

## Today's Bandwidth Testing Services (BTSeS)

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

### Research Objects

- Popular Bandwidth Testing Websites
- Commercial Bandwidth Testing Apps
- Important Bandwidth Testing Interfaces

### Unsatisfactory Performance

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

### Key Insight: How Today's BTSeS Combating Noises?

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*.

Test duration: often 20 - 30 s

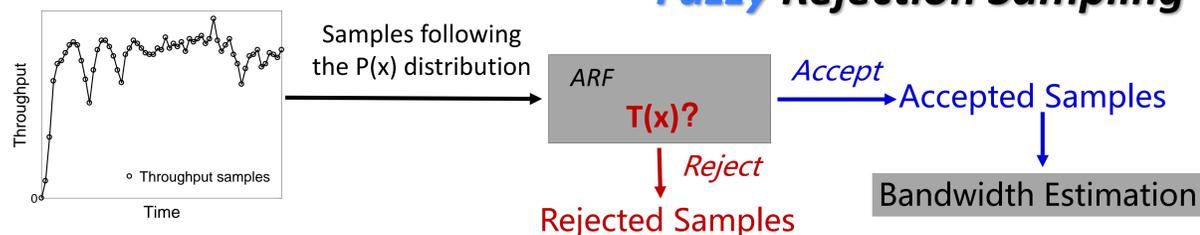
Using long test duration (*temporal redundancies*) to wait for the coming of sufficient desired samples.

Most of today's BTSeS use excessive temporal and spatial redundancies for combating noises

Most of today's BTSeS use excessive temporal and spatial redundancies for combating noises

Can we accommodate and exploit the noises rather than exhaustively suppress the impact of them?

### New Methodology

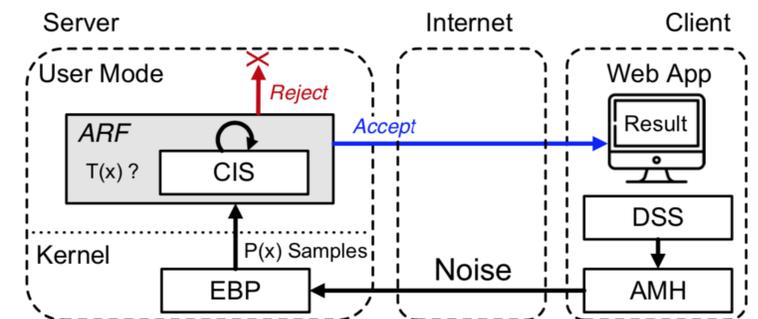


### Fuzzy Rejection Sampling

## FastBTS: Making BTS Fast, Light, and Accurate

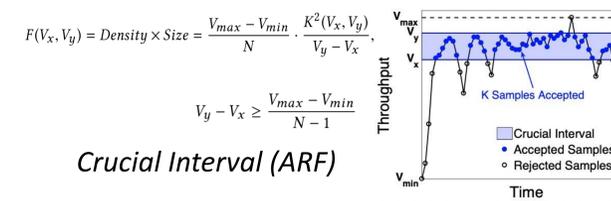
### Architecture of FastBTS

- A fast and lightweight BTS implementing fuzzy rejection sampling
- Accommodating and exploiting noises

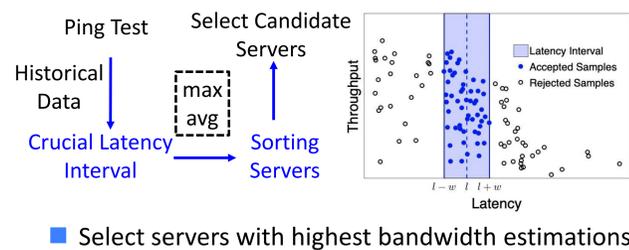


### Core Techniques of FastBTS

#### Crucial Interval Sampling (CIS)

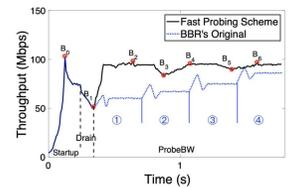


#### Data-driven Server Selection (DSS)



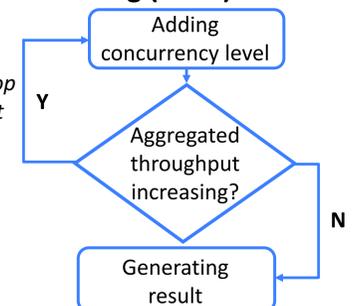
#### Elastic Bandwidth Probing (EBP)

- Leveraging & improving BBR to realize elastic bandwidth probing
- Making the crucial interval always effective

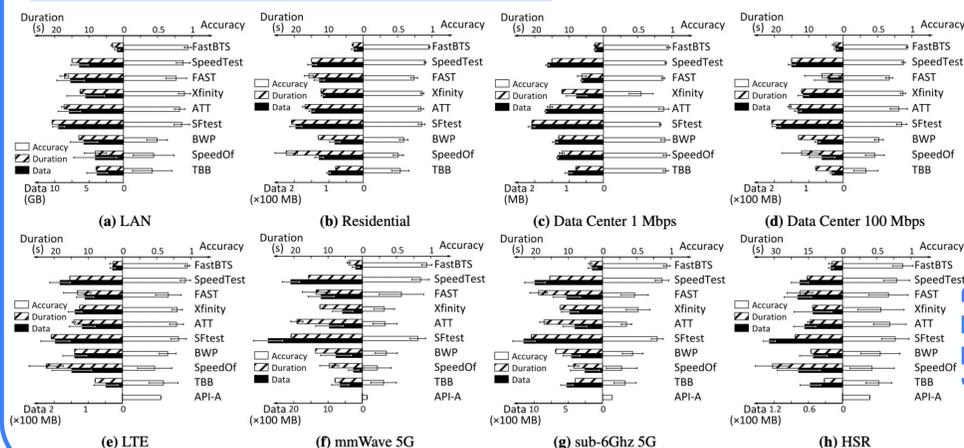


#### Adaptive Multi-Homing (AMH)

When shall we stop adding extra test servers?



### Implementation & Evaluation



FastBTS (with only 30 servers) achieves comparable accuracy compared with the production system of SpeedTest.net with ~12,000 test servers, incurring 5.6x shorter test duration and 10.7x less data usage on average.

Online Demo System  
<http://fastbts.thucloud.com/>