

Althea

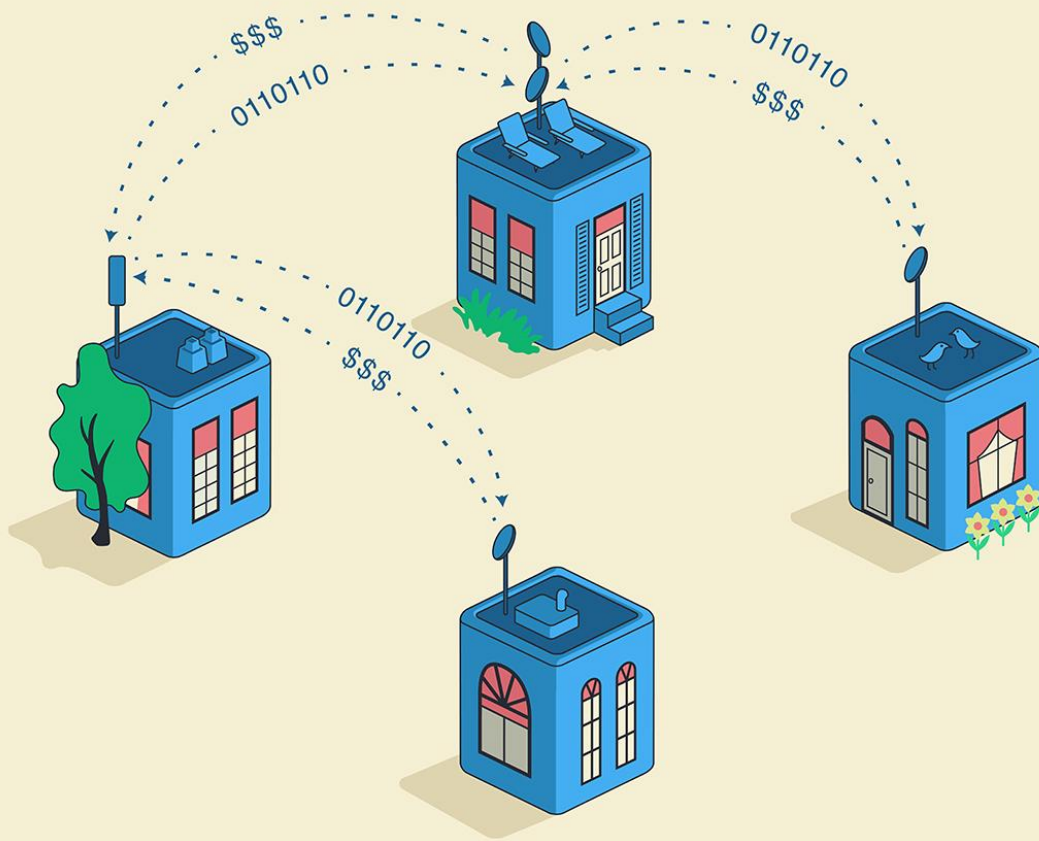
---

# Building an ISP in a box

By Justin Kilpatrick



- **Why is the last mile so expensive?**
- **WISPs or Wireless ISPs build out networks for less than \$300 in capital costs per user**
- **May only have a single employee for hundreds of users**
- **The last mile is profitable to build, but expensive to coordinate**

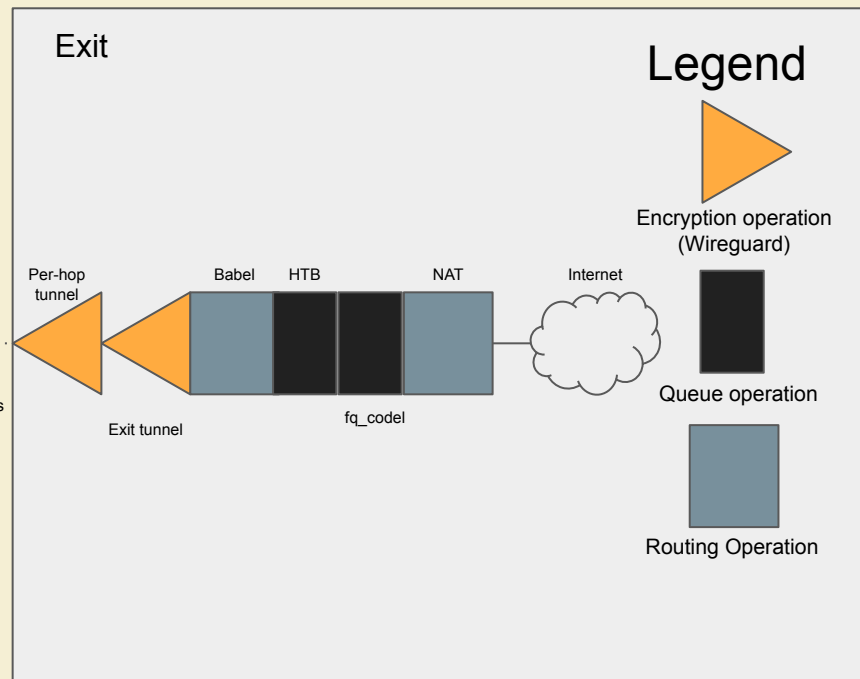
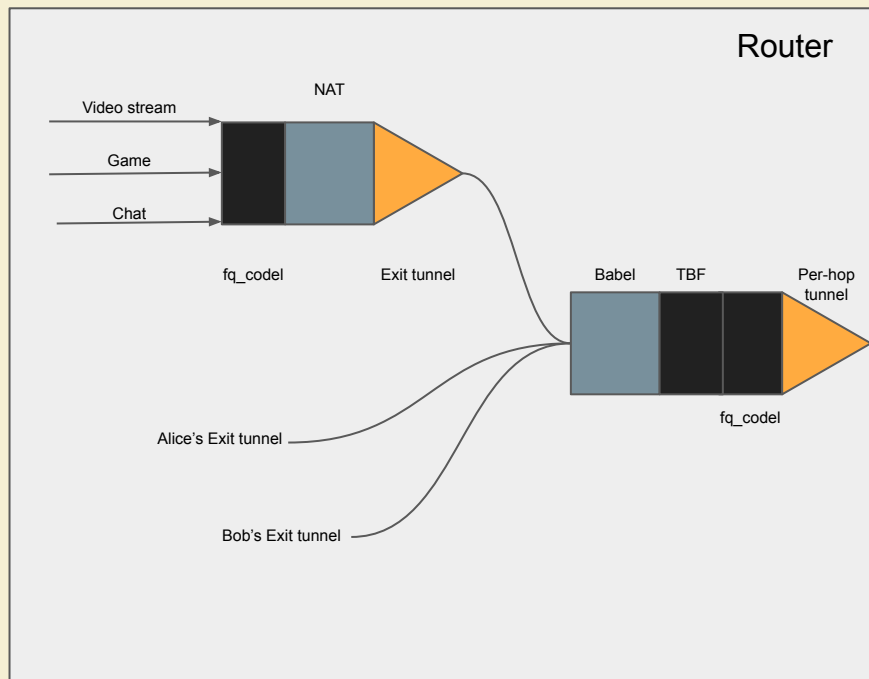


- **Althea is a decentralized ISP**
- **Commoditize bandwidth the same way the cloud commoditized server time**
- **Zero software setup, minimal hardware setup**
- **Separate customer service through local 'network organizers'**
- **Reduce the cost and increase access by automating coordination**

---

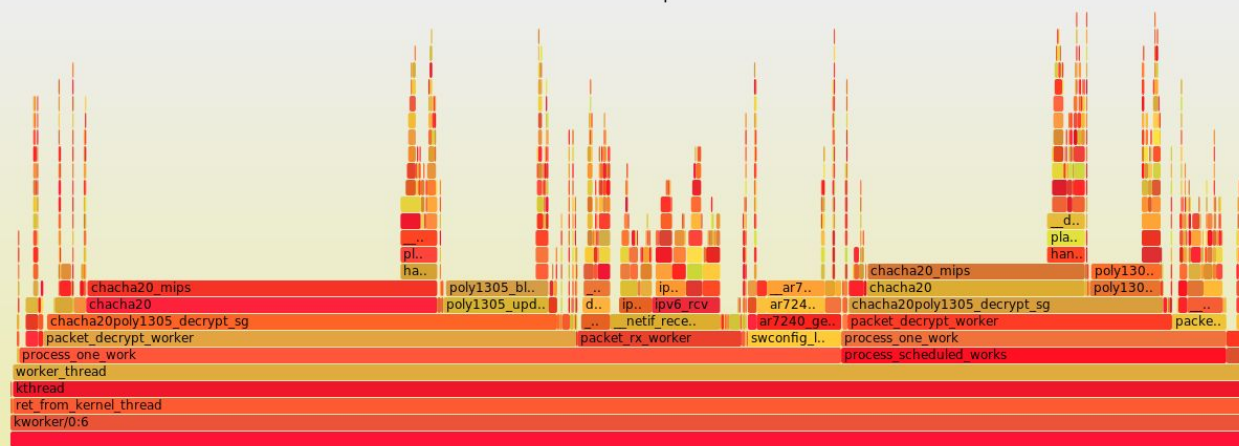
## Building an entire ISP, in one very small box

- An OpenWRT distribution installed on home routers
- Routing with link state detection and price tracking
- Automated QOS and congestion management
- Peer to peer autonomous billing
- Automated incident resolution
- All in 128MB ram and 16MB storage



Flame Graph

Search

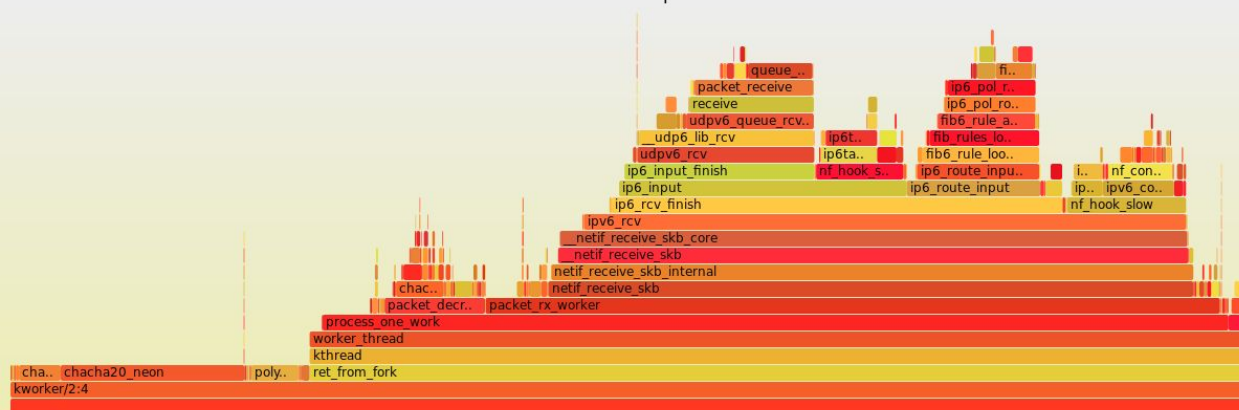


## Wireguard Perf

- WireGuard is an in kernel high performance VPN
- It's sessionless nature and low overhead make Althea possible

Flame Graph

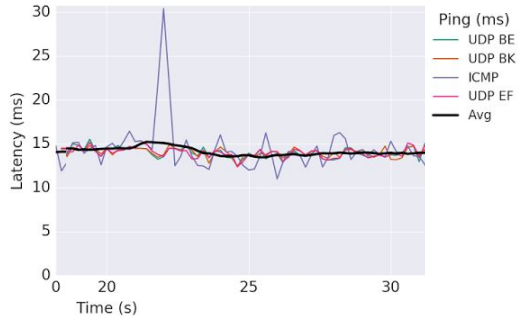
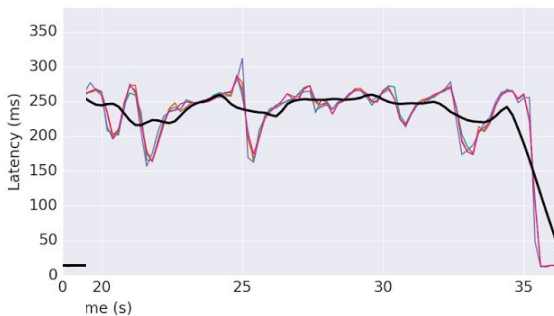
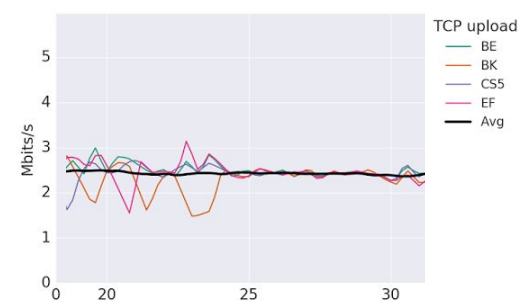
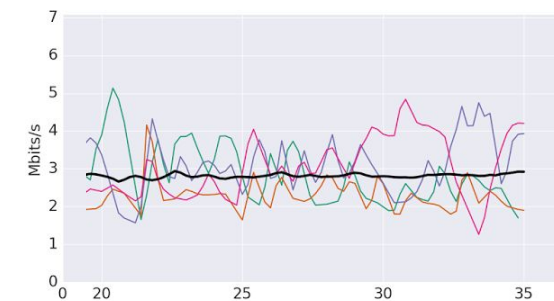
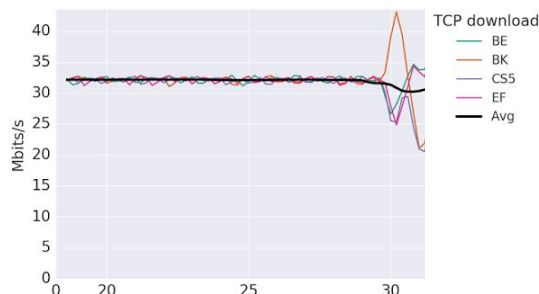
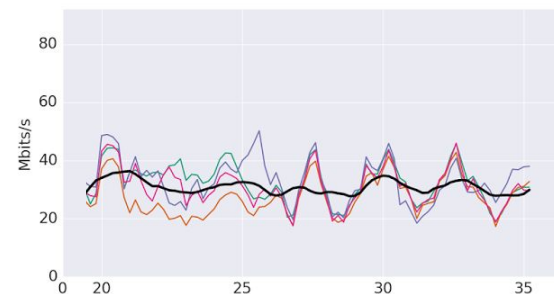
Search



- For the ARM router (bottom) kernel routing is far slower than the cryptography

Response Under Load  
oad, ping (unscaled versions)  
00\_lede\_cake\_unlimited

Response Under Load  
upload, ping (unscaled versions)  
00\_lede\_piece\_cake\_135\_12



## FQ\_Codel

- Solves the traffic management problem without requiring any human involvement
- Applied at two levels to fairly distributed bandwidth between user connections on each router and between routers
- Althea can't use hardware acceleration anyway

# Babel

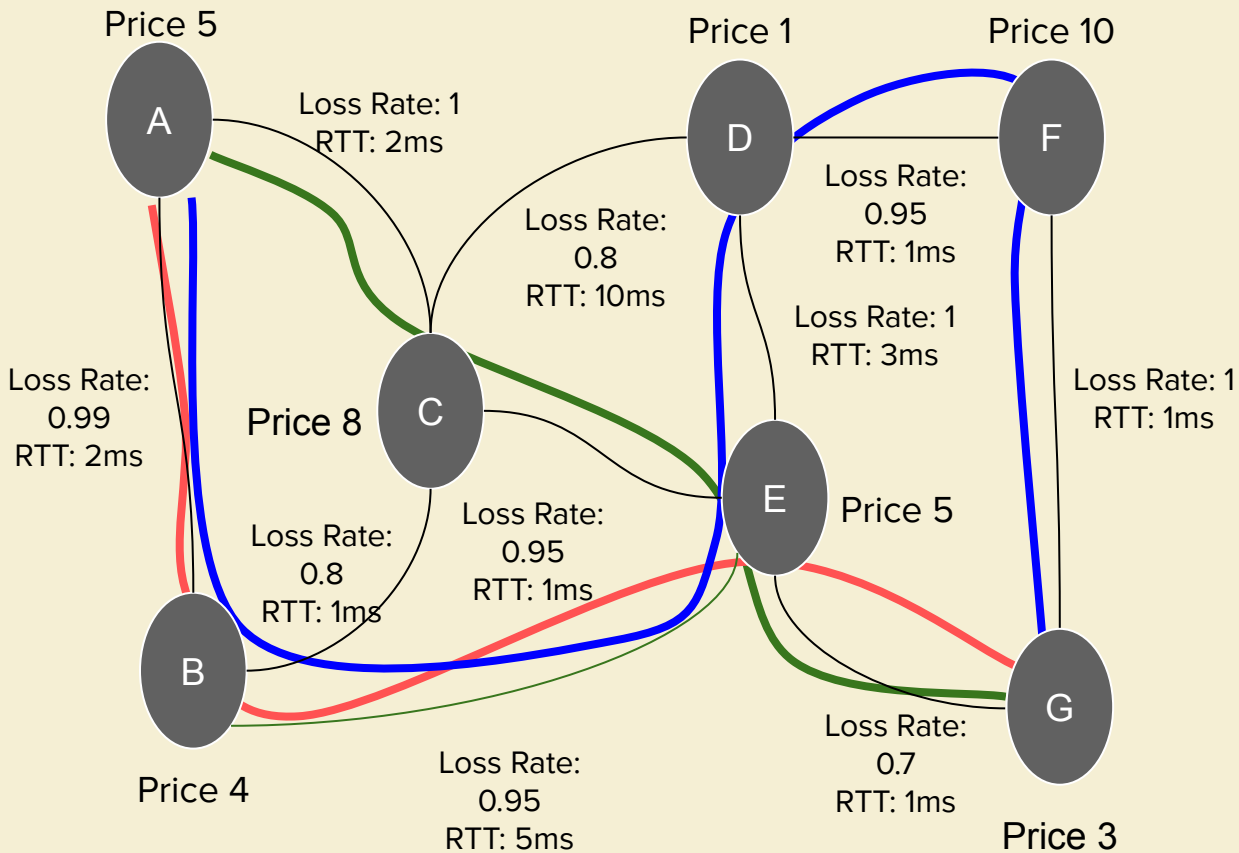
- Open source
- Distance vector with link state detection
- An exhaustively defined spec with multiple implementations
- Extendable
- Inserts routes directly as kernel routes

# The key Insight, multi-vector distance vector

- We want to create a network that is also a market for bandwidth
- If we phrase all purchase criteria as distance vector metrics the output of Bellman-Ford routing becomes the 'best buy'
- But we have to keep each vector separate so that it can be evaluated individually. You can't know what price to pay if you have a single 'metric' field

## Verifiable metrics

- We can make an efficient automated market out of distance vector, but distance vector is trivially spoofable.
- If we assume we have an encrypted connection to our destination we can measure some network properties objectively
- Round trip time and cumulative packet loss can be determined by cooperating nodes on each end, then compared to the advertisement



## Example network

- For routing decisions we re-aggregate the metrics as the sum of logs
- Exact outcome dependent on user preference
- Kernel routing only allows one route per destination at a time
- While route computation is  $O(\text{neighbors})$  memory usage is  $O(\text{network} * \text{neighbors})$

# Routing - Challenges

- Babel does not (and should not) operate in real time, meaning packet loss and rtt as the network advertises will be a sample over a longer period than the actual validation sample.
- Overhead is actually quite practical for a last mile network. 4k nodes generates 1.8mbps in overhead traffic per node. Can be tuned lower in exchange for worse worst case convergence and/or subnetted

# Billing

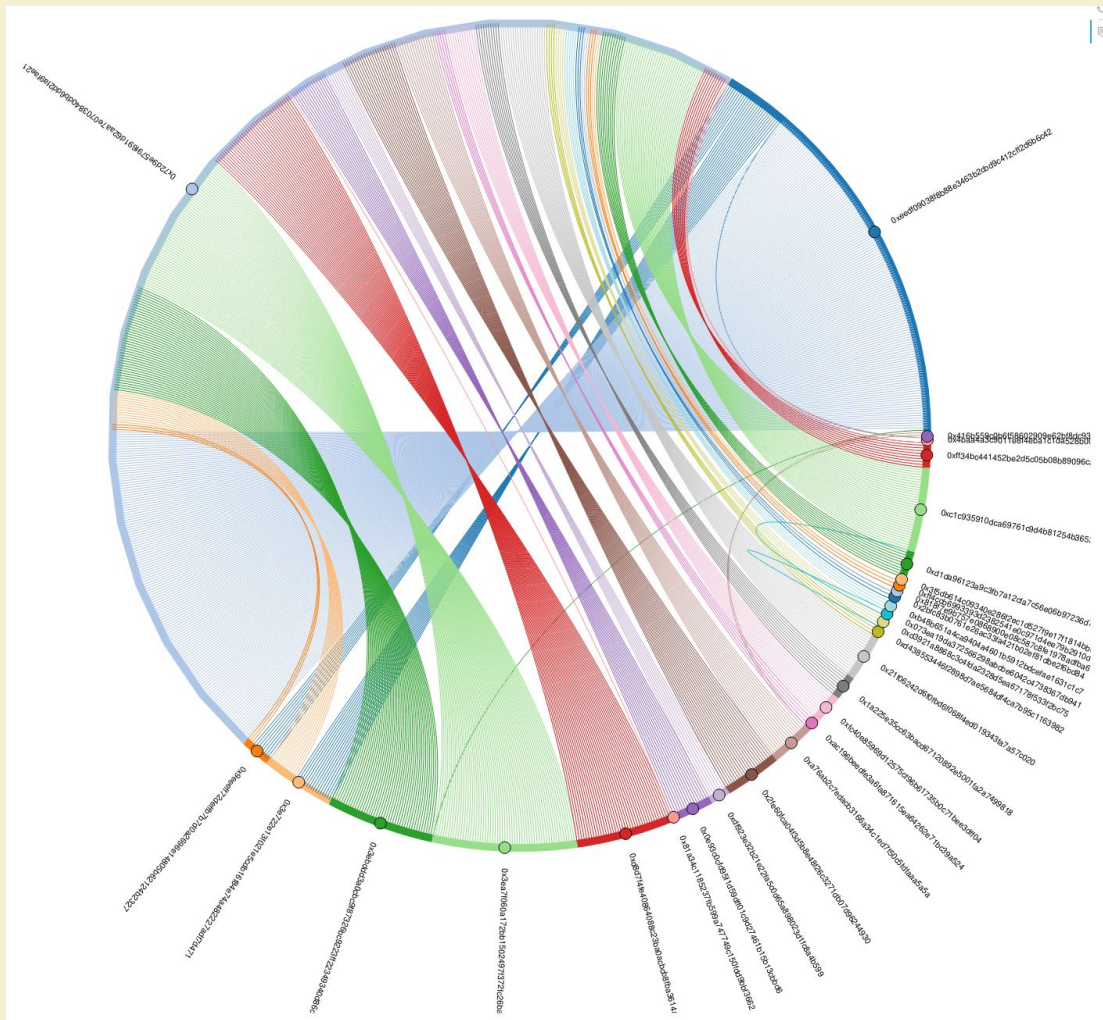
- Cryptocurrency based payments provide a nice way to avoid being a networking startup AND a financial startup
- The money is actually **inside** the router, this is both very convenient and troublesome.
- Depending on the internet to pay for internet results in a sometimes delicate dance of being both an application and a network protocol

# The Key insight, pay per forward

- We're trying to automatically sell and buy bandwidth, how do we term that sale such that it's possible for everyone to independently verify and agree that the exchange was completed?
- What we want is a subset of fault tolerant consensus. Each party needs to observe their own data and determine who to pay, how much, and how much they should expect to be paid. Nodes must agree on these values at all times despite possibly having different inputs due to packet loss

## The Key insight, pay per forward

- If we define that each node is paid per packet it forwards packet loss and other on-the-wire disruptions resolve to overpayment
- For example node A looks at its tx counter to determine how much to pay, node B looks at its rx counter to determine how much it expects to be paid. Packet loss means that  $tx \geq rx$  therefore B may see more payment than it expects but that still produces the consensus we need



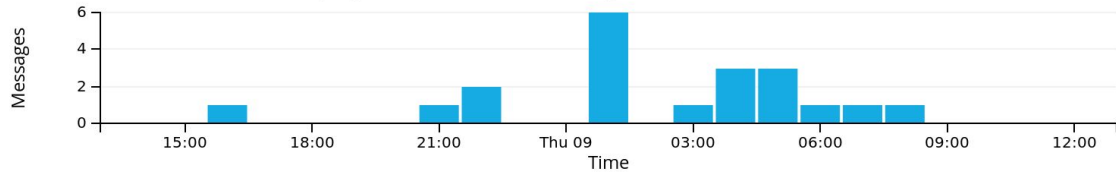
## Payment map

- Taken over 1 week in our production network
- Interactive at <https://bit.ly/2JCxenN>
- Pay per forward combined with the high ratio of download traffic means most payments are to or from only a few nodes
- ‘Exit’ nodes get paid for traffic they ‘upload’ to users, in an exception to the pay per forward model

## Billing - challenges

- Lightweight architecture portable cryptocurrency implementations
- Payments and payment verification must be incredibly resilient
- Fees amounts are variable
- Payment consensus needs to be resilient to execution problems. No ECC on random routers

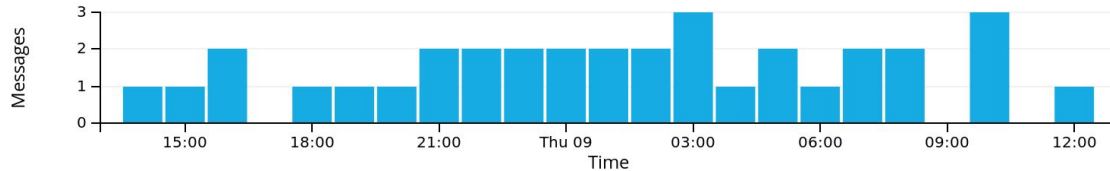
Portland exit bandwidth payment Failures (1 day)



7 minutes ago

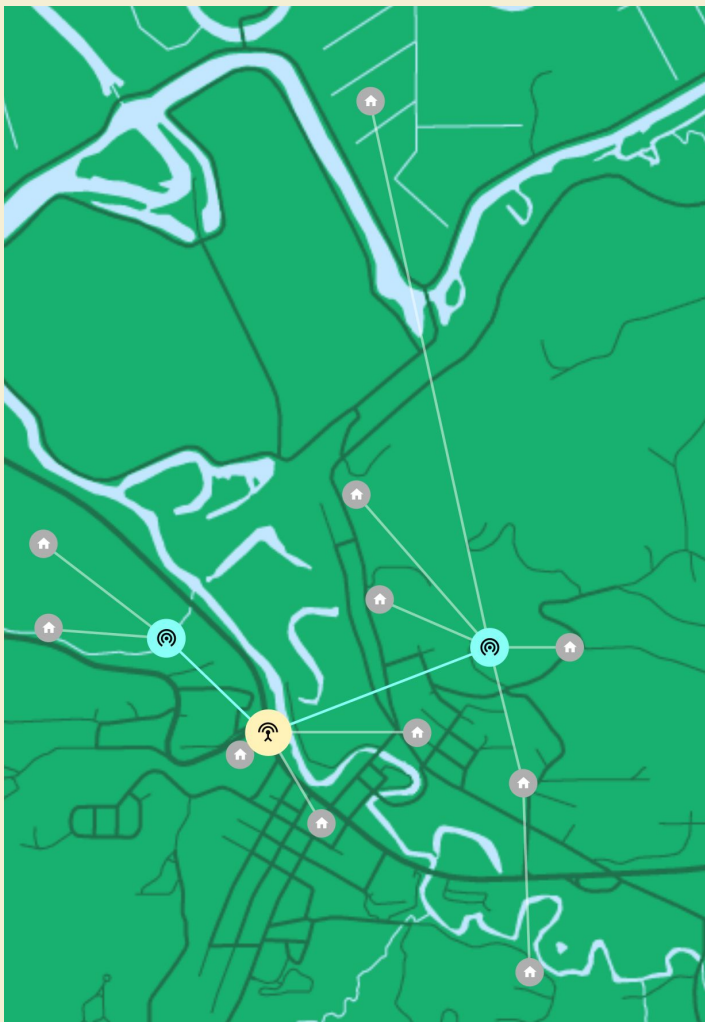


Successful bandwidth payments Portland Exit



2 minutes ago



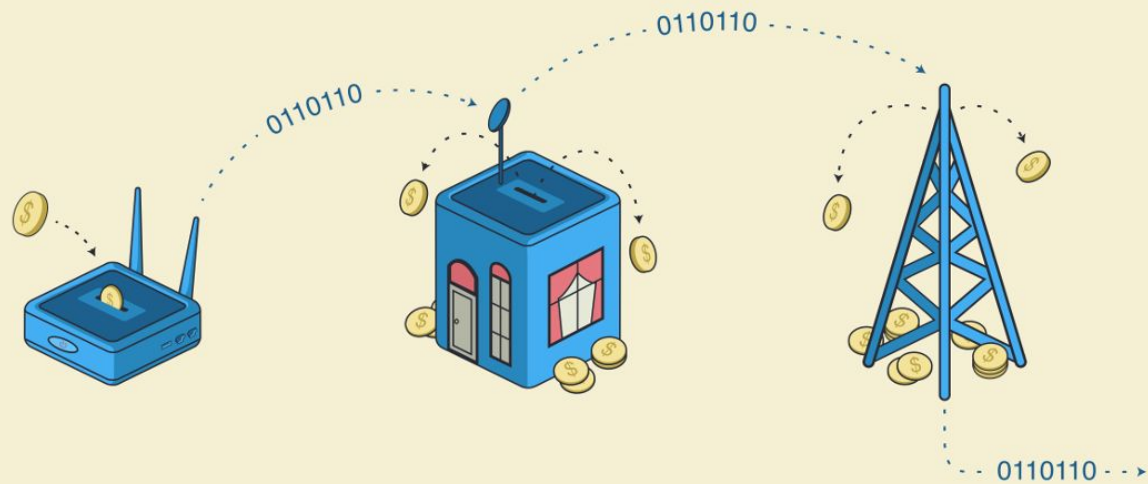


## Tales from the field

- Average user downtime  
15min / week
- Only one downtime  
event >1hr in the last  
year
- Speed / Latency  
100mbps / 10ms
- Patching, 0 failures for 4  
months
- User satisfaction  
Pretty good
- Pay per byte billing has  
upsides and downsides

## Particularly memorable incidents

- Long distance latency inversion combined with lack of TCP multipath
- Gardening = Downtime
- Web libraries are not designed to be used in unstable networks
- Time between billing consensus in the lab and in prod 5 months
- No time based properties are allowed in billing



Althea

---

# Thank you!

@AltheaOrg

justin@altheamesh.com