# A Practical Congestion Attack on Tor Using Long Paths

Towards De-anonymizing Tor

Nathan S. Evans[1] Christian Grothoff[1] Roger Dingledine[2]

[1]University of Denver, Denver CO
[2]The Tor Project

August, 12 2009

## Why attack Tor?

- Tor is the most popular and widely used free software P2P network used to achieve anonymity on the Internet:
  - Tor has a large user base
  - The project is well supported
  - Generally assumed to give users strong anonymity

Our results:

*All the Tor nodes involved in a circuit can be discovered, reducing Tor users level of anonymity and revealing a problem with Tor's protocol*

## Tor General Information

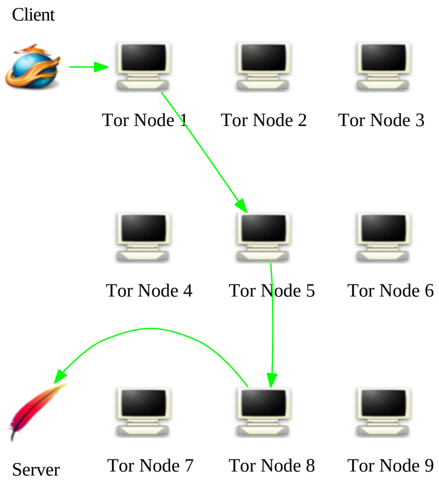
- Tor stands for “The onion router”
  - Encrypts data multiple times and is decrypted as it travels through the network a layer at a time: like peeling an onion
- Tor is a P2P network of mixes
- Routes data through network along a “circuit”
- Data is encrypted as it passes through nodes (until the last hop)

## Routing

- Data is forwarded through the network
- Each node knows only the previous hop and the next hop
- Only the originator knows all the hops
- Number of hops is hard coded (currently set to three)

Key security goal: No node in the path can discover the full path

## Routing Example



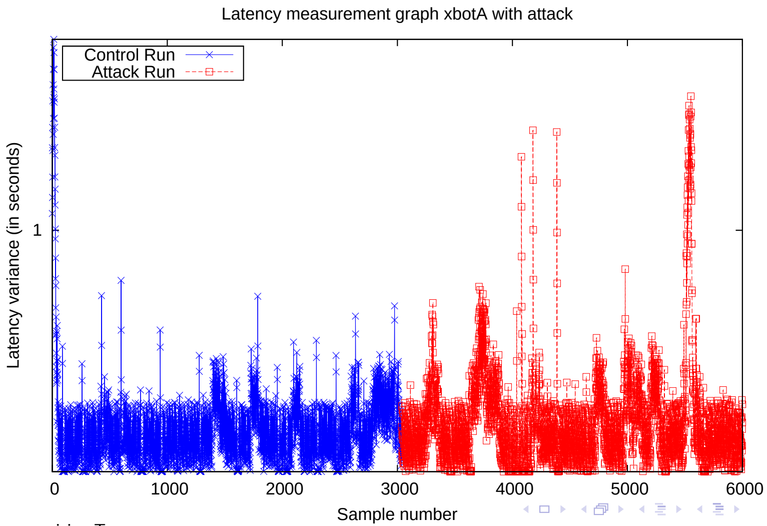
## Previous work

- Murdoch and Danezis wrote “Low Cost Traffic Analysis of Tor”
- Goal is to discover all the Tor routers involved in a given circuit
- Based on being able to tell the added load of one normal Tor connection
- Send a certain sequence down a tunnel, monitor each Tor router to see if it is involved
- Their attack worked reasonably well with the 13 Tor routers they used in 2005 (with 15% false negative rate)

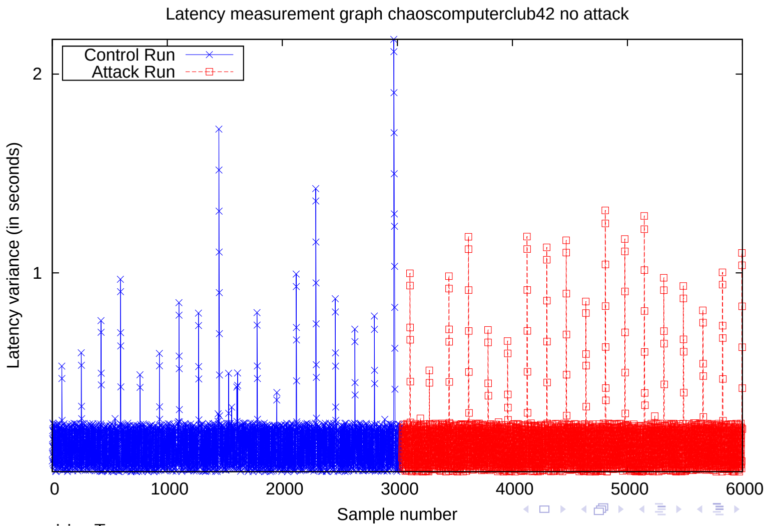
## Problems With Previous Work

- Too inaccurate with today's 1000+ routers
- Must identify all the separate routers in the circuit
- Attempting to measure small effects, large fluctuations that occur in actual current network give false positives
- We replicated their experiments, found method to be much less effective on today's network

## M and D Results - With Attack



## M and D Results - Without Attack



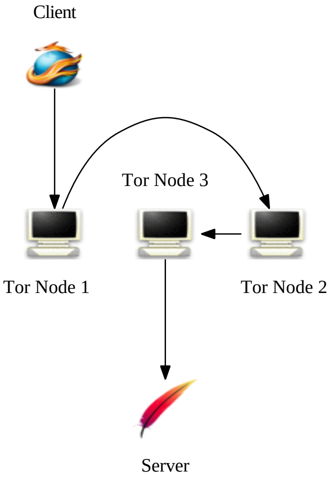
## M and D Testing

- Used same statistical methods for correlation
- Used same source code for attacks
- In our tests, highest correlations seen with false positives
- Attack may be viable for some Tor nodes
- Improved statistical methods may improve false positives

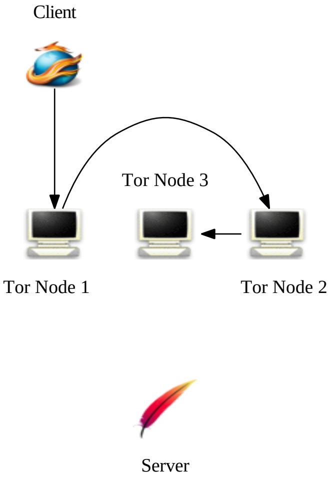
## Our Basis for Deanonymization

- Target user is running Tor with privoxy with all the default settings
- Three design issues enable users to be deanonymized
  1. No artificial delays induced on connections
  2. Path length is set at a small finite number
  3. Paths of arbitrary length through the network can be constructed

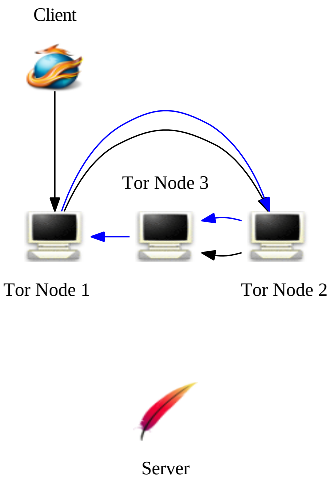
## Regular Path Example



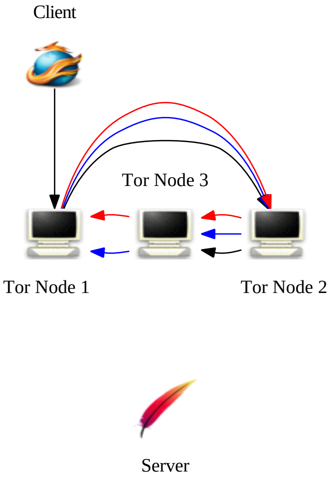
## Circular Path Example 1/5



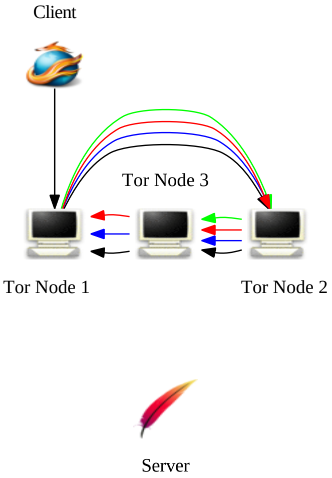
## Circular Path Example 2/5



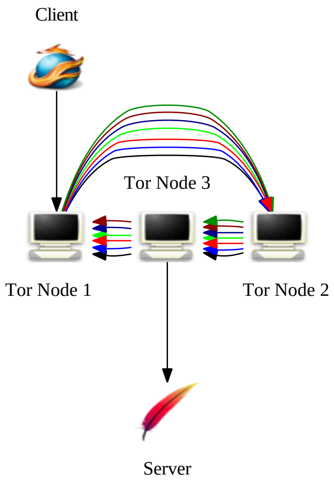
## Circular Path Example 3/5



## Circular Path Example 4/5



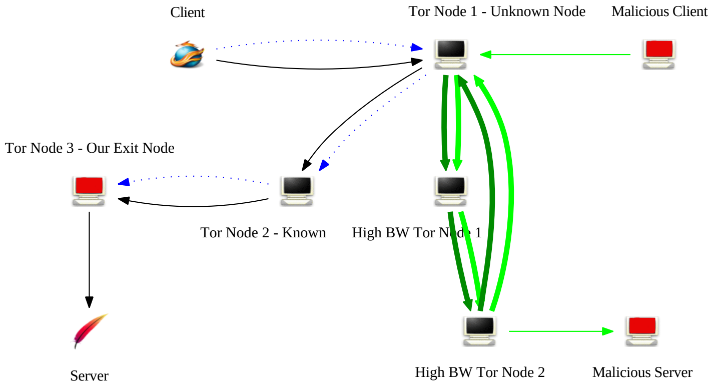
## Circular Path Example 5/5



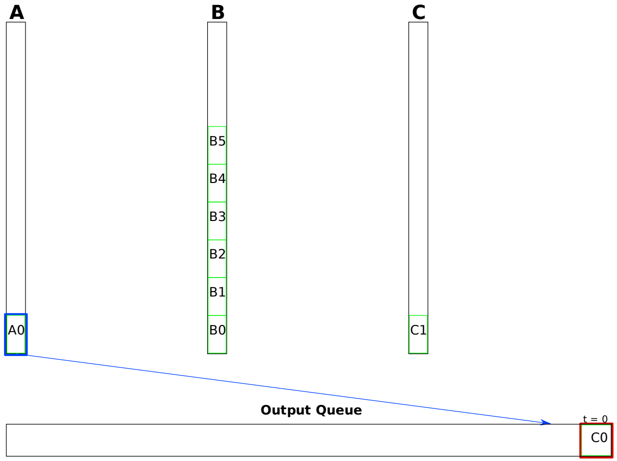
## Attack Implementation

- Exit node “injects” JavaScript “ping” code into HTML response
- Client browses as usual, while JavaScript continues to “phone home”
- Exit node measures variance in latency
- While continuing to measure, attack strains possible first hop(s)
- If no significant variance observed, pick another node from candidates and start over
- Once sufficient change is observed in *repeated* measurements, initial node has been found

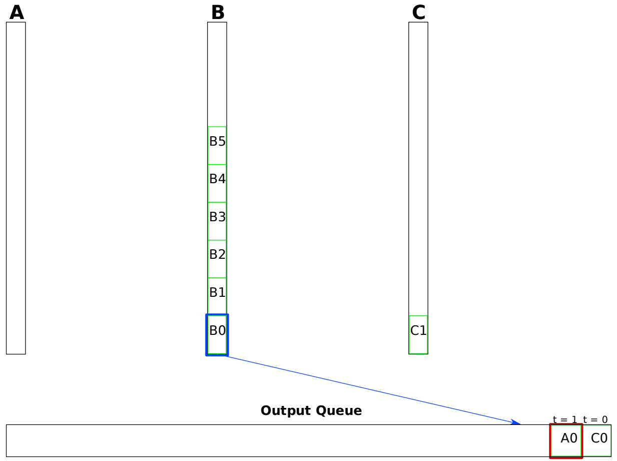
## Attack Example



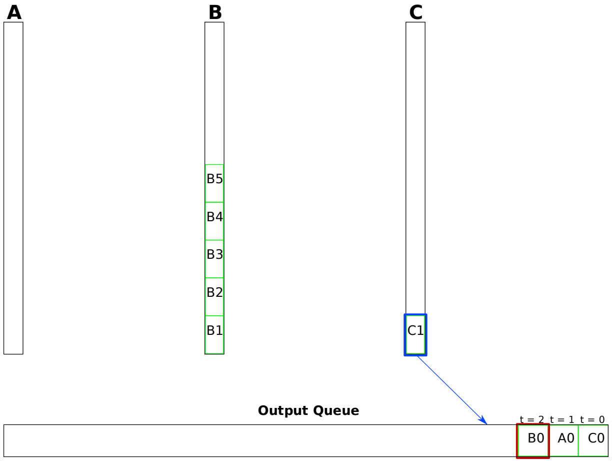
## Queue example 1 (3 circuits)



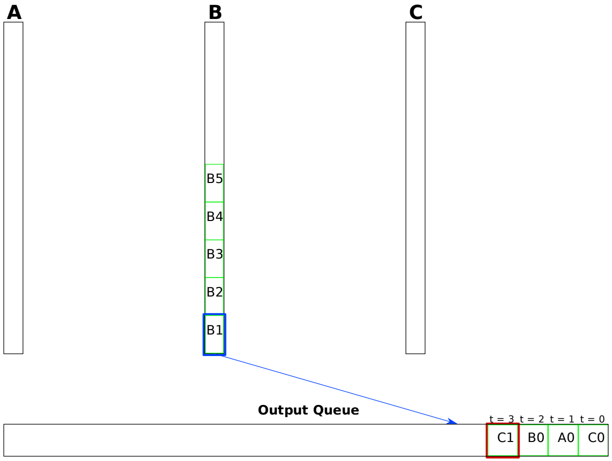
## Queue example 2 (3 circuits)



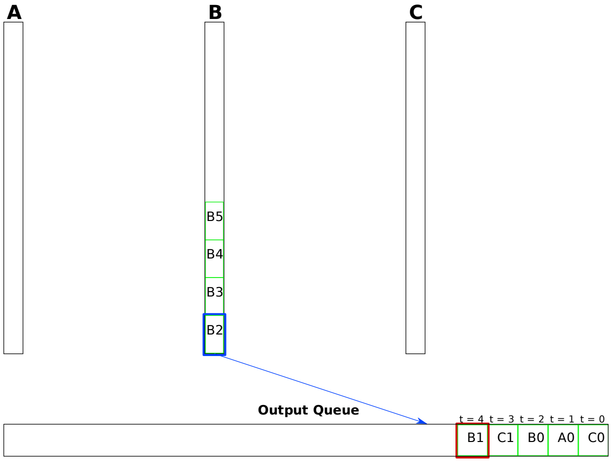
## Queue example 3 (3 circuits)



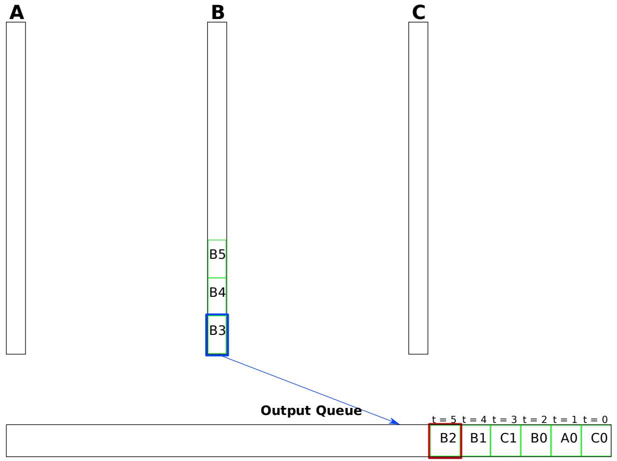
## Queue example 4 (3 circuits)



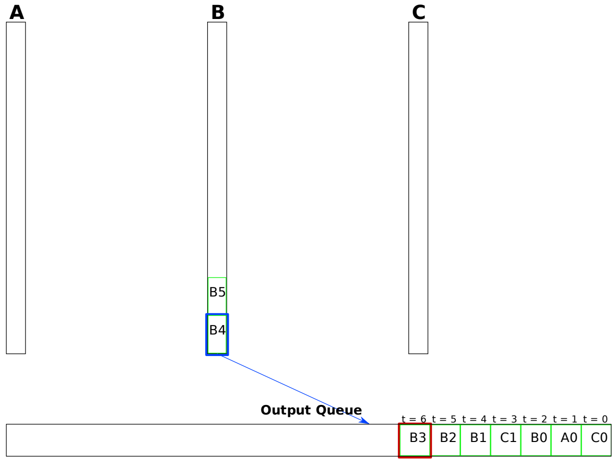
## Queue example 5 (3 circuits)



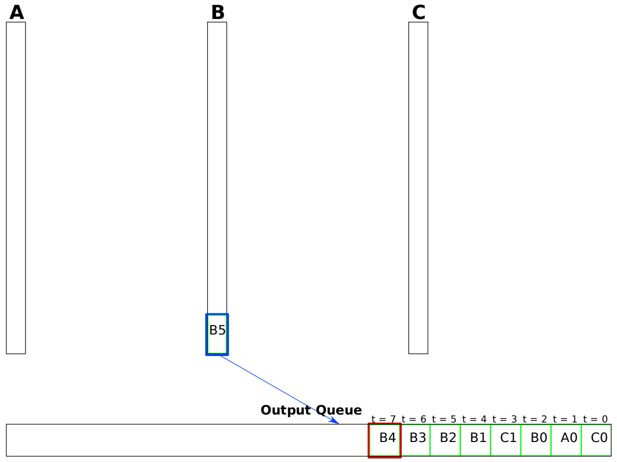
## Queue example 6 (3 circuits)



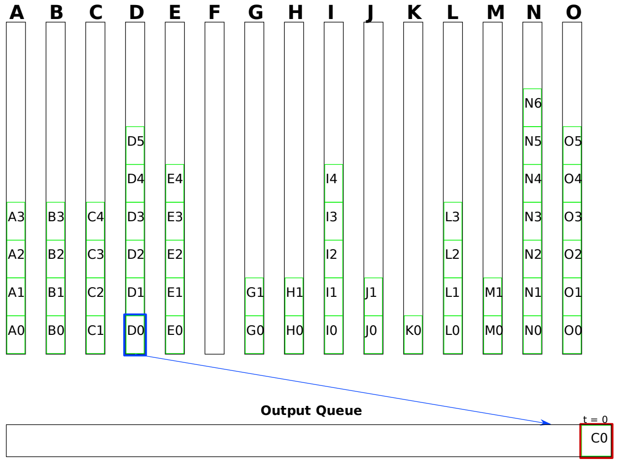
## Queue example 7 (3 circuits)



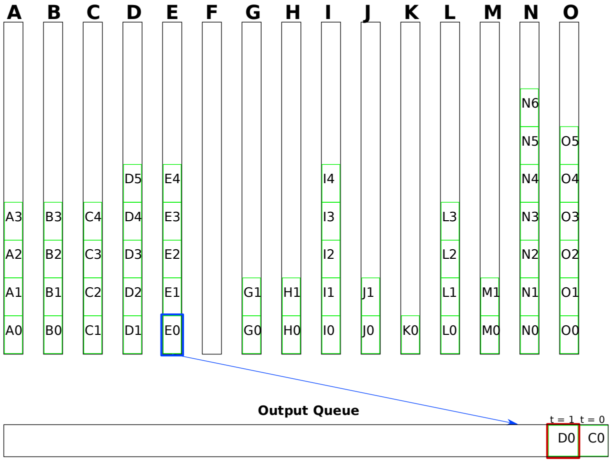
## Queue example 8 (3 circuits)



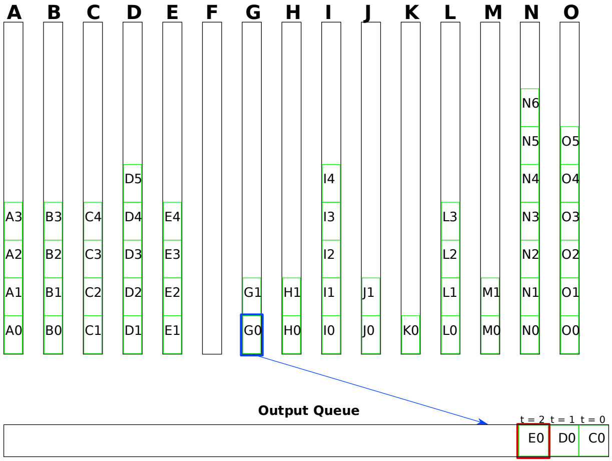
## Queue example 1 (15 circuits)



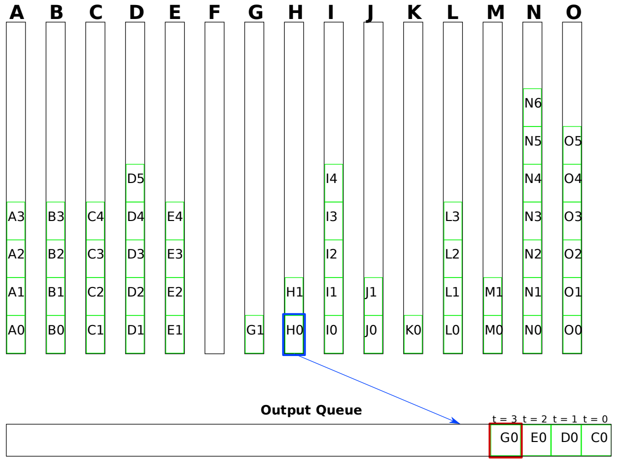
## Queue example 2 (15 circuits)



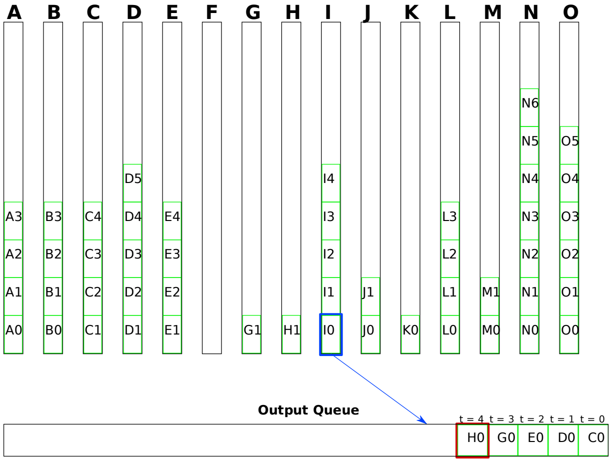
## Queue example 3 (15 circuits)



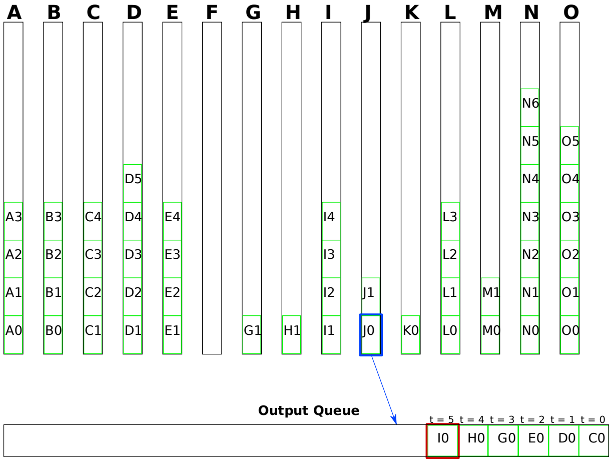
## Queue example 4 (15 circuits)



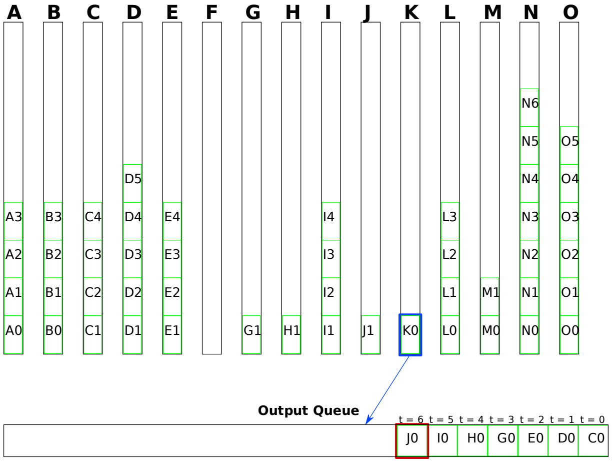
## Queue example 5 (15 circuits)



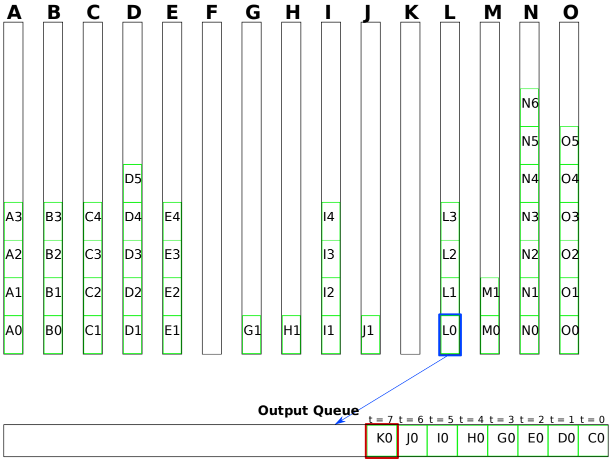
## Queue example 6 (15 circuits)



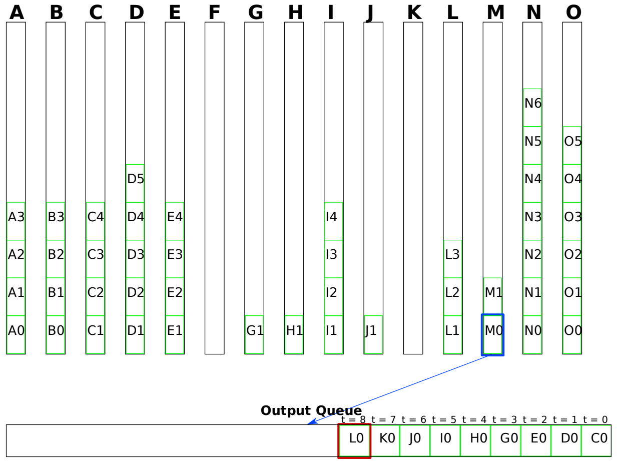
## Queue example 7 (15 circuits)



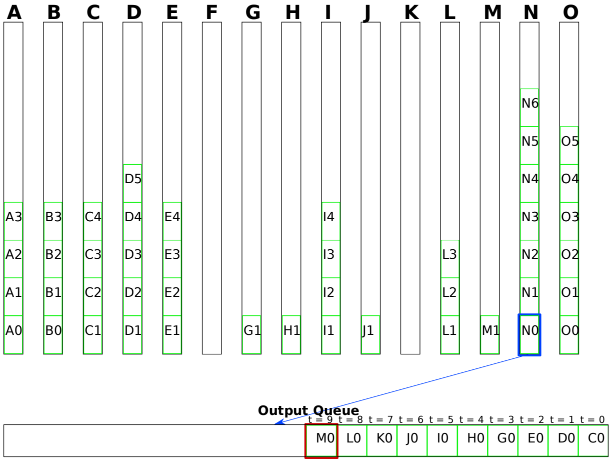
## Queue example 8 (15 circuits)



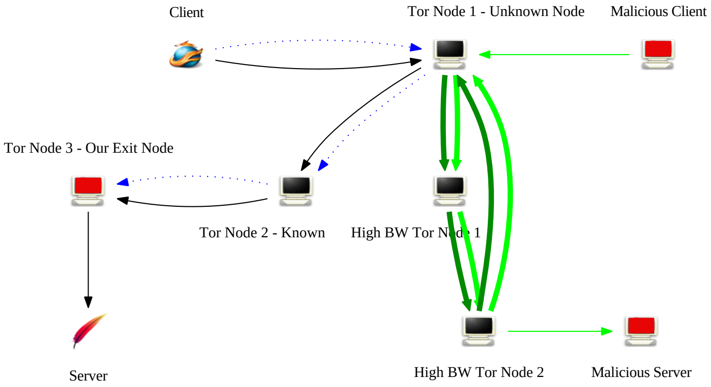
## Queue example 9 (15 circuits)



## Queue example 10 (15 circuits)



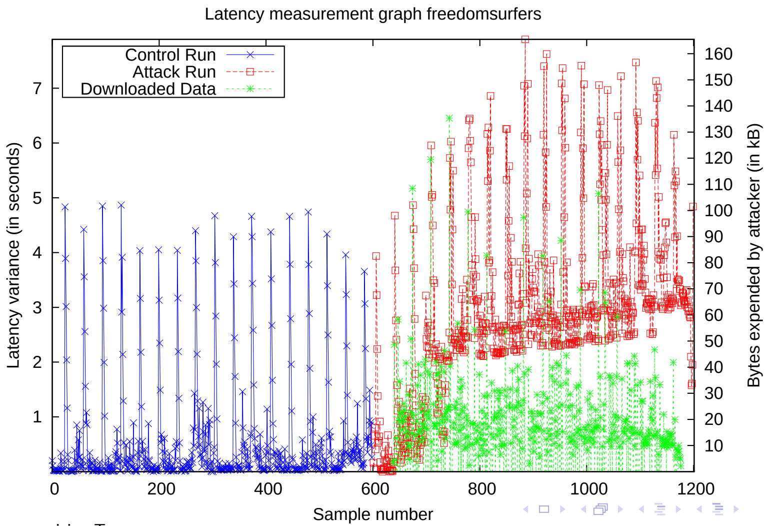
## Attack Example



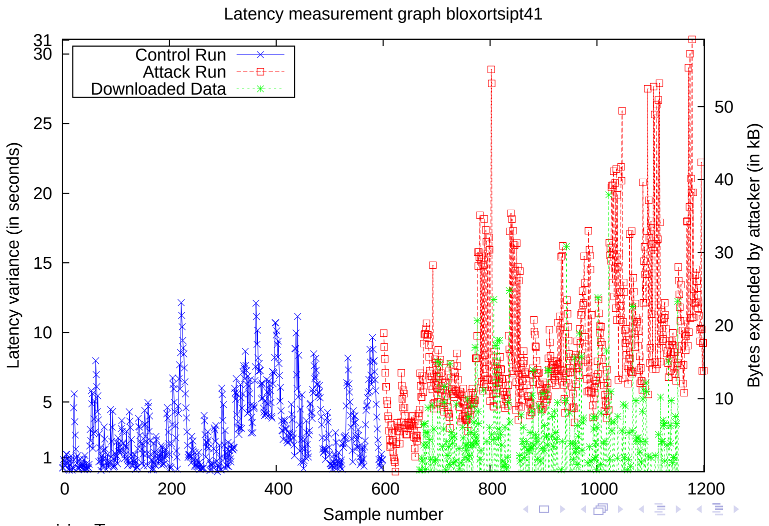
## Attack Implementation

- Modified exit node
- Modified malicious client node
- Lightweight malicious web server running on GNU libmicrohttpd
- Client side JavaScript for latency measurements
- Instrumentation client to receive data

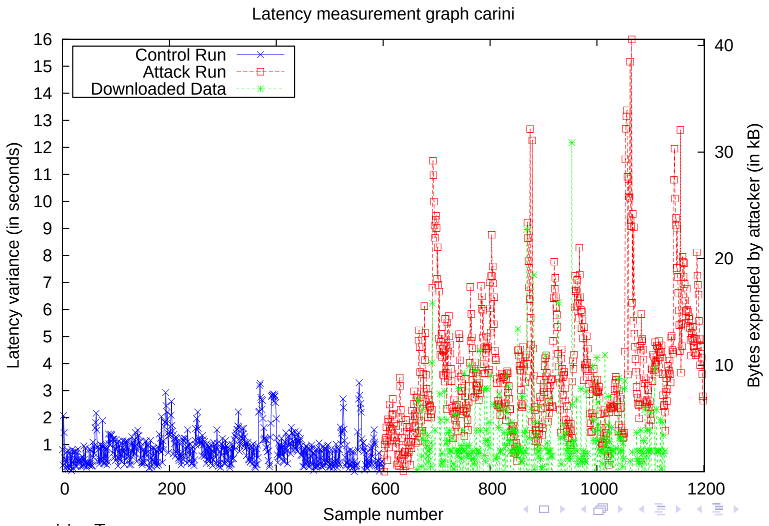
## Gathered Data Example (1/8)



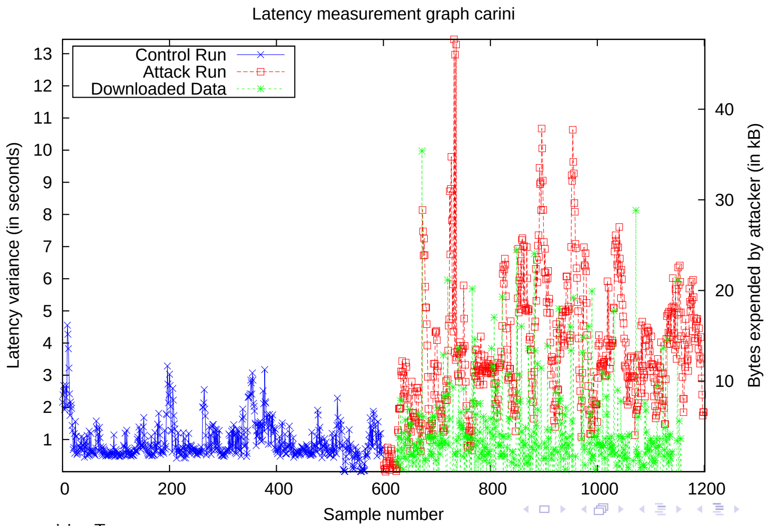
## Gathered Data Example (2/8)



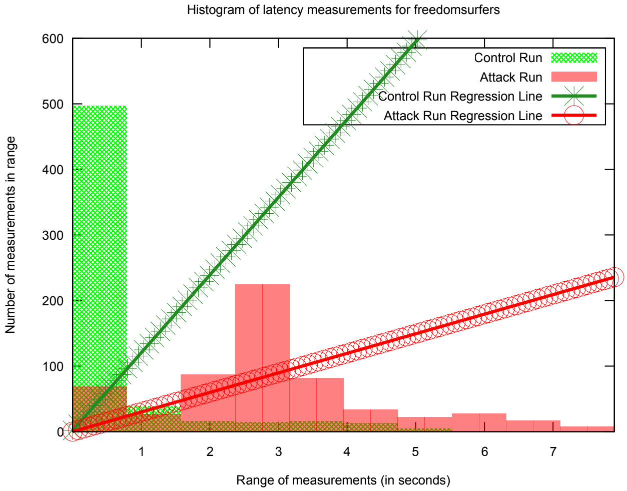
## Gathered Data Example (3/8)



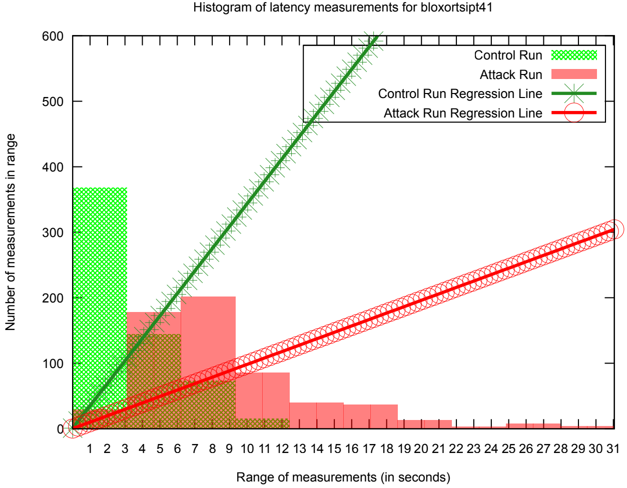
## Gathered Data Example (4/8)



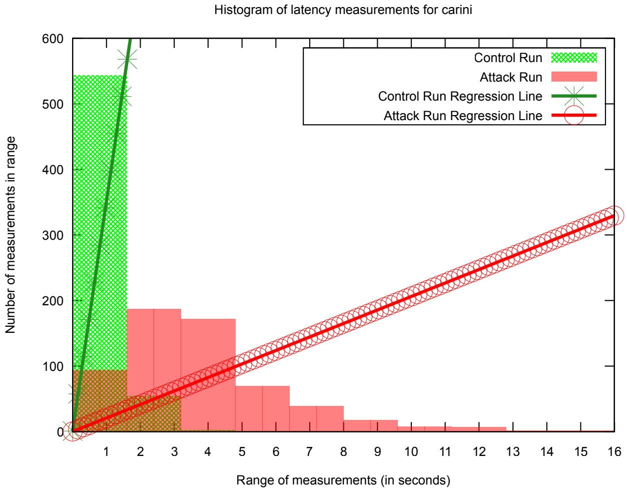
## Gathered Data Example (5/8)



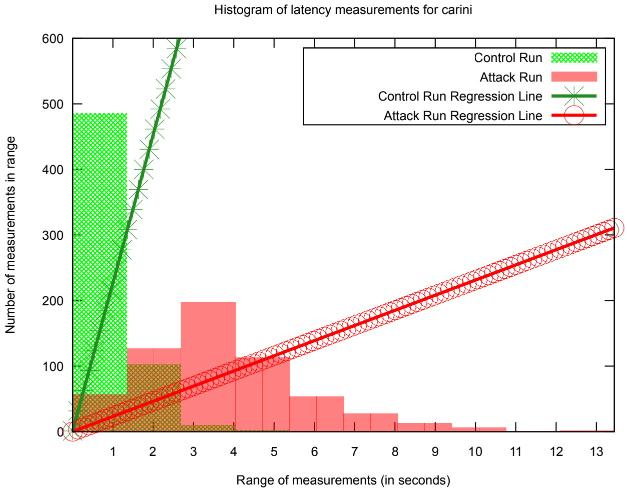
## Gathered Data Example (6/8)



## Gathered Data Example (7/8)



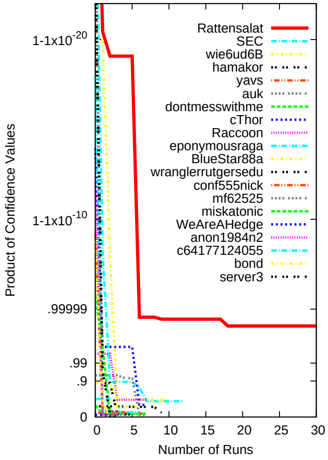
## Gathered Data Example (8/8)

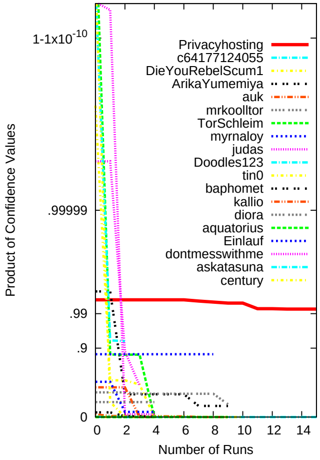


## Statistical Analysis

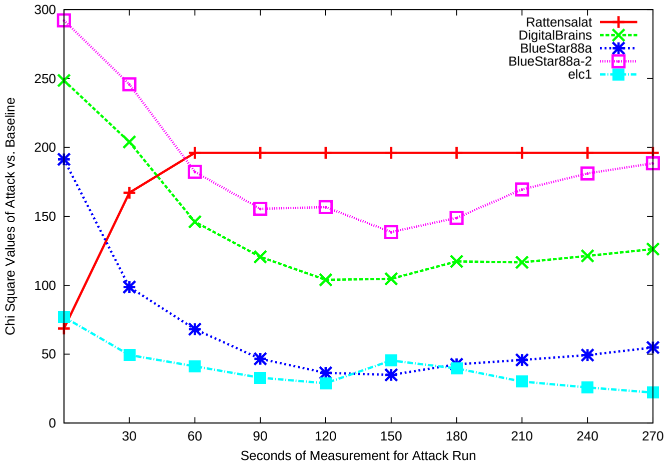
- Use modified $\chi^2$ test
- Compare baseline distribution to attack distribution
- High $\chi^2$ value indicates distribution changed *in the right direction*
- Product of $\chi^2$ confidence values over multiple runs
- Iterate over suspect routers until single node stands out

## Cumulative Product of $\chi^2$ p-values





## Convergence of $\chi^2$ Values



## What We Actually Achieve

- We do identify the entire path through the Tor network (same result as Murdoch and Danezis)
- We do achieve this on the modern, current Tor network
- Attack works on routers with differing bandwidths
- This means that if someone were performing this attack from an exit node, Tor becomes as effective as a network of one-hop proxies

## Why Our Attack is Effective

- Since we run the exit router, only a single node needs to be found
- Our multiplication of bandwidth technique allows low bandwidth connections to DoS high bandwidth connections (solves common DoS limitation)

## Fixes

- Don't use a fixed path length (or at least make it longer)
- Don't allow infinite path lengths
- Induce delays into connections (probably not going to happen)
- Monitor exit nodes for strange behavior (been done somewhat)
- Disable JavaScript in clients
- Use end-to-end encryption

## Attack Improvements/Variants

- Use meta refresh tags for measurements instead of JavaScript
- Parallelize testing (rule out multiple possible first nodes at once)
- Improved latency measures for first hop to further narrow possible first hops

## Conclusion

- Current Tor implementation allows arbitrary length paths
- Current Tor implementation uses minimally short paths
- Arbitrary path lengths allow latency altering attack
- Latency altering attack allows detection of significant changes in latency
- Significant changes in latency reveal paths used

## Questions?

?