Proof of Backhaul: Trustfree Measurement of Broadband Bandwidth

Peiyao Sheng, Nikita Yadav, Vishal Sevani, Arun Babu, SVR Anand, Himanshu Tyagi and Pramod Viswanath
Open Networking

• **1990s**: Heterogeneous networks linking computers
  - TCP/IP: decentralized routing and congestion control
  - Web 1.0

• **2010**: Giant content delivery networks
  - Centralized data centers, cloud computing, caching
  - Web 2.0
Tail Winds of Decentralization: Private 5G

Components to setup private 5G networks are ready!
Blockchains: Stitching Together

- Low friction way to stitch things together
  - Open and trustless
  - Tokenization of incentives
Network Telemetry

• Centralized Telemetry
  • Monitor and optimize network performance

• Decentralized Telemetry
  • Open: no powerful servers (any device)
  • Trustfree: no trusted parties (Byzantine resistance)
  • Network meritocracy: incentive compatibility
Proof of Backhaul

A cryptographic proof system establishing that each party is contributing appropriately towards enabling backhaul bandwidth
Centralized Measurements

Speedtest: speedtest.net

- Nearby powerful servers (high bandwidth, low latency, low packet loss)
- A dedicated foreground service to flood the connection
Centralized Measurements

Speedtest: speedtest.net

- **Not open**: High barrier to entry to be a challenge server
- **Not trustfree**: Need to trust the challenge server for sending data and the prover for timing measurements
Traffic Aggregation

• Multiple challengers send packets simultaneously
• Packets arrive at the network core around the same time
• Aggregated to an equivalent powerful challenger

Open but not trustfree
Interactive Measurement

- Pathchar [97’]

\[ RTT_i = \sum_{k=1}^{i} \frac{B}{\theta_k} + \text{delay} \]
\[ \theta_i = \frac{B}{RTT_i - RTT_{i-1}} \]

Trustfree but not open:
For high-bandwidth provers (>100Mbps), needs a very low jitter path between the challenger and the prover
Combining Aggregation and Interactivity

• Aggregate traffic from multiple challengers

• Prover sends a timing signal upon receiving all the packets
Attacks

• **Withholding**: corrupted challengers can refuse to send the packets

• **Rushing**: corrupted prover can collude with a subset of challengers to get packets from an external channel
Trustfree Proof of Backhaul

- Open: Use Traffic Aggregation

- Trustfree: A Byzantine Fault Tolerant (BFT) interactive measurement scheme
Formal Security Properties

- Soundness: no prover can inflate the bandwidth
- Approximate Completeness: if the prover is not corrupted, the protocol will output a bandwidth $\theta'_p \geq \alpha \theta_P$

- Accuracy rate

$$\alpha = \frac{n - 2f}{n - f}$$

where $n$ is the number of challengers, $f$ is the number of Byzantine faults
Protocol Primitives

• Unforgeable packets
  • Digital signatures

• Robust timing measurement
  • Median is bounded by honest reports

• Short witness
  • Hash and Merkle tree
Multichallenger PoB Protocol
Multichallenger PoB Protocol
Multichallenger PoB Protocol

Blockchain (Verifier)

Server (Challenger)

$\text{sign}(m, sk_1)$

$\text{sign}(m, sk_i)$

$\text{sign}(m, sk_n)$
Multichallenger PoB Protocol
Multichallenger PoB Protocol
Multichallenger PoB Protocol

1. Verifiable traffic aggregation: Multiple challengers send unforgeable traffic
2. Short packet receipts: Prover commits received packets using Merkle root
3. Local Verification: Challengers verify that their respective challenge traffic was received
4. Robustification: Take median of the RTT measurements
Design Scope Exploration

- Packets: UDP / TCP
- Crypto primitive: with / without signature
- Threat model: with / without access to extra link

| TABLE II: Comparison of different protocols in design landscape |
|---------------------------------------------------------------|
| **Packets** | **Crypto primitive** | **Rushing attack** | **Accuracy**               |
| PoB         | UDP               | signature         | Yes                        | $(1 - 2\beta)/(1 - \beta)$ |
| PoB-TCP     | TCP               | signature         | Yes                        | $1 - \beta$               |
| PoB-PRG     | UDP               | pseudorandom generator | Yes                  | $(1 - 3\beta)/(1 - \beta)$ |
| PoB-shuffle | UDP               | signature         | No                         | $1 - \frac{(1+\delta_b)^\xi}{(1-\delta_g)(1-\beta)^t}$ \(\delta_0<\delta_1, \delta_2 \leq 1\) |
System View

- Practical factors: network jitter, synchronization error, computation overhead
- Lightweight: small amount of challenge data
- Open: geographically spread challengers with low bandwidth
- Secure under attacks

| Technique                        | Secure | Challenger BW < Backhaul BW | Accuracy |
|----------------------------------|--------|-----------------------------|----------|
| Pathchar                         | x      | ✓                           | Low      |
| Packet dispersion based          | x      | x                           | -        |
| Secure BW estimation             | ✓      | x                           | -        |
| Multichallenger PoB              | ✓      | ✓                           | High     |

| Backhaul BW (Mbps) | Challenger BW (Mbps) | Challenge Data (MB) | Attack | Measured BW (Error %) | Guaranteed BW (Mbps) |
|--------------------|----------------------|---------------------|--------|----------------------|---------------------|
| 250                | 25                   | 3.44                | -      | 246 (1.6%)           | 184                 |
| 500                | 20                   | 6.86                | -      | 475 (5%)             | 356                 |
| 750                | 75                   | 10.31               | -      | 705 (6%)             | 529                 |
| 1000               | 100                  | 13.75               | -      | 921 (8%)             | 691                 |
| 250                | 32                   | 3.44                | Rushing| 331 (0.6%)           | 249                 |
| 250                | 32                   | 3.44                | Withholding | 241 (3.6%) | 181                  |
Thanks!

Full paper: https://arxiv.org/abs/2210.11546
Github: https://github.com/multichallengerpob/proof-of-backhaul
Email: peiyaosheng@gmail.com