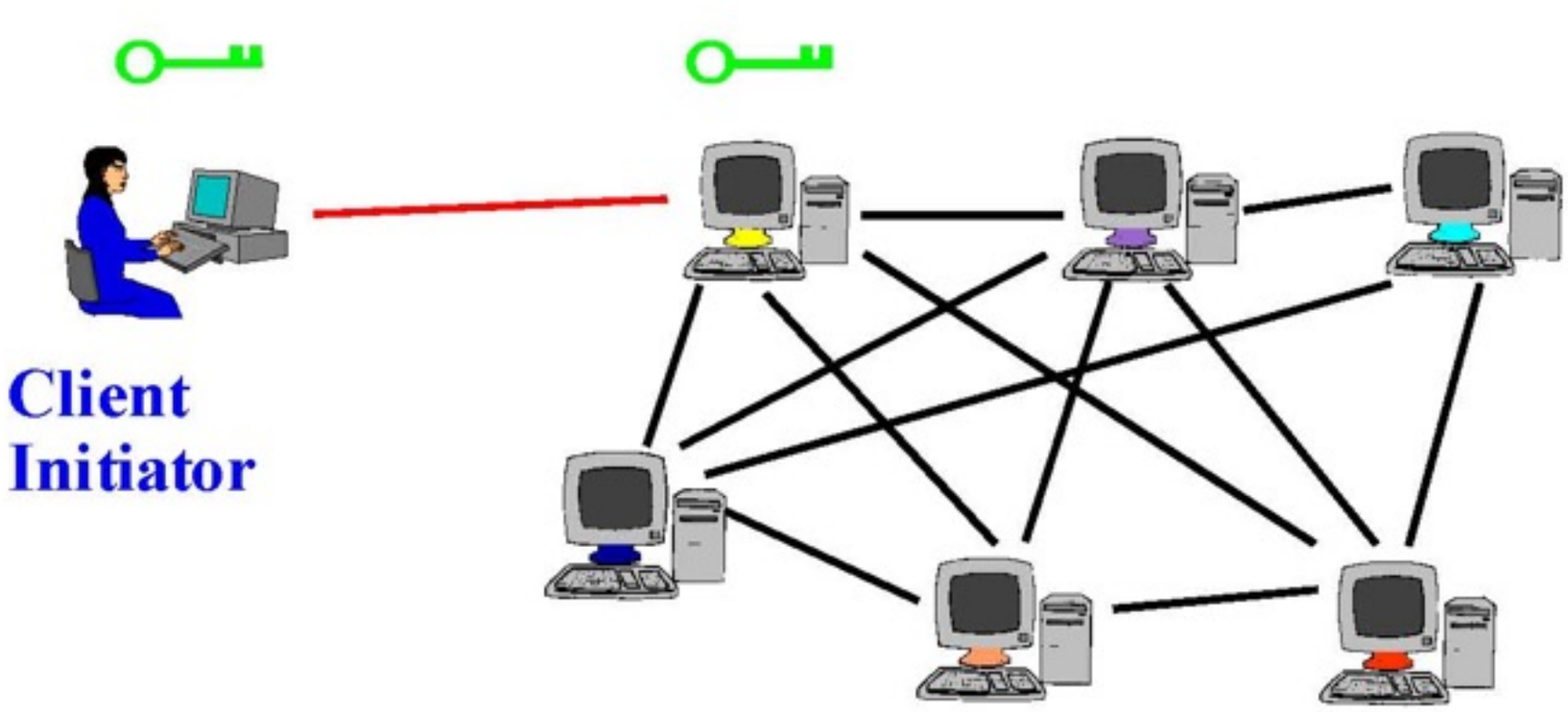
Acknowledgments: Some of the slides are fully or partially obtained from other sources. Reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

Tor

- Second-generation onion routing network
  - http://tor.eff.org
  - Developed by Roger Dingledine, Nick Mathewson and Paul Syverson
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
- About 420 routers in 2006 [Bauer]
- About 3000 in 2012
- “Easy-to-use” client proxy
  - Freely available, can use it for anonymous browsing

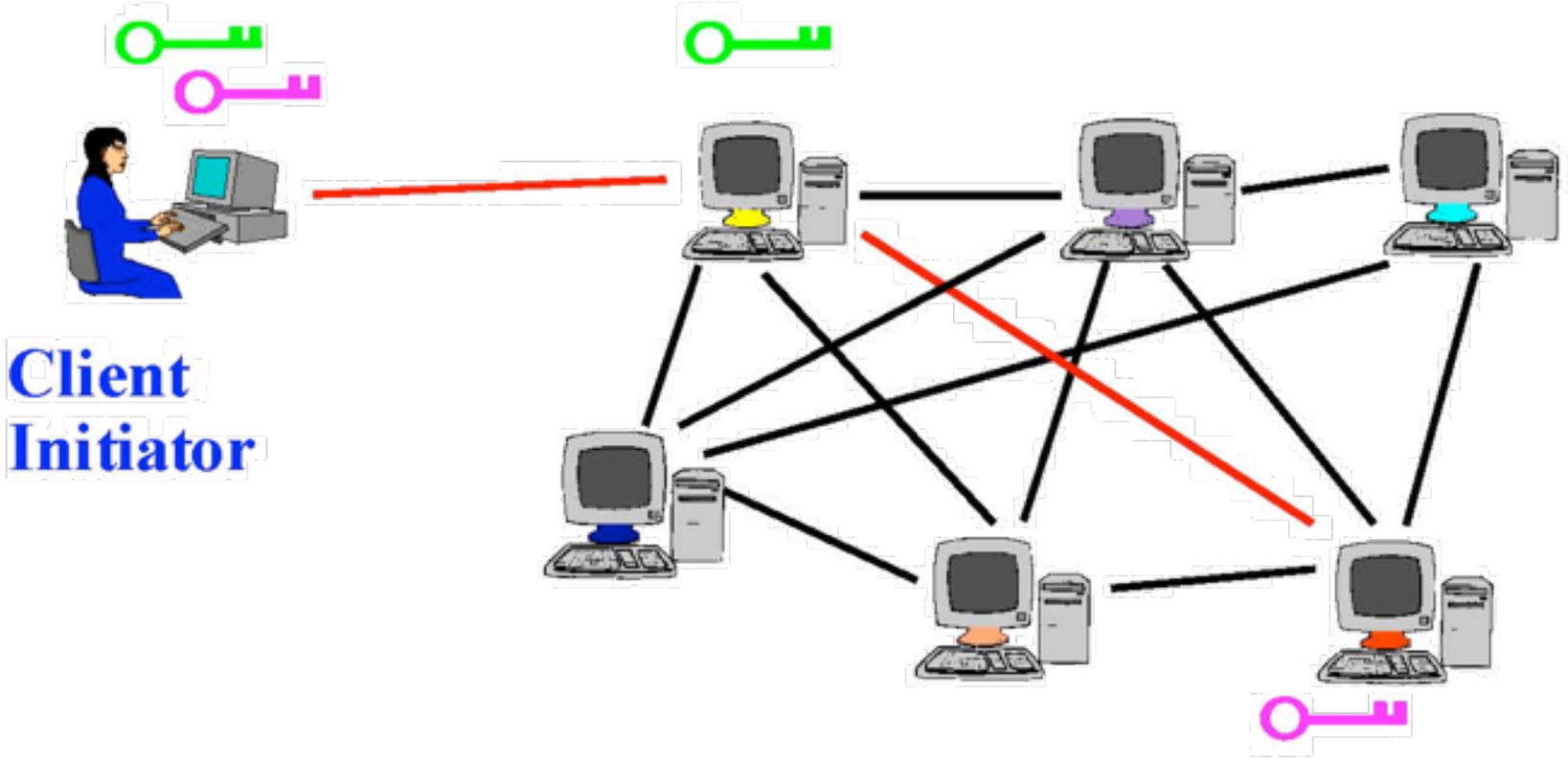
Tor Circuit Setup (1)

- Client proxy establish a symmetric session key and circuit with Onion Router #1



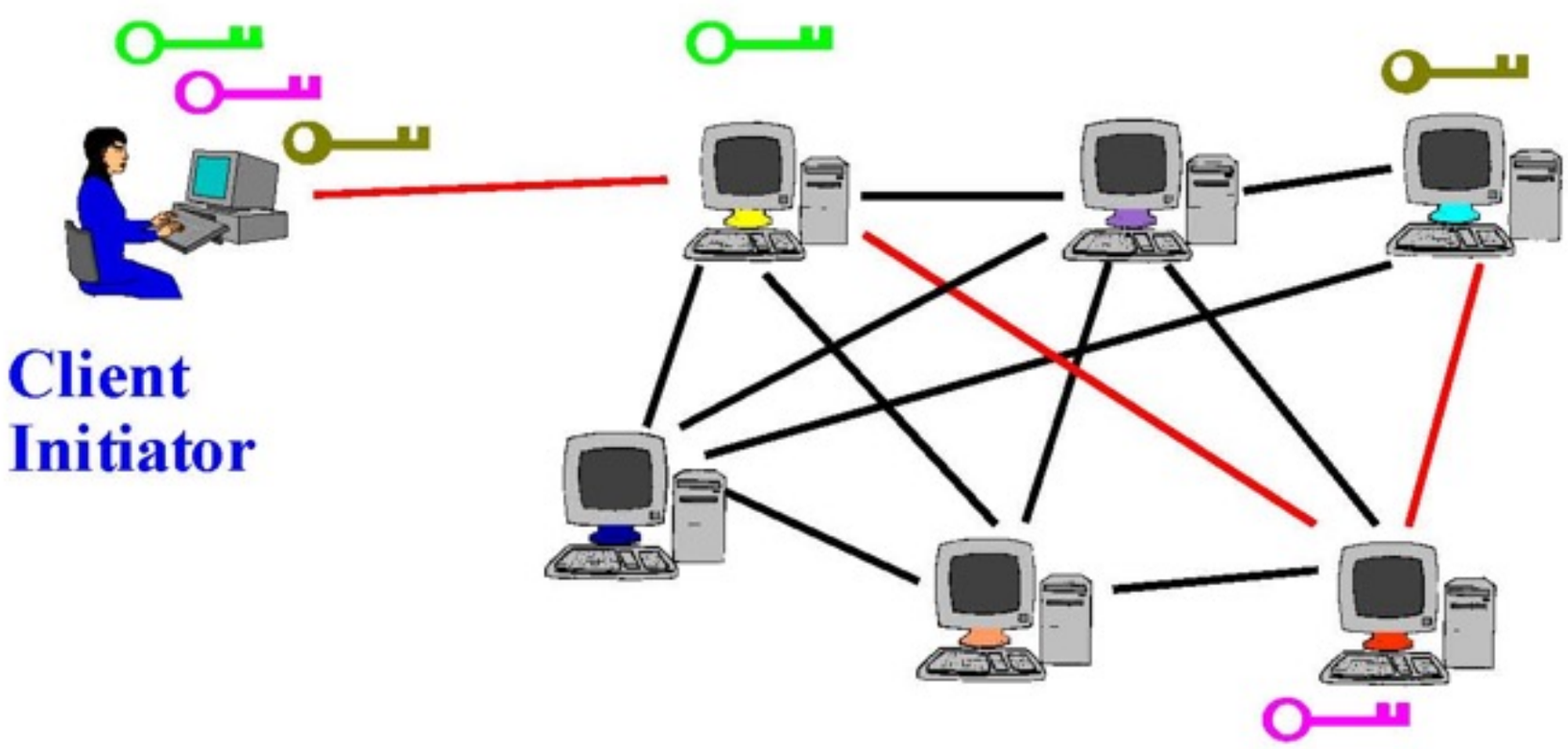
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
  - Tunnel through Onion Router #1 (don't need)



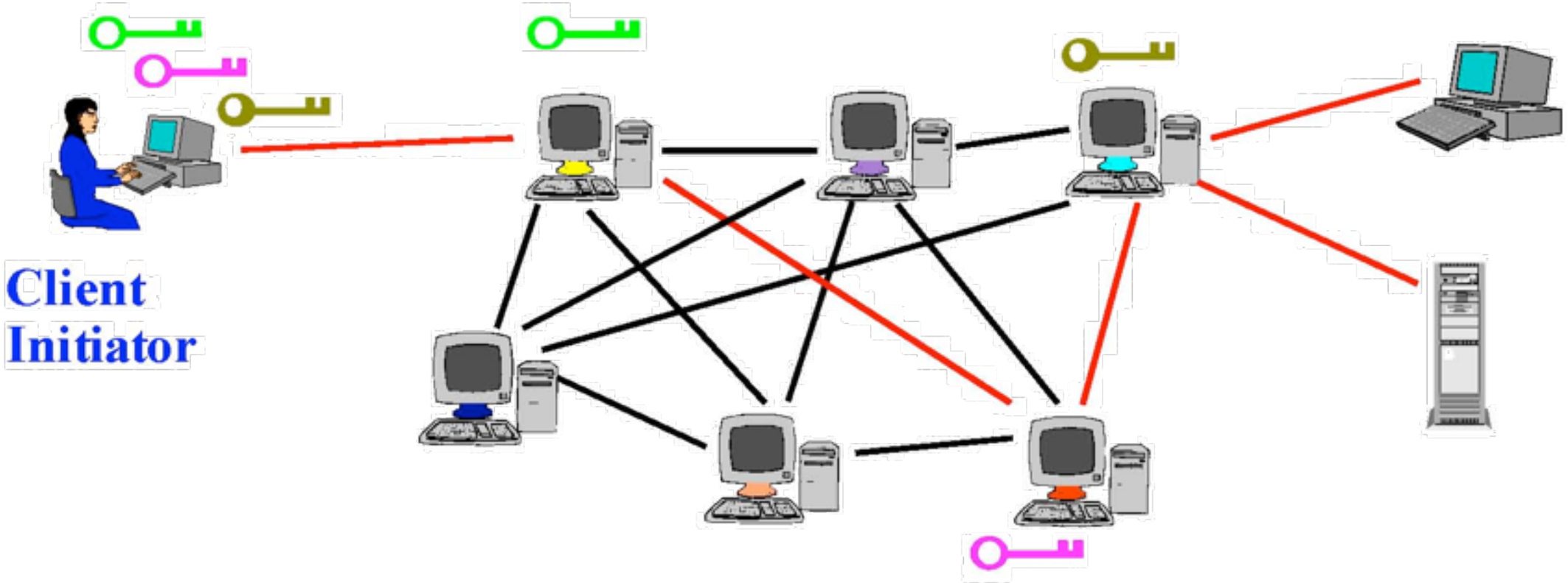
Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
  - Tunnel through Onion Routers #1 and #2



Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit
  - Datagrams are decrypted and re-encrypted at each link

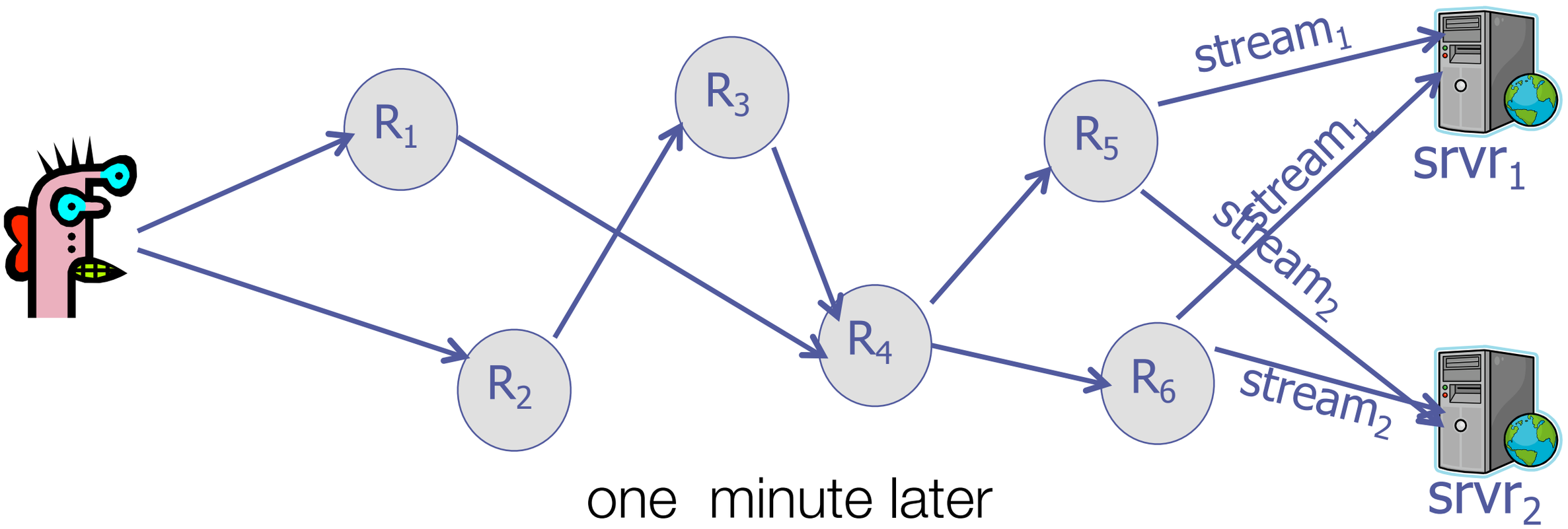


Tor Management Issues

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn’t need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active onion routers, their locations, current public keys, etc.
  - Control how new routers join the network
  - Directory servers’ keys ship with Tor code

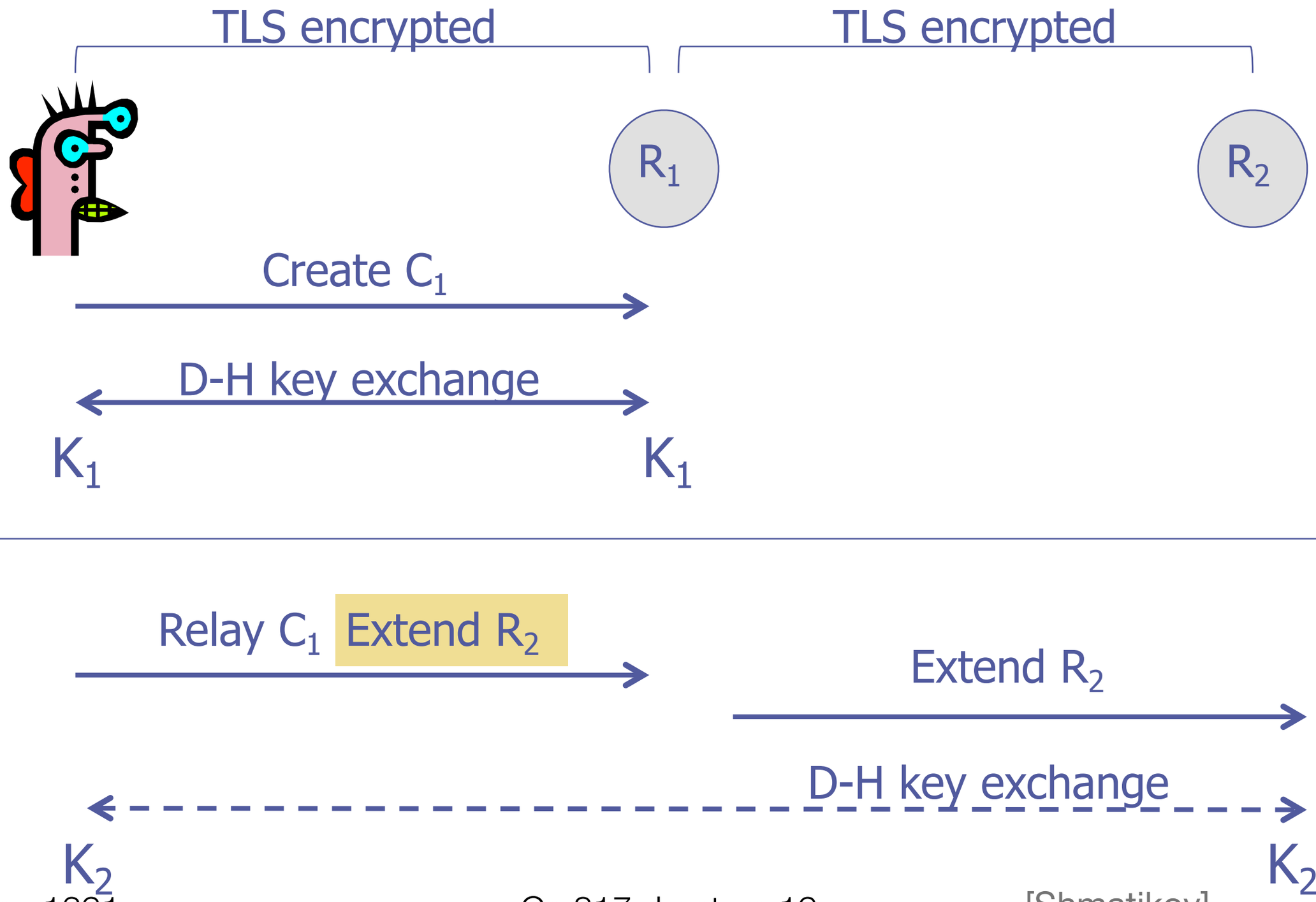
The Tor design

- Trusted directory contains list of Tor routers
- User's machine preemptively creates a circuit
  - Used for many TCP streams
  - New circuit is created once a minute

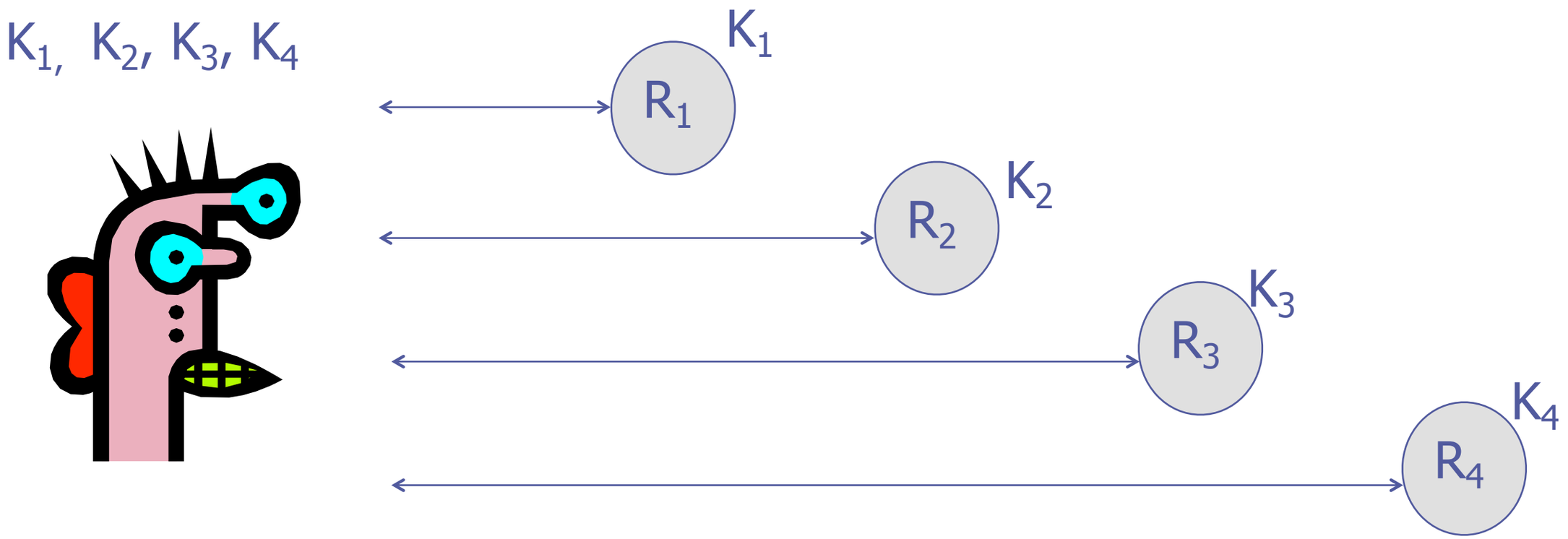


one minute later

Creating circuits

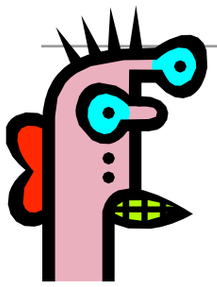


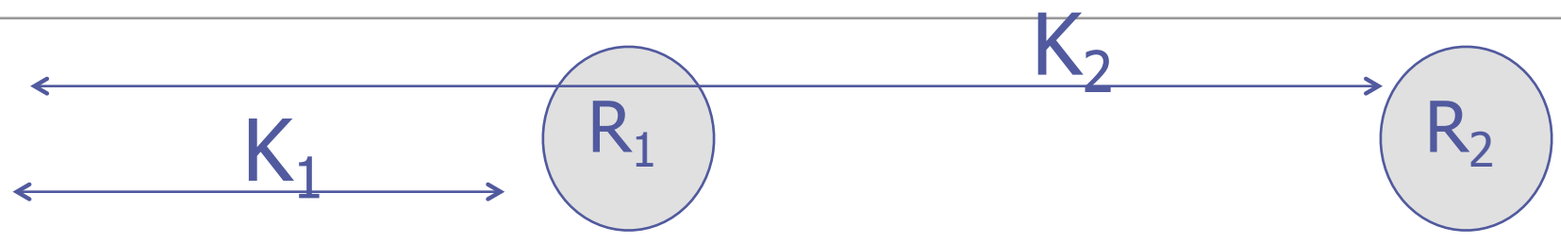
Once circuit is created



- User has shared key with each router in circuit
- Routers only know ID of successor and predecessor

Sending data





Relay C_1 Begin site:80

Relay C_2 Begin site:80

TCP handshake

Relay C_1 data HTTP GET

Relay C_2 data HTTP GET

HTTP GET

Relay C_1 data resp

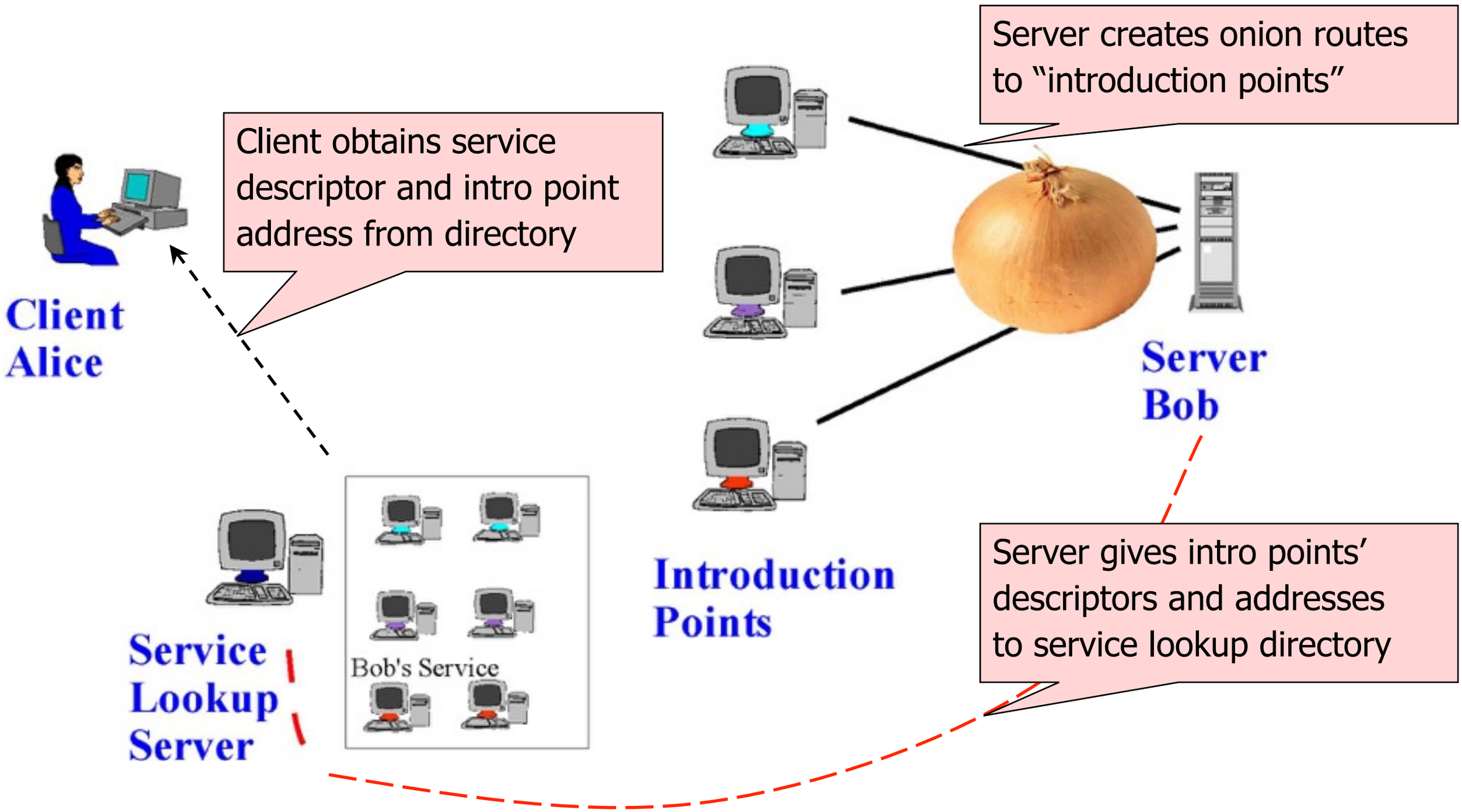
Relay C_2 data resp

resp

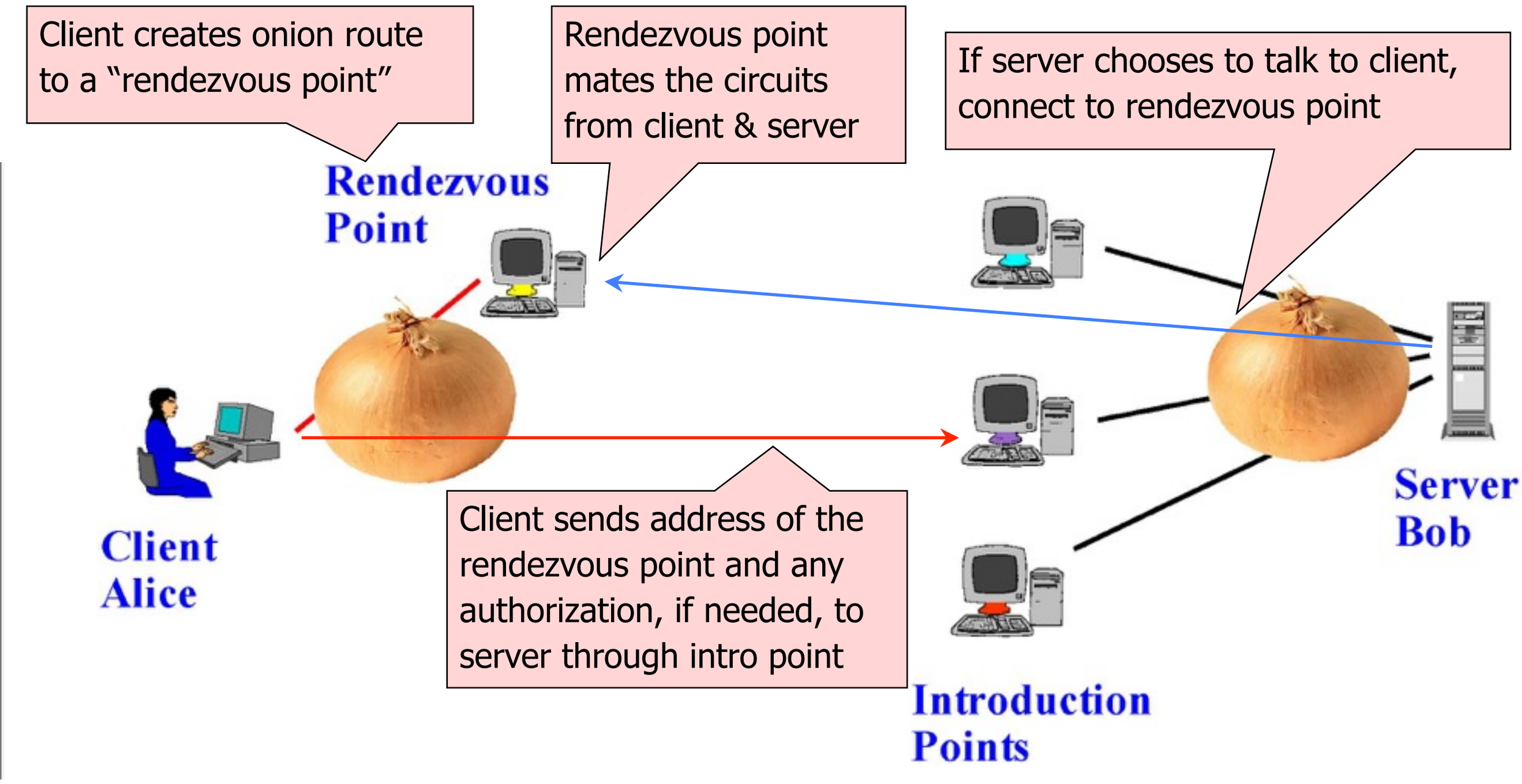
Location Hidden Servers

- Goal: deploy a server on the Internet that anyone can connect to without knowing where it is or who runs it
- Accessible from anywhere
- Resistant to censorship
- Can survive full-blown DoS attack
- Resistant to physical attack
  - Can’t find the physical server!

Creating a Location Hidden Server



Using a Location Hidden Server



Properties

- Performance:
  - Fast connection time: circuit is pre-established
  - Traffic encrypted with AES: no pub-key on traffic

- Downside:
  - Routers must maintain state per circuit
  - Each router can link multiple streams via CircuitID
    - all streams in one minute interval share same CircuitID

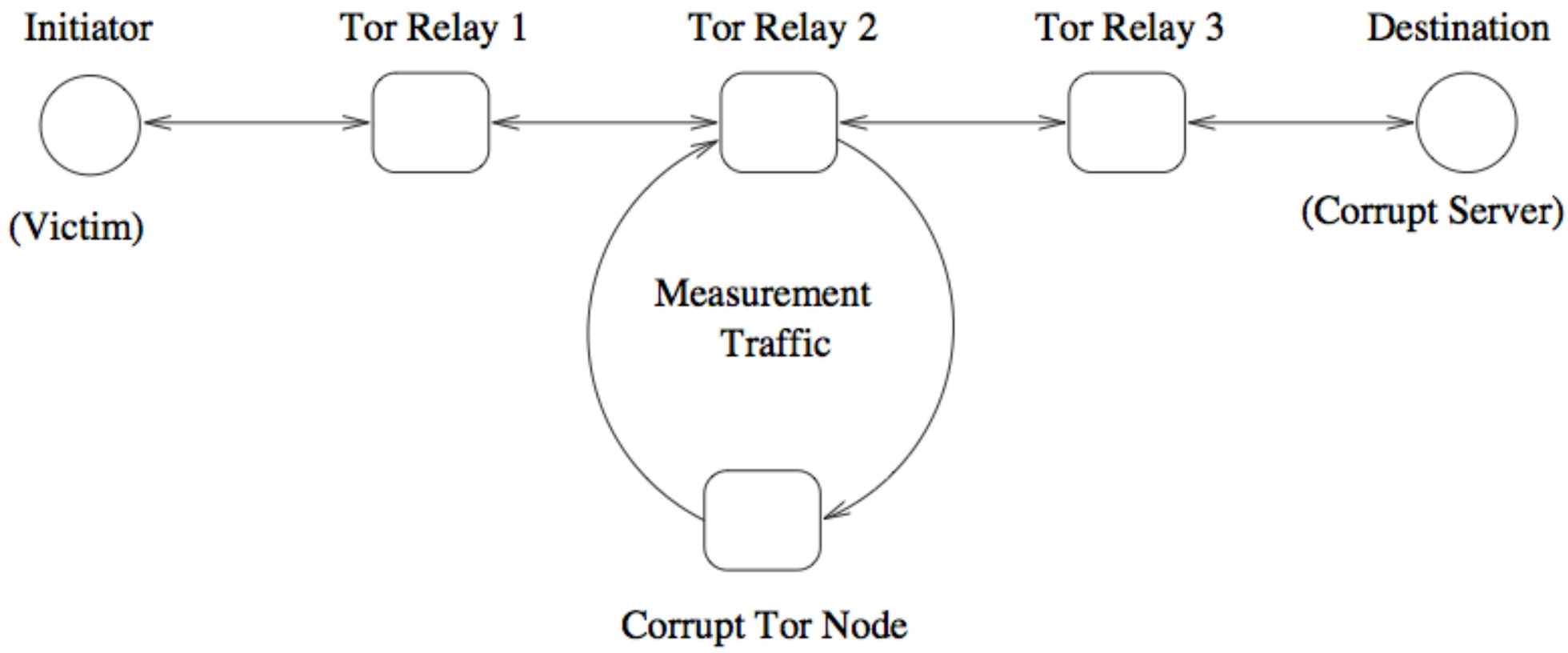
Low-Cost Traffic Analysis of Tor

Steven J. Murdoch, George Danezis
University of Cambridge, Computer Laboratory

The holes in Tor

- No explicit mixing
  - Cells are stored in separate buffers for each stream
  - Output in a round robin fashion (for fairness and best effort service)
  - No explicit delay, reorder, batching or drop
  - It means the load on the Tor node affects the latency of all connection streams routed through it
  - The higher the load, the higher the latency
- Streams from the same initiator use the same circuit
  - Can be used to test whether two streams accessing two server belong to the same user

Attack Setup

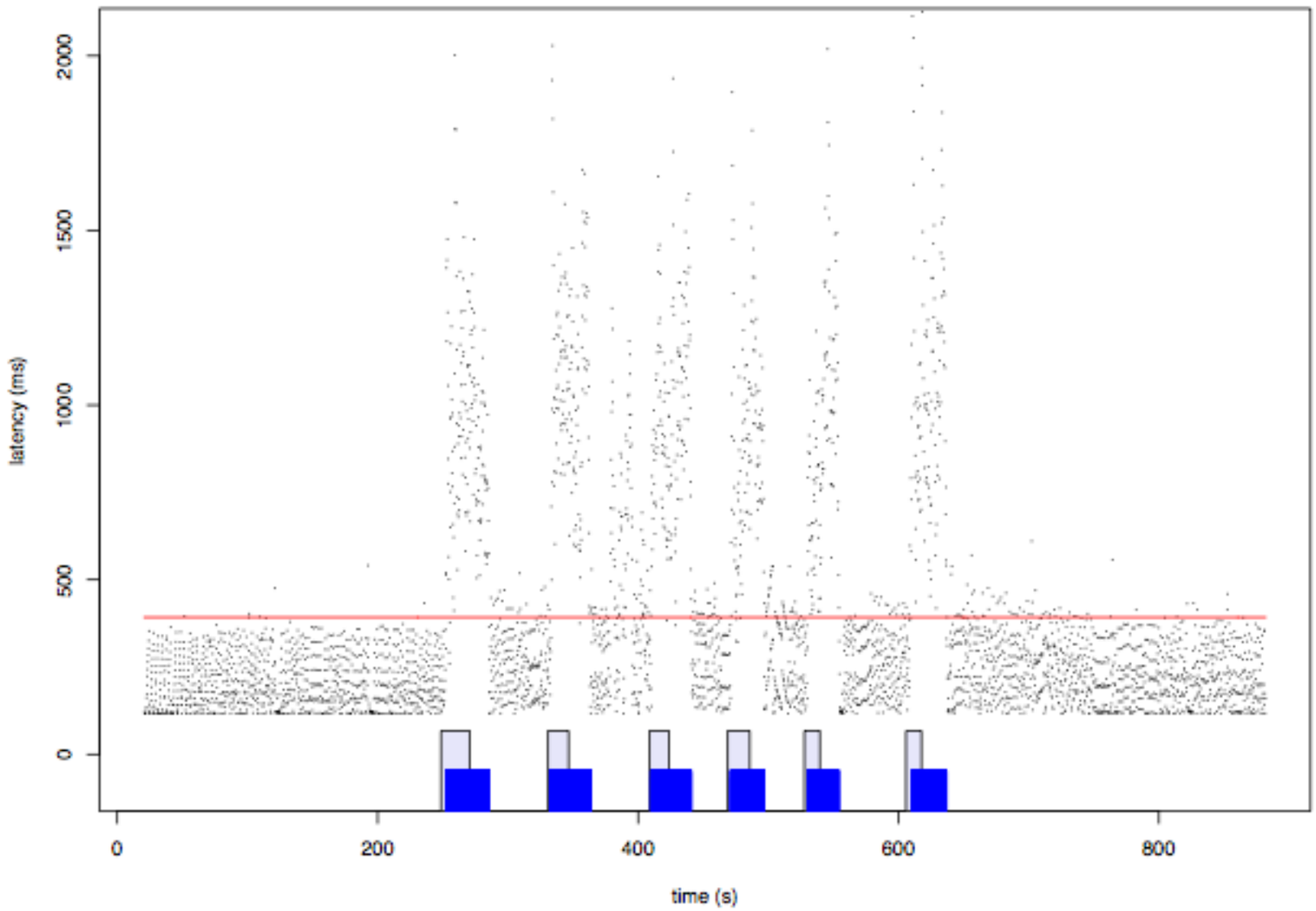


Experimental Setup

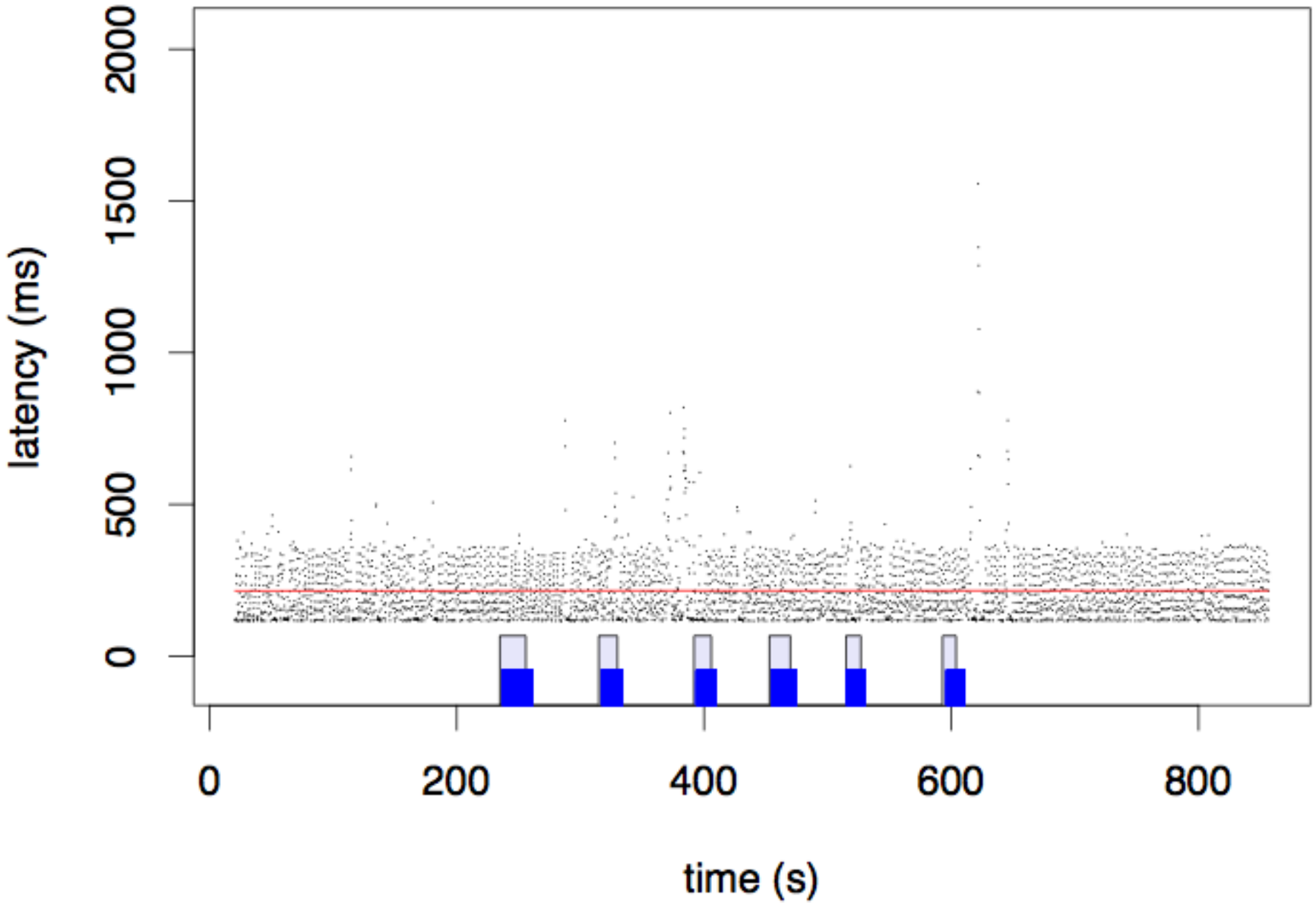
- Probe server in U. of Cambridge
- Victim and corrupt server on PlanetLab
- Tor v0.0.9

- Probe every 0.2 secs
- Corrupt server send data as fast as Tor allows
  - send 10 to 25 secs
  - Wait 30 to 75 secs
- Data from probing 13 Tor nodes was collected

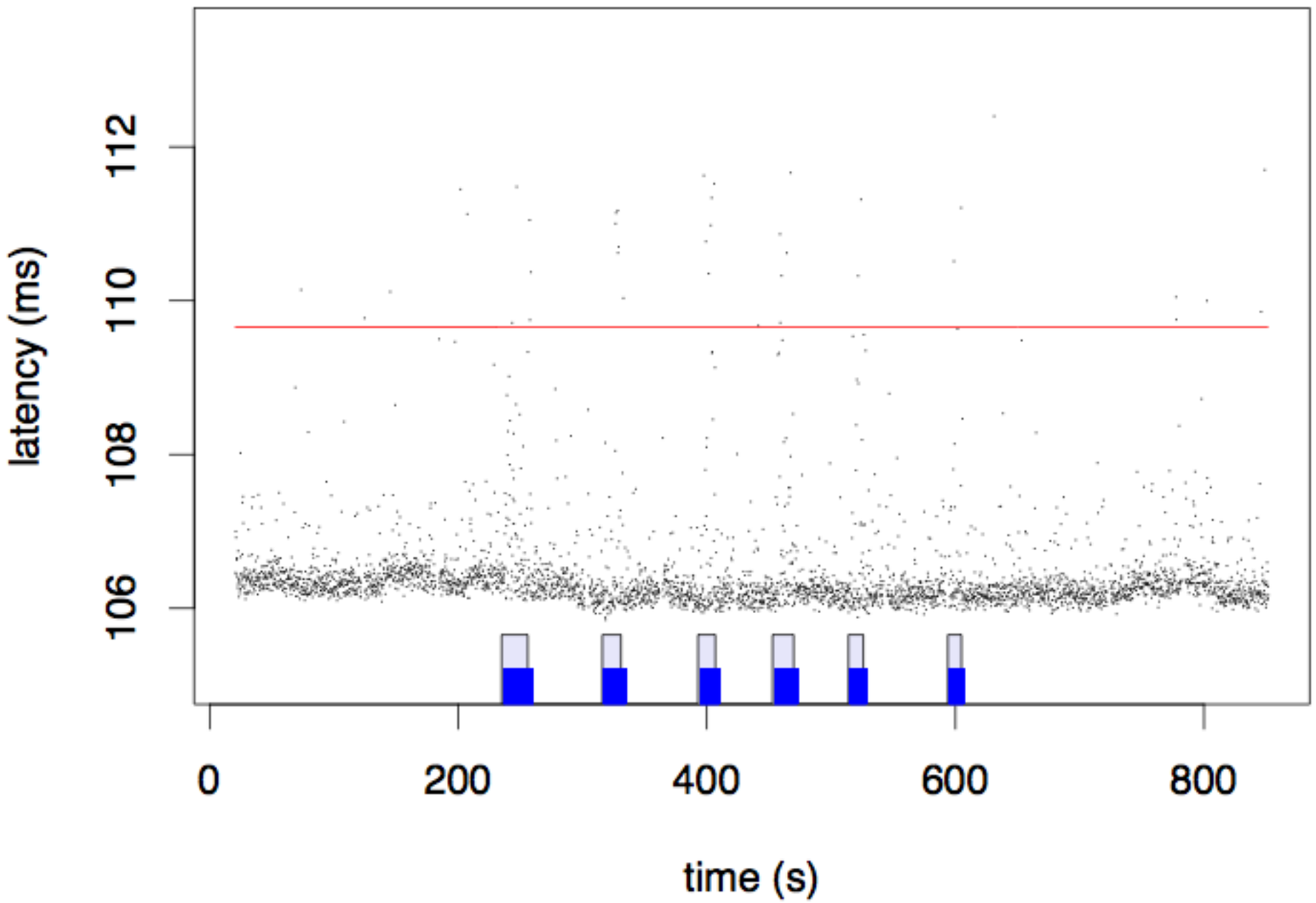
Probe Result (Node K)



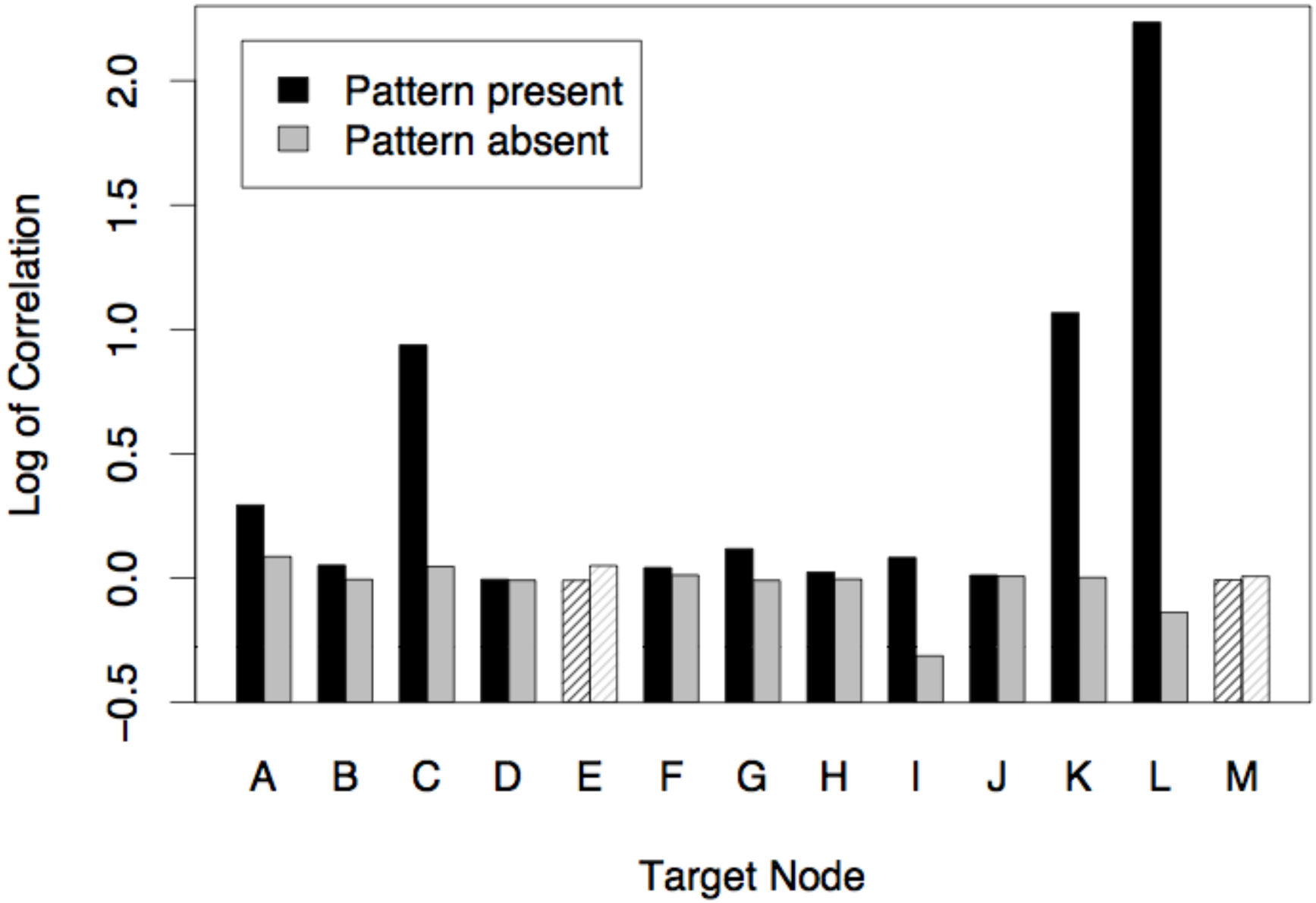
Probe Result (Node K) (no traffic pattern)



False Negative Node E (no significant correlation)



Summary of Correlation



Discussion

- Attackers can use this timing characteristic to observer without direct access to the Tor nodes
- Higher volumes of traffic degrade the performance of the attack

Variants of the Attack

- If no total control over the corrupt server
  - Server is observable, take advantage of a known traffic pattern on the server
  - Otherwise, alter the load on the server by modulating DoS attack
- Identify the entry point by estimating how much the induced traffic pattern is shifted as it travels through the network.

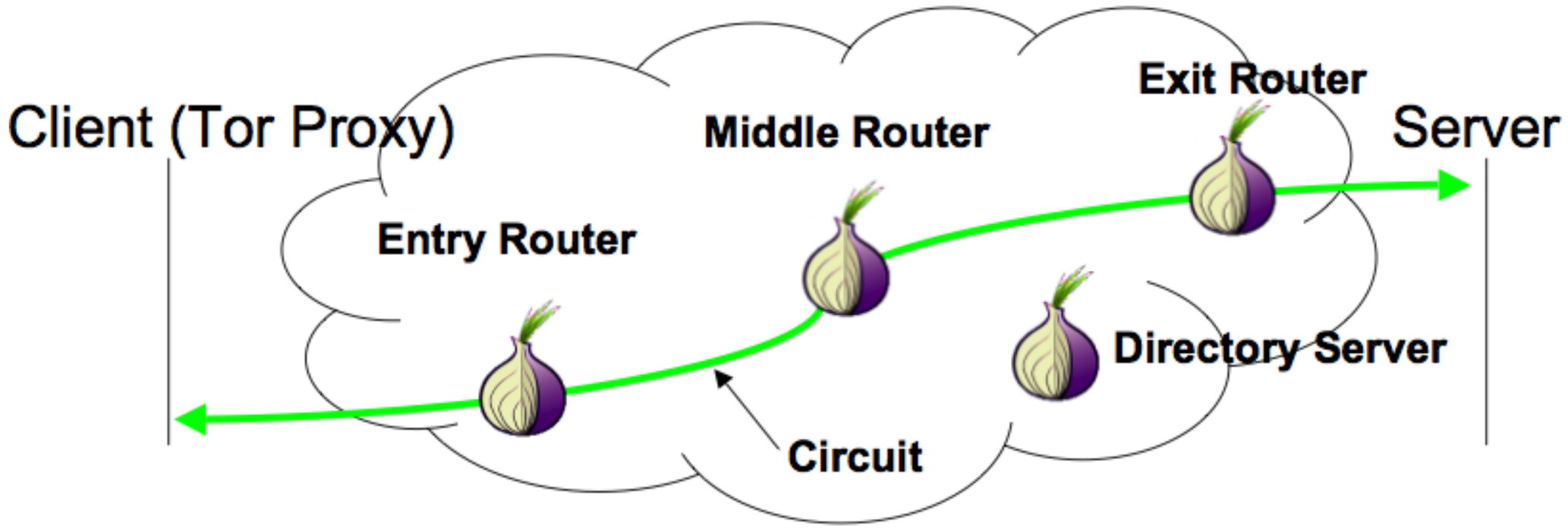
Low-Resource Routing Attacks Against Tor

Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno,
Douglas Sicker, Workshop on Privacy in the Electronic Society,
2007.

Striking a Balance: Performance vs. Anonymity

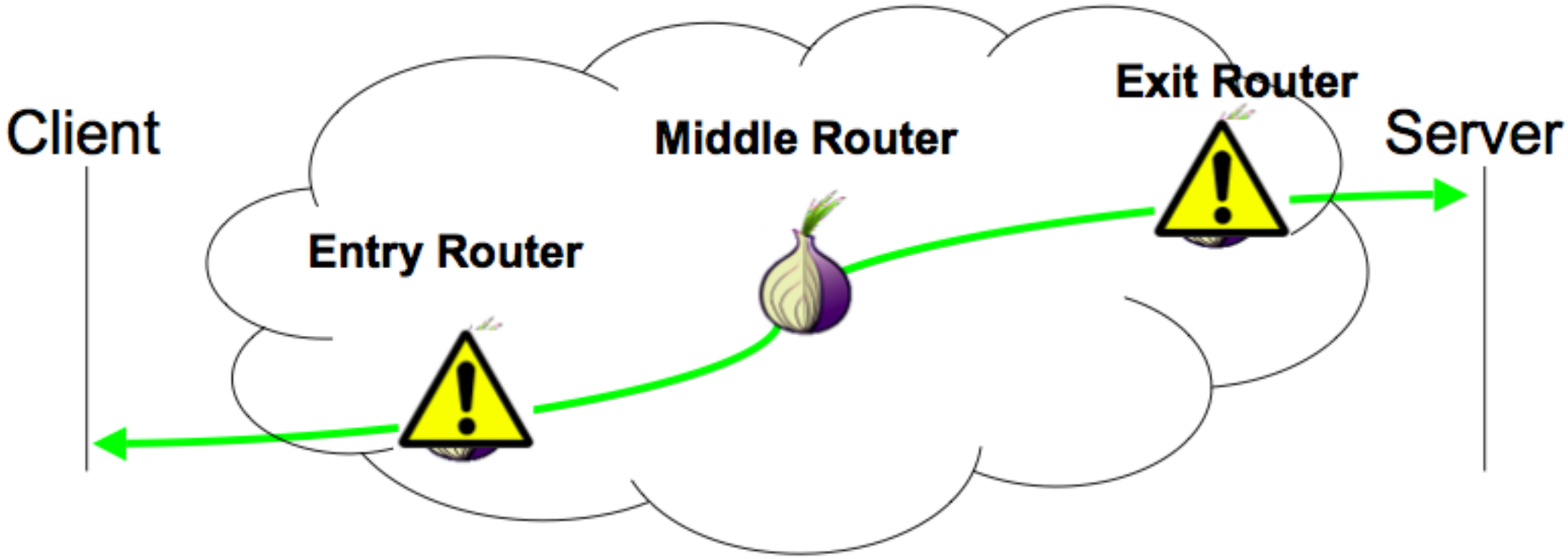
- Two observations:
  - Perfect anonymity is desirable, but not practical
  - Users demand certain performance standards from any anonymous network
- Tor attempts to balance the traffic load to ensure a relatively low latency and high throughput service, but:
- What impact does this attempt at load balancing have on Tor's ability to provide strong anonymity?

Tor and Onion Routing



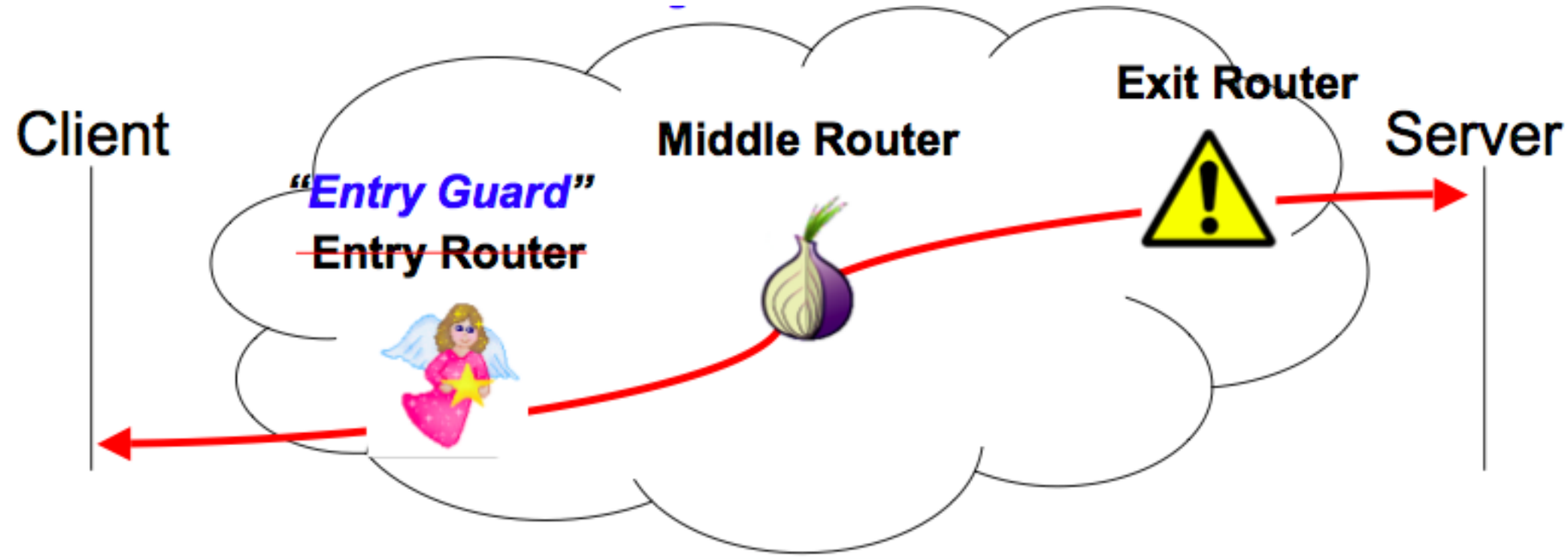
- Tor provides anonymity for TCP by tunneling traffic through a circuit of three Tor routers using a layered encryption technique
- Only the entry router knows the client's identity, and only the exit router knows the final destination server
- To easily correlate client requests to a server's response, an adversary must control the entry and exit routers for a circuit

Traditional Onion Routing Security Model



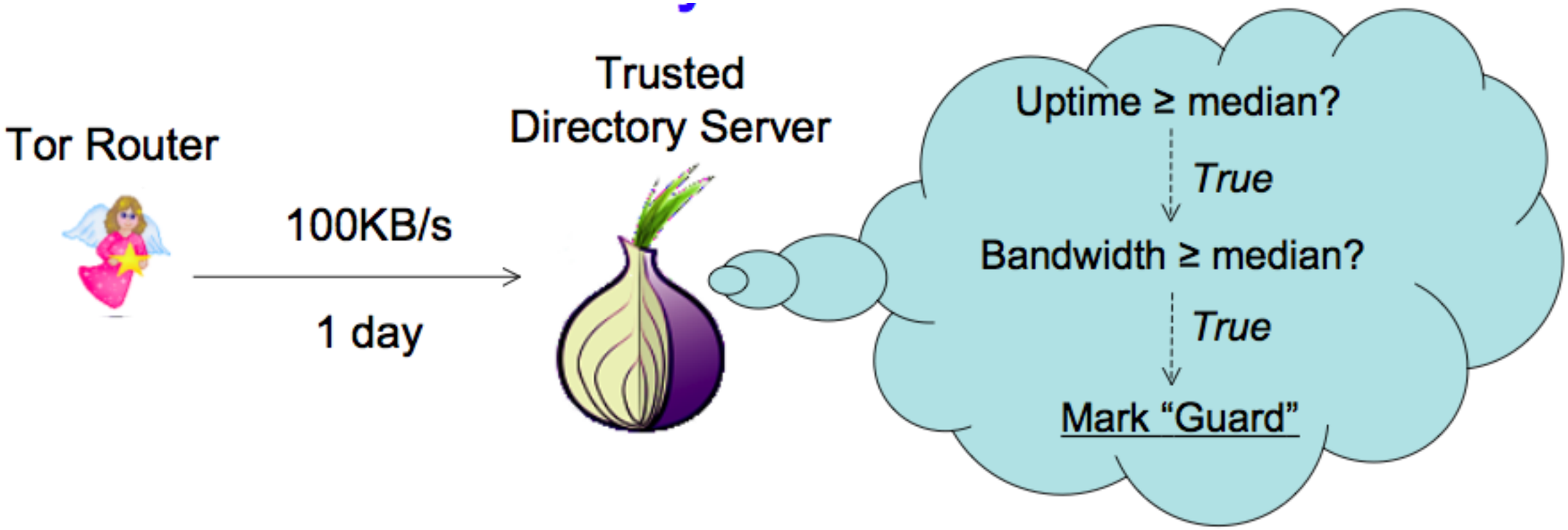
- Traditionally, it has been assumed that an attacker can control an entry/exit router combination for a target circuit with a very low probability:
- Assuming that routers are chosen uniformly at random, this probability is (c/n)^2, where the attacker controls c routers in a network of n total routers

Protecting Circuits: Entry Guards



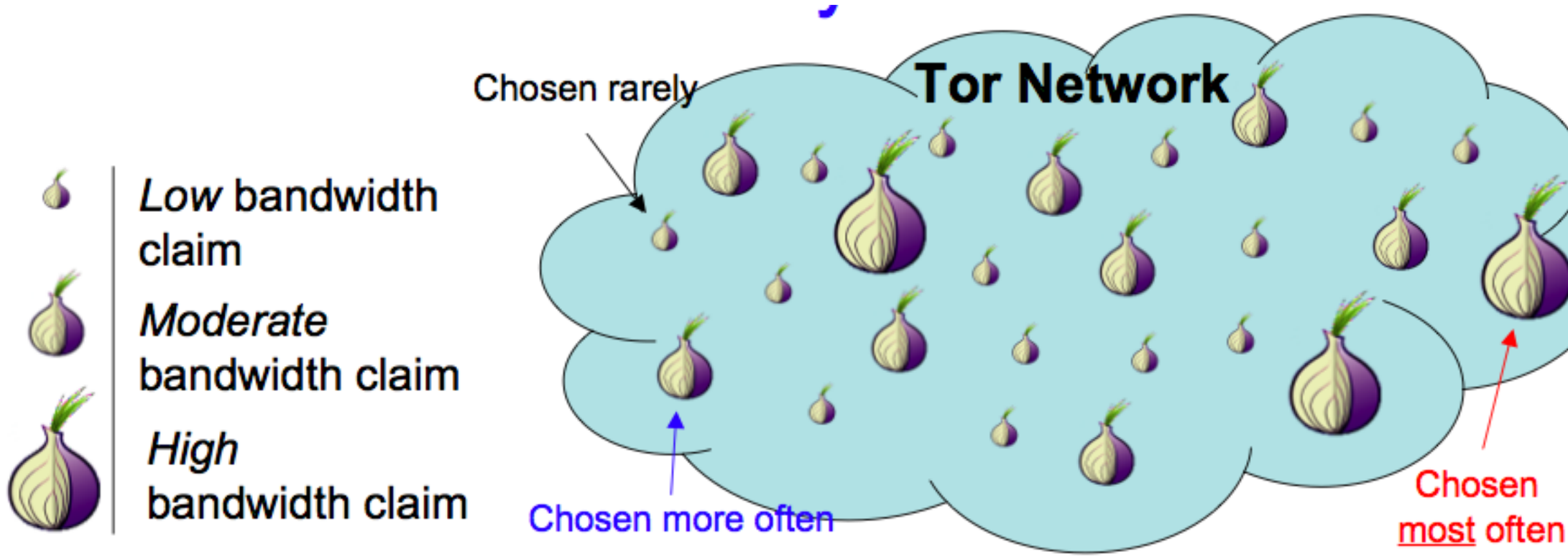
- Øverlier and Syverson demonstrated that an adversary can force new circuits to be built until it controls an entry router
- Entry guards were introduced in May 2006 to limit the likelihood of a malicious Tor router existing at the entry position

Routing in Tor
Part 1: Entry Guard Selection



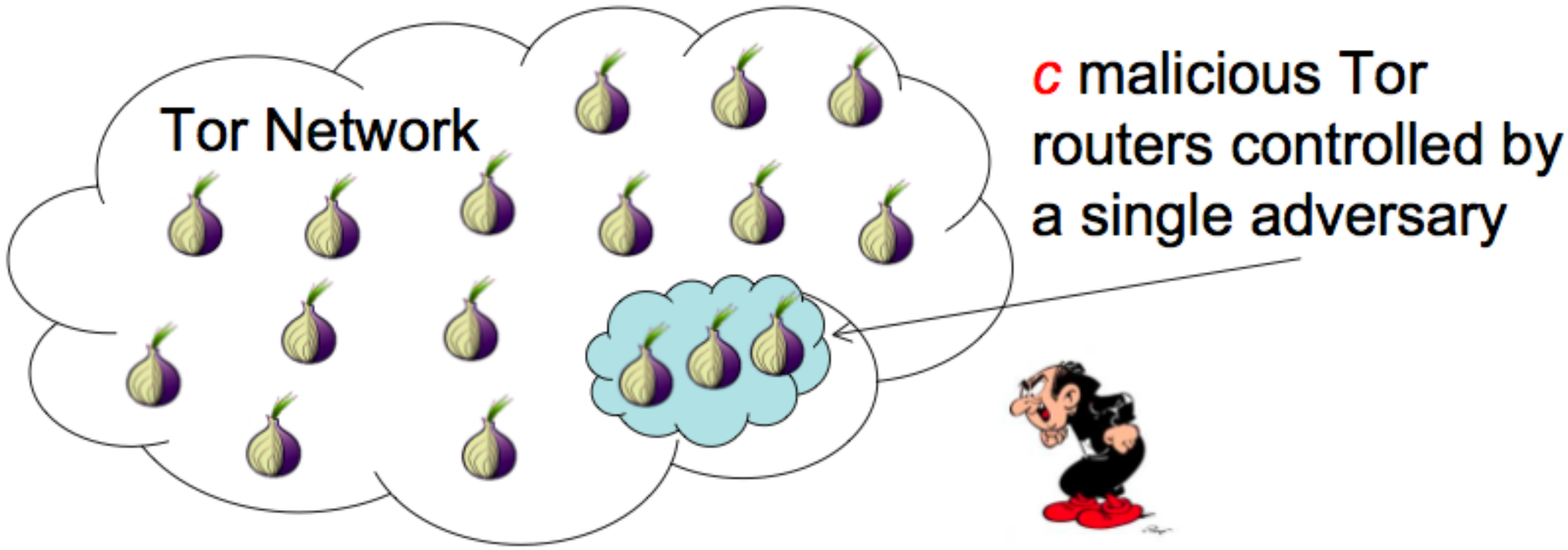
- Tor routers advertise their own bandwidth capabilities and uptimes to the trusted directory servers
- Directory servers mark routers as a potential "Guard" if they advertise uptime and bandwidth at or above the median for all routers; only these routers may be entry guards

Routing in Tor
Part 2: Non-Entry Node Selection



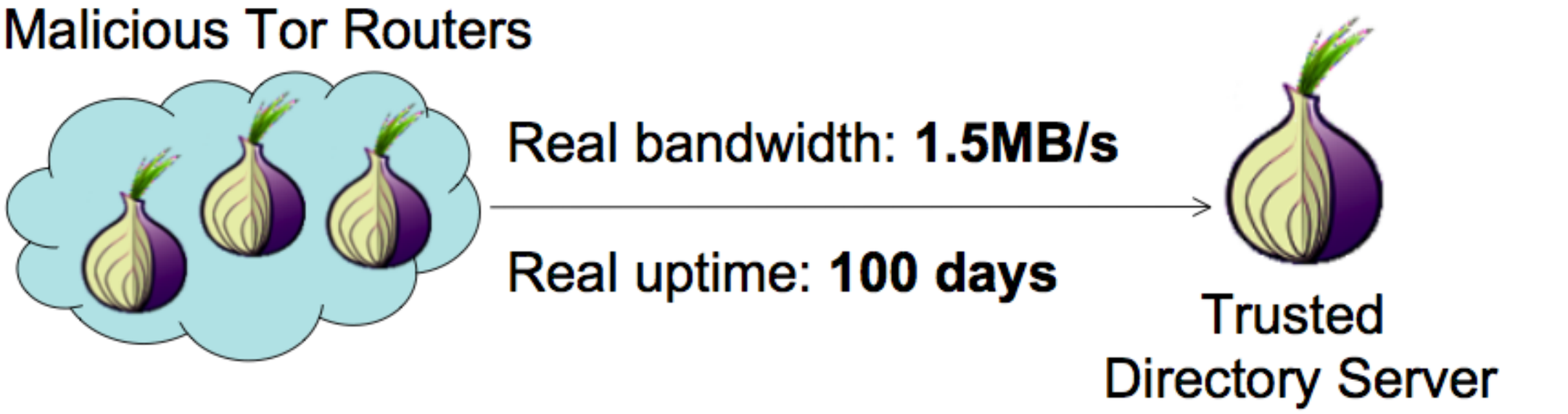
- Routers for the middle/exit positions are chosen to balance the traffic according to each node's bandwidth history; this allows for relatively low latency and high throughput service
- Key feature: Nodes claiming high bandwidth are chosen most frequently

Attack Model



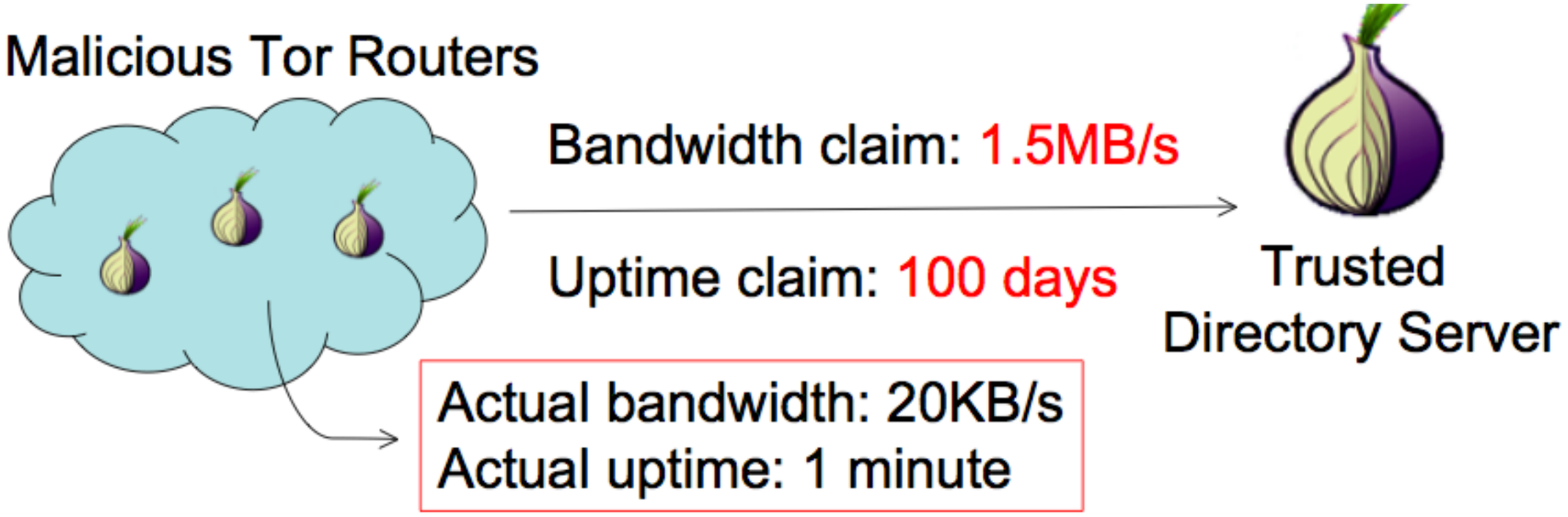
- Adversary: Our attacks assume a low-bandwidth (i.e., residential cable modem) non-global passive adversary controlling n >> c > 1 malicious Tor routers
- Client: The clients run only as Tor proxies (default setting) and join the Tor network for the first time

Compromising Anonymity: Basic Attack



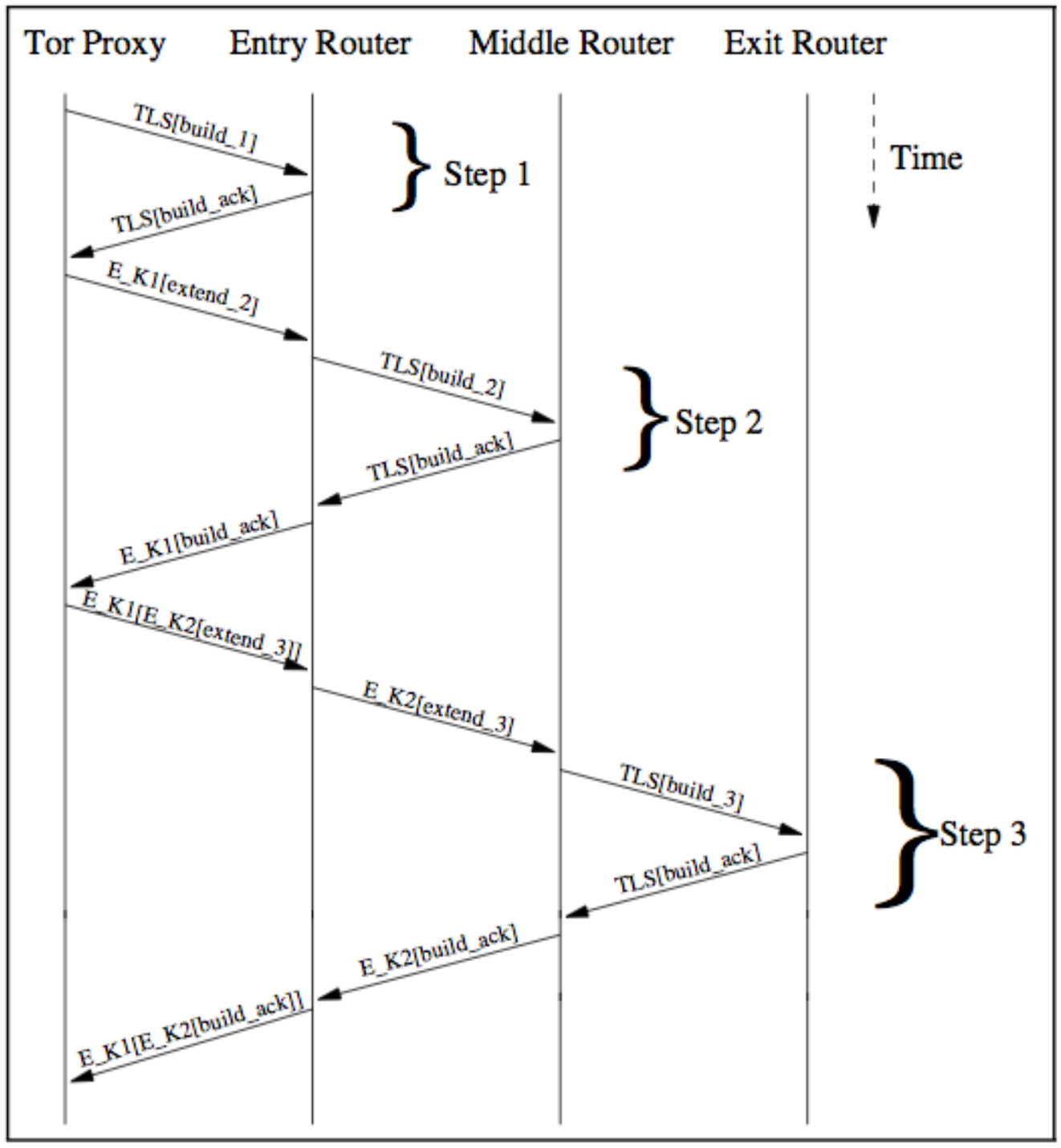
- Very simple: An adversary deploys c high-bandwidth (“fast”) and high-uptime (“stable”) Tor routers; under routing model, increase likelihood of controlling an entry/exit pair

Compromising Anonymity: Resource Reduction



- Any router can claim up to 1.5MB/s and high uptimes
- Currently, Tor directory servers do not verify bandwidth and uptime claims
- Focus malicious router's real bandwidth on accepting new clients and/or targeting specific clients or destinations

New Circuit Linking Technique



- Tor circuits are built deterministically; steps 1-3 must occur in order
- This allows circuits to be correlated before any payload data is sent
- Also allows for circuits to be quickly disrupted, if only one node is malicious on the circuit

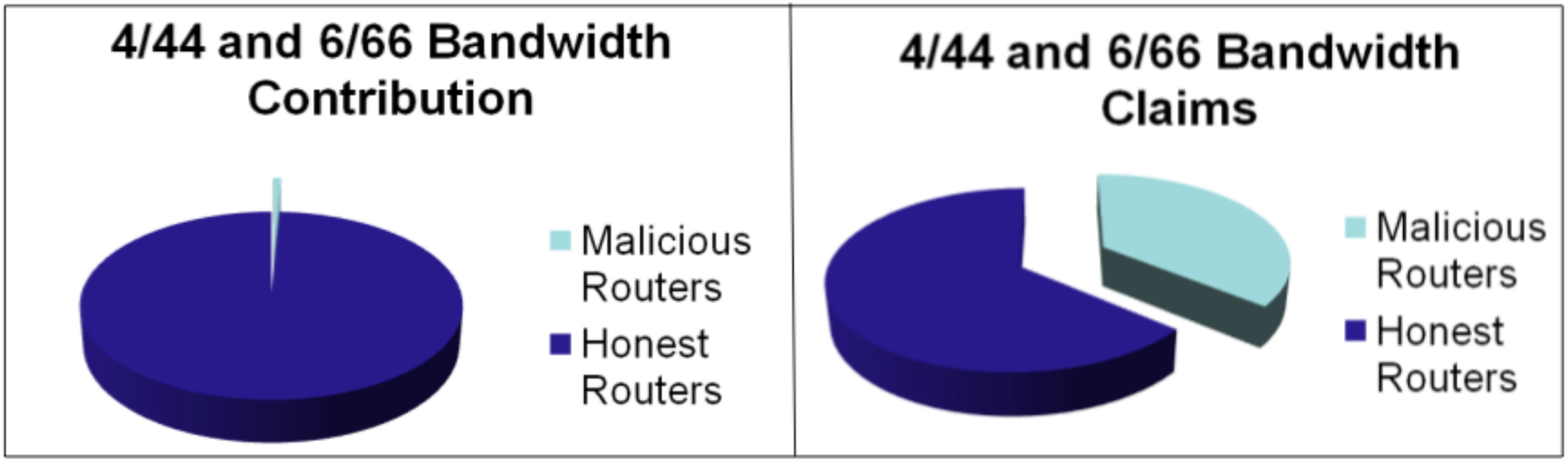
Experimental Methodology

- We evaluated the resource-reduced attack on two isolated Tor deployments with 40 and 60 routers using Tor 0.1.1.23 (August 2006)
- Evaluating the bandwidth distribution from the real network provided a realistic model for the experimental deployments
- We generated traffic using HTTP requests from 60 (40 node network) and 90 (60 node network) clients for two hours

Tier	**Tor Networks**		
	Real Tor	40 Node	60 Node
996 KB	38	4	6
621 KB	43	4	6
362 KB	55	6	9
111 KB	140	13	20
29 KB	123	11	16
20 KB	21	2	3
Total	103.9 MB	10.4 MB	15.7 MB

Malicious Router Configurations

- Varied the number of malicious routers (2/42, 4/44 and 3/63 and 6/66); malicious nodes advertised 1.5MB/s and large uptimes and were limited to 20KB/s



- Malicious routers contributed 0.3-0.8% of network's bandwidth, but advertised 22-36% of the total bandwidth
- Modified malicious Tor router code to prioritize circuit-building requests over data packets

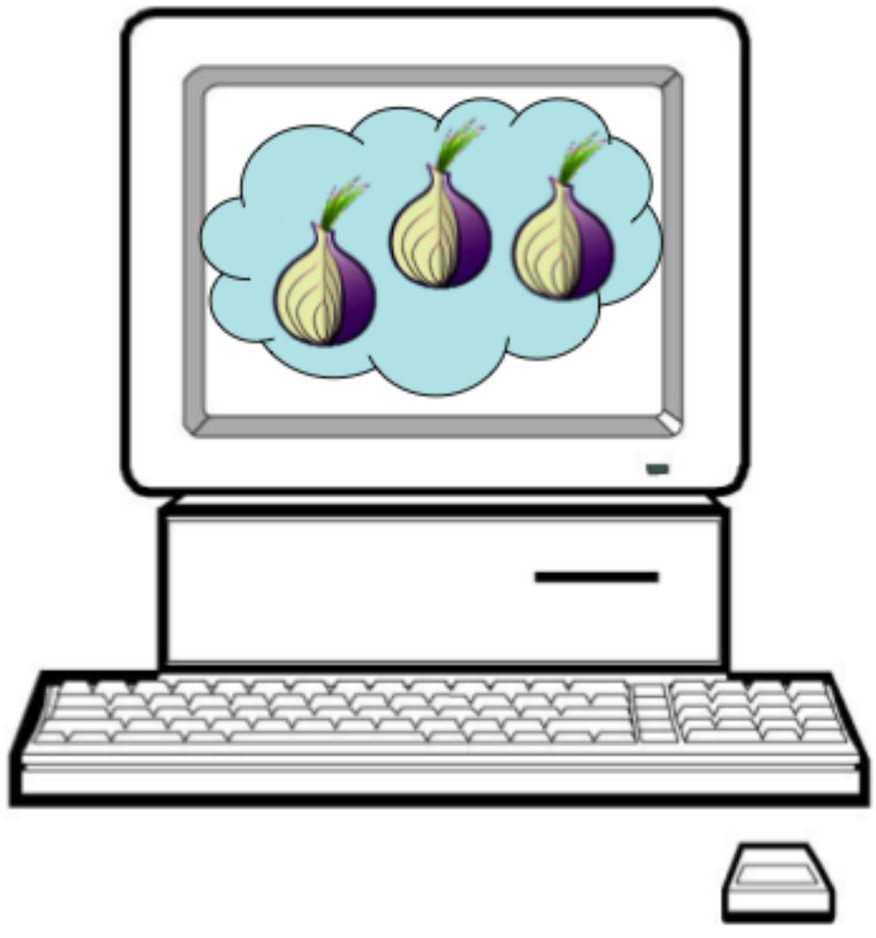
Experimental Results

- "Random Selection" expectation based on (c/n)^2
- The experimental results show a significant increase over the expectation if routers were chosen uniformly at random
- Tor's load balancing optimizations introduce opportunities to reduce the system's anonymity

	Experiments	
	2/42	4/44
Random Selection	0.12%	0.63%
Experimental	8.90%	33.55%
Improvement	7,565%	5,190%

	Experiments	
	3/63	6/66
Random Selection	0.15%	0.70%
Experimental	11.06%	46.36%
Improvement	7,097%	6,530%

Attacking Entry Guards



- There is currently no limit to how many routers may run on a single host
- Run several routers on a local machine, advertising high bandwidths and uptimes
- The global median uptime value will increase to that of the malicious routers
- Now only malicious nodes can be entry guards
- Resource claims must be trusted

Solutions: Raising the Bar

- Idea: The directory servers should monitor uptime (easy) and bandwidth (more difficult)
- Tor specification proposal #107: Uptime sanity checking
- Tor specification proposal #108: Base ‘Stable’ flag on mean time between failures

- Idea: Limit the number of Tor routers that may be run at an IP address
- Tor specification proposal #109: Two routers per IP

- Each proposal will be implemented in a future Tor release

Acknowledgments/References

- [Shmatikov] CS 378 - Network Security and Privacy, Vitaly Shmatikov, University of Texas at Austin, Fall 2007.
- [Song] Lei Song, Michigan Tech University, summer seminars 2007.
- [Bauer] Presented by Kevin Bauer at Workshop on Privacy in the Electronic Society Alexandria, VA, USA , October 29, 2007 (Low-Resource Routing Attacks Against Tor, Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker, 2007)