DRAFT PAPER 24.9.2017

work in progress

Korhal Networks

Interplanetary trustless message bus

In this paper we describe a decentralized message bus, in which a message can be delivered to any device identified by the hash of its public key. A crypto-currency incentivised, overlay network routes messages according to an adaptation of the Kademlia-DHT. We specifically solve direct access to IoT devices on mobile or customer networks without the need for an intermediate “cloud” for nat traversal and/or trust delegation.

Motivation and Ideology

Before even having unfolded, planet scale connectivity of IoT devices is already held back by “innovative” attempts to create lock-in and monopolization in 3 dimensions:

1. Friction to obtain internet connectivity.
2. Delegation of control to “clouds” owned and controlled by someone else
3. Centralization of data streams

Large scale Internet of Things deployment not only depends on ‘Things’ to become internet enabled, but conversely depends on Internet being available. The 3G M2M network is not lucrative for carriers and they’re trying hard to erode net neutrality in order to gain a lock-in mechanism they can exploit. A more realistic innovation friendly connectivity schema is WiFi. It’s wide spread, well understood and does not require a radio license to operate. Unless locked down eventually by the regulators, WiFi is also adaptable by enthusiasts. Free and open operating systems for WiFi-equipment are available and relatively mature.

~~WiFi is a true decentralized network, in the sense that there is no common authority. Every cell is its own isolated island with no easy way to connect to a cell operated by someone else, unless you negotiate terms with them through human contact. There are some larger organizations that have homogenous terms throughout their network, but they are indeed again all geared towards monopolization. However, one open and innovation friendly organization must not go unmentioned: German ‘Freifunk’ has a charitable approach, providing completely free access to the internet for cities. It does however rely on donations, grants and people spending their free time and money to build the network.~~

The second, more dangerous problem of IoT is the delegation of control, or more dramatically rightful owners losing control over their device. Usually, the owner may only transact with a proprietary cloud service, which then executes the owner' commands on their behalf without any boundaries. The cloud operators effectively possess the device providing no way for the actual owner to consent to executed commands. Vast effort and investment flows into creating “unified” cloud solutions which may then control devices multi-vendor. Whereas zero investment went into ensuring that the rightful owner of these devices obtains those capabilities for themselves.

Lastly, the most lucrative way to monopolize in IoT is in data streams. We’ve only just scratched the surface of what can happen when devices can exchange data and transact with each other, but already there are numerous startups claiming to “enable” value transaction between vendors. This cake is only so large - and the winner of that battle will certainly not be the actual owner of the device.

What we present in this paper is essentially a deconstruction of all of these 3 gatekeeper business models, and an alternative way of providing the same solution through a radically different, decentralized, monopolization-resistant, economic model.

The distributed hash router

Bitcoin has paved the way to understanding decentralized value consensus: While the technology behind bitcoin - the blockchain - is not new or particularly complicated, they have managed to show that a simple signature chain can be used to create solid consensus without requiring a central authority. The idea is now so widely accepted, that professional investors started trading BTC. With governments failing to regulate monopolies, decentralization now is the greatest instrument available to “seize the means of production”.

Korhal Networks contains something called “distributed hash router”. It serves the same function as the big internet exchange points we have today, but instead of being large hubs - like in Frankfurt, Amsterdam, New York - it is a virtual entity that exists only in shards distributed among all peers of the network. Similar to how the original internet was designed, anyone can become such a hub. However, in this approach, neutrality is *systematically guaranteed* by encryption. The middleman is taken out of the equation and replaced by proven mathematical concepts. Operating such a hub is also not out of goodwill, but actually incentivized by the system through reward tokens.

Resistance to control by a single entity is ensured by making transport fee transparent and requiring consense among the routing path. The winner of a traffic fee is determined by the best route to a destination. Identification of peers in a message exchange is pseudonymous through their public key. The network has no way to identify peers and discriminate by any aspect of the message other than its size and number of hops to the destination.

Large scale deployment of the system is driven by incentivization of builders through tokens they earn by routing messages. A difficult route to an island will cost more, incentivizing more relay builders to build routes, rather than incentivizing them to occupy already well covered spots and discriminate by content.

As a practical application, we expect first adopters to be WiFi hotspot providers willing to add a contract option to their WiFi access that is based on the Korhal Networks pricing consensus. For them, it opens up new opportunities to earn tokens from IoT devices which otherwise would have no way to connect to the internet with such homogenous contracts.

As enthusiasts and professional hotspot builders expand the network, intermediate hops will become feasible. Those are not directly connected to the internet, but earn tokens by transporting a message
to the nest internet hop, from a remote destination.

Effectively we'll build a mesh network over large areas with hundreds of thousands of subcontractors whose payout will be determined by the actual market price of the traffic rather than us keeping a cut.

Trusted Delegation beyond the cloud

However, all of this is just the groundworks to the grand finale of overcoming "cloud" centralization.
In order to counter the business model that is behind controlling devices that are owned by others, it is essential that we give those owners an alternative path to controlling devices over the internet - without the necessity for an intermediate 'cloud'.

Trust delegation is a combination of well known concepts from computer science. Broken down into its parts the functionality becomes more clear :

Any command happens through smart contracts, which cryptographically guarantee that the original owner of the device has consented to that execution. Through signature chaining similar to the x509 digital certificates, we can delegate the actual execution to anyone, even someone we don't know or trust.

Let's make an example using a smart lock. In "traditional" smart locks- put in quotes, because cloud enabled smart locks are pretty young actually - you would have 4 parties:

A lock, a control cloud, a building owner, a janitor.

Delegating authority over opening that lock to the janitor is easy: the owner just opens some webpage by the vendor. The cloud then ‘tells’ the lock to allow that janitor's key. Or if you’re unlucky and the system is really terribly engineered, it will actually just allow the janitor to send a command to the cloud, which opens the lock. Either way is incredibly scary, with the cloud having full authority over the lock, instead of the owner. It’s not even a good system if you would do a thought experiment and replace all actors in the system with humans: A bouncer, a club owner, a guest, and some random dude with a suitcase that the bouncer has to ask every time someone wants to enter. Usually what club owners do is a tad smarter: they give guests a token of their status that everyone involved understands.

In the case above, a proper solution is actually much simpler than you’d think.

Target	Public key of the entry door lock
Allow	Janitors public key
Transaction	Open
signed by	Owner's private key

A smart contract that the owner can create and hand to the guest as a file, not even having to directly talk with the bouncer. Or the owner of the building to the janitor, same thing.

The token / Proof of Transport

It is commonly accepted by now, that the blockchain technology is not usable for message transport because it’s core idea is that every node in the system is compute heavy. It must be compute heavy to be trustworthy, because trust is determined by proof of work. Etherium has the raiden network, and others like IOTA are attempting to take a different angle at value exchange on small devices. Korhal Networks is not about transacting value. It is about assigning value to transactions, more like IPFS does assign value to storage.

The fee of transport is defined by a market of how badly someone needs to transport a message versus how hard it is to transport it. If it’s easy to transport, lots of relays will be available and the price drops. If there is only one transport, that transport can determine the price. However, unlike in a traditional ISP business, if the relay demands extortionary prices, someone else will be incentivized to build another relay, even the sender.

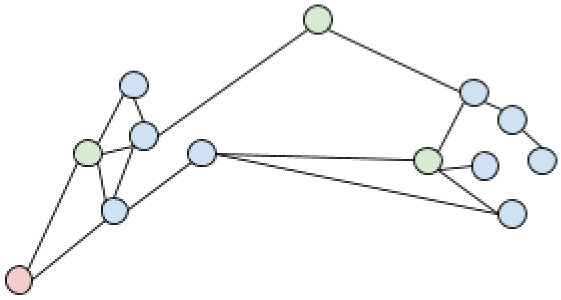
Value of the token in government issued money is determined on an open stock exchange trade. As a baseline, it should be roughly correlated to over demand of transports versus available relays. That means, if lots of people want to send messages, but the infrastructure doesn’t exist yet, the value of the token goes up. So people will build infrastructure to earn tokens until the demand and supply equalize.

Technical paper to be integrated begins here

Distributed Hash Router / Proof of Path

Message transport in the network is built over streams of messages, very similar to tcp. This allows negotiating a connection once and then efficiently transporting data without renegotiating. A sender/receiver pair may choose to create a new stream for each message for improved anonymity, but the sum of the initiation fees will be higher.

When opening a stream, the network determines the path to the destination via previously announced routing information that is stored in a Kademlia-DHT[1] with a twist. Because of the nature of allowing arbitrary physically transport to mesh up, some of our edge-nodes will only really have physical connections to a limited number (or even just one) of peer nodes. A node can choose to send masked contacts, which allow other nodes to route on their behalf.



Origin in red. Nodes it is required to have contacts in in green. Those are not directly reachable and will be forwarded to by nodes in the path.

.

```
Sending a Contact C
  C.id = my public key
  For each Kademlia subtree mask X not containing our id C.id
    If we have a peer P in subtree X
      Set C.mask to all 1’s
      Amend P.id to C.hops
      Sign Contact with my private key
```

[1] "Kademlia - Parallel and Distributed Operating Systems Group." Accessed September 20, 2017. https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf.

```
      Send Contact to P
    Else
      Pick the next closest Peer P to X
      Set C.mask to the rest that does not match
      Amend P.id to C.hops
      Sign Contact with my private key
      Send Contact to P

Receiving a Contact C {id, hops, mask}
  For each hop H
    Assert H was signed by previous H-1 or C.id
  If C.mask is all 1
    If C.id is already present in binary tree as C2
       If C.hops is shorter than C2.hops
         Replace
       Else
         Ignore
    Else
     Insert
  Else
    If we have a peer P in subtree C.mask
      Set C.mask to all 1's
      Amend P.id to C.hops
      Sign Contact with my private key
      Send Contact to P
    Else
      Pick the next closest Peer P to X
      Set C.mask to the rest that does not match
      Amend P.id to C.hops
      Sign Contact with my private key
      Send Contact to P
```

Proof of Delivery

The Korhal Networks protocol must regulate fee payment to relays in the system. When a sender A sends a message to B, using C as a relay, then A must pay a fee to C. 2 out of 3 parties might be low power systems not connected to the internet. They cannot verify payment of a fee by validating the integrity of a shared ledger like blockchain. Instead they must rely on the stochastic probability of incomplete information.

In an ideal world with everyone in the system well behaved, we would simply have A send a voucher along with the message that can be redeemed by anyone forwarding the message. C would then forward the message to B and collect the vouchers in a database to be redeemed later when it has internet. It might even use the network itself to forward the voucher to its owner, who then redeems it.

Now let’s imagine a completely bad world, in which nobody plays fair. A wants to send but not pay, B wants to receive but not pay, C wants to get paid but not do any work. These can be segmented into just two reward categories.

	A (sender)	B (receiver)	C (relay)
Exchange data	wants	wants	does not want
Exchange fee	does not want	does not want	wants

The asymmetric reward structure would make it impossible to operate if the majority of participants are ill-intended, which is extremely likely in a scenario with 3 participants, but extremely unlikely in a large scale community like bitcoin. All known proof systems fail if the verifier has no access to friendly nodes. This is why systems that simply use a directed acyclic graph are extremely weak in offline scenarios. Fortunately thanks to the mesh property of the network, actually all nodes sort of have access to the internet through other hops.

References