May 2020

# DISTRIBUTED POLICY MANAGEMENT FOR SERVICE PROVIDER CHAINS

Nagaraj Kenchaiah

Niranjan M. M

# DISTRIBUTED POLICY MANAGEMENT FOR SERVICE PROVIDER CHAINS

AUTHORS:
Nagaraj Kenchaiah
Niranjan M. M

## ABSTRACT

Techniques are described herein to provide distributed end-to-end policy management across a chain of service provider networks (i.e., administrative domains). The techniques leverage an agent-centric framework for a fully distributed peer-to-peer network that allows nodes to maintain decentralized tamper-proof hash chains (e.g., Holochain). With this framework, the techniques are able to quickly and conveniently indicate network policies across a chain of service providers, in a distributed manner, and guarantee that requirements of the policies are met along the chain of service providers.

## DETAILED DESCRIPTION

Internet-based applications and services are often provided to a network by a chain of administrative domains. For example, administrative domains of service providers in a chain of service providers (referred to herein as a “SP chain”) may convey packets for a given Internet Protocol (IP) service or application. Often, an SP chain provides static network parameters (i.e., a fixed maximum bandwidth) and the various administrative domains forming the chain are unable to communicate to coordinate any changes (e.g., to propagate filter rules across the SP chain). Consequently, it may be difficult, expensive, and/or inefficient for a network (i.e., a home network or enterprise network) to obtain connectivity for an IP application or service with suitable network parameters, especially for networks with shifting and dynamic needs (i.e., networks that intermittently or infrequently need increased upload speed, bandwidth, etc. for a particular service or application). Moreover, the foregoing issues may make it difficult to achieve privacy in an SP chain for differential service offerings.

As a specific example, enterprises may need to intermittently export large amounts of data (e.g., hundreds of gigabytes or one or more terabytes) to the Cloud for big data analytics (e.g., network assurance, wireless service assurance, etc.), but may only intermittently need increased upload speeds for these exports and would rather not pay for the increased upload speeds between uploads. As another example, a home network user might want increased bandwidth to watch a three-dimensional (3D) movie for two hours, but might not want to pay for this increased bandwidth before and after watching the 3D movie. Typically, there are not convenient, efficient, and secure techniques available to allow these temporary on-demand (i.e., requirement driven) end-to-end IP connectivity service requests to be indicated and guaranteed across an SP chain.

Moreover, often an SP chain may have a limited ability to identify and defend against network threats/attacks, such as a distributed denial of service (DDoS) attack (e.g., since limited communications between administrative domains limit the SP chain's ability to collectively filter traffic from an attacker). For example, with DDoS attacks, an attacker attempts to compromise host devices and use those infected host devices to make machines or network resources (e.g., an application server, a client, a router, a firewall, or a network list of an entire network etc.) unavailable to their intended users. DDoS attack can be mitigated by blocking traffic from the attacker(s) (e.g., with filter rules); however, currently, there are not many techniques that allow for communication across an SP chain to coordinate such efforts (e.g., to propagate filtering rules).

Some techniques utilize blockchain techniques to try to provide communication across SP chains. However, recently, blockchain has been discovered to have a number of drawbacks/limitations. For example, since blockchain techniques require data to be replicated on all blockchain nodes, blockchain techniques may create scalability issues. Moreover, the number of transactions may limit scaling. Additionally, convergence time may be relatively long for some blockchains so that, for example, adding transactions to a blockchain might take a couple of minutes in some uses cases. Still further, time must be synchronized across blockchain nodes since timestamps are part of transactions and are used during merging of ledgers (thus, nodes that are not time-synchronized may create ledger issues).

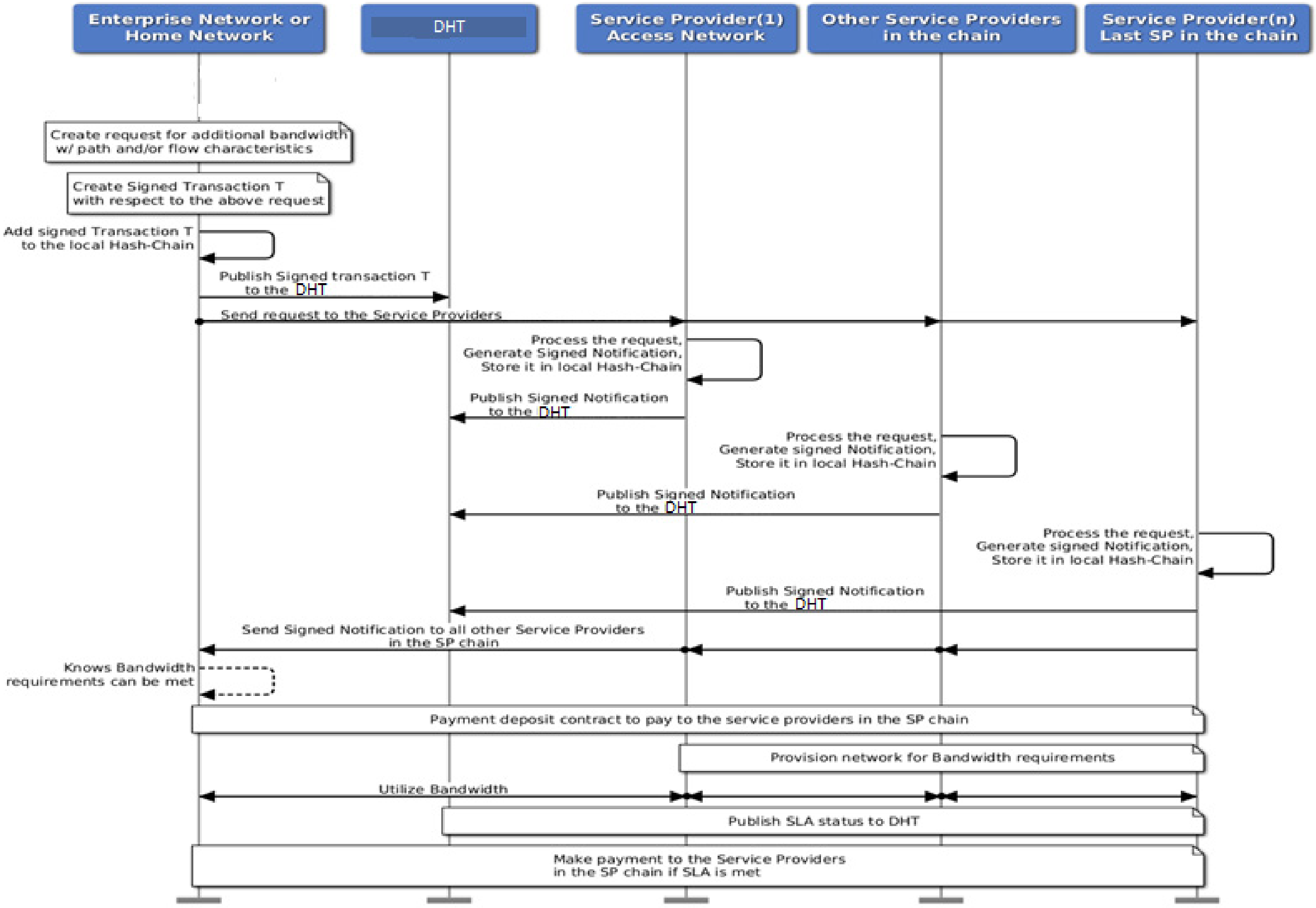
In view of the foregoing, the techniques presented herein enable communication across an SP chain to provide distributed end-to-end policy management across multiple service providers networks (i.e., administrative domains). To achieve this, the techniques leverages an agent-centric framework for a fully distributed peer-to-peer network that allows nodes to maintain decentralized tamper-proof hash chains (e.g., a Holochain framework). With this framework (referred to herein as "agent-centric framework," "the framework", "agent-centric hash chain framework," etc.), the techniques are able to quickly and conveniently indicate network policies across a chain of service providers, in a distributed manner, and guarantee that requirements of the policies are met along the chain of service providers. Notably, since the end-to-end policy management is distributed, the techniques may be easily scalable. Moreover, the framework ensures that the end-to-end policy management is efficient (e.g., since convergence is unnecessary).

Generally, these techniques are generic in nature so that they are applicable to any end-to-end or multi-node policy that should be verified across multiple service provider networks (i.e., administrative domains). That is, the techniques presented herein provide a convenient and useful method to indicate, in a distributed manner, any network policy across domains of an SP chain and to guarantee, in a distributed manner, that the requirements are met. Thus, although the techniques presented herein are discussed with respect to satisfying on-demand bandwidth requests and defending against DDoS attacks, the techniques can also be used with for a variety of other functions/purposes, such as service level agreement (SLA) management and verification (e.g., via performance metrics). That said, among other advantages, these techniques can allow an SP chain to accommodate additional bandwidth for specific duration (e.g., which subscribers can request and pay for accordingly) and/or to defend against attacks (e.g., a DDoS attack). The techniques may also ensure privacy of and/or allow granularity for an SP chain by, for example, managing fixed and variable costs of an SP chain and/or adjusting the services in the SP chain on a per-customer basis (which may be valuable to customers and service providers).

As a specific example, Figure 1 below provides a sequence diagram that illustrates how the techniques can determine if IP service/application requirements are met across an SP chain by leveraging a fully distributed peer-to-peer agent-centric framework that allows nodes to maintain decentralized tamper-proof hash chains (e.g., Holochain). Generally, the framework provides a fully distributed way of data sharing and access, with secure peer-to-peer network communication (i.e., no centralized server, no ledgers, no intermediaries, no miners). Each node stores its own immutable hash chain to maintain ordered transactions based on a time sequence at each node (and, thus, is agent-centric) and can communicate with other nodes to propagate data. For example, when a node receives a message, it can broadcast the message to some or all of its peers, which can propagate the message to their peers, creating an exponential rate of propagation. The nodes can each share entries, metadata, neighborhood health, and peer addresses.

Additionally, the framework may utilize digital signatures that provide authenticity and ownership of data and a distributed hash table (DHT) to provide de-centralized data storage (e.g., so that data can be hosted by entities other than a centralized authority system). The DHT utilizes cryptographic hashes for content-addressable storage and validates with the hash-chains and digital signature before storing transactions.

Generally, in a DHT, nodes coordinate amongst themselves to balance and store data in the network without any central coordinating party. DHTs are both fault tolerant and resilient when key/value pairs are replicated, but require that information to be evenly distributed across the network. Thus, DHTs utilize consistent hashing, where a key is passed through a hash algorithm that serves as a randomization function, ensuring that each node in the network has an equal chance of being chosen to store the key/value pair. Typically, DHTs only use the hash of the data itself to confirm authenticity, provenance, timelines, or integrity of data sources. However, in the framework leveraged by the presented techniques, validation rules can be embedded as a condition for the propagation of data, which keeps data bound to signed source chains (e.g., the framework may use a Holochain DHT). This provides similar consistency and rule enforcement to blockchain ledgers in an asynchronous manner while alleviating bottlenecks caused by consensus requirements. That is, the framework DHT leverages signed source chains to ensure tamper-proof immutability of data and to verify data origins and provenance. The framework DHT also emulates aspects of a graph database by enabling nodes to connect links to other hashes in the DHT tagged with semantic markers, which may help locate hashes for retrieval from the DHT.


Enterprise Network or Home Network
DHT
Service Provider(1) Access Network
Other Service Providers in the chain
Service Provider(n) Last SP in the chain
Create request for additional bandwidth w/ path and/or flow characteristics
Create Signed Transaction T with respect to the above request
Add signed Transaction T to the local Hash-Chain
Publish Signed transaction T to the DHT
Send request to the Service Providers
Process the request, Generate Signed Notification, Store it in local Hash-Chain
Publish Signed Notification to the DHT
Process the request, Generate signed Notification, Store it in local Hash-Chain
Publish Signed Notification to the DHT
Process the request, Generate signed Notification, Store it in local Hash-Chain
Publish Signed Notification to the DHT
Send Signed Notification to all other Service Providers in the SP chain
Knows Bandwidth requirements can be met
Payment deposit contract to pay to the service providers in the SP chain
Provision network for Bandwidth requirements
Utilize Bandwidth
Publish SLA status to DHT
Make payment to the Service Providers in the SP chain if SLA is met

As mentioned above, Figure 1 illustrates how the techniques presented herein can leverage the framework (e.g., Holochain framework) to determine if IP service/application requirements are met across an SP chain. This example illustrates how the techniques can handle an on-demand bandwidth request, but as mentioned, this is merely an example and the techniques are applicable to a variety of parameters. Initially, an enterprise or home network implementing the techniques presented herein creates a request for the service providers in an SP chain provide additional bandwidth for a flow or set of flows for a specific duration. The enterprise or home network may also convey path characteristics, like downstream or upstream delay tolerance, loss tolerance, jitter tolerance, minimum and maximum bandwidth in the request. That is, the enterprise or home network can create a transaction T with a general or specific request.

Once the request is created, a node of the decentralized framework associated with the home or enterprise network signs transaction T and adds it to its local hash-chain. Although Figure 1 illustrates the node as a device in the home or enterprise network, the node could also be a service provider node directly connected to the home or enterprise network. Next, the signed transaction T is published to the DHT and the request is conveyed (using peer-to-peer communication protocols) to all other service providers in the SP chain. In response to the request, each service provider in the SP chain processes the request and publishes a signed notification response to the DHT (e.g., indicating they can and will accommodate the bandwidth requirement). Assuming all of the service providers in the SP chain agree, the last service provider in the SP chain also publishes a signed notification to every other Service Provider in the chain, allowing every service provider on the SP chain to know the request has reached the last service provider. At this point, the enterprise or home network will also know if the bandwidth requirement can be fully or partially met by the service providers in the SP chain.

If the requested path characteristics will be honored by all the service providers in the SP chain, the enterprise or home network can execute a payment deposit contract to pay to the service providers in the SP chain. Once the payment deposit contract is executed, service providers in the SP chain provision their network to accommodate the path characteristics conveyed by the enterprise or home network and the enterprise or home network can use the applications that need additional bandwidth for the requested duration. Moreover, during the requested duration of time, service providers in the SP chain can publish network performance metrics for SLA monitoring. If the SLA requirements are met for the requested duration, the payment deposit contract will be executed to pay to the service providers in the SP chain.

Consequently, the techniques may allow multiple businesses or enterprises to be part of the same end-to-end request. Moreover, the techniques can track service provider compliance with requests without an out-of-band and/or centralized mechanism. Still further, during execution of these techniques, fixed and variable costs of the SP chain can be managed privately because the framework may implement generic validation rules for the whole SP chain and service provider specific validation rules that include fixed and variable costing, security policies, services offered etc. That is, the techniques presented herein resolve on-demand end-to-end provisioning in an SP chain (e.g., bandwidth demands) in an efficient, private, secure, and authentic manner.

If, instead, the transaction T of Figure 1 was intended to protect against a DDoS attack, transaction T could request to enforce filter rules (e.g., conveyed in Boarder Gateway Protocol (BGP) flowspec). The transaction T could include DDoS attack details such as attack type, total dropped packet count, average dropped packets per second etc. and would be signed and stored in local Hash-Chain. As mentioned above, once a transaction T is stored in the local Hash-Chain, the transaction T is published to the DHT and conveyed (using peer-to-peer communication protocols) to all other service providers in the SP chain.

After the transaction T is published and conveyed, service providers in the SP chain could use DDoS detection techniques to analyze flow records, packet samples collected from the attackers IP addresses (e.g., conveyed in the BGP flowspec) to check if a DDoS attack has been launched on a target/victim (e.g., the home or enterprise network initiating the request). Then, DDoS traffic from the attacker will be filtered and the target/victim network will know which service providers in the SP chain agreed to honor the filtering rules. Moreover, if service providers did not agree, it might be indicative that the service providers are compromised. Since the framework leveraged by the techniques presented herein provides a tamper-proof manner of conveying information in full distributed computing environment, the conveyed information is not only authentic, it is immutable. Thus, among other advantages, when the techniques implemented herein are utilized to defend against DDoS, the techniques may gather DDoS attack information and application-level statistics in a distributed manner and acquire application level acknowledgement that filtering rules are being enforced.

In summary, techniques are described herein to provide distributed end-to-end policy management across a chain of service provider networks (i.e., administrative domains). The techniques leverage an agent-centric framework for a fully distributed peer-to-peer network that allows nodes to maintain decentralized tamper-proof hash chains (e.g., Holochain). With this framework, the techniques are able to quickly and conveniently indicate network policies across a chain of service providers, in a distributed manner, and guarantee that requirements of the policies are met along the chain of service providers.