OpenWave

A peer-to-peer distributed-network routing system.

Document v0.7.0
Revision 20180310

Jordan Mack
jordan.mack@openwave.io
www.openwave.io

Abstract

OpenWave is a peer-to-peer distributed-network routing system that is incentivized, transport agnostic, censorship resistant, fully encrypted, and transparently backwards compatible with existing networks and software without modification. OpenWave allows users to participate in an open-market mesh network where each running node is rewarded for providing network data transport to other users and devices. By directly incentivizing the free flow of information, OpenWave will extend beyond all borders, regardless of the transmission medium or content.

OpenWave is free from control or ownership by any group or government. All software and hardware will remain open source. All research will be released publically without patents or royalties.

Table of Contents

Prerequisite Knowledge

In order to understand this document, one should have the following knowledge:

- Familiarity with WiFi wireless networking technology.[1]
- Basic understanding of cryptocurrency.[2]
- Basic understanding of smart contracts.[3]
- Basic understanding of a cryptographic hash function.[4]
- Basic understanding of symmetric key cryptography.[5]
- Basic understanding of public key cryptography.[6]
- Familiarity with existing mesh network solutions. (Recommended.)

Overview

OpenWave is a distributed peer-to-peer “decentralized internet” project that is designed to include privacy features and censorship resistance. The end goal is to create a self-sustaining system where users can participate in a marketplace to buy or sell global network access. OpenWave is designed to run beside the existing internet, or on top of it. A project of this scale and scope has numerous challenges and hurdles that must be overcome. OpenWave takes a unique approach to tackling these challenges and has many key features at its core that differentiate it from other projects with similar goals.

OpenWave is incentivized from start to finish. We believe that there is no such thing as a free lunch, and projects that rely on “good will” participation will ultimately fail. Careful consideration has been taken to ensure that every aspect of the project has a basis for incentivization in a market-based economy.

OpenWave is designed to be transport agnostic. The incentive system in OpenWave rewards participants for moving data from “point A” to “point B.” It does not dictate the medium or underlying protocols used to accomplish the task. As technology continues to improve, OpenWave will evolve to adopt the best available solutions.

OpenWave is censorship resistant and fully encrypted. All information flowing through the OpenWave network is transported with the minimal amount of knowledge required for a successful network. Every piece of user data is encrypted at the protocol level.

OpenWave is transparently backwards compatible. A network solution functions best when it encompases as many existing use cases as possible. OpenWave is designed to work with all major operating systems and allow all network enabled applications to continue functioning without modification through a backwards compatibility layer.

[1] (Wikipedia n.d.)
[2] (Wikipedia n.d.)
[3] (Wikipedia n.d.)
[4] (Wikipedia n.d.)
[5] (Wikipedia n.d.)
[6] (Wikipedia n.d.)

OpenWave is self-sustaining, decentralized, and free from centralized ownership. OpenWave is not owned by any single group or entity. All software is free, open source, and available for reuse without limitation. OpenWave releases all research freely and will never attempt to copyright or patent any part of the project.

Approach

In each industry there are often multiple market leaders. Each leader may command a specific niche, demographic, or geographic region. It is very possible that multiple solutions will become leaders worldwide in the “decentralized internet” space. Once these solutions gain a foothold, market penetration becomes more difficult. Rather than directly competing to be the single market leader, OpenWave aims to be a solution that works synergistically with existing and emerging network solutions and technologies.

One of the biggest challenges with emerging blockchain technology is the lack of user friendliness. A positive user experience is a key aspect for the success of any project, especially a project of this nature. Participation is needed from both technical and non-technical users. OpenWave will focus on providing plug-and-play solutions that “just work” right out of the box. Advanced users will have access to a rich set of configuration options, but even a non-technical user should be able to get up and running in five minutes or less.

Motivation

The motivation for OpenWave is multifaceted. The centralized structure of the internet has created many problems that continue to grow as the internet becomes more and more intertwined with our everyday lives. Many of these issues can be solved through decentralization using a peer-to-peer networking solution. OpenWave aims to solve these issues by creating an open platform where users are directly incentivized to provide services that are relevant today and proving to be more relevant each passing day.

- **Network Decentralization:** The backbones of the internet are controlled by a handful of large corporations.[7] This level of centralization leads to vulnerability to distributed attacks and widespread outages when problems occur. Decentralization will increase the resiliency of the internet by automatically providing alternative routes that mitigate damage to the total system. Even though individual nodes would be susceptible to attacks, the entire network would remain functional since there would be no single point of failure.
- **Privacy:** All information that flows over the internet is recorded and illegally spied on by governments around the world.[8] In some cases the ISPs sell the data, and in other cases the backbone fiber optic cables are covertly, illegally tapped. OpenWave limits the ability for this activity by sending the data directly through peers, rather than through a centralized backbone. OpenWave requires complete encryption of all data, and transparently adds encryption to applications that do not support encryption. OpenWave goes one step beyond encryption,

[7] (Dyn Blog n.d.)
[8] (Wikipedia n.d.)

seamlessly supporting advanced privacy routing features at the packet level, making data flow through the network in a seemingly random manner.

- **Censorship Resistance:** Censorship of information and ideas by corporations and governments has become an all too common occurrence. OpenWave has no central authority, and no single point of control, thus it is resistant to censorship. Information flows directly from node to node, and if any node chooses to censor, alternative routes are mapped automatically.
- **Global Roaming Network Access:** Those who travel regularly are familiar with the repetitive process of locating a reliable internet connection wherever they roam. OpenWave will address this need by creating a unified platform that allows the user to connect from any location an OpenWave node exists with lower costs, based on actual usage rather than set fees.
- **Backup Network Access:** Internet connection problems occur frequently. ISPs regularly have both scheduled maintenance and unexpected outages. OpenWave can function as a secondary connection in the case of an internet outage. This provides an ideal solution, as there are no monthly fees. Users will only incur negligible costs when they need to use the OpenWave network.
- **Last Mile Network Access:** Many areas of the world still do not have internet access, and the infrastructure does not exist to bring it to them.[9] OpenWave can extend the "last mile," providing these regions with lower-cost solutions than can be provided by a traditional ISP.
- **IoT Internet Connectivity:** Millions of low power IoT devices will require internet access. Those deployed in the field will be forced to rely on low bandwidth cellular connections (LTE Cat M1). OpenWave can address this same need by lowering the costs of the required hardware, and lowering the cost of network access.

OpenWave Components

- **OpenWave Open Source Software**
    - **OpenWave Client:** A lightweight client enabling the user to connect to the OpenWave Network.
    - **OpenWave Core:** The primary server software suite implementing the OpenWave Protocol.
    - **OpenWave Directory:** A distributed database server that implements the OpenWave Directory Protocol. This server holds transient metadata about the OpenWave network, which is required for the network to operate.
- **OpenWave Hardware**
    - **OpenWave Server:** A low-cost, plug-and-play node solution designed for end users. Designed specifically to work out of the box, a user can be connected in five minutes or less.
    - **OpenWave Field Server:** A mountable lockbox with an optional solar panel and embedded battery and a low cost IoT OpenWave Server. This product is designed for mass deployment by field techs for large scale, high coverage installations.
- **OpenWave Token (OW)**
    - The native token of the OpenWave network, which is used for all payments and rewards.
    - Users will be able to purchase OW tokens on standard exchanges, or instantly exchange select other cryptocurrencies for OW tokens using a built-in, instant exchange.

[9] (Wikipedia n.d.)

- Users will be able to purchase OW tokens directly through the end-user software at a competitive rate, using standard issue credit and debit cards.

Types of Nodes

The OpenWave network is comprised of multiple node types. Each node type is not mutually exclusive. Depending on the desired use case, a node may comprise a single node type, or multiple node types at once. (Note: In this revision of the document a "node" is used interchangeably to mean a physical server or an application function. This will be clarified in a later revision.)

- **Client Node:** An end user or device requiring network access. In its most simple form, this can be a user that connects to an access node via WiFi. No additional software is required. All configuration will be handled through a browser splash page, similar to how a hotel or coffee house offers free WiFi access. However, this will provide only the most basic usage. Additional functionality will be gained by use of a browser plugin, and/or software, which can be installed on the computer of the end user. This functionality includes configuration of which network the client connects to, how the connection operates, how user data is encrypted, and manages the user's OW tokens automatically.
- **Access Node:** A node that facilitates the actions of a Client Node, but runs on the server of the operator. This is provided as a convenience to onboard users to the system more easily, but requires that the user trust the operator. A common use case for this would be an OpenWave node operator accessing his own node from a device that cannot run the client software. This node type would most frequently coexist on a Relay Node or Gateway Node.
- **Relay Node:** A node that facilitates the transport of data to other nodes. Relay Nodes require a fee to be paid by the user in order to transmit data and will accumulate earnings over time. Relay Nodes do not require an internet connection, but must be able to reach a Gateway Node with internet access or Payment Node in order to properly receive payments.
- **Gateway Node:** A node that facilitates dataflow to and from specific networks. The most common network destination will likely be the Internet itself. Other networks can be bridged in seamlessly, allowing the user to access them without any additional software. These networks may include Tor[10], IPFS[11], Swarm[12], and any other distributed or mesh-based networks. Gateway Nodes will require a fee similar to Relay Nodes.
- **Payment Node:** A payment node is a special kind of Gateway Node, which only allows the transmission of payment data. The purpose of this is to allow low-power IoT devices with low-bandwidth cellular connections (LTE Cat M1) to participate in the network in a beneficial way, even though they lack the ability to transmit large amounts of data. In more remote regions, many Relay Node hops may be required to access a Gateway Node. A Payment Node helps to more quickly facilitate payments on the network by reducing the amount of hops required. Payment Nodes will require a fee similar to Gateway Nodes.
- **Directory Node:** A node stores and transmits distributed database data required for the OpenWave network to operate. This includes the node mapping and name services. Directory Nodes will commonly be paired with all of the above-listed node types.

[10] (The Tor Project n.d.)
[11] (Labs n.d.)
[12] (ethersphere n.d.)

- **Exit Node**: A node which exists for the sole purpose of providing a final "exit" point to the internet. An Exit Node provides legal shielding to Gateway Nodes by mixing user data together over a single IP address, before being transmitted to their internet destination. An Exit Node would require a high bandwidth connection, making a data center the most viable location. The concept of an Exit Node was inspired directly by the Althea White Paper.[13] It is a project goal to build in compatibility with Althea Exit Nodes directly into Gateway Nodes, rather than developing this feature in parallel.

Network Overview

OpenWave is a peer-to-peer network with a distributed mesh topology. It is designed with a flexible architecture that isn't tied to any single underlying technology. This will allow the platform to evolve to adopt new technologies as they emerge in the future.

OpenWave is designed to be transport agnostic. This means that it does not matter how the data gets from "point A" to "point B," as long as it conforms to the OpenWave Protocol. This could include exotic radio, laser or satellite transmission. However, consumer wireless technologies will be encouraged as the primary method of transport. Their wide availability and relatively low costs are essential to maintaining a low barrier to entry for everyday users.

OpenWave nodes form a mesh network by connecting to each other utilizing an ad-hoc WiFi connection, or by utilizing a VPN connection. WiFi connections are established automatically with nearby OpenWave nodes. VPN connections are set up by two cooperating node operators to establish a bridge over longer distances.

The OpenWave network will consist primarily of two distinct building blocks, the OpenWave Protocol, and the OpenWave Directory Protocol. These two protocols operate symbiotically to facilitate operation of the OpenWave network.

Payment Overview

OpenWave uses a token-based system to incentivize nodes and allow users and devices to pay for the network resources they consume. All payments are made utilizing an escrow system through cryptocurrency smart contracts. This system is completely automated, allowing resolution without any form of human intervention.

Payments are facilitated by use of the OpenWave Token (OW). This token will be based on ERC20[14], or a similar token standard, which in turn will utilize an existing smart contract platform. All current smart contract platforms currently require an internet connection, and therefore some internet connectivity will be (indirectly) required to properly facilitate payments.

[13] (Tremback and Kilpatrick n.d.)

[14] (Wikipedia n.d.)

OpenWave Tokens will be purchasable through the client software interface for convenience purposes. Tokens will be traded on major cryptocurrency exchanges, and will be exchangeable for other coins and tokens, such as Bitcoin, or for traditional fiat currency, such as USD.

The OpenWave payment model utilizes a type of batched escrow payments to reduce the amount of internet traffic required to verify payments. A single escrow contract is used to pay for a large chunk of data usage at a single time. As the user continues to utilize data, chunks of the escrow amount are released by the user to the provider. An internet transaction must be utilized to open and close the escrow contract, but the release of individual chunks are directly peer to peer, and do not require an internet transaction. If the full amount of data is not utilized, the remaining balance is refunded automatically after by the smart contract after a specified amount of time.

OpenWave Tokens are deflationary by nature. All tokens are brought into existence during the token contract genesis and are not created through inflationary mining. Nodes that successfully operate generate revenue exclusively through from direct payments other users. Node operators receive 100% of the payments, and OpenWave does not tax these payments in any way.

Hardware Overview

OpenWave will produce hardware with a focus on ease of use and seamless adoption. By supporting only a limited set of hardware, more development time will be able to be dedicated to forward development rather than fixing compatibility issues. This approach will also reduce the scope and volume of support requests and will ultimately lead to better user experience.

All hardware will remain open source. Power users are free to build their own devices and install OpenWave software, or they can save themselves time by simply purchasing a dedicated device. More casual users will benefit by having a greatly simplified experience. Even a novice technical user will be able to get up and running in five minutes or less.

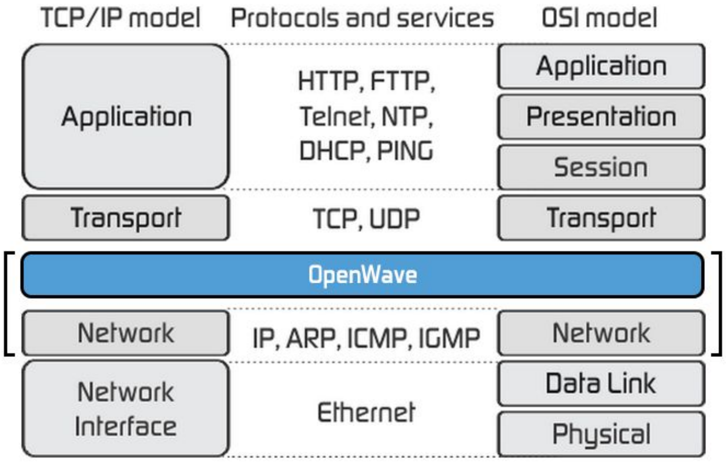
The basic node hardware will be comprised of a low-cost computer (such as a Raspberry Pi[15]), a WiFi adapter, an Ethernet adapter, a high gain antenna, a miniature LCD screen, and an attractive custom case to house the assembly. This basic kit will be pre-configured and ready to use out of the box by home users.

Alternate versions of the node hardware will be developed for specific use cases, such as outdoor or industrial and commercial deployment. The majority of internal electrical components and software will remain the same, but the external housing assembly will change depending on the intended use case. For example, a housing designed for an outdoor setting will include a tamper resistant design and will be designed to be mounted to an external structure. It may also include a solar panel and battery for areas without power. A node designed for a commercial installation would include a minimalistic housing that is designed to be mounted above a t-bar ceiling.

[15] (“What Is a Raspberry Pi?” n.d.)

OpenWave Protocol

The OpenWave Protocol defines how connected nodes communicate and route data through the network. When viewed on the OSI model[16] or TCP/IP model[17], OpenWave operates below the transport layer, operating directly with the network layer. Applications built without native OpenWave integration will continue to operate normally because the OpenWave layer is transparent to them.



Data Encryption

At the base level, the OpenWave protocol is based off asymmetrical cryptography.[18] Similar to how cryptocurrency operates using public and private key pairs, OpenWave utilizes a similar procedure to keep data private and secure. Every single packet of user data that flows over the OpenWave network is encrypted by the source node using the public key of the destination node. This ensures that only the destination node is able to decrypt the contents of the packet. The packet is also signed using the private key of the source node. This ensures that packets cannot be forged to appear as if they came from a different source node.

When any network enabled application sends data over OpenWave, it is automatically encrypted. It does not matter if the application was designed to support network encryption or not. Encryption is inherent to the design of the OpenWave network. If an application is accessing a resource external to the OpenWave network, the data remains encrypted while being transported, but is decrypted at the final hop, the OpenWave Gateway Node, before being forwarded to the external resource.

IP Address Allocation

OpenWave nodes are identified by their unique IPv6 address, which is generated directly from their private key. The private key is used to derive the public key. The public key is run through a hashing

[16] (Wikipedia n.d.; “OSI Model - Wikipedia” n.d.)
[17] (Wikipedia n.d.)
[18] (Wikipedia n.d.)

algorithm to generate an unpredictable hash. This hash is then run through a translation function which maps it to a unique IPv6 address.

Conceptual Example of IP Address Derivation Sequence:

```
Private Key -> Public Key -> Hash() -> Translation() -> IPv6 Address
```

Because the IPv6 address is derived from the private key in a way that is unpredictable and random, the odds of an IP address collision are negligible. The full IPv6 range contains 3.4e38 addresses. The reduced ULA range (usage described below) contains 2.65e36 addresses. For perspective, the highest Bitcoin hashrate recorded to date was 24,293,141 TH/s, recorded on February 8th, 2018.[19] Bitcoin uses the SHA-256 algorithm, which is sufficiently comparable for the purposes of this document. At this hash rate it would take 3.47 billion years to exhaust the reduced ULA address range.

If two nodes share the same private key, then they will also share the same IPv6 address on the OpenWave network. This could happen if a node operator accidentally copies the disk image of one node to another node, and forgets to generate a new private key. This will cause an IP address conflict that will notify the operator to correct. Failure to correct the error will may cause routing errors which will cause neighboring nodes to automatically route around the problematic nodes, and in some cases blacklist the nodes. The misconfigured nodes may not be able to serve all requests properly, but the OpenWave network will be largely unaffected.

OpenWave nodes cannot be assigned IPv4 addresses. Any application that attempts to access an IPv4 resource is automatically assumed to be a resource on the internet, or another network.

OpenWave addresses are allocated in the IPv6 unique local address (ULA) range of fd00::/7 similarly to how Cjdns allocates their addresses.[20] This design allows Cjdns IP addresses to intermingle with public IPv6 internet addresses without risk of a collision. OpenWave will adopt this same design initially, but it is the long term intent to open the full range of IPv6 addresses for allocation once network segregation models are robust enough to prevent collisions.

Service Messages

A Service Message is a category of data transmission that is used by nodes to organize and coordinate the transmission of Data Messages, similar to ICMP[21] on the Internet. Service Messages do not incur any cost for transmission on the network because they forbid the transmission of any meaningful data. Service Messages are transmitted without encryption in order to ensure that they conform to the specification, and do not carry any meaningful data which could allow a malicious user to obtain free usage of the system.

Examples of Service Messages:

- Ping Node
- Traceroute

[19] (Blockchain.info n.d.)
[20] (DeLisle n.d.)
[21] (Wikipedia n.d.)

- Query Node Details
    - Check Node Status
    - Check Node Capabilities
    - Check Data Rates
    - Check Data Allocation (Balance)
    - Purchase Data
    - Request Refund for Unused Data

Service Messages are generated automatically by nodes during normal operation as they coordinate with peer nodes. They do not carry the same privacy considerations as an end user. However, the unencrypted nature of the messages can give clues about the destinations of the encrypted Data Messages. For scenarios where higher security is required by a user, a Service Message can be embedded in a Data Message. This allows the Service Message to be encrypted during transmission, but it will incur a transmission fee.

Any data that is transmitted without fee creates the potential for malicious abuse. Initial versions of OpenWave will implement rate limiting for requests per physically connected node. Future versions of OpenWave may eventually require small fees for Service Messages.

Data Messages

A Data Message is a category of data transmission used by nodes to transmit meaningful data through the network. Data Messages represent the bulk of all data transmission. Data Messages require a fee to be paid to all nodes that are tasked with relaying the data to its destination. All Data Messages are encrypted, and the contents of the data can only be decrypted by nodes which have a corresponding private key.

Gateway Services

A Gateway Service is a network resource that is not located on the native OpenWave network. Examples include Tor, IPFS, Swarm, and the internet. A Gateway Node publishes a list of the Gateway Services it provides through the OpenWave Directory Protocol, and nodes which need to access this resource can do so natively without having to install additional software.

Gateway Services may or may not carry fees in addition to base OpenWave transmission fees. Different Gateway Services may require different levels of computing resources. Therefore, each Gateway Service can be configured with a different fee level, allowing the market to dictate fair market value.

Route Sessions

Every Data Message transmitted requires standard header data. In many cases this header data is the same from message to message. Client Nodes can create a Route Session to register the common header data with a short identifying code. The identifying code is then used in place of the header data, reducing the amount of overhead that must be passed with each request.

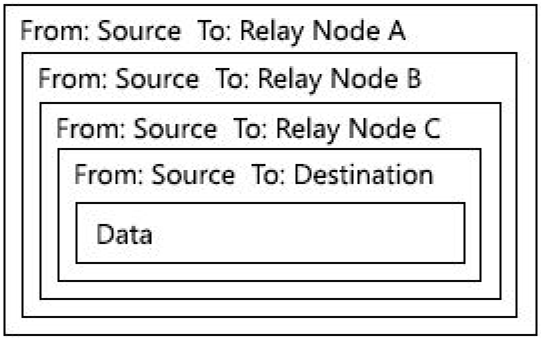
Compact Routing

Compact Routing builds upon Route Sessions to establish a more efficient way to transmit data with reduced overhead. A Data Message can be sent through multiple Relay Nodes, containing only the Route Session identifier, the payload, and the validating signature of the source. The payload itself is encrypted using the public key of the destination node, so it cannot be decrypted by any other nodes that relay the message.

Compact Routing is considerably more space efficient, but exposes the destination node to all nodes it is relayed through. When a higher level of privacy is required, Fractal Routing can be used to encrypt the final destination from nodes which relay the data.

Fractal Routing

Fractal Routing is a Data Message structure where one or more Data Messages are nested within each other, similar to a matryoshka doll.[22] Each nested Data Message is encrypted to one specific node. When the message is received, it is decrypted and the payload examined. If the payload destination is the current node, it will process the message. If the payload destination is another node, it will relay the payload to the next destination.


From: Source To: Relay Node A
From: Source To: Relay Node B
From: Source To: Relay Node C
From: Source To: Destination
Data

When Fractal Routing is used, relay nodes are completely unaware of the final destination of the payload. All that can be determined is the next hop in the relay chain.

Fractal Routing allows multiple messages to be nested within each message. The messages included can be of any type, including Service Messages or Data Messages. This opens a tremendous level of flexibility to the system, allowing for superior efficiency and privacy.

Segregated Payload

Segregated Payload is when a message references the payload of another message. This allows for further flexibility at the message level and allows for a single payload to be stored and reused temporarily.

[22] (Wikipedia n.d.)

Consolidated Routing

Consolidated Routing is a method to increase the efficiency of identically broadcasted messages utilizing Compact Routing and Segregated Payload. The source node creates a message that is addressed to multiple destination nodes and contains a single payload. This can be used to facilitate bandwidth intensive activities that may otherwise saturate the network.

Node Aliasing

Node Aliasing is the ability for an OpenWave node to create additional private/public keys and IP addresses. These additional addresses are always secondary to the primary address, and in most cases are not published to ODP. A node can create a node alias for itself, or request a node alias be created on a relay or destination node. Node aliases are designed to be temporary and act as a privacy feature similar to the IPv6 privacy extensions.

When requesting a node alias on a relay node, the private key can either be provided to the node or retained by the source node. If the private key is provided, the relay node will process the messages it receives directly. If the private key is not provided, it will forward the encrypted packet back to the source node, effectively acting as a proxy.

Redundant Nodes

Nodes which serve identical data, such as static images or video files, can utilize Redundant Node functionality to provide the same benefits as a Content Delivery Network. A node operator would set up OpenWave Nodes in multiple geographic locations, each of which serves the same content. The operator would then publish the details of all the nodes in a single ODP entry. When a Client Node queries ODP, they will receive the complete list of nodes available and utilize the most optimal connection.

OpenWave Protocol Routing

Routing Algorithm

Nodes will route packets between each other using a custom algorithm based on the Babel routing protocol.[23] OpenWave will introduce additional metrics, which will be used to generate the values used to determine viable routes.

Example additional metrics may include:

- Data rates
- Reliability
- Availability
- Available network services
- Available balance

[23] (Wikipedia n.d.)

Data Rates

Data rates are the fees for blocks of data transmission, a data block being 1MB of data transmission. Nodes charge only for transmitting and do not charge for receiving data. The data rate will be variable within configurable ranges, based on utilization and availability. Defaults will be set, but the ranges will be configurable by the operator.

As utilization changes, the price will automatically adjust within the configured ranges. As the device approaches full capacity, the advertised rate will increase exponentially. If the node becomes non competitively priced, data will automatically route through other paths.

Gateway services will each have individually configurable rates. Different gateway services consume different amounts of resources, and are subject to different amounts of demand.

Once data is purchased at a specific rate, future price increases do not affect it. The data rate must be honored at the sale price, even if it remains unused for a long period of time.

Data Metering

When data is purchased on a node, the transaction is recorded by both the client and the node itself. The amount of data transferred is also tracked by both the client and node to ensure that there are not large discrepancies. If a significant discrepancy does occur, the client will automatically blacklist the node from future use.

Data Block Purchases

Data blocks are 1MB increments, but purchases will occur in larger blocks using a form of pre-authorized payment escrow. As data is utilized, blocks of payment will automatically be released from escrow by the client, creating a transparent process that will not impact the user experience.

OpenWave Protocol Routing Privacy

The flexibility of the OpenWave messaging system allows data to move throughout the network indirectly and randomly. This enables routing strategies which can extend privacy features beyond encryption, reaching the topology layer.

Randomized Routing

Randomized Routing is a privacy feature that utilizes Fractal Routing to send sequential Data Messages through different Relay Nodes at random. Rather than always using the fastest or least expensive route, multiple routes are mixed for a single data stream. The net effect being that different pieces of a single transmission are sent through different Relay Nodes.

Hidden Nodes

Hidden Nodes are OpenWave Nodes which do not publish their own details to ODP. OpenWave nodes that are connected to the Hidden Node directly are aware of their presence, but do not publish them to ODP as a connected node. A Hidden Node cannot be connected to unless the Source Node is provided with the information required to establish a static route. This would typically be distributed in a specially formatted file containing the needed data. A Hidden Node is a privacy feature that would generally be used on a Source Node or the Destination Node.

Obfuscated Routing

Obfuscated Routing is an advanced privacy feature that combines Fractal Routing, Node Aliasing, Randomized Routing, and Hidden Nodes all at the same time. Not only is all data encrypted at every hop, but the flow of data is completely obfuscated using multiple techniques. By combining Randomized Routing, Node Aliasing, and Hidden Nodes, the source and destination of data streams cannot be accurately calculated by analyzing network topology.

OpenWave Directory Protocol

The OpenWave Directory Protocol (ODP) is a distributed database that holds transient metadata about the OpenWave network. This data is essential for network to operate, but changes regularly. All of this data is regenerated at regular intervals and in no way fatal if lost.

ODP primarily hosts two types of data - node data and connection data. Node data is published information about OpenWave nodes, such as the public key of the node and a list of the connected nodes. Connection data is information published about how to connect to a specific OpenWave resource and includes OpenWave Naming System data.

Node Data

Examples of ODP node data:

- Public Key: The public key associated with the queries node
- Connected Nodes: A list of all nodes connected
- Metadata:
    - Available Services
    - Node Capabilities
    - Data Rates

Connection Data

Examples of ODP connection data:

- Resource Name
- Application Layer Services (HTTP, HTTPS, DNS, FTP, SSH, etc.)
- Network Type (OpenWave, Internet, IPFS, etc.)
- Network Protocol (HTTP, HTTPS)
- TCP Port

- IP Address (IPv4 or IPv6)

OpenWave Naming System

The OpenWave Naming System (OWNS) is a decentralized protocol for associating named strings to network resources. This system is similar to the DNS system and backwards compatible to an extent, but extends the capabilities beyond the original limits of the DNS system. The scope of OWNS includes full connection details in addition to basic name->address resolution and common features that are desired but not present in the original DNS specification. These features include the ability to specify a network, protocol, port, or redirect to another resource.

One of the goals of OWNS is to create unified accessibility to the emerging and pre-existing naming systems that exist today. For example: In 2011, the first fork of Bitcoin was a decentralized naming system known as Namecoin.[24] Because the system is not accessible in any meaningful way, it remains largely devoid of usage. OWNS can provide access to Namecoin through an OpenWave Service compatibility layer.

Storage and Query Format

Record data is encrypted utilizing the original query string. The query string is then hashed, and this hash is used as the key for clients to locate this data.

Conceptual Example of a Hashed Query:

```
QUERY CONNECT openwave.io -> SHA3-256() ->
a6d3c3f7263dac270cf5685539af2cc2beb2f6478b494d98a7bdb4a8fd8fec41
```

Conceptual Example of an Encrypted Record Data:

```
{"address": "fd19:40a6:357c:a565:f4d4:cc89:0193:0afb"} -> AES-256(“QUERY CONNECT
openwave.io”) -> <encrypted binary data>
```

The hash and encrypted data are generated on the source node before publishing. Plain text data is never transmitted.

Conceptual Example of Published Data:

```
{
    "key": "a6d3c3f7263dac270cf5685539af2cc2beb2f6478b494d98a7bdb4a8fd8fec41",
    "value": <encrypted binary data>
}
```

The queries being published remain unknown to the greater network, because hash functions create unpredictable strings. In order to decrypt record data, you must know the original query string. This

[24] (Namecoin n.d.)

privacy feature makes widespread data mining of ODP substantially more difficult since there is no way to "scan" the entire database.

Payment System

All payments on OpenWave are made using the OpenWave Token (OW). These tokens are sent between devices to pay for data transmission throughout the network. The price of data is completely market based, and users set their own rates.

OpenWave Token

The OpenWave token will be based on ERC20[25], or a similar token standard, which in turn will utilize an existing smart contract platform. The OpenWave token will be the exclusive method of value exchange on the OpenWave platform.

All current smart contract platforms require an internet connection to fully propagate transactions. However, OpenWave payments will not require a direct internet connection. A client node can initiate transactions through any nearby payment node, even if the payment node does not allow direct internet access.

The OpenWave token will initially be tied to a single underlying smart contract platform. However, the long term goal is to be fully agnostic towards any underlying platform. Once Atomic Cross Chain Trading[26] has been accepted industry wide, the OpenWave token will be made available on multiple, interchangeable smart contract platforms.

OpenWave Tokens are deflationary by nature. All tokens are generated during the token contract genesis and are not created through inflationary mining. Nodes that operate successfully generate revenue exclusively through direct payments by other users. Node operators receive 100% of the payments, and OpenWave does not tax these payments in any way.

OpenWave Token Purchases & Sales

OpenWave tokens will be available for purchase through two primary avenues: cryptocurrency exchanges, and directly through the OpenWave software.

OpenWave tokens purchased through exchanges can be transferred to the client node to be used. Any form of currency supported by the exchange can be used to purchase OpenWave tokens. OpenWave tokens accumulated on server nodes can also be transferred to exchanges to be sold.

The OpenWave software will also allow purchases of OpenWave tokens directly through the client interface. This will provide a convenient way for users to obtain tokens quickly without having to deal with the formality of an exchange. Users will be able to pay using select cryptocurrencies or traditional credit cards or debit cards. Tokens purchased by this method will be priced slightly above market to cover the costs of providing this convenience service.

[25] (Wikipedia n.d.)
[26] (“Atomic Cross-Chain Trading” n.d.)

Data Blocks

A data block is a unit representation of data purchased by client for network access. Data blocks are variable in size and metered in increments of 1MB or larger. A smaller size allows for more granular usage tracking, but larger sizes allow for more data to be purchased with less required payment overhead. The size management of the data blocks will be handled by the client node, automatically adjusting based on usage patterns.

Data Block Purchases

Purchases of data blocks will occur in batches, using a form of pre-authorized payment escrow. This pre authorization will utilize a smart contract to ensure that X available data blocks have been purchased and are available for use on the node. The full amount starts in a locked escrow state. As data is utilized successfully, the client will automatically send unlock codes allowing the node to receive payment for the blocks in escrow.

After the escrow has been validated, no further internet connection is required until the funds are claimed or released. This reduces the frequency of transactions required to purchase data. The unlock codes serve as a form of instant payment that is off-chain and does not require network validation until it is closed. This method serves a similar purpose to technologies such as Lightning Network[27], but does not require participation in a layer 2 network.

All funds in an escrow smart contract will expire at a predetermined date. In the event that a node disappears, or unused data blocks remain, unused funds will be returned to the client. The node can also release the unused funds from escrow at the request of the client. There is no way for a node to disappear with user funds without providing service.

Acknowledgements

References

“Atomic Cross-Chain Trading.” n.d. Bitcoin Wiki. Accessed March 10, 2018. https://en.bitcoin.it/wiki/Atomic_cross-chain_trading.

Blockchain.info. n.d. “Bitcoin Hash Rate.” Blockchain.info. Accessed February 8, 2018. https://blockchain.info/charts/hash-rate.

DeLisle, Caleb James. n.d. “Cjdns White Paper.” GitHub. Accessed February 8, 2018. https://github.com/cjdelisle/cjdns/blob/master/doc/Whitepaper.md.

Dyn Blog. n.d. “A Baker’s Dozen, 2016 Edition.” Accessed February 8, 2018. https://dyn.com/blog/a-bakers-dozen-2016-edition/.

ethersphere. n.d. “Ethersphere/swarm.” GitHub. Accessed February 8, 2018. https://github.com/ethersphere/swarm.

Labs, Protocol. n.d. “IPFS Is the Distributed Web.” IPFS. Accessed February 8, 2018. https://ipfs.io/.

“Lightning Network.” n.d. Accessed February 13, 2018. https://lightning.network/.

Namecoin. n.d. “Namecoin.” Namecoin. Accessed February 8, 2018. https://namecoin.org/.

[27] (“Lightning Network” n.d.)

“OSI Model - Wikipedia.” n.d. Accessed February 8, 2018. https://en.wikipedia.org/wiki/OSI_model.

The Tor Project, Inc. n.d. “Tor Project | Privacy Online.” Accessed February 8, 2018. https://www.torproject.org/.

Tremback, Jehan, and Justin Kilpatrick. n.d. “Althea Mesh White Paper.” Althea. Accessed February 8, 2018. http://altheamesh.com/documents/whitepaper.pdf.

“What Is a Raspberry Pi?” n.d. Raspberry Pi. Accessed March 9, 2018. https://www.raspberrypi.org/help/what-is-a-raspberry-pi/.

Wikipedia. n.d. “Babel (protocol).” Accessed February 9, 2018a. https://en.wikipedia.org/wiki/Babel_(protocol).

———. n.d. “Cryptocurrency.” Accessed February 8, 2018b. https://en.wikipedia.org/wiki/Cryptocurrency.

———. n.d. “Cryptographic Hash Function.” Accessed February 8, 2018c. https://en.wikipedia.org/wiki/Cryptographic_hash_function.

———. n.d. “ERC20.” Accessed February 18, 2018d. https://en.wikipedia.org/wiki/ERC20.

———. n.d. “Global Internet Usage.” Accessed February 8, 2018e. https://en.wikipedia.org/wiki/Global_Internet_usage.

———. n.d. “Internet Control Message Protocol - Wikipedia.” Accessed February 8, 2018f. https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.

———. n.d. “Internet Protocol Suite.” Accessed February 8, 2018g. https://en.wikipedia.org/wiki/Internet_protocol_suite.

———. n.d. “Matryoshka Doll.” Accessed February 8, 2018h. https://en.wikipedia.org/wiki/Matryoshka_doll.

———. n.d. “OSI Model.” Accessed February 8, 2018i. https://en.wikipedia.org/wiki/OSI_model.

———. n.d. “Public-Key Cryptography.” Accessed February 8, 2018j. https://en.wikipedia.org/wiki/Public-key_cryptography.

———. n.d. “Public-Key Cryptography.” Accessed February 8, 2018k. https://en.wikipedia.org/wiki/Public-key_cryptography.

———. n.d. “Smart Contract.” Accessed February 21, 2018l. https://en.wikipedia.org/wiki/Smart_contract.

———. n.d. “Symmetric-Key Algorithm.” Accessed February 8, 2018m. https://en.wikipedia.org/wiki/Symmetric-key_algorithm.

———. n.d. “Utah Data Center.” Accessed February 8, 2018n. https://en.wikipedia.org/wiki/Utah_Data_Center.

———. n.d. “Wi-Fi.” Accessed February 8, 2018o. https://en.wikipedia.org/wiki/Wi-Fi.

Revisions

- Document v0.7.0 Revision 20180310: Initial draft release.
- Document v0.5.0 Revision 20180209: Pre-draft release.