

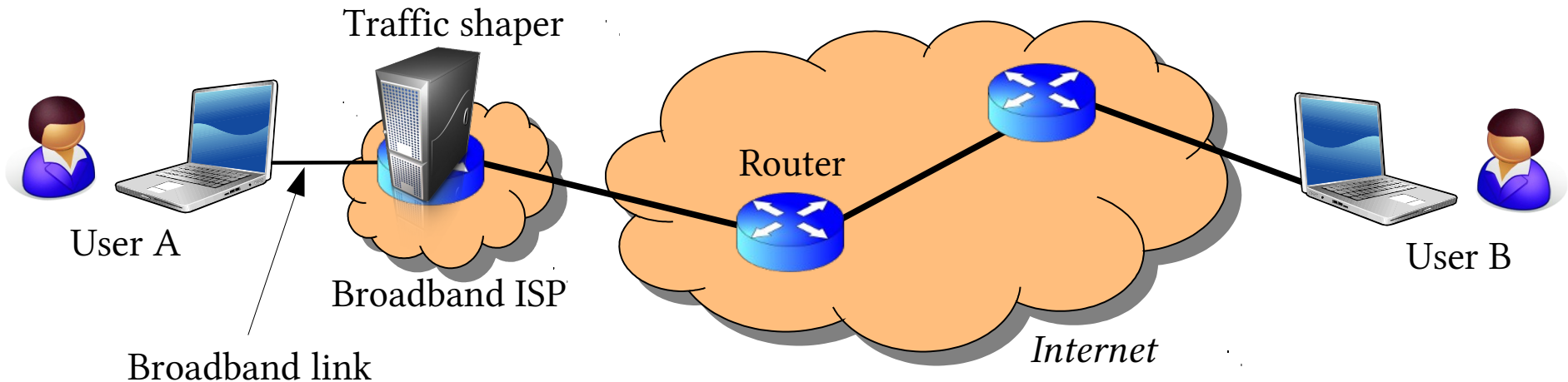


Max
Planck
Institute
for
Software Systems

Glasnost: Enabling End Users to Detect Traffic Differentiation

Marcel Dischinger, Massimiliano Marcon, Saikat Guha,
Krishna P. Gummadi, Ratul Mahajan, Stefan Saroiu

Networks are not transparent today



- ISPs are deploying traffic shapers widely
 - But ISPs often do not inform their customers about this
- Can we enable users to detect traffic shaping?

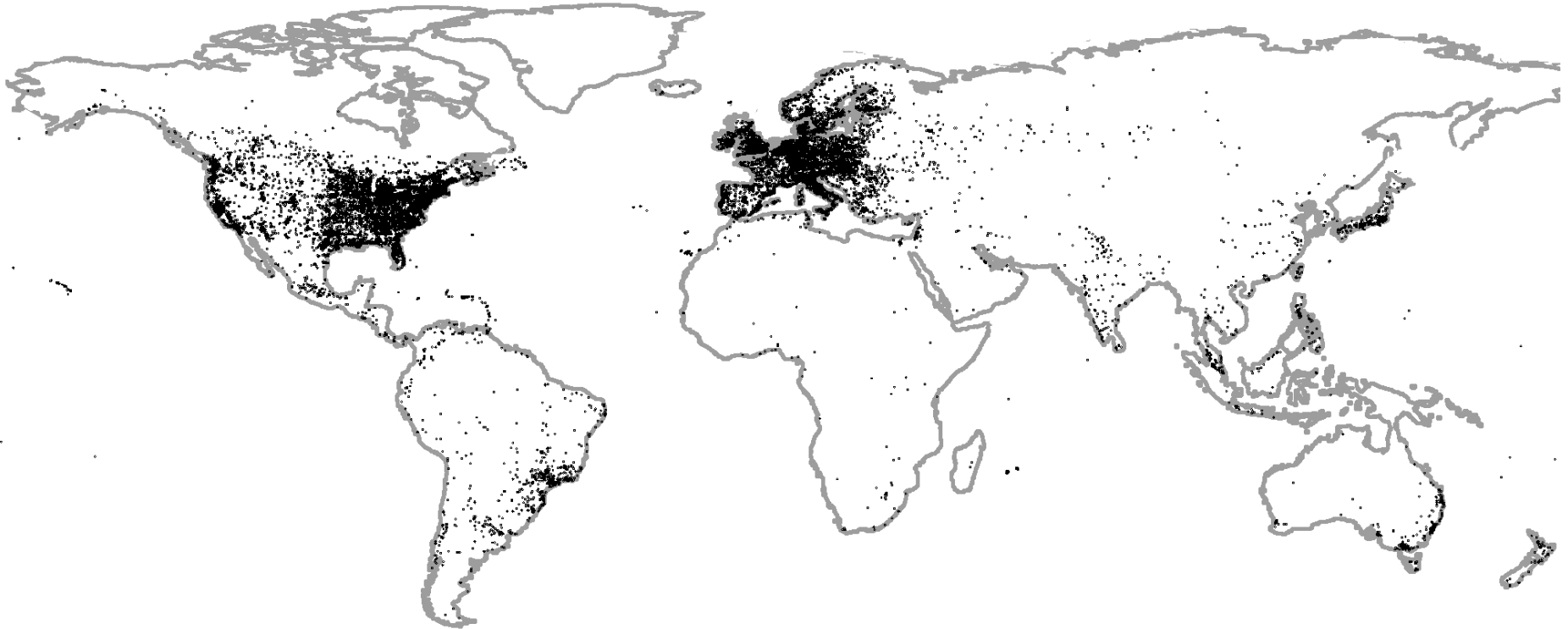


Why transparency is important

- End users can make an informed choice
- Developers can adapt applications to ISP policies
- Regulators can monitor ISPs and hold them accountable
 - Today, regulators rely on information provided by ISPs for their investigations



Glasnost user base



- Since March 2008, **more than 500,000 users** world-wide
- Glasnost results are used by telecom regulators



Rest of this talk

- **The Glasnost system design**
 - **Challenges**
 - **Design choices**
- **Glasnost deployment**
- **Conclusion**

Challenges

1. Tests must be easy to use

2. Tests must be short

3. Tests must be accurate

Challenges

1. Tests must be easy to use

- No complex software installation
- Simple and intuitive interface

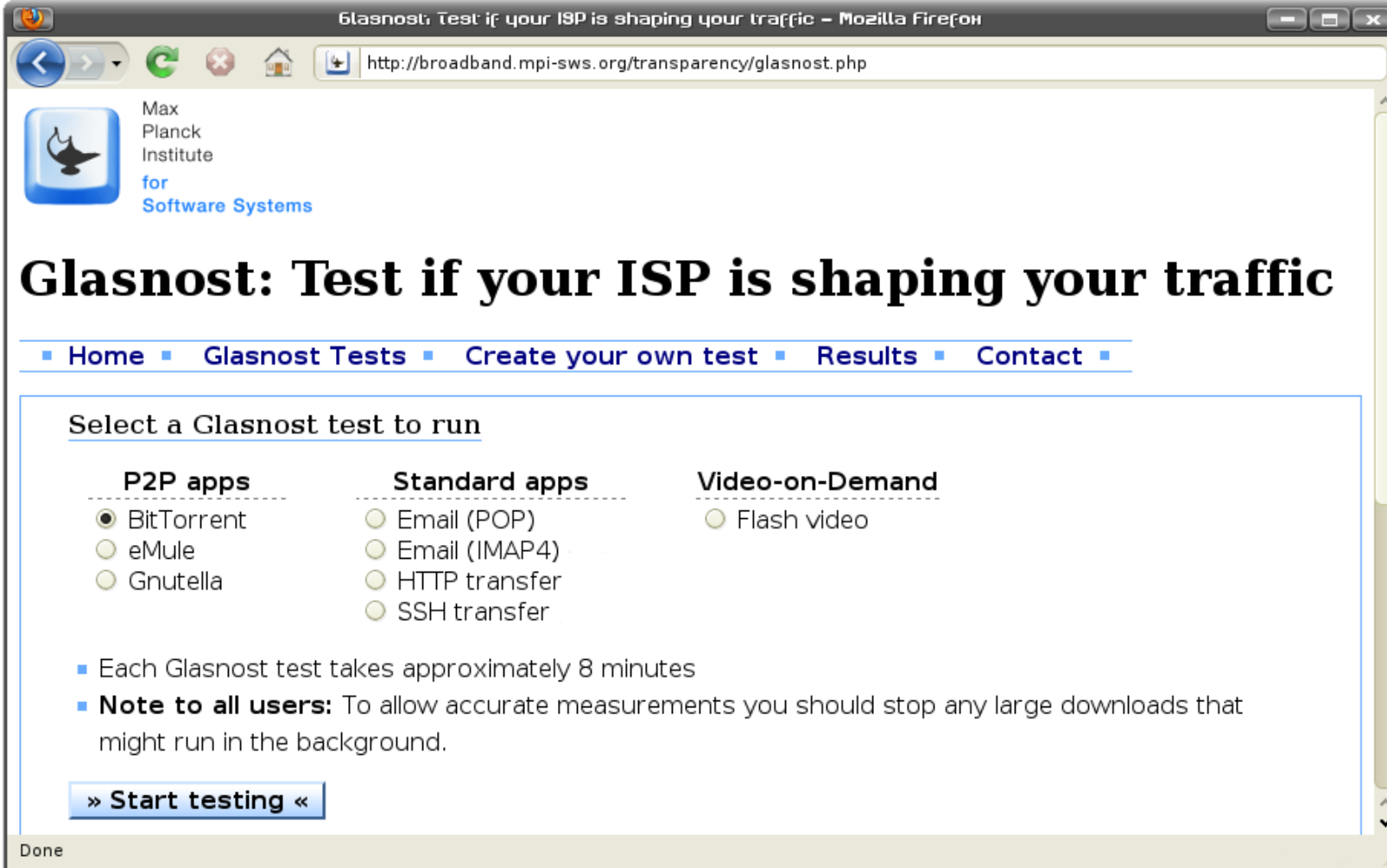
2. Tests must be short

3. Tests must be accurate



Glasnost from a user's perspective

- Users can easily test their own broadband links



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://broadband.mpi-sws.org/transparency/glasnost.php`. The page title is "Glasnost: Test if your ISP is shaping your traffic". The page content includes a navigation menu with links for Home, Glasnost Tests, Create your own test, Results, and Contact. Below the menu, there is a section titled "Select a Glasnost test to run" with three columns of radio button options: "P2P apps" (BitTorrent, eMule, Gnutella), "Standard apps" (Email (POP), Email (IMAP4), HTTP transfer, SSH transfer), and "Video-on-Demand" (Flash video). A note states that each test takes approximately 8 minutes and advises users to stop large background downloads for accurate measurements. A "Start testing" button is located at the bottom of the test selection area.

Max Planck Institute for Software Systems

Glasnost: Test if your ISP is shaping your traffic

- Home
- Glasnost Tests
- Create your own test
- Results
- Contact

Select a Glasnost test to run

P2P apps	Standard apps	Video-on-Demand
<input checked="" type="radio"/> BitTorrent	<input type="radio"/> Email (POP)	<input type="radio"/> Flash video
<input type="radio"/> eMule	<input type="radio"/> Email (IMAP4)	
<input type="radio"/> Gnutella	<input type="radio"/> HTTP transfer	
	<input type="radio"/> SSH transfer	

- Each Glasnost test takes approximately 8 minutes
- **Note to all users:** To allow accurate measurements you should stop any large downloads that might run in the background.

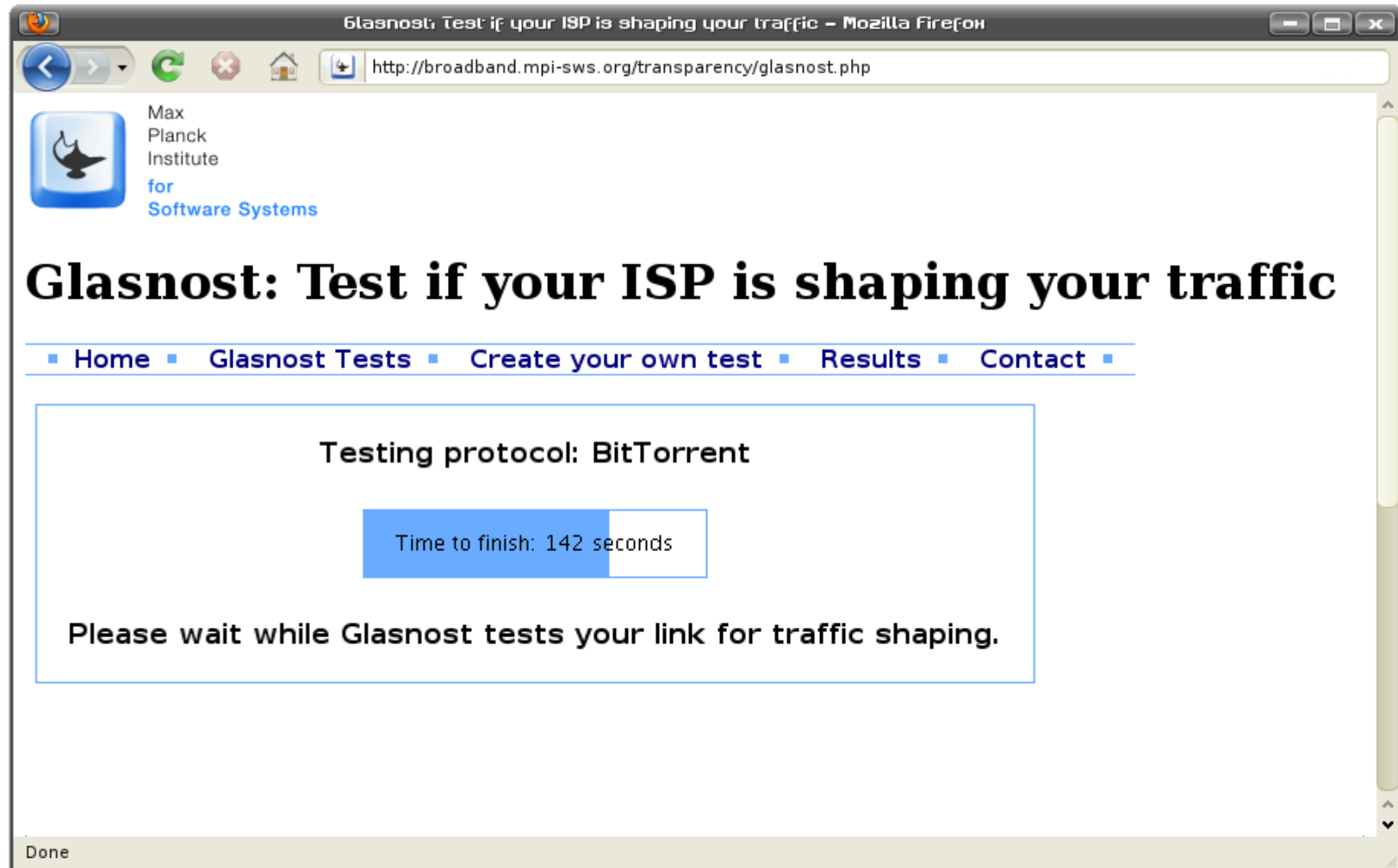
» Start testing «

Done



Glasnost from a user's perspective

- Users can easily test their own broadband links



Glasnost from a user's perspective

- Users can easily test their own broadband links

Max Planck Institute for Software Systems

Glasnost: Test if your ISP is shaping your traffic

- Home
- Glasnost Tests
- Create your own test
- Results
- Contact

Results for your host ([redacted] - [redacted]):

Is your upload traffic rate limited?

Your ISP appears to rate limit your uploads.

Details:

- **Your ISP appears to rate limit your BitTorrent uploads.** In our tests, uploads using control flows achieved up to 1549 Kbps while uploads using BitTorrent achieved up to 159 Kbps.
- **Your ISP appears to rate limit uploads on port 6881.** In our tests,

Done



Challenges

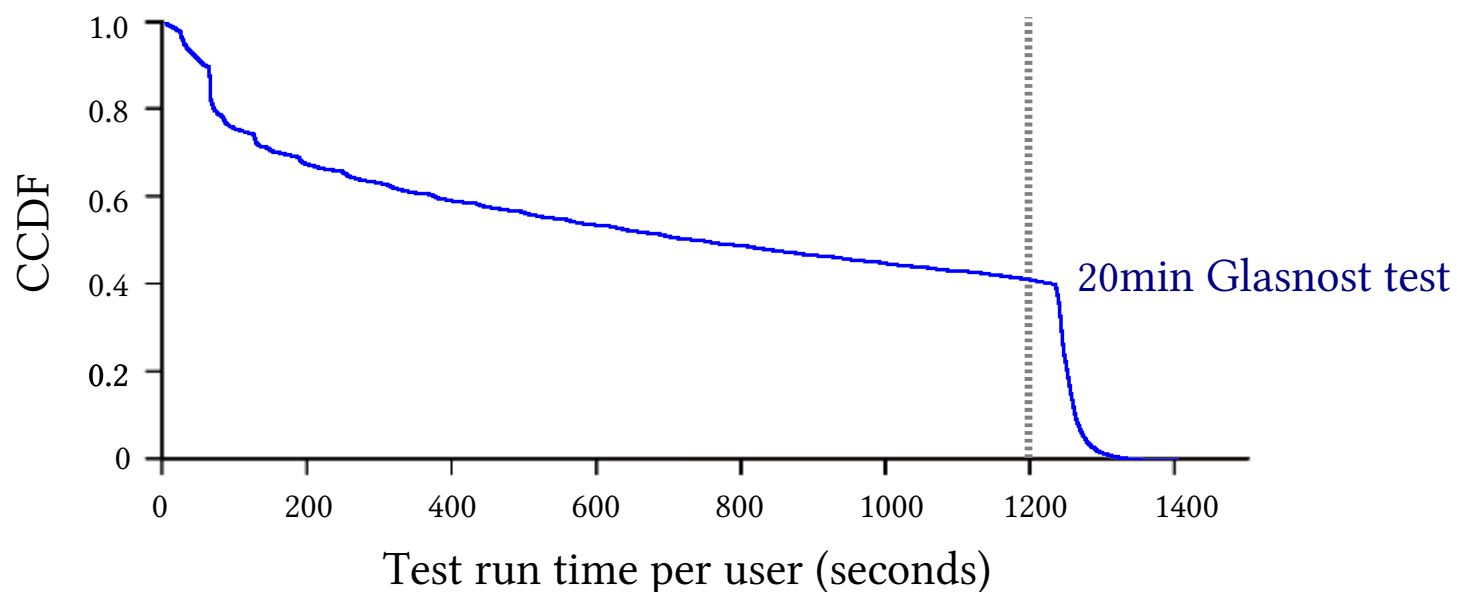
1. Tests must be easy to use

2. Tests must be short

3. Tests must be accurate



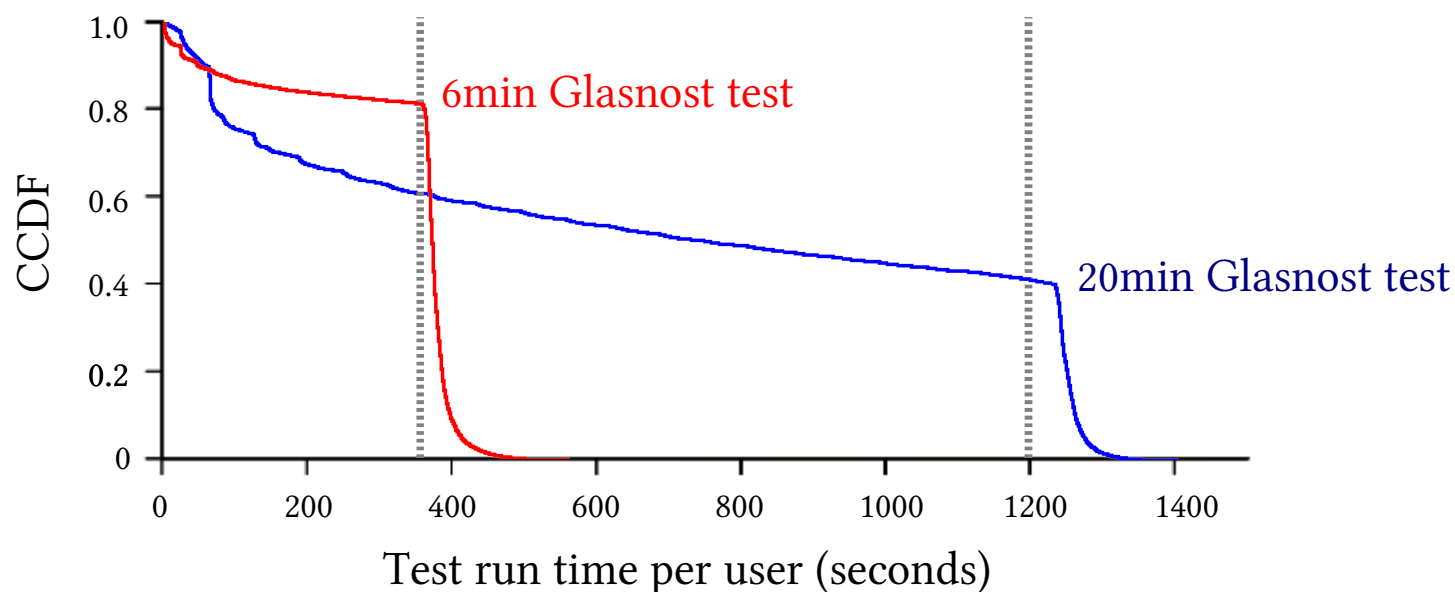
Users are impatient



- > 50% of users abort tests longer than 10 minutes
- **Consequence:** In Glasnost, we use shorter tests to avoid incomplete tests



Users are impatient



- > 50% of users abort tests longer than 10 minutes
- **Consequence:** In Glasnost, we use shorter tests to avoid incomplete tests
 - But, short tests decrease the amount of data we can collect



Challenges

1. Tests must be easy to use
2. Tests must be short
3. Tests must be accurate

Glasnost tests must be accurate

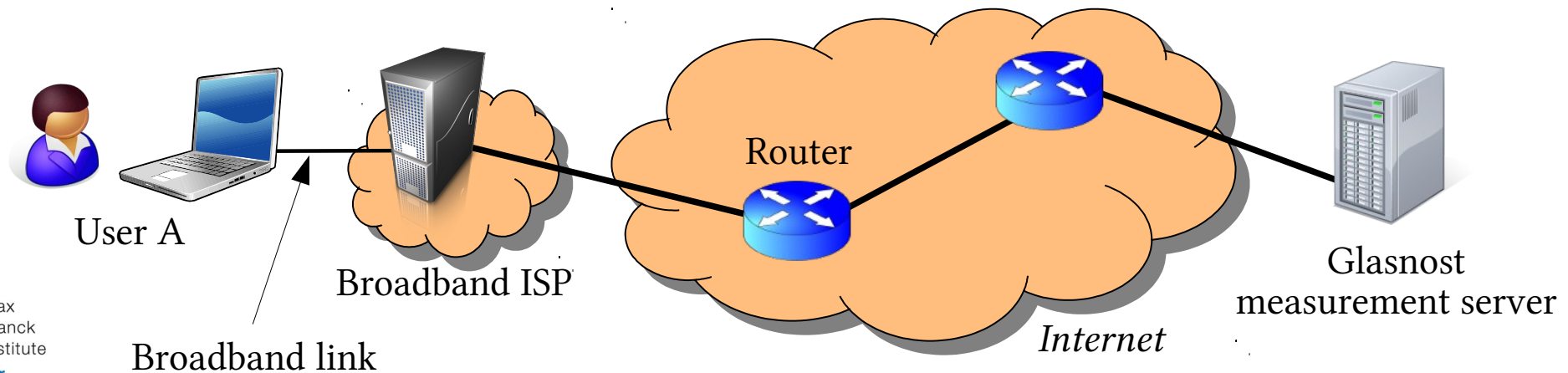
- Tests must be designed to avoid confounding factors
 - Heterogeneous hardware / software configurations
 - Differences in the natural traffic patterns of applications
 - Transient noise from background network traffic

- Test data limited to a single user's access link



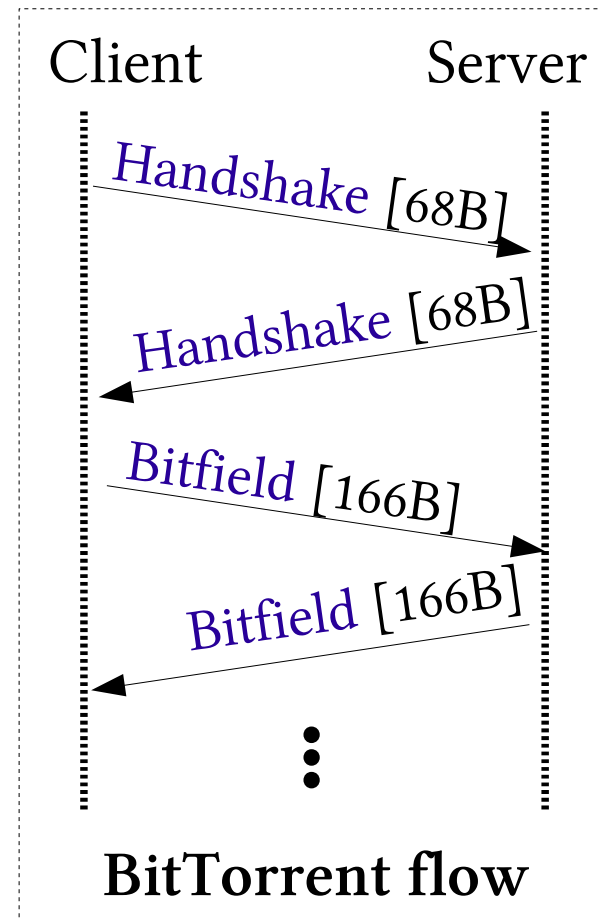
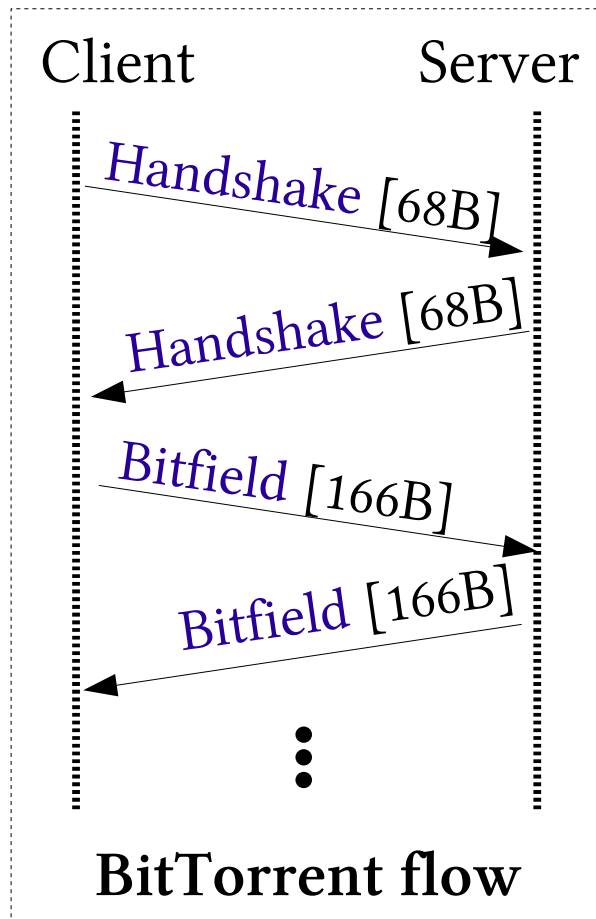
Glasnost test construction

- We perform active measurements in a **controlled** fashion
 1. Glasnost sends a flow that emulates realistic application traffic
 2. Glasnost sends a second flow that varies the payload of the first flow, but keeps everything else the same
- We check for differences in the performance of the two flows



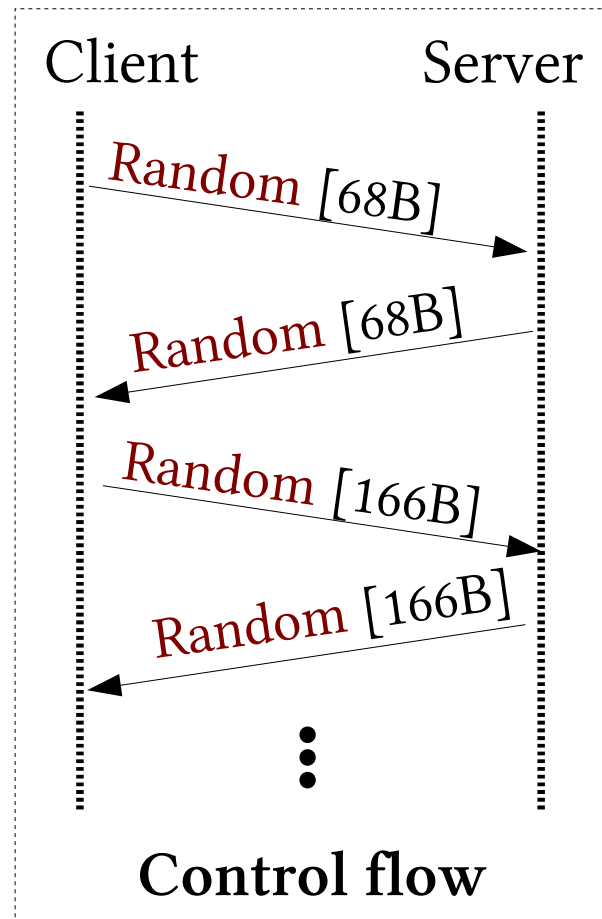
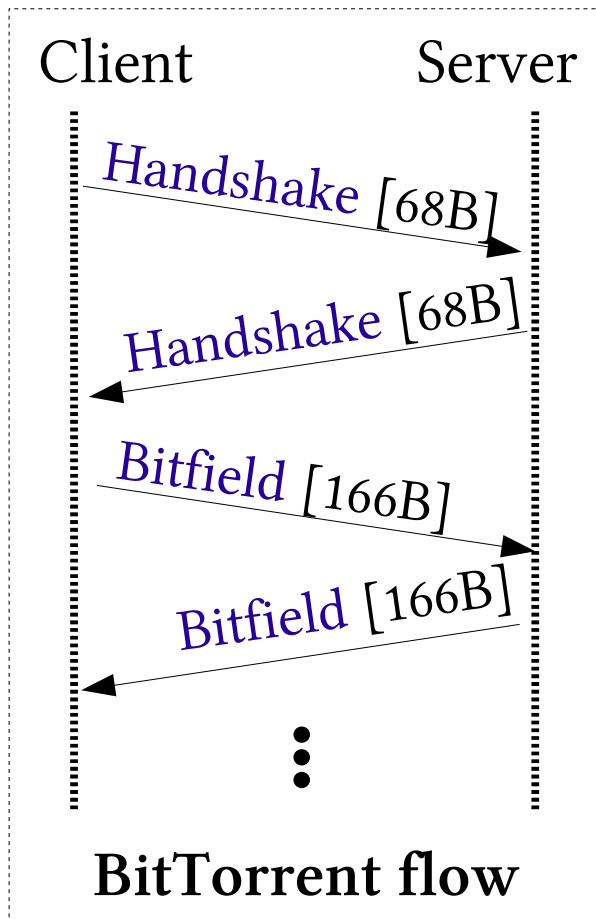
Detecting BitTorrent traffic shaping

- Glasnost compares the performance of a pair of flows



Detecting BitTorrent traffic shaping

- Glasnost compares the performance of a pair of flows



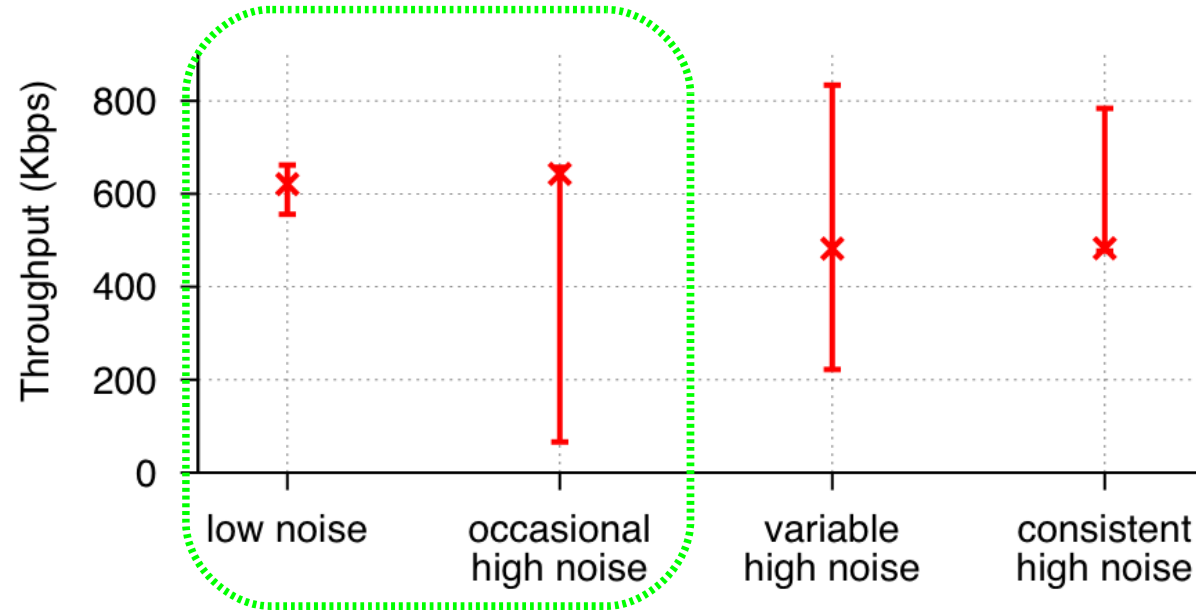
Glasnost tests must be accurate

- Tests must be designed to avoid confounding factors
 - Heterogeneous hardware / software configurations ✓
 - Differences in the natural traffic patterns of applications ✓
 - Transient noise from background network traffic



Identifying noisy tests

- We repeat the flow pairs several times over a few minutes
- Comparing throughputs of repeated flows reveals 4 noise patterns

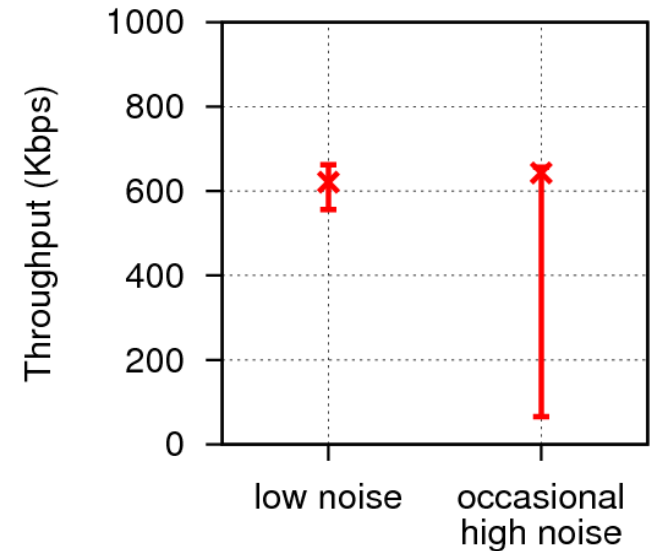


- > 80% of all tests have low noise or occasional high noise



Detecting traffic shaping

- Compare the maximum throughput achieved by the 2 flow types
- Declare traffic shaping when difference is more than a threshold
 - High threshold: more false negatives and fewer false positives
 - Low threshold: fewer false negatives and more false positives



Outline

- The Glasnost system design
- **Glasnost deployment**
 - **BitTorrent rate-limiting**
- Conclusion

Glasnost deployment

- Glasnost currently runs on 20 servers on 9 sites world-wide
 - 3 servers at MPI-SWS, 17 servers hosted by contributors

- Glasnost is part of Measurement Lab
 - An open platform for Internet measurement tools for more transparency
 - Provides measurement server locations around the planet
 - Founded in collaboration with Google, PlanetLab, and other academic researchers



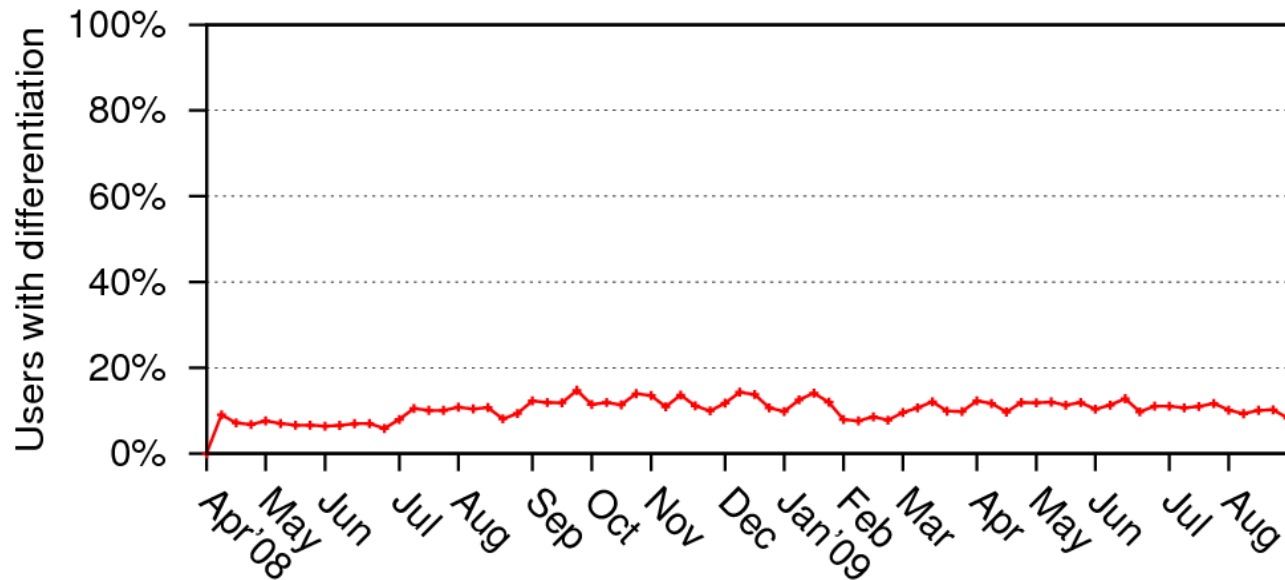
Results on BitTorrent rate-limiting

- How prevalent is BitTorrent rate-limiting?
- Which ISPs rate-limit BitTorrent traffic?
- Are all users of an ISP affected by rate-limiting?
- Do ISPs rate-limit only at peak hours?
- Is rate-limiting based on ports or application protocol messages?
- Do ISPs rate-limit downloads more often than uploads?



1. How prevalent is BitTorrent rate-limiting?

- Roughly 10% of our BitTorrent tests indicate rate-limiting over our 18 month deployment



2. Which ISPs' customers are affected?

- For Jan and Feb 2009 we analyzed measurements from 100,000 users for evidence of rate-limiting
- We found users of major ISPs world-wide to be affected by BitTorrent rate-limit

ISP	Tech.	Country
Bell Canada	DSL	Canada
BT	DSL	UK
City Telecom	FTTH	Hong Kong
Clearwire	WiMax	USA
Cogeco	Cable	Canada
Free	DSL	France
Kabel D	Cable	Germany
ONO	Cable	Spain
Rogers	Cable	Canada
Tiscali	DSL	UK
TM Net	DSL	Malaysia
TVCabo	Cable	Portugal
UPC	Cable	Netherlands



Other results

- Rate-limiting is more common in the upstream direction
 - Rate-limiting is usually based on both packet content and ports
 - Not all tests from traffic shaping ISPs suffer rate-limiting
 - Probably caused by selective deployment of traffic shaping
 - Half of all rate-limiting ISPs do so only at peak hours
 - Confirms announcements by BT, Bell Canada, Kabel Deutschland
- *More details in the paper*

Constructing Glasnost tests for other applications

- ISPs' policies evolve over time
 - Target new applications, use different policies / shaping mechanisms
- Users are interested in testing for many different policies
- Initially, manual test construction
 - Required full understanding of the application protocol and protocol implementation
 - Laborious and error-prone task



Creating additional Glasnost tests

- Automatic test construction with `trace-emulate`
- To construct new tests, users need to
 - Capture a trace of application traffic
 - Feed it to our `trace-emulate` tool
 - Upload the resulting test configuration to Glasnost servers
- `Trace-emulate` extracts key features from input trace for replay



Summary

- Glasnost enables end users to detect traffic shaping
- Glasnost was used by more than 500,000 users world-wide
- Currently 8 tests, including BitTorrent and Flash video
 - Interface to create your own Glasnost tests
- First large-scale study on the prevalence of BitTorrent blocking and rate-limiting



Thank you!

- Questions?

Check out the Glasnost project at
<http://broadband.mpi-sws.org/transparency/>

