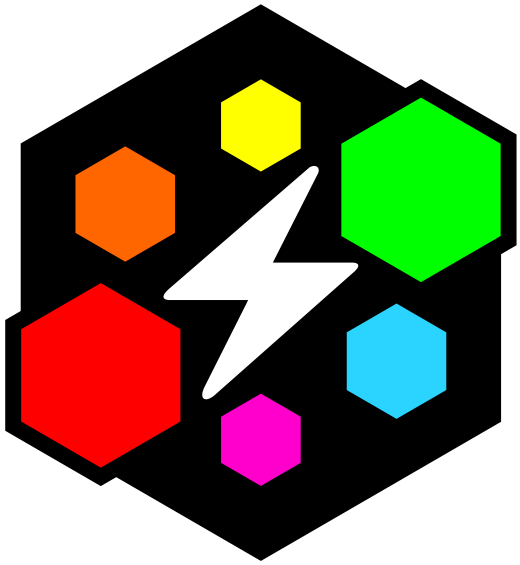


# Indra Routing Protocol White Paper

Onion routed distributed virtual private network protocol with anonymised payments to create scaling incentives.

David Vennik September 2022

## Abstract

The state of counter-surveillance technologies has remained largely unchanged in the 20 years since the inception of the Tor network.

The primary use case has always been obscuring the location information of users from clearnet sites, and the more it has been used for this purpose, the more hostile clearnet sites have become towards this network, due to its frequent use to launch attacks on web services.

With the increasing amounts of value being transported in data packets on the Internet since the appearance of the Bitcoin network, the need for eliminating the risks of geographical correlation between payments and user locations continues to rise.

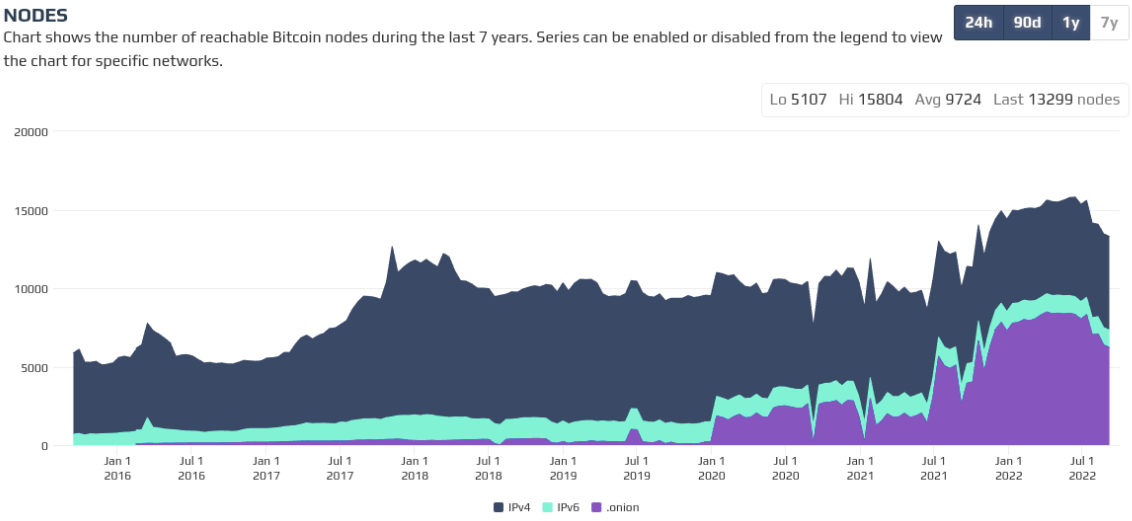
However, without any way for users to pay routers without creating an audit trail, the anonymising networks have not grown in nearly a decade, and thus well heeled attackers have largely been able to keep pace and pluck off high value targets, such as the Carnegie Mellon University

- implicated in part of what led to the arrest of the Silk Road founder, Ross Ulbricht.

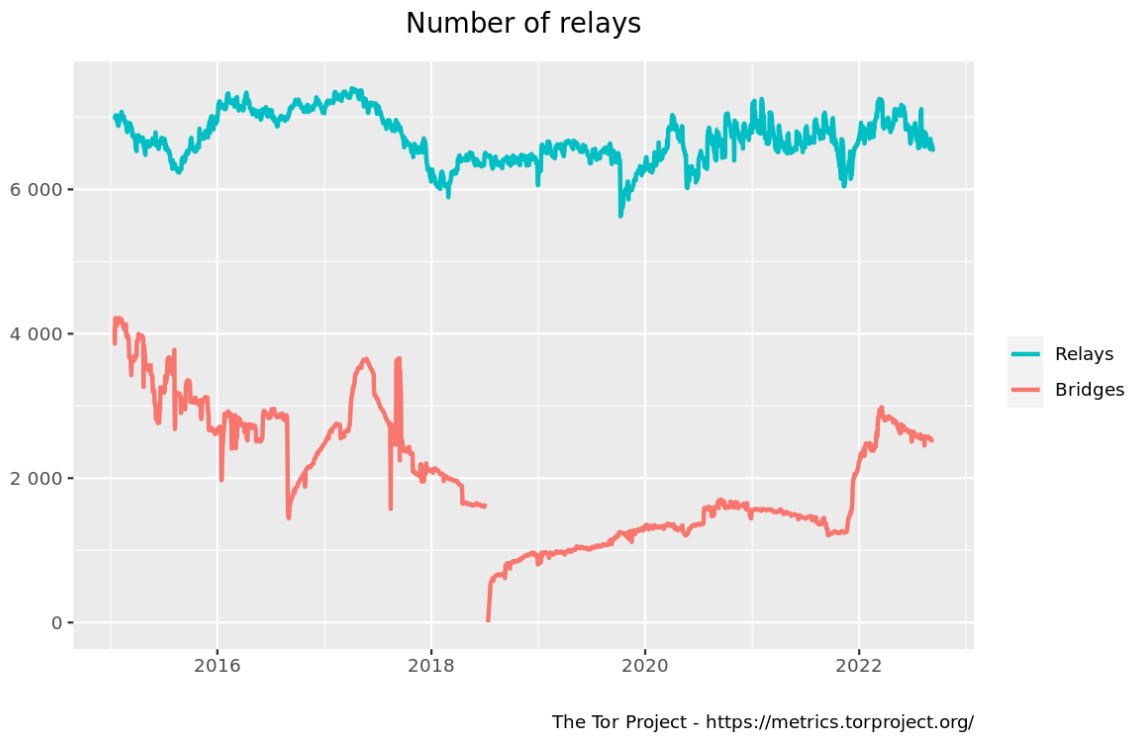
It is the central thesis of this paper to demonstrate how obfuscating correlation between payments and session usage can be achieved and create a marketplace in routing services which can economically increase to a size that is beyond the capabilities of a state sized actor to fund an attack, while also improving latency and stability of routed connections.

## Tor Isn't Scaling, But Bitcoin Needs Onion Routing

For comparison, this is Bitcoin's node count:




Versus Tor in a comparable period:




It is not hard to see:

- Tor is not growing, it has flatlined.
- Bitcoin is growing.

Not only that, you can also see that onion routing is forming an increasingly large component of Bitcoin connectivity.

# Goals of Indra Routing Protocol

Three key elements of the Tor protocol make it less than desirable in general.

1. Establishment of circuits is quite slow, taking a large number of steps to "telescope" into a circuit.
2. Once a circuit is running, when it fails, the failure is opaque to the client side, and there is no way to provide a latency guarantee or connection stability. It is unsuitable for interactive and long living connections.
3. There is no profit motive to drive expansion of router capacity.

Tor is a poor solution for a very limited subset of of the use cases that benefit from the security of route obfuscation. Indra aims to provide what Tor has definitely now failed to achieve for a large majority of internet users: location privacy.

# Anonymised Session Purchase Protocol

The initial purchase of a routing session presents a chicken and egg situation. Without an existing circuit, how does a client node acquire tokens for sessions with onion routers to use for creating hops in an onion route?

For this, we borrow from Lightning Network protocol BOLT#4 which we cannot use by itself as it has no provision for returning an arbitrary package of data, nor the notion of an interacting midpoint in the loop, with the path going back to the sender.

As distinct from this `lightning-onion` protocol, we use ed25519 for signatures and curve25519 for the ECDH key exchange, and encrypt traffic with AES-GCM. Rolling over the cipher is done via hash chains on sequential packets, which constitute the unit of transmission. As noted later, this is 8 Kb, a moderate size but not too small to accommodate a substantial message with no return packets embedded.

## Session Tokens

Session tokens are an arbitrary random value that must be present in the header of encrypted data and is hashed in a chain to provide a counter for the packet sequence within a session, which are combined with the public key of the router to generate the cipher for a packet. This cipher changes as the session progresses by hashing the previous value matching the accounting of the packets in the session.

## Exit Sessions and Charging

When packets are delivered at the exit point, the node then encrypts the return message with the provided cipher which is carried back according to the instructions of the next hop, back through two nodes before reaching the client.

The return message indicates the number of packets that have been consumed by the session. These should be congruent with the exit node's advertised exit data rate or scheme.

### Protocol Specifics

In order to prevent spam, different protocols have different rules. For Bitcoin, for example, the transaction must be valid, and the fee must be within 50% variance of the current fee rate in the mempool, or similar specifications. The router can charge equal fee to the embedded fee in the transaction, or a factor, as the privacy premium. This is then computed against the main data rate by a factor.

To enable this, there may need to be a simple specification for complex requirements that relate to the relayed messages in the exit protocol. Usually a Bitcoin transaction will fit into one packet comfortably, so the immediate return message will signify the satoshis charged as a proportion of the fee rate for the advertise bandwidth.

Similarly, for Lightning, there can be message related fee rate calculations, for lightning, more simple, the router can simply state a fee rate in the same way as clearnet routers. This will be charged to the channel operators, not the carried transactions, so the operators will add this privacy premium to their routing fees on the connection path in addition to their routing fee.

## Path Hop Acknowledgements

In order to ensure the session purchase protocol is properly executed, in each layer of the onion there is special acknowledgement onion messages carrying the payment receipt that confirms delivery, and route backwards to the buyer, who knows then that a hop has succeded.

In this way the buyer can expect 5 acknowledgements to be successful and then receive the session key from the 5th node in the circuit.

The acknowledgement onions are constructed so that nodes do not know what step they are, so each has space for 5 steps, which are masked using methods as described in BOLT#4.

If the 5 acknowledgements are not received within a reasonable time, the buyer then propagates forwards payment reversals up to the last acknowledged payment, reversing the payment and denying the misbehaving node where the route stopped.

### Session Purchase Fees

These are not charged as relay packets, the cost is amortised with the per hop fee paid to propagate the payment forward. Each hop gets a fee, and the fee paid is dictated by the node operator, and is a threshold parameter for the buyer. Nodes will not be chosen to do the purchase if they advertise purchase fees above this threshold. This can be considered to be a question of anonymity set size, rather than a concrete threshold, a proportion of offered rates, representing the relative proportion of nodes that could be selected and process the payment.

# Source Routing

As distinct from TOR, Indra uses source routing, thanks to the magic of the session tokens and ECDH, means that in the event of a route path failing, a new path can be generated with a changed set of intermediate routers when a timeout occurs.

## Latency Guarantee and Path Timeout Diagnostics

For time sensitive interactive applications, these progress detecting onions be used at every step to ensure the moment one hop latency exceeds a prescribed threshold the source routing algorithm can then swap out a different node in the route and thus provide a strong latency guarantee.

Some applications are very time sensitive. Real-time interactive shared environments such as games can have very serious consequences (to the players) when their connection starts to increase in latency putting them at disadvantage against their opponents, and in general, a sluggishness of the interactivity.

Long lived sessions like SSH also can become tiresome when running over Tor when inevitably congestion or downtime hits a hop in the path. For applications where a few seconds stall do not disrupt the protocol, activating path timeout diagnostic onion acknowledgement packets enables the client to determine which hop needs to be replaced.

For additional security, a user can configure the return onions to return via random, multi hop paths, rather than reversing the forward path.

### Acknowledgement Charging

These acknowledgements are essentially reverse onion paths like the forward paths, but only 3 hops each, and with a small payload. They are essentially charged one packet each, per hop, to maintain the flat packet size and eliminate packet size fingerprinting. The client thus can consume as much as 5 packets per forward packet in charges for a path traced transmission, as well as the fees on the exit, for path tracing or high reliability guarantee transmission.

## Circular Paths

One of the key inventions of Indra is the notion of circular paths. These are two hops out to the exit/destination point, and two hops backwards, on a different path, for the return.

By using this circular topology, the source can provide a return path that is not the same as the forward path, and when the path timeout diagnostics are not in play, there is no visible reverse path confirmation timing for large scale network traffic analysis.

That is to say, the packets appear to always only be going forward, and no correlation is easily made between, therefore, forward, or reverse paths, which is not the case with telescoped TOR protocol packets, and for most general purposes in Indra are avoided when traffic achieves the intended forward path without excessive latency.

The return paths also serve as a path to return a new cipher for an exit node, whose messages are carried with the return loop, as well as bandwidth accounting data, via provided ciphers for the return path payload.

Return paths are also used by the exit node to return a value indicating. the number of packets charged out of the exit session (which includes at least one for their return relay)

## Packet Sizes

All packets are 8 Kb in size. Acknowledgement onions increase overhead and reduce data that can be carried. Routers charge for the total payload. By making packets uniform, it is a simple matter of counting packets, and thus the hash-chain counters directly relate to total routed traffic volume.

## Liveness

Because of the circular path and the reverse path after the middle carrying return messages, communication must always be prompted by the client in order for ongoing return messages to occur.

This has a bearing on bandwidth charges, in that each onion sent out will deduct session bandwidth from 5 separate nodes. Bandwidth is really quite cheap, in general, so, it is not really onerous. A person offering relay services on their node probably ends up earning as much as they use, depending on how much bandwidth they proscribe for their own use outside of the service, and depending on the charge regime of the provider (if it is metered or flat charge per time).

Indra uses UDP, eliminating any session overhead, the path acknowledgements and return paths are already defined in the onion packets. TCP and QUIC can be carried across the circuits, as will often be the case for hidden services. Indra aware services can take advantage of the knowledge of the protocol and trigger features of the protocol such as return acknowledgements and the return hops to carry things like acknowledgements of packets received via their hash fingerprint.

There is no need for negotiating connections, data is simply forwarded around on the basis of pre-agreed contracts of service created by the purchase of data sessions, and authenticated by valid headers, which prove relation to the session root code.

This messaging strategy does require a constant request/reply pattern, but a node does not need to send a second request unless the response does not come back within an expected time window, or is expected to have some amount of delay, because the return path is already plotted, and the cipher provided to the exit hop in the circuit.

For some applications, this is fine, such as a terminal session, as while the user is not asking for anything, the listener does not either have to wait for anything. For this type of traffic there can be pings, which are short packets and do not need path diagnostics, so they are cheap for monitoring liveness.

In addition, nodes monitor the state of other nodes in the clear when gossiping status updates and advertisements, and if the (uncharged) traffic of asking for status updates from peers reveals a node is unresponsive this is a back channel that can be used to trigger a circuit path change to route around a dead router.

# Funding model for Indra developers

One of the problems with open source projects is that it can be difficult to find sufficient funds to pay for the maintenance of the software. In order to provide an income stream that can fund this development, it becomes possible that a set of nodes are designated in the delivered binaries that can be required in a 3rd hop in purchases to charge a proportion, something like 1%, maybe less if traffic volumes are very large.

In this way, these funds can be accumulated and used to maintain the software and improve it.

**End**