Privacy-preserving datagram delivery for ubiquitous systems

David Evans

May 21, 2008

Outline

1. Introduction
2. Our approach
3. Evaluation
4. Conclusions

1. Introduction

The Problem

- Many useful ubiquitous (and transport) applications are context-aware
- Context-aware applications build models of the world
- These models contain information about people
- People worry about…
  - Where this information is stored
  - Who gets to see it
  - How it is used

Datagrams Are Desirable

- Defer retransmission strategy to applications
- Applications with sensors have to handle missing data anyway
- Improved privacy properties compared to streams

Anonymous Datagrams

- Lots of work on anonymous comms (Mixminion, I2P, Tor, ...)
- Work supports either high latency (for, *e.g.*, email) or TCP or is written by d00dz (I2P)
- Goal: build a real (albeit toy) anonymous datagram service and evaluate its performance

2. Our approach

What We Did

- Modified Tor to support UDP
- Only ingress and egress nodes need modification
- Intermediate nodes can't tell whether they are forwarding UDP or TCP traffic

A SOCKS Primer

1. Application requests a *datagram association*
2. Server evaluates the request and responds
3. Application sends its first datagram
4. Server sets up state to forward the datagram and any replies
5. Application tears down the association

Tor Terminology

Circuit A path through the overlay network from ingress to egress nodes

Stream The state an ingress node needs to forward data

UDP With Tor

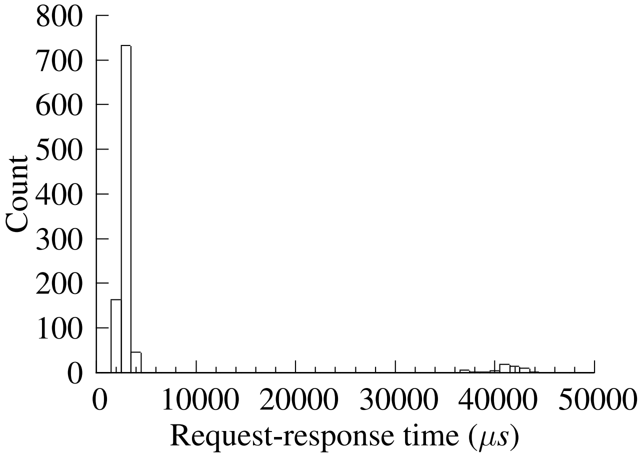
- Beefed up SOCKS support to handle UDP
- Map each datagram association to a “pseudo-stream”
- Use the forwarding internals without change

Tor's Congestion Control

- Aims to protect both the underlying and overlay networks
- Uses transmission windows per stream and per circuit
- Drop datagrams if the circuit window would close
- No congestion control of pseudo-streams

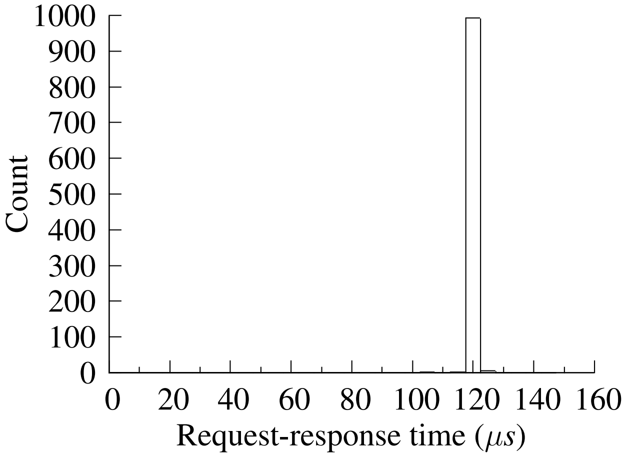
3. Evaluation

Request-response time



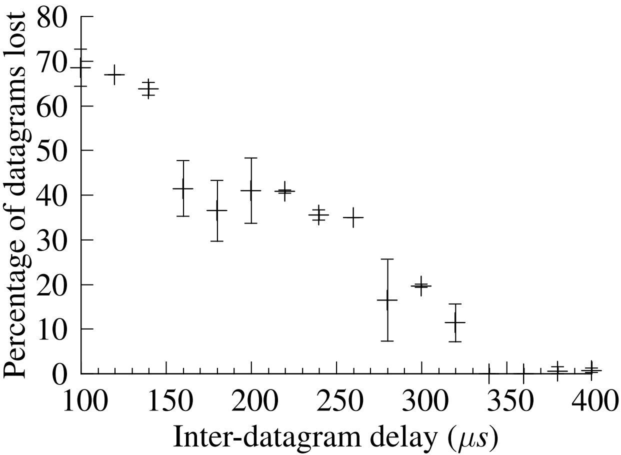
$$\text{Mean} \;=\; 5681\ \mu s$$

Request-response time

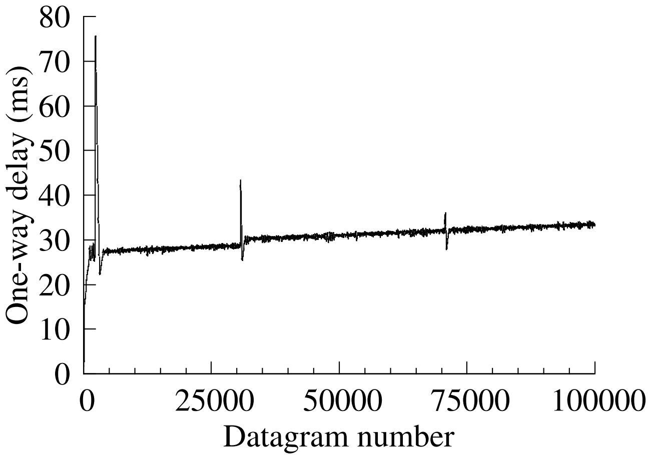


$$\text{Mean} = 124.9\ \mu s$$

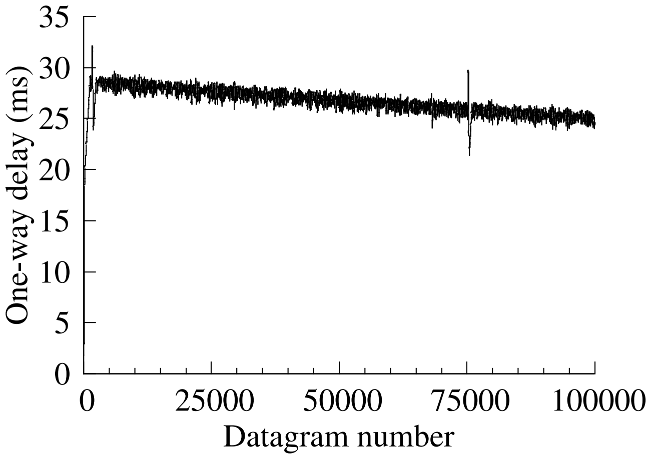
Percentage packet loss



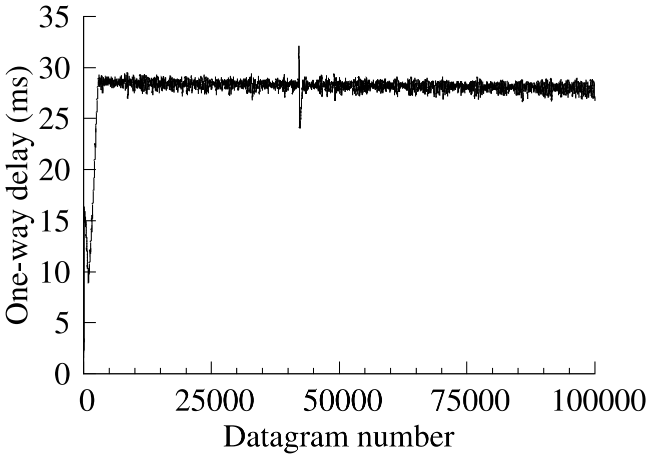
One-way delay, replication one



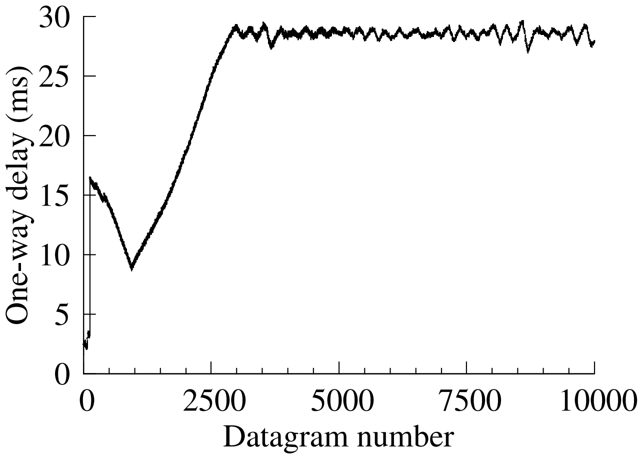
One-way delay, replication two



One-way delay, replication three



One-way delay, replication three



4. Conclusions

Contributions

- Argued that anonymous datagrams are useful for a spectrum of ubiquitous applications
- Provided and evaluated a toy implementation that is incrementally deployable
- Illustrated that the cost may not be all that high
- A more clever solution would have to be justified

The Future

- Better implementation (improved administration, *etc.*)
- Better guarantees that datagrams are good citizens
- Use of non-interactive key exchange protocols for circuit building
- Incorporation into TIME