Witness Chain: Proofs of Bandwidth

Motivation and High-level Description

Why do we need it?

Witness Chain provides a Proof of bandwidth (PoB) system, namely a proof system used to establish trust in decentralized wireless networks. The use of these proofs is best illustrated in the context of a decentralized wireless network marketplace with many (small) suppliers of bandwidth and many consumers of bandwidth, although there are many applications beyond it. In such a decentralized wireless network marketplace, the following two big trust challenges arise:

1. Does a supplier really have the available bandwidth to support its offerings?
2. How can we ensure fair payments and service as the quality of bandwidth varies?

These questions arise in different forms in different use cases of interest. The following list of example use cases will help set the context:

- When connecting to a supplier hotspot, a user wants to connect to the one which has higher available bandwidth. How can it verify reliably which one has what bandwidth to offer?
- A carrier wants to allow data offloading for its users to hotspots hosted by individual suppliers. It can use a modification of the roaming protocol to allow its users to attach to these hotspots. Since the data is routed through the home network in roaming, the carrier will verify the usage itself and trust the usage bill provided by the supplier. However, how does he know that the supplier is giving appropriate QoS for the service? In a centralized network, the carrier has visibility of its own RAN performance and also does field tests to verify the QoS. But in this decentralized setting, can the carrier verify that its users are getting the services they are paying for.
- An individual user is connecting to a hotspot hosted by another individual supplier with no prior reputation. The supplier provides individual a USD 10 coupon to avail up to 10GB data at 100Mbps. How can the individual user trust that the supplier will provide the promised throughput, and even account for the usage correctly? What if the supplier simply declares 10GB usage when the actual usage is only 8GB? What if the throughput provided was much lower than 100Mbps.
- A supplier hotspot has been providing a high quality network to many individual users for a long time. It has always met its promised bandwidth quality. How can it build a reputation around its high performance in a provable manner?

Witness Chain is a toolkit of basic proofs that can be used to address these challenges.

What is it?

Witness Chain is a cryptographically secure proof system to allow assertion of network service quality and quantity for various links. This proof system is intended to be used in the trustfree decentralized network setting outlined above. Below we describe the basic outline of this proof system and the individual proofs implemented.

- **Proofs:** A proof is a two-party protocol (program) where the first party (the verifier) wants to verify a fact about the second party (the prover). The prover may try to cheat to bluff the verifier. A secure proof is the one where the prover cannot cheat.

  More concretely, in our setting the prover is a node in a network (the hotspot or an intermediate node) which needs to prove the performance of a network link incident on it to a verifier who can route traffic through that link. The verifier and prover can either be the sender of the challenge traffic, be an intermediate note, or the receiver for the challenge traffic.

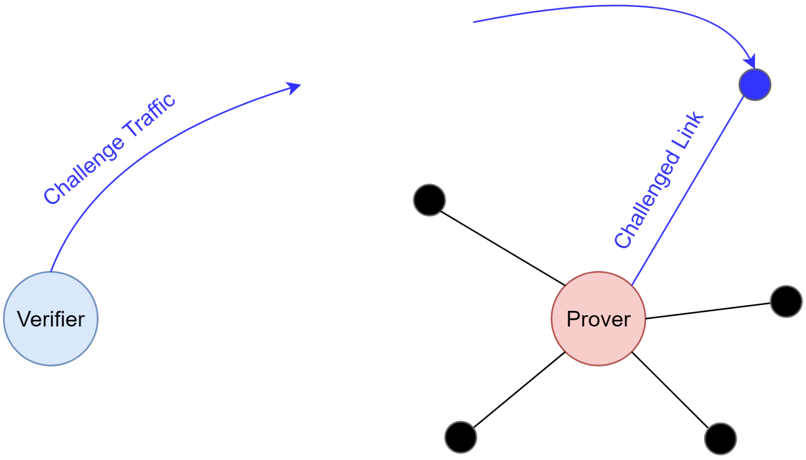

Fig 1. A PoB system with one prover and one verifier.

- **Multiple Provers:** In the basic description, the verifier is itself interested in evaluating the performance of the link associated with the prover, and has no incentive to cheat. The verifier is the sender and initiates the traffic, which goes over the internet and reaches the prover, who is the receiver. Often we are interested in settings where the verifier is not one of these but an external entity (such as the blockchain ledger!), and both the sender and the receiver are provers. In these multiprover proof systems, a subset of malicious provers may try to fool the verifier. We need to device multiprover PoB systems where the verifier cannot be fooled despite a subset of provers being malicious.

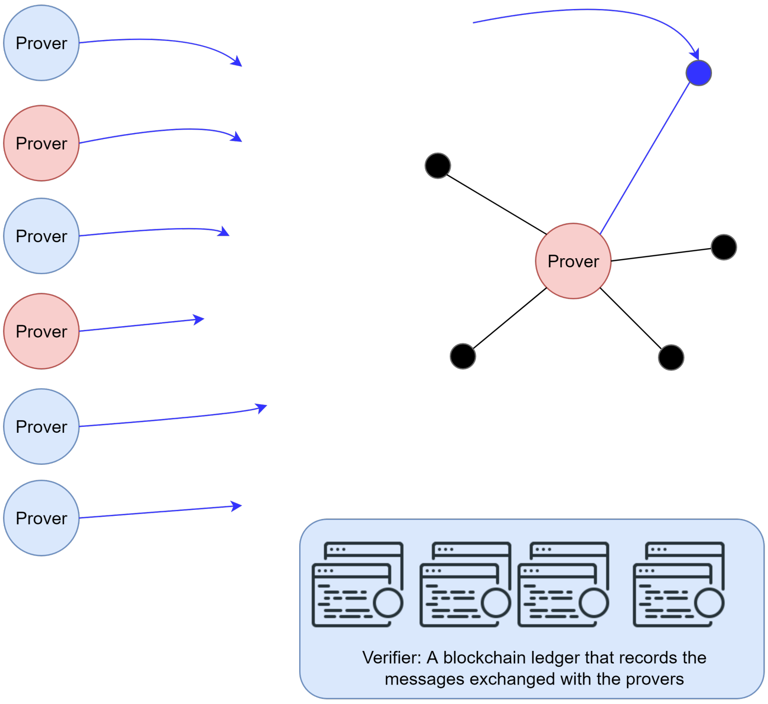
Prover
Prover
Prover
Prover
Prover
Prover
Prover
Verifier: A blockchain ledger that records the messages exchanged with the provers

Fig 2. A mutiprover PoB system.

- **Measurements and Proofs:** It is important to distinguish between local measurements and a provable measurement in this discussion. Even in existing networks, telemetry plays an important role for optimizing the network performance. The ability of a hotspot, WiFi or 5G, to measure the usage and performance for different flows it supports exists even now (although it may not be prevalent in home WiFi networks). However, such a measurement is local and cannot be proven to others. Proof of bandwidth provides a proof system that allows network nodes (such as a hotspot) to prove their networking performance to each other. Nonetheless, we often use local measurement services of the device as a blackbox entity and use them in our protocols. Our protocols build over these local measurements, make them provable, and even enable reconciliation when multiple network nodes make measurements related to the same flow.

- **Proof of Backhaul:** This is a mutiprover proof system where a WiFi or 5G hotspot, which has a wired/fiber backhaul connection, wants to establish the amount of available bandwidth at the hotspot backhaul. Towards that, multiple senders send challenge traffic (measurement probe) to the hotspot. In the trusted setting, one can use a standard measurement probe for this – a server close to the hotspot can send a certain amount of traffic to it and the prover can calculate what is the time taken to receive that data. In our decentralized, trust-free setting, many issues arise:
    - The hotspot need not give correct measurements of the timing.

- ○ The server sending the data may not be "very close" to the hotspot. Here being very close translates to the requirement that the bottleneck link is the backhaul of the hotspot. In fact, we want an open system where someone with a much lower throughput at the backhaul than the hotspot being challenged can verify the backhaul.
- ○ The server may collude with the hotspot and give a favorable outcome.

Our proof of backhaul addresses these issues by setting-up a multiprover proof system with the following features:

- ○ The challenge traffic from multiple senders are combined to allow them to collectively measure the throughput of backhaul larger than their own throughput.
- ○ A cryptographic scheme is used to ensure that the hotspot, even when colluding with a subset of senders, can succeed only if it has the claimed backhaul throughput.

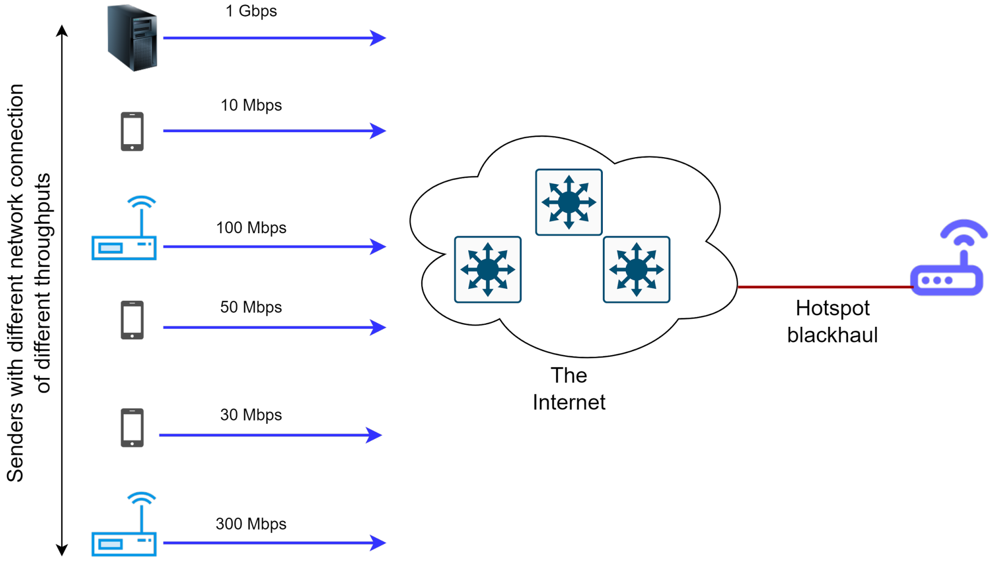

Fig 3. Proof of Backhaul setup.

- **Proof of Service:** This is a two-prover proof system where one party is the supplier (the hotspot providing the network) and the other party is the consumer (the user consuming the network). The supplier provides network service to the consumer, and they want to ultimately prove the delivery of the service to the blockchain verifier to settle the payment. The service must be delivered in accordance with a service level agreement (SLA) which determines quality of service (QoS) and price based on the usage for a given QoS. In a trusted setting, the supplier completes the service and gives the bill. For example, it can say that it served 100GB at 100Mbps. However, in the decentralized setting a malicious supplier can falsely claim delivery. Our proof of service allows the consumer to independently measure the usage; it can even be supplemented with other proofs to provide the consumer with an estimate of the throughput (see the proof of wireless QoS below). If the consumer's and supplier's measurements deviate from the beyond expectations, proof of service allows for renegotiation of price. Of course, either party can disconnect anytime if they are not satisfied. Instead of accounting for the whole service as a single unit, we account for it in small units and release micropayments (which are simply proofs of payment which can be used in the final settlement).

In the description above, the proof of service does not entail anyone besides the supplier and the consumer. No one outside needs to verify it. But once the procedure concludes, the very fact that the consumer promised to pay the supplier for the service can be viewed as a proof that the service was delivered as per the SLA. This way proof of service can be accepted by an external verifier (the blockchain ledger) as well.

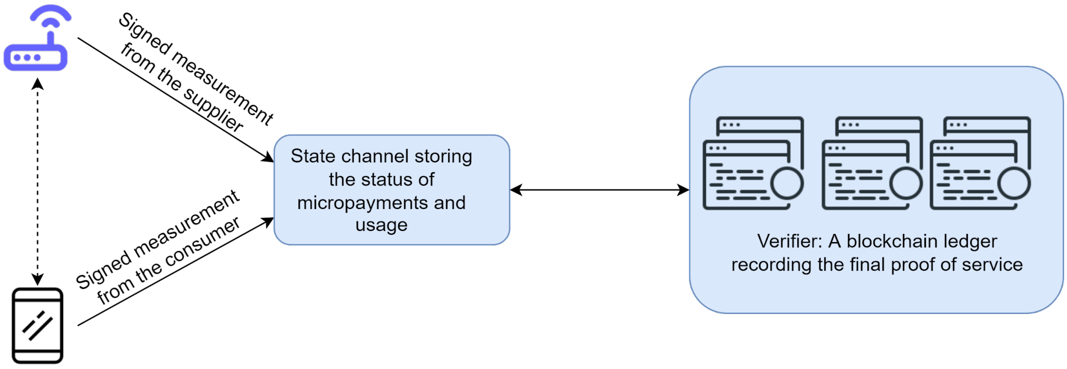
Signed measurement from the supplier
Signed measurement from the consumer
State channel storing the status of micropayments and usage
Verifier: A blockchain ledger recording the final proof of service

Fig 4. Proof of Service setup. The supplier and the consumer both make independent measurements of service and its quality, and the proof system helps them reconcile.

- **Proof of Wireless QoS:** This is a single-prover proof system where the hotspot is the prover and the connected user (the consumer) is the verifier. The verifier can send traffic to the prover for uplink communication or may receive traffic from the prover for downlink communication. The verifier needs proof that the prover hotspot is giving it the QoS that it is claiming; say, the throughput, latency, and packet dropped rate are as claimed.

  (Details to be elaborated after implementation.)

- **Proof of Route:** This is a multiprover proof system with the verifier being either the connected user (the consumer) or the blockchain. The verifier needs to know that the route followed by its traffic is the same, or has the same bottleneck link, as the one which was earlier evaluated. For instance, once the proof of backhaul is done, the user wants to make sure that its traffic is routed from the same path.

  (This is work in progress – we need to extract a signature for the path from the IP headers and maybe delay profile for links along the path.)