# Datagram Testing Plan

Steven J. Murdoch
2012-03-16

## 1 Introduction

This document discusses current considerations and plans for evaluating approaches for moving to a datagram transport in Tor. For background, refer to Murdoch [3].

### 1.1 Simulation vs Emulation

An important design choice for the experiments is whether we run experiments on simulated Tor software, or on the real (but modified) Tor implementation. The advantage of the former is that it should be easier: only the timing of cells needs to be simulated and nodes have full visibility of the state of other nodes. The advantage of the latter is that the simulation is more realistic.

In the absence of a suitable model of Tor, it has been decided to modify the existing Tor implementation. The Tor network can then be emulated using ExperimenTor [1] and Shadow [2]. This does require implementing a datagram variant of Tor sufficiently well to allow clients to route traffic with realistic performance characteristics.

### 1.2 End-to-end vs Hop-by-hop

A design choice for a datagram variant of Tor is the end points of the reliability protocol. Options include OP to destination server (i.e. Tor routes IP packets), OP to exit node (i.e. the exit node reassembles TCP streams and re-segments them), or hop-by-hop (i.e. each node re-assembles streams). OP to destination server is the most versatile, but makes implementing exit-policies difficult. OP to exit node is the closest approximation of the end-to-end principle for Internet design, but means that any dropped packet will need to be retransmitted all the way along the circuit. Hop-by-hop is not as versatile as either of the end-to-end variants, but by providing a reliable tunnel to all nodes it means that Tor's existing cell cryptography and relay protocol could remain unchanged.

The ease of implementation, coupled with the fact that evaluation has shown good results from the hop-by-hop approach make it the most suitable for further development. End-to-end variants may however still be worthwhile to examing if time is available.

### 1.3 Transport Protocol

Reardon's original implementation used the TCP Daytona stack, which is not available under a Tor compatible license and so cannot be used. However there are a number of more suitable alternatives.

#### 1.3.1 µtp

The libutp implementation is widely deployed in BitTorrent, although its performance goal of yielding to TCP is not ideal.

#### 1.3.2 FreeBSD network stack

The FreeBSD network stack is being ported to userspace by Kip Macy, and will provide TCP and SCTP.

#### 1.3.3 SCTP

A userspace SCTP stack is available[1].

#### 1.3.4 CurveCP

The CurveCP stack is available, although has not had as extensive testing as the other options.

## 2 Development plans

Future plans will need to encompass four areas.

### 2.1 Network model

A model of the simulated Tor network will need to be developed. This should include node characteristics (CPU, possibly memory) and network characteristics (capacity, latency, other traffic usage).

### 2.2 User/server model

A model will need to be developed of the demands users put on the network, and how servers respond to requests.

### 2.3 Tor implementation

Tor will need to be updated to use each of the transport protocols to be tested.

### 2.4 Metrics and Experimentation

Finally, performance metrics will need to be developed, and experiments run.

## 3 Schedule and participants

Participants who have offered to help are:

- Rob Jansen (Shadow) working with Kevin Bauer (ExperimenTor) on modeling Tor.
- Kevin and Mashael AlSabah will be getting Joel Reardon's code working on current Tor
- Rob will be focusing on hop-by-hop transports, starting with μTP. He is currently busy on another project, but will be able to return in early May.
- Mashael will focus on end-to-end transports, including analyzing how it affects security
- Steven Murdoch will try to pull everything together and do everything else

---

[1] `http://sctp.fh-muenster.de/sctp-user-land-stack.html`

# References

[1] Kevin Bauer, Micah Sherr, Damon McCoy, and Dirk Grunwald. ExperimenTor: A testbed for safe and realistic Tor experimentation. In *4th USENIX Workshop on Cyber Security Experimentation and Test*, August 2011. `http://crysp.uwaterloo.ca/software/exptor/`.

[2] Rob Jansen and Nicholas Hopper. Shadow: Running Tor in a box for accurate and efficient experimentation. In *Symposium on Network and Distributed System Security (NDSS)*, 2012. `http://shadow.cs.umn.edu/`.

[3] Steven J. Murdoch. Comparison of Tor datagram designs. Technical report, November 2011. `https://www.cl.cam.ac.uk/~sjm217/papers/tor11datagramcomparison.pdf`.