HyperNebula: A Peer-to-Peer Overlay Network Based on Internet Protocol Address Identification

Alyssa McKeown

January 30, 2023

Abstract

In this paper, we describe HyperNebula, a peer-to-peer overlay network. HyperNebula has multiple features not found together in any previous network overlay protocol. It offers built-in Sybil attack resistance by opting for an Internet Protocol (IP) address-based node identifier rather than entropy. Furthermore, it does not include key-value lookup, thus reducing complexity and increasing performance in applications where storage is not needed. This paper examines the architecture of HyperNebula and its security features.

1 Background

Peer-to-peer (P2P) networks have become increasingly popular due to their scalability, reliability, and censorship resistance. P2P networks are composed of interconnected nodes that form a distributed network. Such a network has been utilised in various applications including file sharing, streaming media, distributed computing, and blockchain technology.

2 Introduction

The proliferation of P2P networks has enabled users to share data and resources in a decentralised manner. However, these networks are vulnerable to various attacks, including Sybil attacks, which can be used to disrupt network operations.

There are several existing peer-to-peer overlay network protocols. One of the most widely used for blockchain protocol applications is Kademlia, a Distributed Hash Table (DHT) overlay network protocol [1]. Kademlia is efficient and low latency but lacks built-in Sybil attack resistance. Some proposals have been made to implement resistance, such as a trust-based system [2]. However, these systems are poor choices for trustless blockchain protocols.

In order to address these issues of trust-based protocols, we present HyperNebula, a P2P overlay network with built-in Sybil attack resistance that does not require a trust-based system. Furthermore, rather than a DHT architecture, we propose an Internet Protocol (IP) address-based protocol for generating node identifiers.

3 Architecture

HyperNebula is communication protocol agnostic and can be implemented on top of User Datagram Protocol (UDP), Transmission Control Protocol (TCP), or Reliable User Datagram Protocol (R-UDP). This provides added versatility where applications can prioritize performance over reliability, or vice versa.

Each HyperNebula node is assumed to have either an Internet Protocol version 4 (IPv4) address or an Internet Protocol version 6 (IPv6) address. Node IDs are then constructed from these addresses. Rather than use the digest – a fixed length output of a cryptographic hash function – of the full address, as done in Chord [2], we instead use the digest of the individual address octets.

Nodes are sorted into h-buckets – First In, First Out (FIFO) collections of nodes —based on these octet digests, thus resulting in nodes being aware of more peers nearest their most significant IP address octets. Due to a larger delta between the most significant octet results at a more considerable distance, a diverse overlay network is created. Furthermore, the resulting overlay network results in most nodes being aware of their closest neighbours within the first h-bucket. Because of this, when a node is contacted, it is usually possible to contact their closest nodes in only one additional hop.

These nodes are sorted into 128 separate h-buckets based on the distance between them. The maximum number of nodes in an h-bucket is defined as H.

Distance is calculated by interpreting each node ID as an unsigned 128-bit integer. This distance can be expressed as $|x - y|$.

When a node is to be added to a full h-bucket, we check if the least recently seen node responds to a PING request. If the node responds, we discard the new node and retain the existing node. If the least recently seen node does not respond, we discard it and append the new node to the h-bucket.

3.1 Node Identifiers

In HyperNebula, nodes have 128-bit node identifiers (IDs). In the case of IPv4 addresses, we first calculate the Keccak-256 digest of each IP address octet, which we will refer to as D. In the case of IPv4:

$$D_0, D_1, D_2, D_3$$

Next, we take four bytes of the three most significant octet digests and two bytes of the least significant octet digest. Finally, we append the two bytes of the remote node's port.

In the case of IPv6 addresses, we do not generate any digests. Instead, we ignore the first two octets and assume them to be global unicast. We then append the two port bytes, and use the 128-bit result as our node ID.

3.2 Remote Procedure Calls

HyperNebula contains only two remote procedure calls (RPC). The PING RPC is used to determine if a node is online. REQUEST NODES instructs the remote node to respond with a message containing every known node. Whenever a node receives an RPC, the sender is added to the node's list of nodes and sorted into an appropriate h-bucket.

4 Sybil Attacks

A Sybil attack is when a malicious actor creates multiple identities to gain control over a distributed network. The goal is to gain an unfair advantage over other users in the network, for example, by manipulating the network to their benefit or by launching a denial-of-service attack on other users.

Multiple proposals have been made to introduce Sybil resistance to existing peer-to-peer overlay networks [2][3]. These proposals rely on trust management, which in many cases is undesirable in decentralised P2P networks. Furthermore, these proposals have proven both insufficient and difficult to implement.

Rather than trust management, HyperNebula uses IP addresses for node ID generation which offer built-in Sybil attack resistance. Since an attacker cannot arbitrarily create large quantities of IP addresses as they can with simple numbers, they cannot execute a Sybil attack on the network without immense resources.

5 Summary

With its novel node identifier scheme, HyperNebula is the only peer-to-peer overlay network protocol to combine built-in Sybil attack resistance with low-latency routing. It is an ideal protocol for any application that demands high performance and does not require key-pair storage, a notable example of which is blockchain protocols.

References

[1] Maymounkov, P., Mazières, D. (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds) Peer-to-Peer Systems. IPTPS 2002. Lecture Notes in Computer Science, vol 2429. Springer, Berlin, Heidelberg.

[2] Stoica, Ion and Morris, Robert and Karger, David and Kaashoek, M. and Balakrishnan, Hari. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. ACM SIGCOMM Computer Communication Review, vol. 31

[3] Jean-Philippe Eisenbarth, Thibault Cholez, Olivier Perrin. Ethereum's Peer-to-Peer Network Monitoring and Sybil Attack Prevention. Journal of Network and Systems Management,

2022, Special Issue on Blockchains and Distributed Ledgers in Network and Service Management, 30 (4), pp.65.

[4] Lin Cai, Roberto Rojas-Cessa. Containing Sybil Attacks on Trust Management Schemes for Peer-To-Peer Networks. 2014 IEEE International Conference on Communications (ICC)