Fast and Light Bandwidth Testing for Internet Users

Xinlei Yang, Xianlong Wang, Zhenhua Li, Yunhao Liu,
Feng Qian, Liangyi Gong, Rui Miao, Tianyin Xu
Outlines

1. Background
2. Motivation
3. State-of-the-Art
4. Novel Design
5. Evaluation
6. System Demo
7. Conclusion
1. Background

Bandwidth testing services (BTSes) are widely used

- Core component of many network applications
- Cited by government reports & trade press
- Handy measurement tools for Internet users
1. Background

- BTSes are becoming increasingly important
  - Virtual Network Operators (VNO) catching on
  - Wireless access becoming ubiquitous
  - Bandwidth-hungry apps (e.g., UHD videos, VR/AR) emerging
2. Motivation

- Today’s BTSes are not satisfactory
  - Long test duration
  - Excessive data usage
  - Low accuracy for most BTSes

<table>
<thead>
<tr>
<th>Example</th>
<th>mmWave 5G, 1.15-Gbps downlink bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTSes</td>
<td>Duration (s)</td>
</tr>
<tr>
<td>Speedtest.att.com</td>
<td>19.1</td>
</tr>
<tr>
<td>Sourceforge.net</td>
<td>20.8</td>
</tr>
<tr>
<td>Fast.com</td>
<td>13.5</td>
</tr>
<tr>
<td>SpeedTest.net</td>
<td>15.7</td>
</tr>
</tbody>
</table>
2. Motivation

 Today’s BTSes are not satisfactory

- Long test duration
- Excessive data usage
- Low accuracy for most BTSes

Example

mmWave 5G, 1.15-Gbps downlink bandwidth

<table>
<thead>
<tr>
<th>BTSes</th>
<th>Duration (s)</th>
<th>Data Usage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedtest.att.com</td>
<td>19.1</td>
<td>1.37 GB</td>
<td>0.42</td>
</tr>
<tr>
<td>Sourceforge.net</td>
<td>20.8</td>
<td>2.75 GB</td>
<td>0.81</td>
</tr>
<tr>
<td>Fast.com</td>
<td>13.5</td>
<td>1.20 GB</td>
<td>0.68</td>
</tr>
<tr>
<td>SpeedTest.net</td>
<td>15.7</td>
<td>1.94 GB</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Can bandwidth testing be fast, light, and accurate simultaneously?
3. State-of-the-Art

**Popular Bandwidth Testing Websites**
- Alexa
- Google
- 18 popular bandwidth testing websites

**Commercial Bandwidth Testing Apps**
- WiFiMaster
- A popular Android/iOS app with 800 million users

**Important Bandwidth Testing Interfaces**
- Android 11
- 5G-oriented bandwidth testing Android SDK APIs
3. State-of-the-Art

Research methodology

- **Small-scale study**
  1. Network traffic tracing
  2. System reverse engineering

- **Large-scale benchmarking**

<table>
<thead>
<tr>
<th>Device</th>
<th>Location</th>
<th>Network</th>
<th>Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-1</td>
<td>U.S.</td>
<td>Residential broadband</td>
<td>88–96 Mbps</td>
</tr>
<tr>
<td>PC-2</td>
<td>Germany</td>
<td>Residential broadband</td>
<td>91–97 Mbps</td>
</tr>
<tr>
<td>PC-3</td>
<td>China</td>
<td>Residential broadband</td>
<td>90–97 Mbps</td>
</tr>
<tr>
<td>Samsung GS9</td>
<td>U.S.</td>
<td>LTE (60Mhz/1.9Ghz)</td>
<td>60–100 Mbps</td>
</tr>
<tr>
<td>Xiaomi XM8</td>
<td>China</td>
<td>LTE (40Mhz/1.8Ghz)</td>
<td>58–89 Mbps</td>
</tr>
<tr>
<td>Samsung GS10</td>
<td>U.S.</td>
<td>5G (400Mhz/28Ghz)</td>
<td>0.9–1.2 Gbps</td>
</tr>
<tr>
<td>Huawei HV30</td>
<td>China</td>
<td>5G (160Mhz/2.6Ghz)</td>
<td>0.4–0.7 Gbps</td>
</tr>
</tbody>
</table>
### 3. State-of-the-Art

#### Summarizing

<table>
<thead>
<tr>
<th>BTS</th>
<th># Servers</th>
<th>Bandwidth Test Logic</th>
<th>Duration</th>
<th>Accuracy (Testbed / 5G)</th>
<th>Data Usage (Testbed / 5G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBB</td>
<td>12</td>
<td>average throughput in all connections</td>
<td>10 s</td>
<td>0.59 / 0.31</td>
<td>42 MB / 481 MB</td>
</tr>
<tr>
<td>SpeedOf</td>
<td>116</td>
<td>average throughput in the last connection</td>
<td>8–230 s</td>
<td>0.76 / 0.22</td>
<td>61 MB / 256 MB</td>
</tr>
<tr>
<td>BWP</td>
<td>18</td>
<td>average throughput in the fastest connection</td>
<td>13 s</td>
<td>0.81 / 0.35</td>
<td>74 MB / 524 MB</td>
</tr>
<tr>
<td>STest</td>
<td>19</td>
<td>average throughput in all connections</td>
<td>20 s</td>
<td>0.89 / 0.81</td>
<td>194 MB / 2,013 MB</td>
</tr>
<tr>
<td>ATTTest</td>
<td>75</td>
<td>average throughput in all connections</td>
<td>15–30 s</td>
<td>0.86 / 0.53</td>
<td>122 MB / 663 MB</td>
</tr>
<tr>
<td>Xfinity</td>
<td>28</td>
<td>average all throughput samples</td>
<td>12 s</td>
<td>0.82 / 0.67</td>
<td>107 MB / 835 MB</td>
</tr>
<tr>
<td>FAST</td>
<td>~1,000</td>
<td>average stable throughput samples</td>
<td>8–30 s</td>
<td>0.80 / 0.72</td>
<td>45 MB / 903 MB</td>
</tr>
<tr>
<td>SpeedTest</td>
<td>~12,000</td>
<td>average refined throughput samples</td>
<td>15 s</td>
<td>0.96 / 0.92</td>
<td>150 MB / 1,972 MB</td>
</tr>
<tr>
<td>Android API-A</td>
<td>0</td>
<td>directly calculate using system configs</td>
<td>&lt; 10 ms</td>
<td>NA / 0.09</td>
<td>0 / 0</td>
</tr>
</tbody>
</table>

3. State-of-the-Art

Reflection of bandwidth testing

Ideal case

Real case

Noises
(congestion control, link sharing, etc.)

User’s access link bandwidth
3. State-of-the-Art

- Combating noises

**Space Dimension**

*Speedtest.net*

**Time Dimension**

*Fast.com*

Our strength is in our hosted servers

The accuracy and high-quality performance of Speedtest is made possible through the 11,000+ servers around the world that host our Speedtest server daemon. This robust network of servers enables us to ensure that our users get local readings wherever they are on the planet.

Using large-scale test server deployments (*spatial redundancies*) to ensure high-quality network connections, *largely reducing noises.*

Using long test duration (*temporal redundancies*) to wait for the coming of sufficient desired samples.

Test duration: often 20 - 30 s
Most of today’s BTSes use excessive **temporal and spatial redundancies** for **combating** noises.

Large-scale network deployments, long test duration, and excessive data usage.

Can we **accommodate and exploit** the noises rather than exhaustively suppress the impact of them?
4. Novel Design

- Re-consider BTS through rejection sampling

- Target Distribution
- Proposal Distribution
- Acceptance-Rejection Function (ARF)
- Rejected Samples
- Accepted Samples
4. Novel Design

Modeling the bandwidth testing process

Samples following the $P(x)$ distribution

$ARF$ $T(x)$

Accept

Rejected samples

Rejected samples

Bandwidth Estimation

Accepted samples

Rejection Sampling
4. Novel Design

- Modeling the bandwidth testing process

![Throughput samples](chart.png)

Samples following the P(x) distribution

**Fuzzy Rejection Sampling**

\[ ARF(T(x)) \]

- Accept
  - Accepted samples
  - Bandwidth Estimation
- Reject
  - Rejected samples
4. Crucial Interval Sampling (CIS)

**Key Findings:**
- Noise samples are scattered across a wide throughput interval.
- Desired samples tend to concentrate within a narrow interval.

**Crucial Interval: (ARF)**

\[ F(V_x, V_y) = Density \times Size = \frac{V_{\text{max}} - V_{\text{min}}}{N} \cdot \frac{K^2(V_x, V_y)}{V_y - V_x}, \quad V_y - V_x \geq \frac{V_{\text{max}} - V_{\text{min}}}{N - 1} \]

A throughput interval (1) whose density is as high as possible, and (2) which contains as many samples as possible.
4. Crucial Interval Sampling (CIS)

Crucial intervals converge quickly

Accepted Samples
intensify the crucial interval by making it denser

Rejected Samples
help better “contrast” the crucial interval

Both accepted and rejected samples are exploited to make bandwidth tests fast and light.

Video available at https://youtu.be/lgZOy59im7M
4. Crucial Interval Sampling (CIS)

- Convex hull acceleration

- **Brute-force mechanism**
  Walking through all the throughput samples to find the crucial interval. \(O(N^2)\)

- **Convex hull acceleration**
  Dynamically maintaining a convex hull for quickly finding the crucial interval. \(O(N \log N)\)
4. Elastic Bandwidth Probing (EBP)

- **BBR**: emerging congestion control mechanism with a built-in bandwidth probing scheme
- Leveraging and improving BBR to realize elastic bandwidth probing
- Making crucial interval always effective

![Graphs showing throughput over time with crucial intervals and accepted/rejected samples.](image)
4. Novel Design

- Architecture of FastBTS

- Fuzzy Rejection Sampling

- Crucial Interval Sampling

- Elastic Bandwidth Probing
4. Novel Design

- Architecture of FastBTS

- Crucial Interval Sampling

- Fuzzy Rejection Sampling

- Elastic Bandwidth Probing

- Traffic-driven Server Selection

- Adaptive Multi-homing

Cannot saturate user’s access link bandwidth
4. Data-driven Sever Selection (DSS)

Ping-based server selection

Low latency ≠ high throughput

Historical performance-based server selection

Select servers with highest bandwidth estimations

Ping test

Historical Data

Crucial latency interval

Select candidate servers

max avg

Sorting servers

Throughput

Latency

Latency Interval
Accepted Samples
Rejected Samples
4. Adaptive Multi-Homing (AMH)

Under-estimating user’s bandwidth (e.g., 5G)

Adding concurrency level with fixed threshold

Adaptive Multi-Homing

Adding concurrency level

When shall we stop adding extra test servers?

Aggregated throughput increasing?

Generating result

Y

N
5. Evaluation

- Testbed networks

LAN

Residential broadband

Datacenter network

LTE network

mmWave & Sub-6Ghz 5G network

HSR cellular network
5. Evaluation

- Major results
  - **FastBTS vs. others on testbed networks:** 5%–72% higher average accuracy, 2.3–8.5× shorter test duration, 3.7–14.2× less data usage.
  - **FastBTS vs. SpeedTest.net in real world:** FastBTS (with only 30 servers) achieves comparable accuracy compared with the production system of SpeedTest.net with ~12,000 test servers, incurring 5.6× shorter test duration and 10.7× less data usage on average.
6. System Demo

- Case 1: PC + Wi-Fi (~100 Mbps)

  **SpeedTest.net**

  - Duration: 15.0 seconds
  - Result: 95.18 Mbps
  - Data usage: 176 MB

  **FastBTS.thucloud.com**

  - Duration: 3.1 seconds
  - Result: 99.25 Mbps
  - Data usage: 37 MB

Videos available at: https://youtu.be/QbHO27RvzbU
6. System Demo

- Case 2: smartphone + Sub-6Ghz 5G (~500 Mbps)

**SpeedTest.net**
- **Duration:** 15.0 seconds
- **Result:** 484 Mbps
- **Data usage:** 936 MB

**FastBTS.thucloud.com**
- **Duration:** 4.1 seconds
- **Result:** 543.07 Mbps
- **Data usage:** 168 MB

Videos available at: https://youtu.be/VGN32d3dIAU
We reveal how today’s commercial bandwidth testing services actually work as well as their pros and cons based on in-depth investigations and large-scale benchmarking tests.

We present FastBTS, a novel bandwidth testing solution that accommodates and exploits network noises to make bandwidth tests fast and light. With only 30 test servers, FastBTS achieves comparable accuracy compared with SpeedTest.net with ~ 12,000 servers, while incurring 5.6× shorter test duration and 10.7× less data usage on average.

We have released all the source code at https://FastBTS.github.io and an online demo system at http://FastBTS.thucloud.com.

Thanks!
Q & A