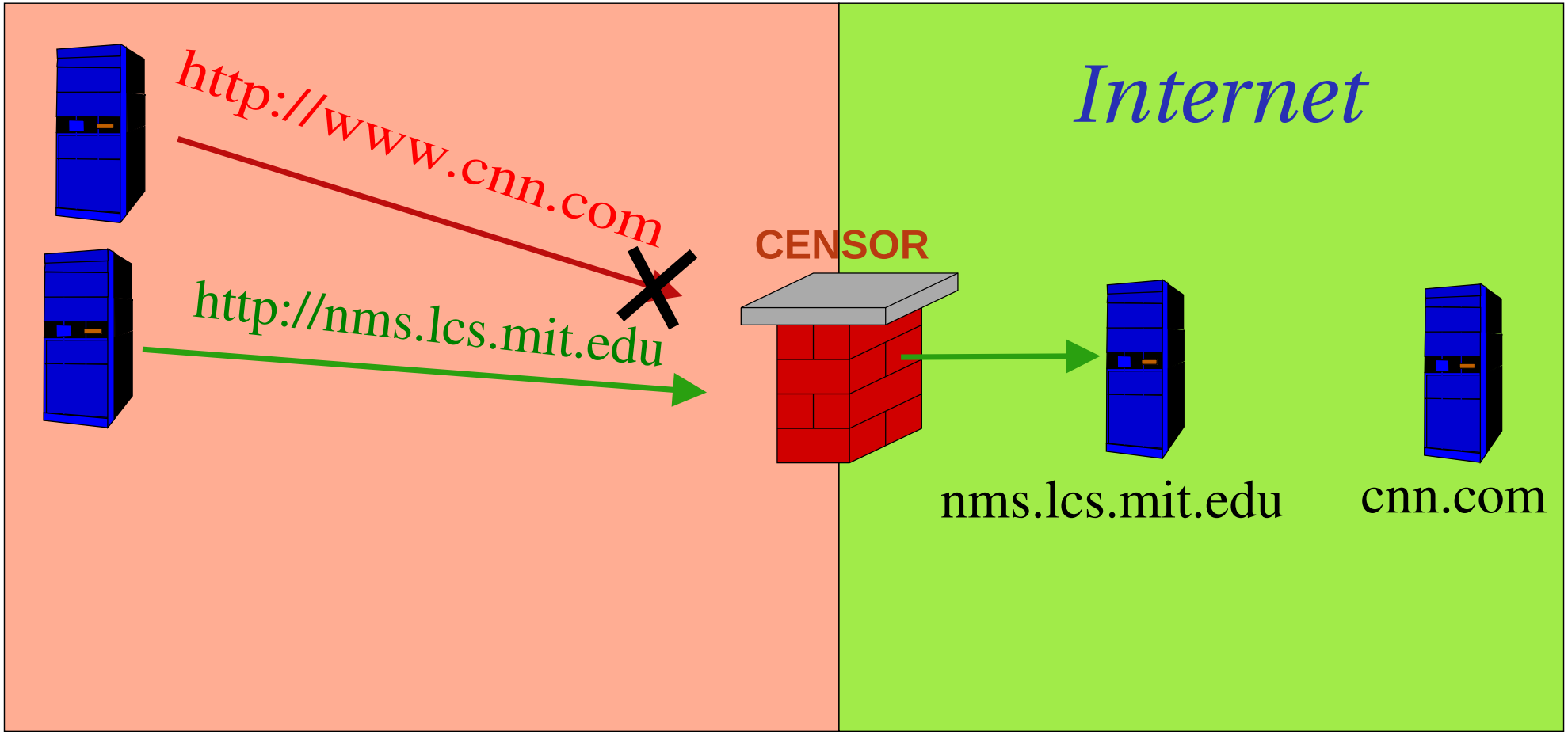
Infranet: Circumventing Web Censorship and Surveillance

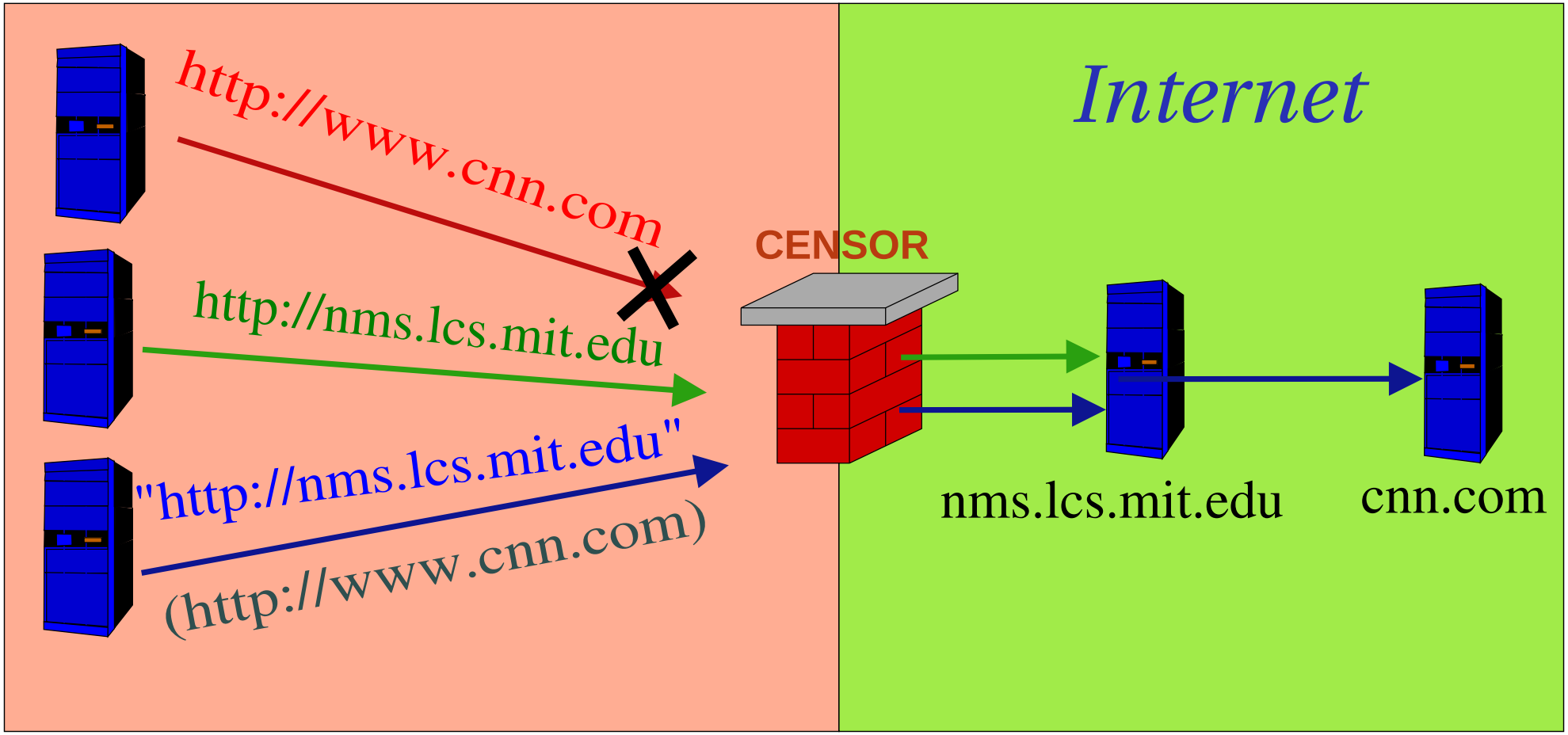
Nick Feamster, Magdalena Balazinska,
Greg Harfst, Hari Balakrishnan, David Karger
M.I.T. Laboratory for Computer Science

http://nms.lcs.mit.edu/infranet/

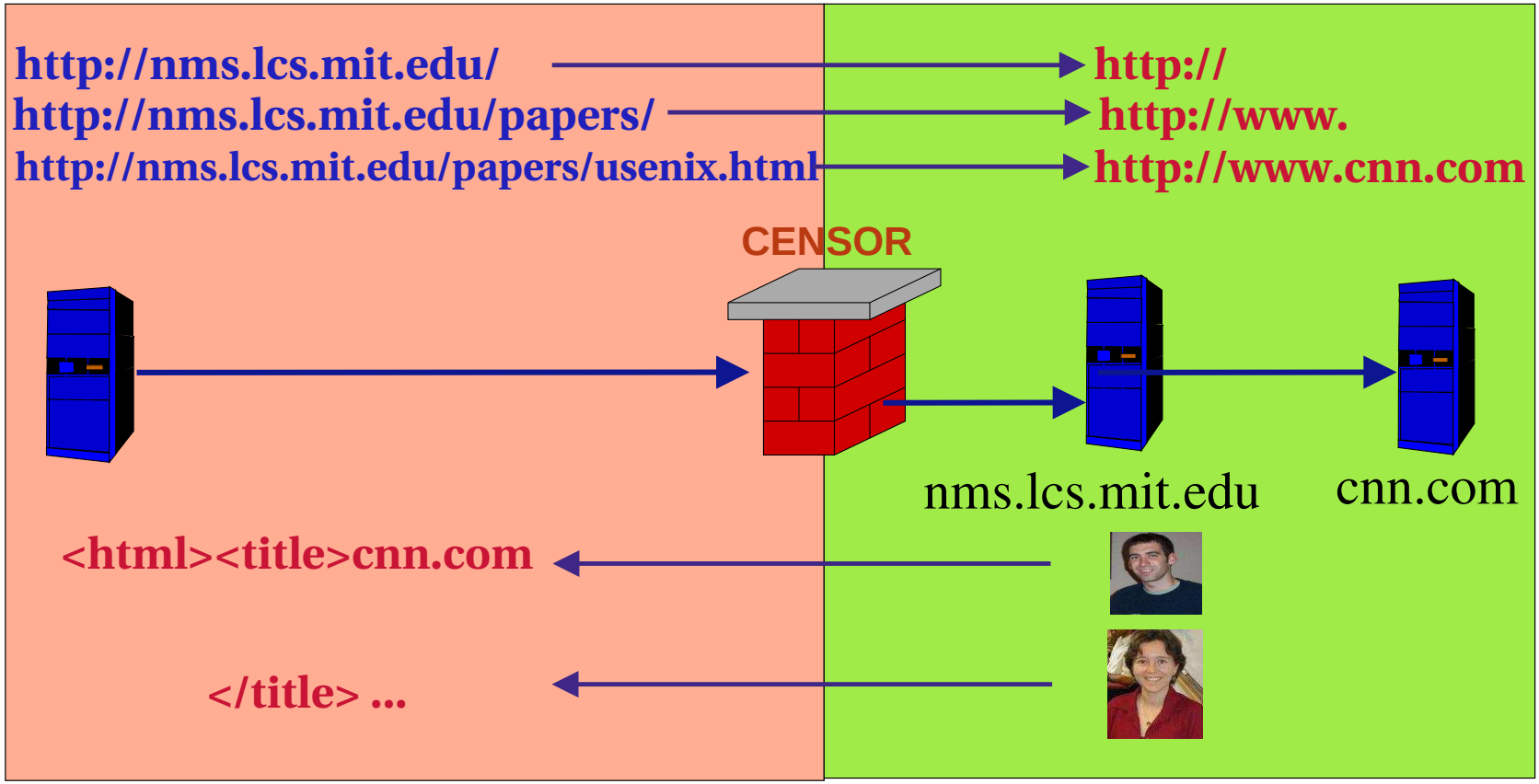
The Big Picture



The Big Picture



How Infranet Works



- Use Infranet requester proxy (on localhost)
- Upstream request in sequence of HTTP requests
- Downstream response in images

Censor

- Restrictive government, corporate firewall, etc.
- Discovery Attacks: Notice unusual-looking Web traffic.
  - monitors Web access for "inappropriate use"
  - watch Web traffic for inappropriate access attempts
  - *watch for suspicious looking Web access patterns*
  - *watch for use of circumvention software*
- Disruptive Attacks: Keep the endpoints from talking.
  - blocks access to certain Web sites
  - *attempts to block access to circumvention software (e.g., blocking SSL, disrupting communication, etc.)*

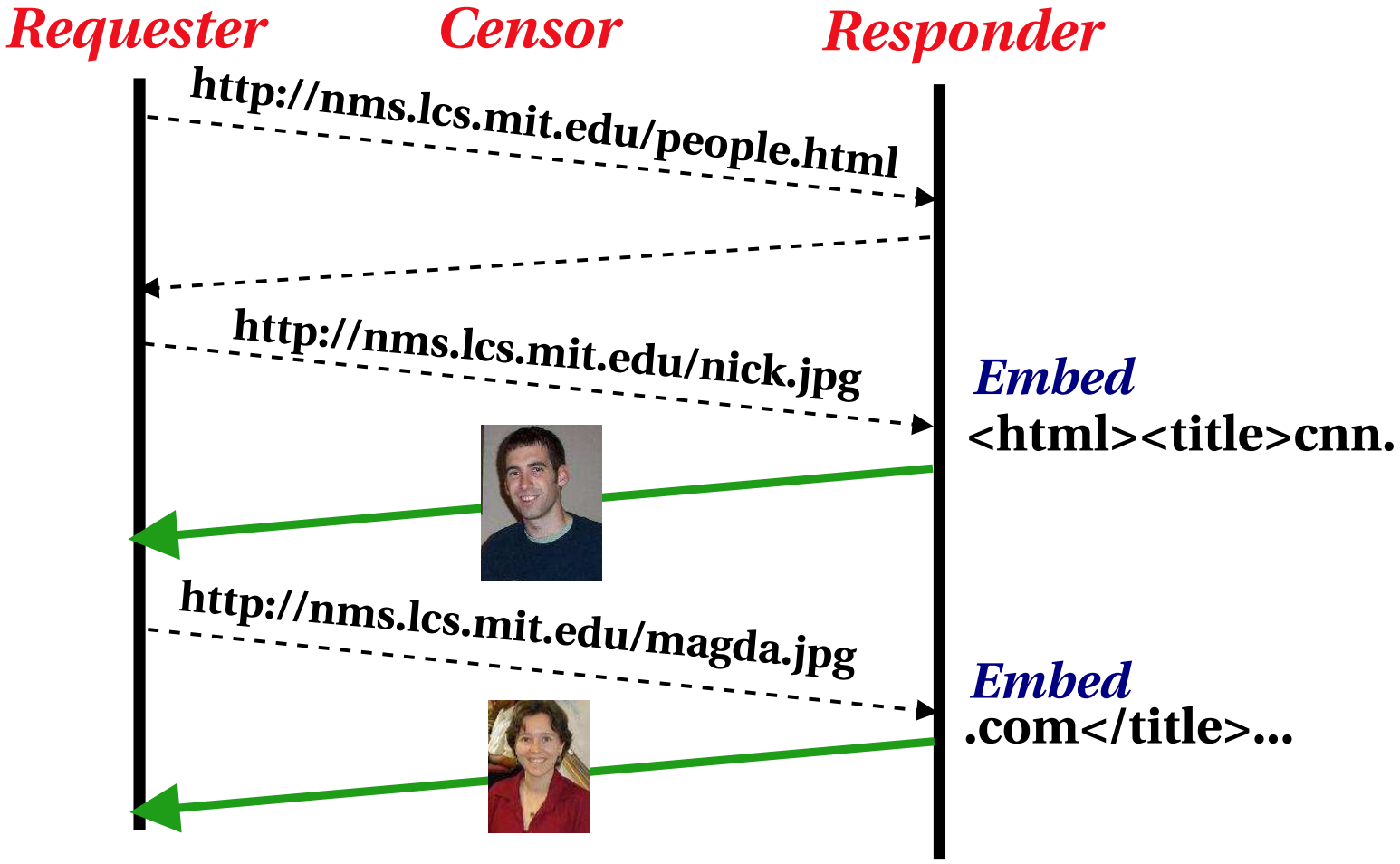
Design Goals

- *Deniability for clients*
  - Can’t confirm that a client is intentionally retrieving censored data
- *Statistical deniability for clients*
  - Web traffic doesn’t look unusual
- *Covertness for servers*
  - Can’t discover a server that is serving censored content
  - Defense against blocking
- *Communication Robustness*
  - Should be difficult to disrupt request/transfer of censored content
- *Reasonable Performance*

Related Systems: Triangle Boy, Peekabooty, etc.

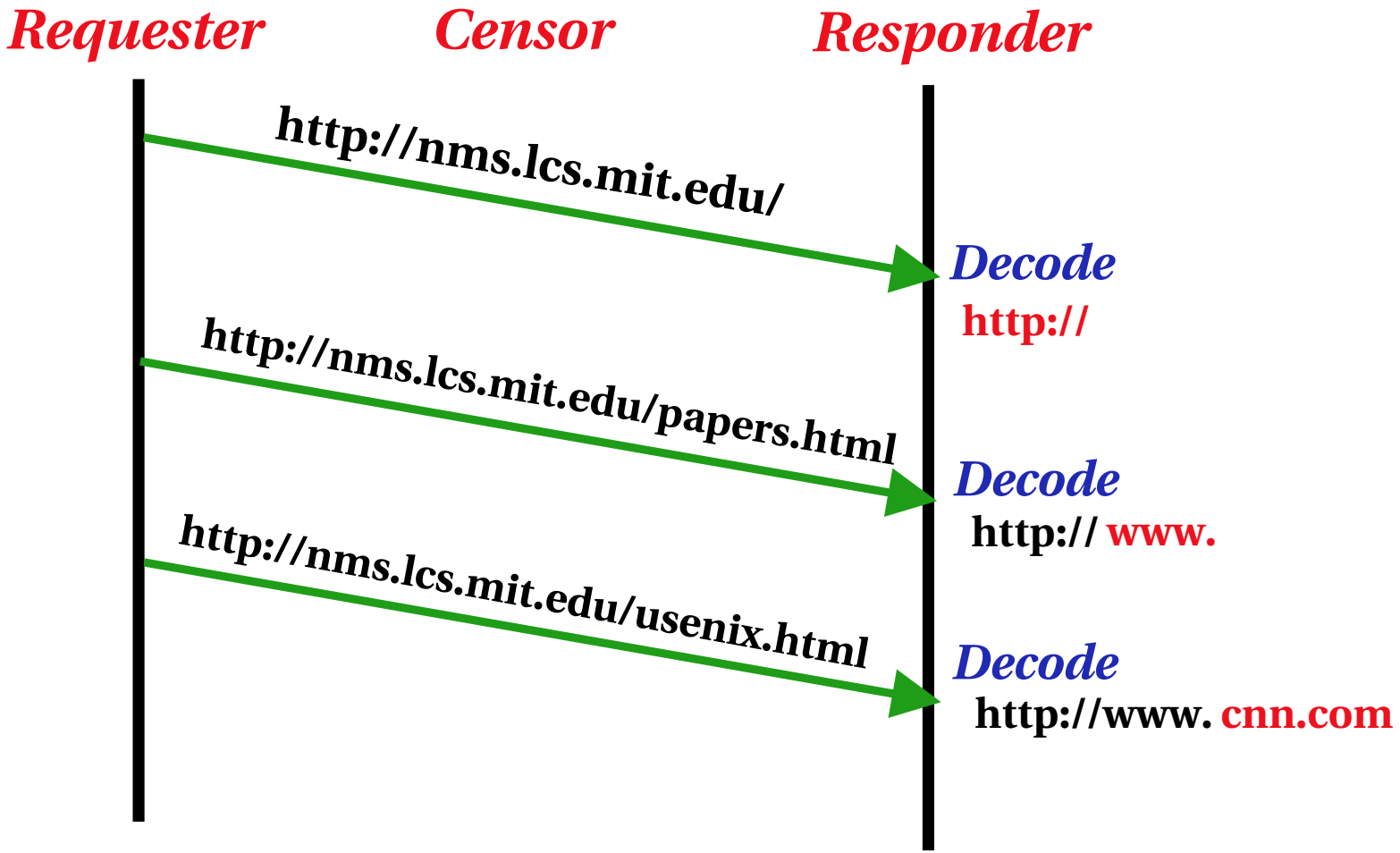
- *Deniability for clients*
  - Existing systems rely on SSL, vulnerable to fingerprinting
- *Statistical deniability for clients*
  - SSL traffic looks suspicious
  - No attempt to conceal suspicious traffic patterns
- *Covertness for servers*
  - Servers make no attempt to conceal their existence
  - Suspicious traffic patterns may result in discovery and blocking
- *Communication Robustness*
  - SSL can be blocked (e.g., unsigned server certificates)

Downstream Communication ("Downloading")



- Embed data in images, recover by shared secret
- Steganography is not ideal: can't reuse cover image
- Web cams are wonderful.

Upstream Communication ("Requesting")



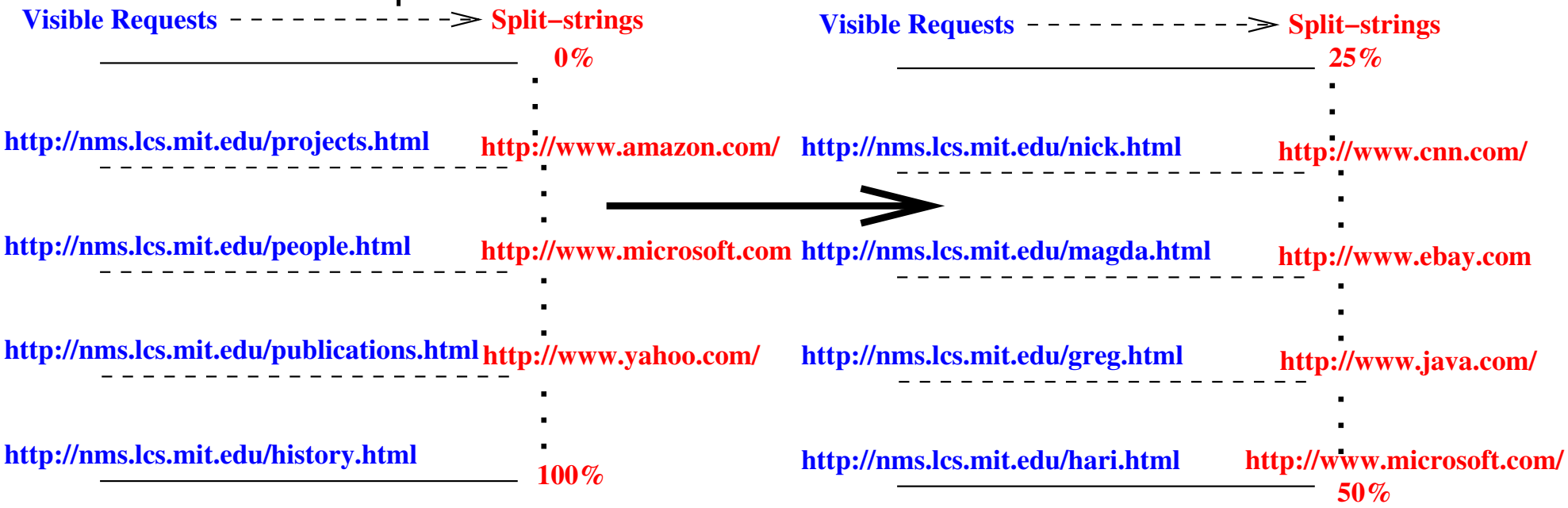
- Hidden message => sequence of HTTP requests
- Mapping function: secret, critical to deniability

Simple Schemes: Covertness/Bandwidth

- Odd/Even Links
  - *Covertness:* Requester may ask for any one of half of the links at any given time
  - *Bandwidth:* 1-bit per visible HTTP request
- Links modulo k
  - *Covertness:* Requester asks for any of N/k links
  - *Bandwidth:* lg(k) bits per visible HTTP request
- Static Mapping
  - *Covertness:* potentially quite bad...
  - *Bandwidth:* M bits per request

Range-Mapping: Web Surfing, 20 Questions-Style

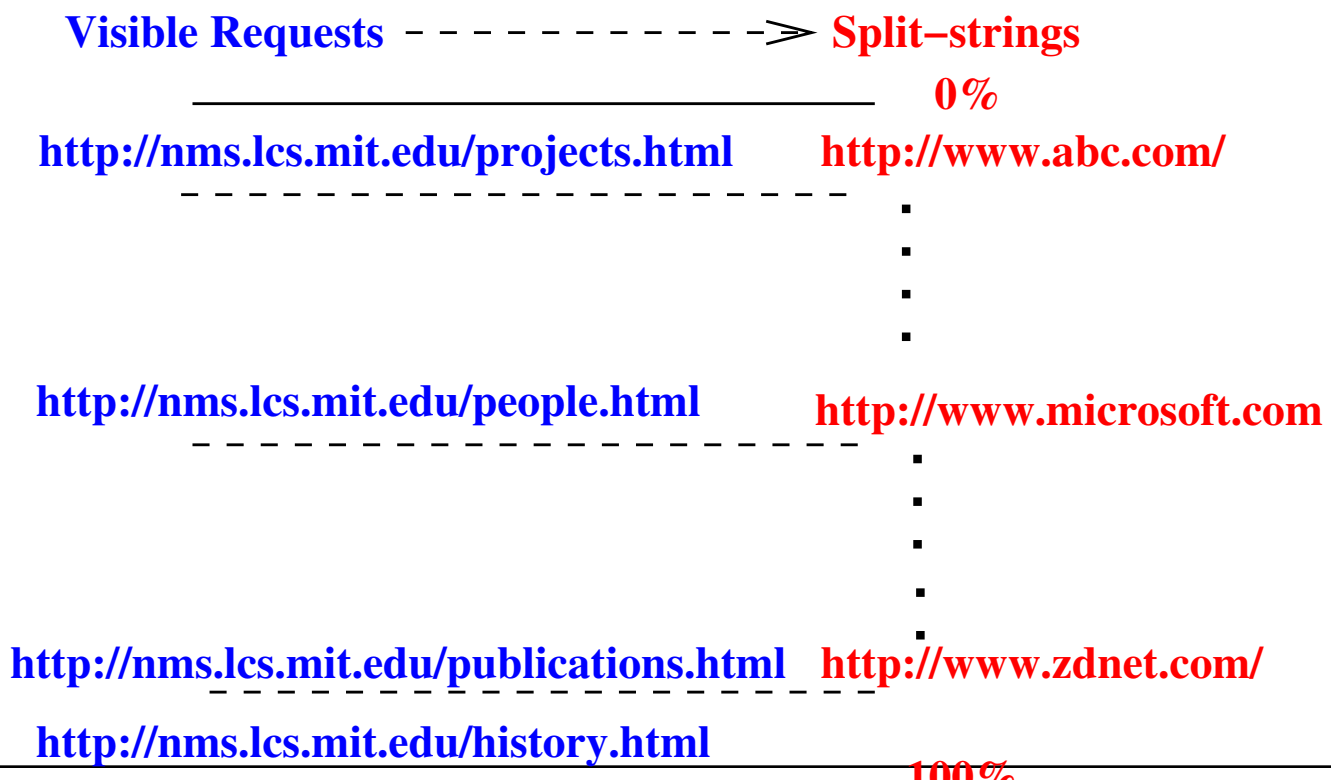
- Assume: Some set of censored URLs are commonly requested
- Responder tells requester
  - the boundaries (split-strings) for ranges in this set, and
  - the mapping between visible HTTP requests and split-strings
- Requester tells responder
  - a visible HTTP request



but...not all requests are equally likely!

Getting Statistical Deniability

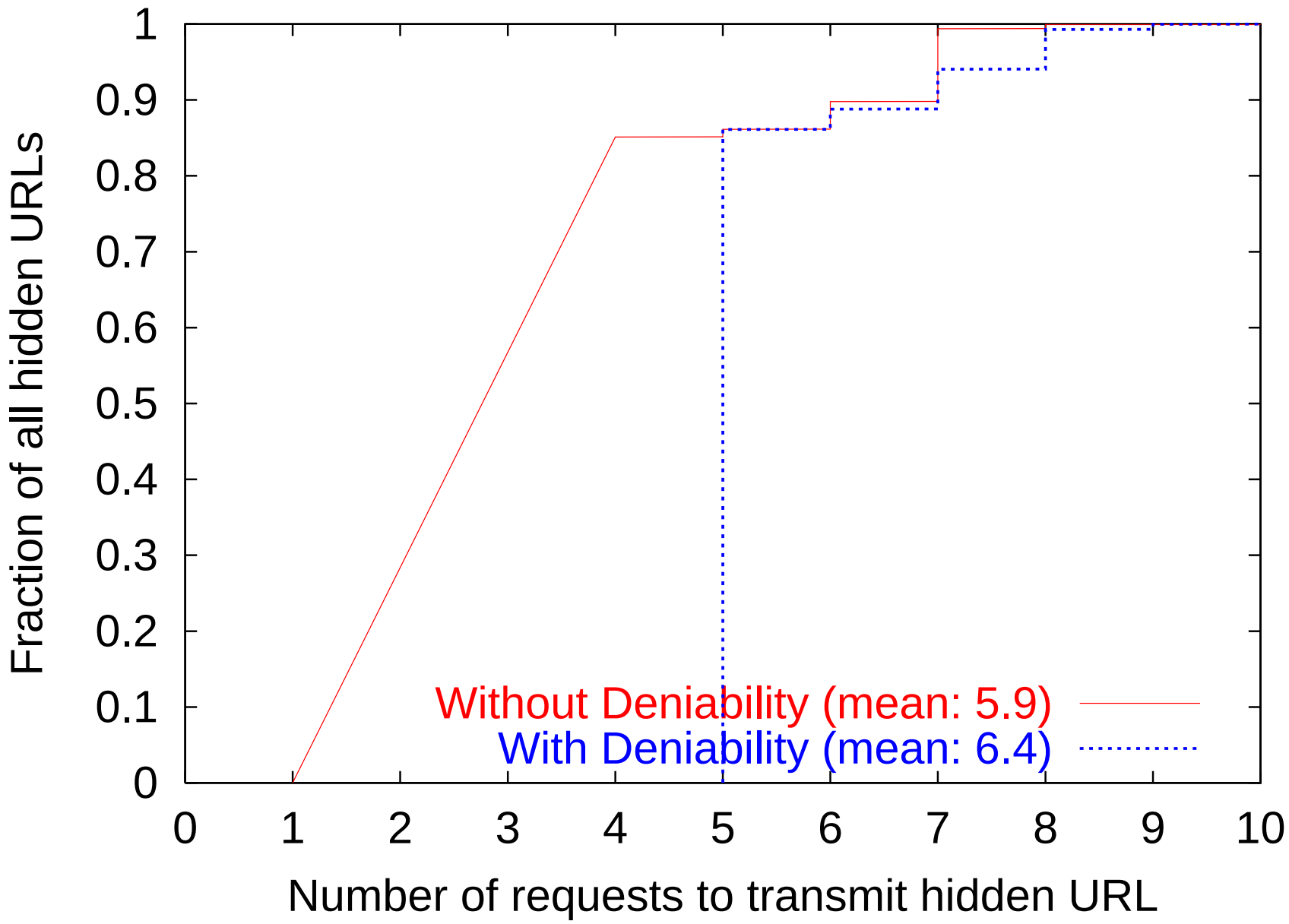
- Divide the corpus according to more likely visible HTTP requests.
- Alphabetic coding says that our expected number of requests is the same!



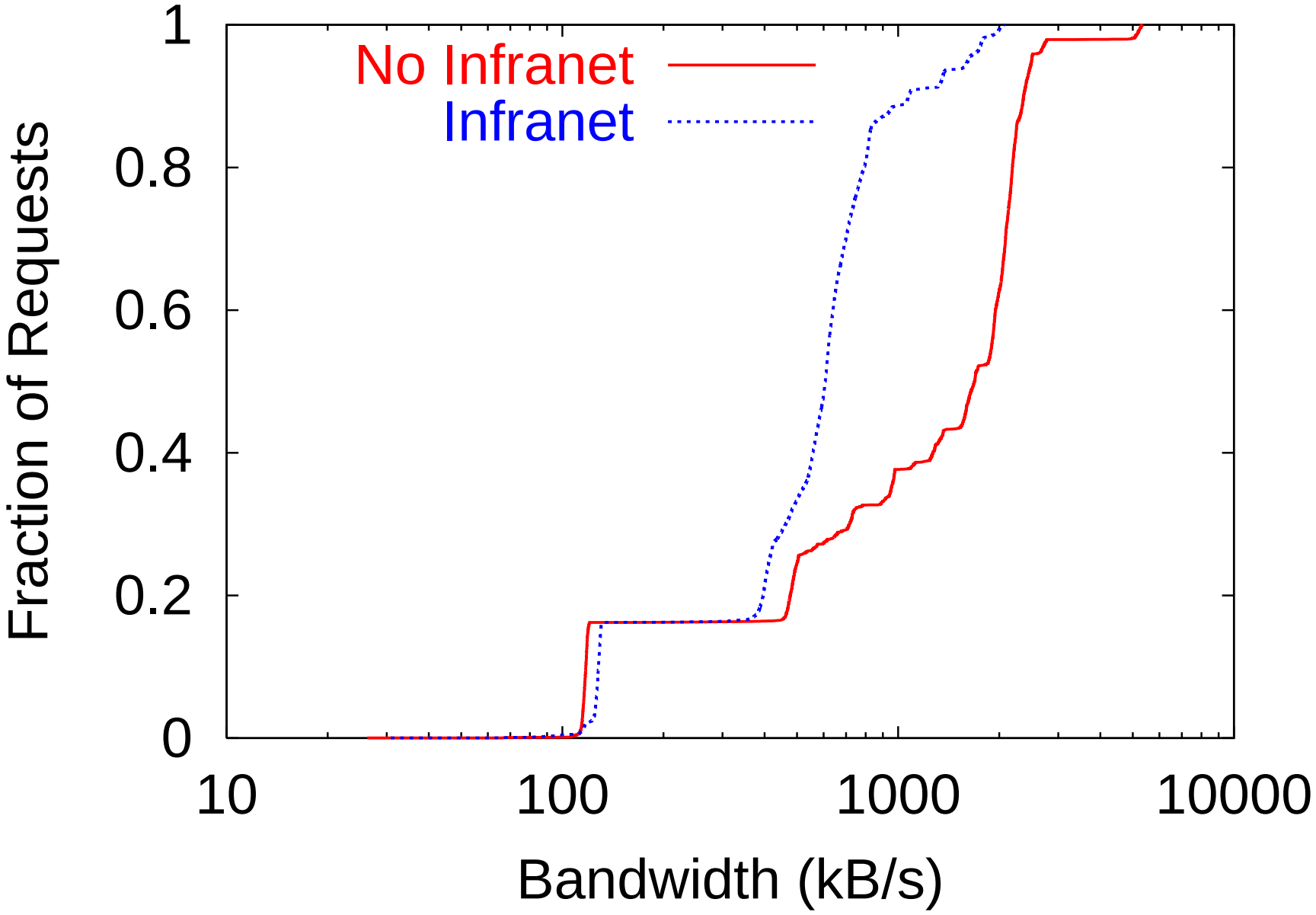
Range-Mapping

- Search through set of frequently-requested censored URLs achieves good upstream bandwidth.
- Division of ranges according to conditional request probabilities achieves deniability and covertness.
- Idea can be applied over the space of all strings.

Statistical Deniability is Free



Server Covertness is Not Free



Conclusion

- Infranet hides censored requests and responses in innocuous-looking HTTP request/response streams
  - client deniability
  - server covertness
  - reasonable robustness
- Future work
  - robustness
  - software distribution
  - server discovery

http://nms.lcs.mit.edu/infranet/