

Development of a User-Centered Network Measurement Platform

Craig Wills, Mark Claypool, Artur Janc and Alan Ritacco
Worcester Polytechnic Institute
Worcester, MA USA

Presentation at the ISMA Workshop on Active Internet
Measurements

La Jolla, CA USA
February 2010

Motivation

- Would like to broaden the set of network measurement points beyond university and research labs provided by an infrastructure such as PlanetLab.
- Focus on doing network performance evaluation in a browser—why?
- Because browsers are ubiquitous and make a natural candidate for an end-user performance platform—low *impediment* for user participation.
- Looking for more active user participation—try to focus on *incentives* of interest to them while learning information of value to the network research community.
- Thus far we have investigated measurement capabilities of using Flash and JavaScript, as well as using signed Java applets.

Complement to Other Approaches

1. PlanetLab, Ark—flexible measurement platforms with limited vantage points.
2. NETI@home, DIMES, DipZoom—allow measurements from any node in the Internet, but limited incentive for general populace participation.
3. Speedtest services—provide performance measures to users, but not designed to inform network research.
4. Illuminati, BitTorrent plugins—placement of measurement software on visited sites and tools allowing measurement independent of user participation.

Our user-centered approach aims to focus on applications and sites of interest to users—which in itself is valuable information—while gathering performance data. Key is to present tests and results in forms useful to users whether it be “health checks”, games or videos.

Challenges to Using Flash/JavaScript

1. HTTP-only—no UDP, ICMP;
2. scripts cannot only connect directly to the origin server; and
3. no persistent storage on the client—must transmit to the server.

Contributions

In contrast to existing Speedtest services, which measure download/upload throughput and RTT from an existing server using Flash.

- Use *explicit* techniques to perform measurements to the origin server in both Flash and JavaScript;
- Compare the relative accuracy of these techniques across different operating system and browser environments;
- Develop *implicit* techniques that allow reliable measurements of download throughput and RTT to arbitrary third-party servers;
- Develop a technique for measuring the jitter of a streaming Web server; and
- Perform distributed set of browser-based network measurement tests.

Approach

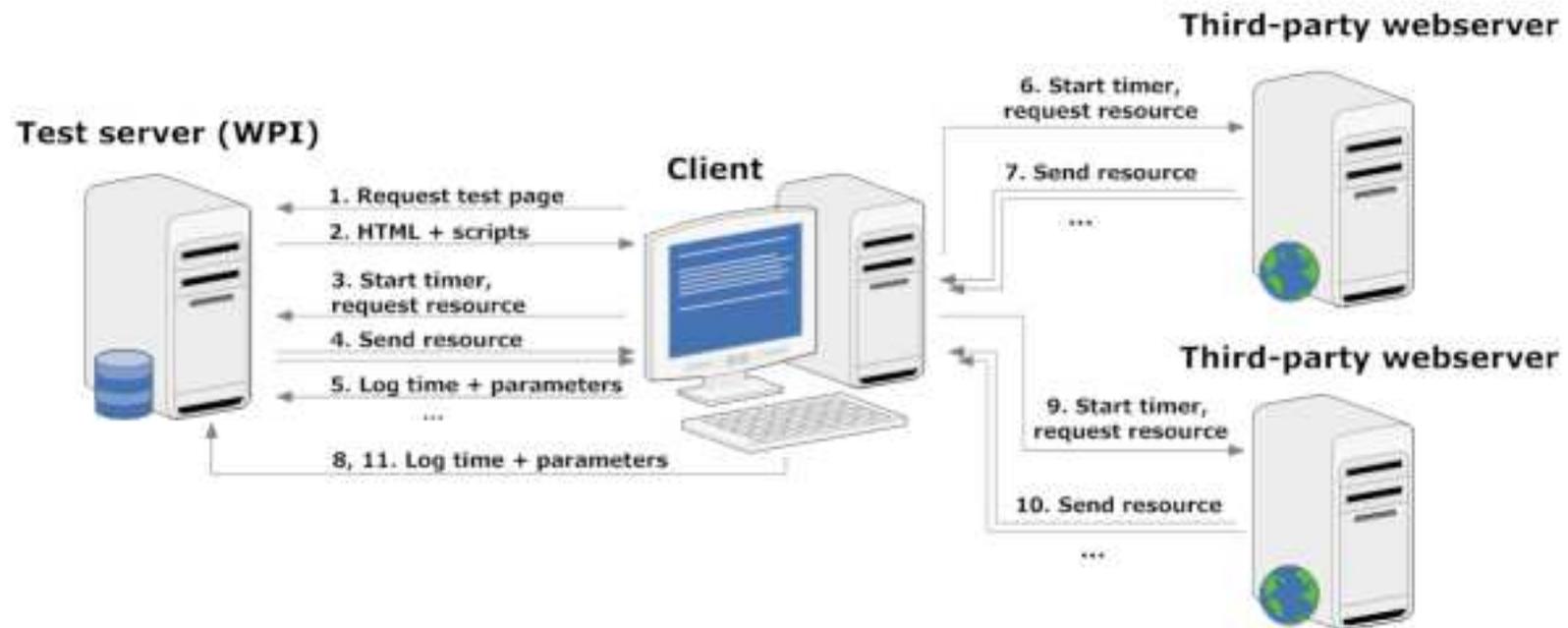
Developed techniques with both Flash and JavaScript.

- Explicit Communication Methods—measurements performed with the origin server.
 - Download/upload throughput, RTT (one-byte file), jitter
- Implicit Communication Methods—measurements not performed with the origin server
 - JavaScript—add third-party object to DOM tree and then implicitly measure download time. Can obtain download throughput and RTT.
 - Flash—use implicit timing of `loadPolicyFile()` for known object on a third-party server.

Network Performance Measurement Capabilities by Technique

| Technique | Category | Origin Server | | | | Third-party Server | | |
|------------------------|----------|---------------|-------------|-----|--------|--------------------|-------------|-----|
| | | Dwnld Tput | Upload Tput | RTT | Jitter | Dwnld Tput | Upload Tput | RTT |
| JavaScript: XHR | Explicit | ✓ | ✓ | ✓ | ✓ | | | |
| JavaScript: DOM | Implicit | ✓ | | ✓ | | ✓ | | ✓ |
| Flash: URLRe-quest | Explicit | ✓ | ✓ | ✓ | ✓ | | | |
| Flash: loadPoli-cyFile | Implicit | ✓ | | ✓ | | ✓ | | ✓ |

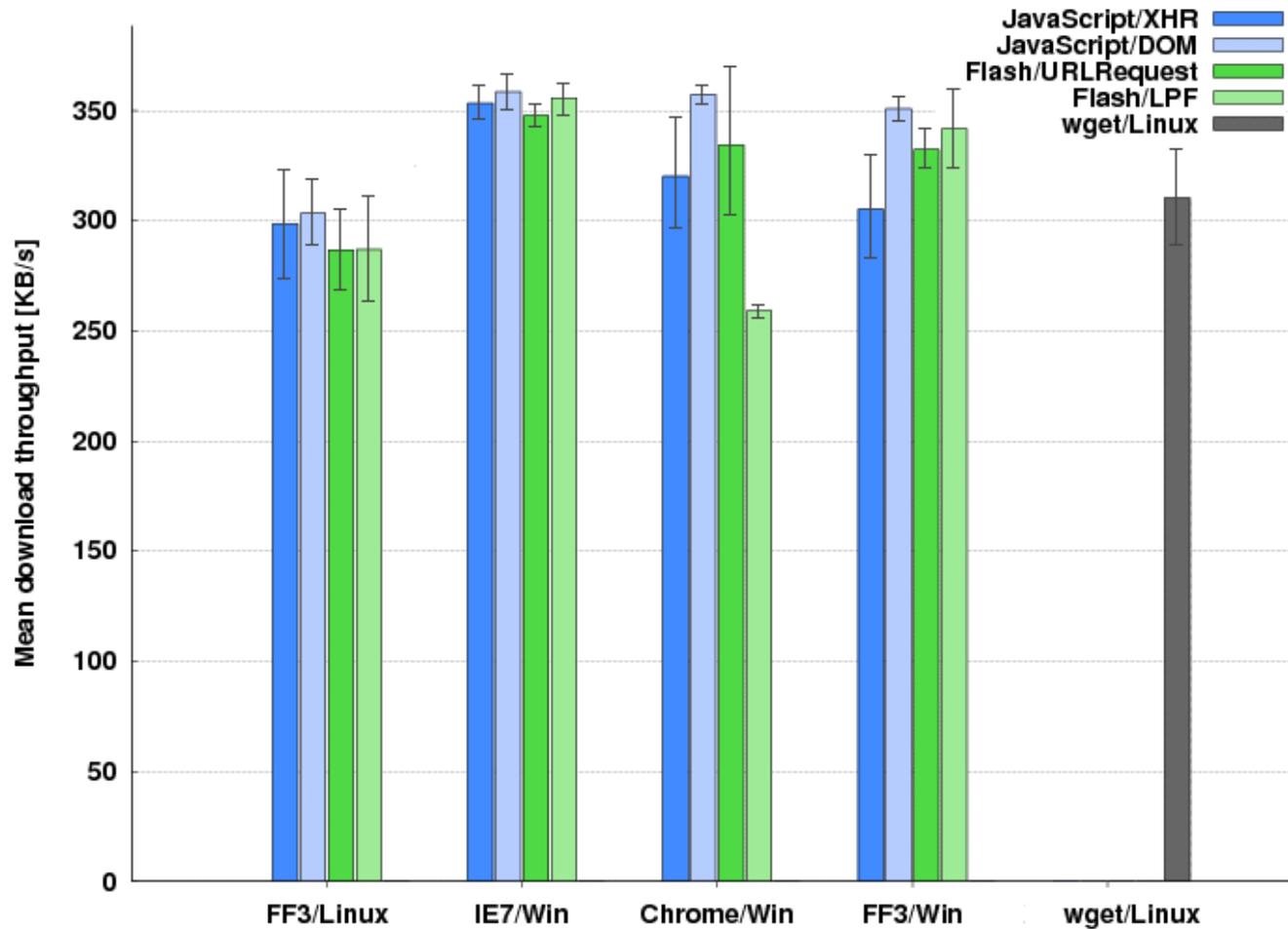
Test Platform System Design



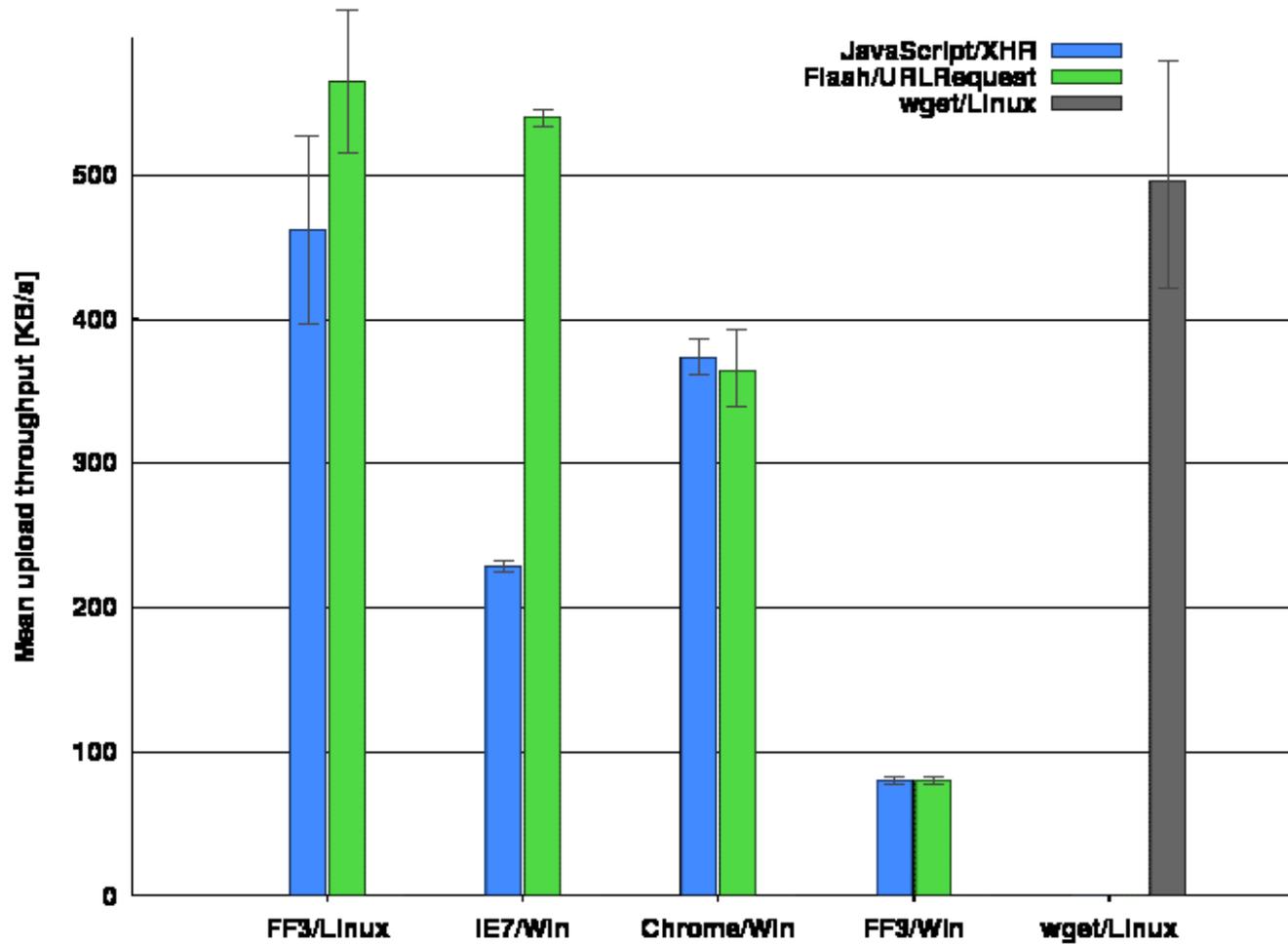
Performed from clients in a campus residential network on the U.S. West coast to server at WPI (U.S. East coast). Non-overlapping tests repeated every 12 minutes.

Used 2MB file for upload and download.

Controlled Download Throughput (Means and Confidence Intervals)

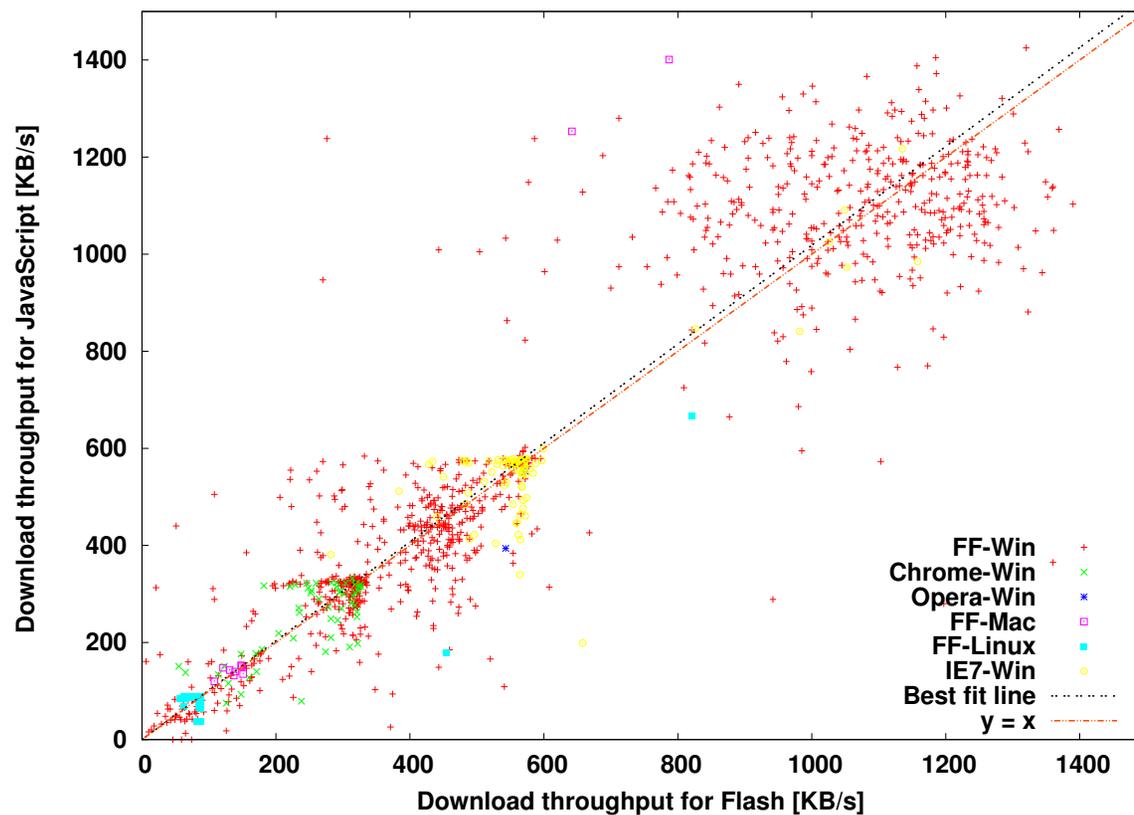


Controlled Upload Throughput (Means and Confidence Intervals)

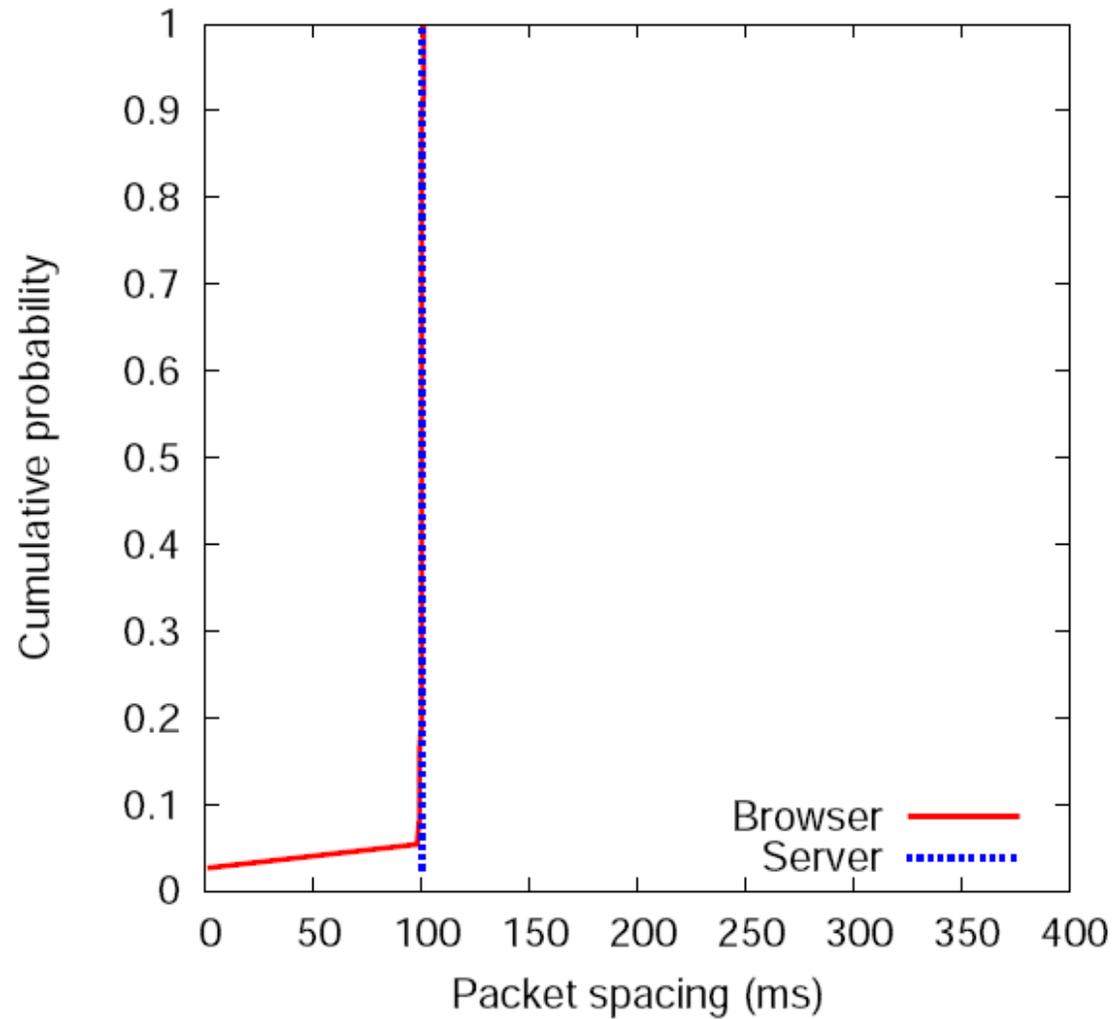


Comparison of Download Throughput Measured by JavaScript and Flash Methods for Broadband Connections

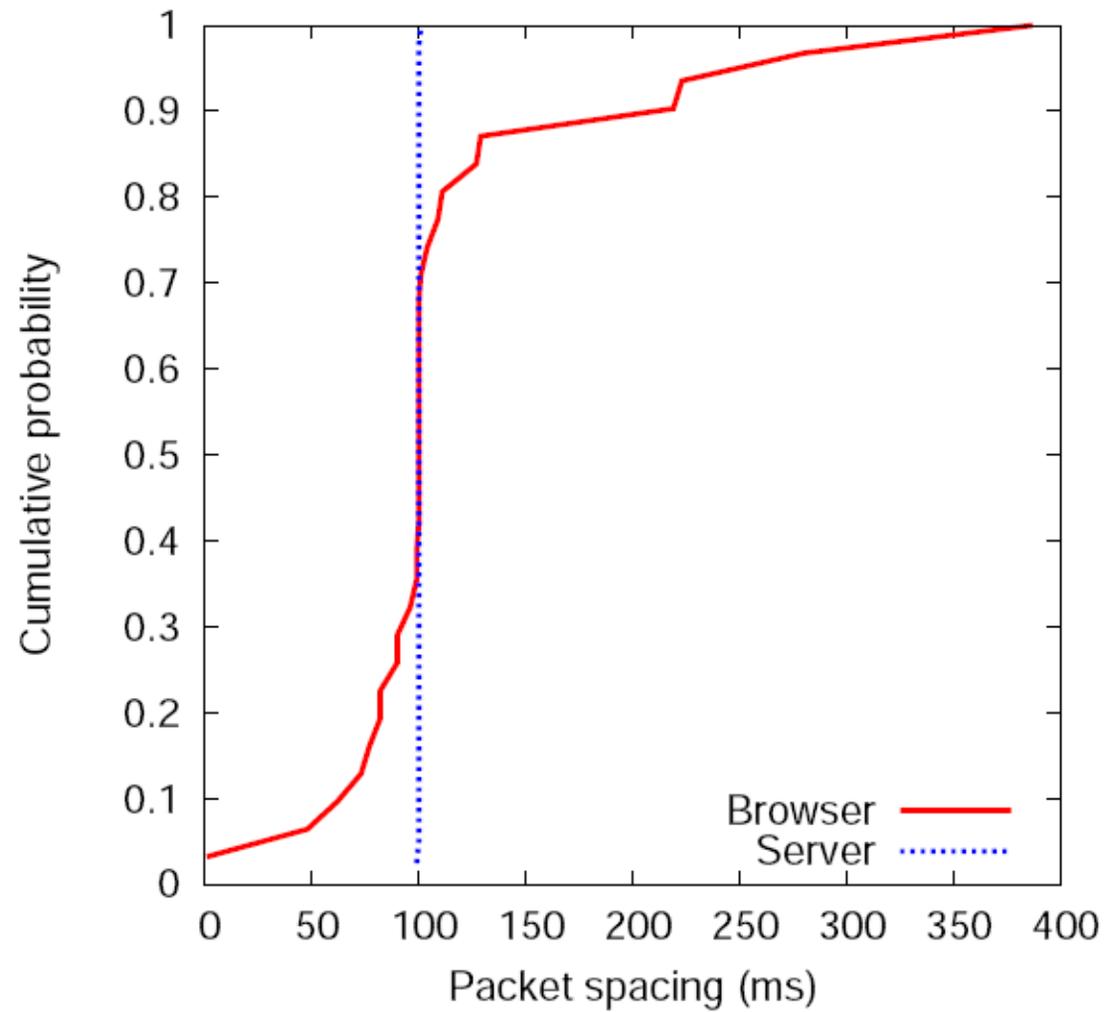
3610 total measurements from broadband ISP users.



Jitter—Packet Streaming without Interference



Jitter—Packet Streaming with Interference



Signed Java Applet Approach

Ability to gather a variety of information, such as:

- host and OS information
- LAN—wired/wireless connectivity, number and types of devices (PCs, game consoles, printers, Tivo, etc)
- Internet throughput measurements
- DNS performance

Similarities to ICSI's Netalyzer Project, but also focus on local network/devices and orient results toward user interests.

Have initial tool in place with use by home users with different types of connectivity (cable, DSL, FIOS).

Next Steps

Use what we have learned in development of these two measurement platforms to orient measurements and results towards user interests.

Focus presentation of results to make them *user centered* where the focus is not so much on network performance, but translating that performance into what it means for applications/sites of interest to user.

Orientation via:

- games, videos—explore measurements both alongside and within
- diagnostic health check

Look to initially attract users with simpler Flash/JavaScript-based tests with enticements for more comprehensive tests with applet-based approach.

Look to make results available to networking research community.