ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Advanced Network Architectures (M132)

Assignment 5

CJDNS. The secure alternative routing solution

Name:	**Evangelos Siatiras**
e-mail:	**EN2190001@di.uoa.gr**

The Emergence of CJDNS

The internet is neither neutral nor private. As it is referred in this article [1] the US National Security Agency can reportedly collect nearly everything a user does on the net, while internet service providers (ISPs) move traffic according to business agreements, rather than what is best for its customers. So some people have decided to take matters into their own hands, and are building their own net from scratch. Work is underway to construct user-owned wireless networks that will permit secure communication without surveillance or any centralised organisation. They are known as meshnets. Each node in the mesh, consisting of a radio transceiver and a computer, relays messages from other parts of the network. If the data can't be passed by one route, the meshnet finds an alternative way through to its destination. An idea would be a decentralised network that relies on encryption by default.

"Privacy advocates envision a decentralised network that relies on encryption by default"

Encryption is the starting point. Computer researcher Caleb James DeLisle wrote software called CJDNS which allows the Meshnet nodes to use Hyperboria and keep all communications between them encrypted. Instead of letting other computers connect to you through a shared IP address which anyone can use, CJDNS only lets computers talk to one other after they have verified each other cryptographically. That means there is no way anyone can be intercepting your traffic.

Principles of operation

From a technical perspective described in [2] CJDNS is a secure and decentralized multilayer routing protocol which aims to close the loopholes in the protocols designed over thirty years ago and which constitute the fundamental architecture of the modern Internet. CJDNS seeks in particular to effectively solve the problem of the size of routing tables by judicious use of decentralized hashing tables. Its architecture allows among other things the relay of packets without memory reading thanks to headers containing the path to the next router. Routing is done in successive stages which aim to approach the destination using a restricted routing table obtained by "search" type requests from known neighbours. The protocol also offers complete security thanks to an encryption mechanism incorporated into the protocol. The authentication is based on SHA-256. This protocol is reactive, but its decentralized hashing model greatly simplifies the search for the destination. On the other hand, the additional security layer somewhat reduces its performance. It should also be noted that CJDNS is capable of operating via a tunnel and that the addresses it generates do not conflict with the traditional internet since they are located in the block. It is important to note that this protocol does not include a process for discovering new nodes, which means that a new node must manually bind to the network through a peer.

Hyperboria is a virtual meshnet because it runs through the existing internet but is purely peer-to-peer. This means people who use it exchange information with others directly over a completely encrypted connection, with nothing readable by any centralized servers. When physical meshnet nodes are set up, existing Hyperboria connections can simply be routed through them. Now, Hyperboria offers a blogging platform, email services, and even forums similar to reddit. Some form of encryption is already in use across much of the internet, but to be useful it has to be ubiquitous. Web services like Gmail, for example, let you log in using an encrypted connection. But when you send an email it leaves Google's encrypted garden and hits the open web in clear text for anyone to read. With Hyperphoria's peer-to-peer connections, every single link in the chain of communication is fully encrypted. Intermediaries that handle traffic cannot even see what kind of traffic it is, let alone read any email. Use the purpose-built hyperboria.name email service, and your communication becomes untraceable.

Decentralize with CJDNS in Practise

A practical implementation [3] and widely verified for its stability is the so called "IPv6 Overlay Mesh VPN" where every device is assigned a private IP and translated to the public IP by the router. Without port forwarding to a specific private IP, incoming TCP connections or UDP sessions can't tell where to forward to and are dropped. As a result, nothing can connect to a specific device. Every client must connect to various public servers to do anything. IPV4 NAT Jail forces centralization. The simplest solution to this problem is IPv6. The CJDNS package (included in Fedora) implements a global IPv6

mesh by connecting to several peers instead of a centralized server. Each node has a public/private key pair. The IPv6 is the truncated SHA512 hash of the public key, preventing spoofing.

- Packets are end to end encrypted — relays can be untrusted.
- Packets are source routed, allowing seamless upgrades of and experimentation with routing algorithms. (This is safe thanks to anti-spoofing.)
- The data for routing comes from a Distributed Hash Table listing the peers of each node.
- Peers can be explicitly configured as UDP tunnels, or auto configured on ethernet via layer 2 protocol 0xfc00.

With CJDNS installed, IPv6 address is stable and “unspoofable” (standard cryptographic caveats apply), can be used with any IPv6 ready application. The recipient must also use the Cjdns protocol, but this isn’t much of an obstacle since it’s easier to install CJDNS than convince US ISPs to provide usable IPv6.

After the establishment of the connection the client can ping any node in the global IPv6 mesh. Attention is needed as all those nodes can now directly connected by each other. The default Fedora firewall will block all incoming connections by default, but specific and careful configuration is needed for the allowed traffic.

Using CJDNS for the voice calls gives you privacy and authentication. You can use any sip client that supports IPV6. An example is the linphone app included in Fedora.

Because CJDNS is end-to-end encrypted, there can be no CJDNS “gateway”. However, CJDNS has a built-in IPTunnel, so that your router could partner with another CJDNS node to route conventional IPv4 and IPv6 traffic to/from your home network. This is functionally no different than using something like OpenVPN for the same purpose (but is harder for external firewalls to block). There are, in fact, commercial services providing VPN service via CJDNS.

Momentary advantages and Goals for the Future

CJDNS is essentially a TCP/IP replacement protocol with totally decentralized addressing and mesh routing, and with built in end-to-end encryption. It's intended for creating mesh networks to effectively replace the internet, or at least the "last-mile" portion of it (ISPs).[4]

Currently, direct CJDNS links between nodes are quite rare, though, and most of the links on the Hyperboria network (the mesh network established for testing and using CJDNS) are actually over VPN links on the internet. This is a temporary "bootstrap" situation which it is hoped will be eventually replaced as much as is possible by true independent mesh networks with direct, probably primarily wireless, links. Many foresee a future where one first connects to their local community mesh network, then for access to the commercial internet (if needed) one would purchase gateway access from a local provider that they connect to over the mesh.

It should be noted that the VPN connections are not a fundamental or necessary part of CJDNS (and in fact are hoped to be completely unnecessary someday as mentioned), and that although VPN functionality is included with the CJDNS software, that is really just a practical consideration due to that being the primary method of connection during the early stages.

There is indeed some work being done to produce a version of CJDNS for Android, however I believe that the Windows and OpenWRT versions are considered higher priorities.

Instead of a few established players building network infrastructure, DeLisle wants anyone to be able to do it. For him, decentralized internet access in the hands of the people is just a start. The services they use must be decentralized, too. “If people continue to use Facebook, they will continue to be spied on, that’s just the reality of the world.”

References

[1] H. Hodson, “Meshnet activists rebuilding the internet from scratch,” 07 08 2013. [Online]. Available: https://www.newscientist.com/article/mg21929294-500-meshnet-activists-rebuilding-the-internet-from-scratch/.

[2] “Documentation relative au développement d'une solution de monitoring,” [Online]. Available: https://wiki.reseaulibre.ca/documentation/monitoring/documentationPertinente/.

[3] S. D. Gathman, “Decentralize common Fedora apps with Cjdns,” 20 08 2018. [Online]. Available: https://fedoramagazine.org/decentralize-common-fedora-apps-cjdns/.

[4] OmixronNine, “Reddit,” [Online]. Available: https://www.reddit.com/r/cjdns/comments/1n8lox/questions_of_a_beginner/.