Universidade do Minho
Escola de Engenharia
Departamento de Informática

Paulo Edgar Mendes Caldas

Development of a system compliant with the Application-layer Traffic Optimization protocol

January 2021

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Paulo Edgar Mendes Caldas

Development of a system compliant with the Application-layer Traffic Optimization protocol

Masters dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
Pedro Nuno Miranda de Sousa

January 2021

ACKNOWLEDGEMENTS

I would like to firstly thank my advisor, professor Pedro Nuno Sousa, who was always present in any moment I struggled and required input to improve on my work.

I would also like to thank my family for financially and emotionally supporting me through my academic journey, the friends I've made along the way that made me see the best in people, and last but not least my dog Oscar who showed me unconditional love like only a dog could.

I finally also thank you, the reader - as a work unused is no work at all, may you find some value in it.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Paulo Edgar Mendes Caldas

ABSTRACT

With the ever-increasing Internet usage that is following the start of the new decade, the need to optimize this world-scale network of computers becomes a big priority in the technological sphere that has the number of users increasing, as are the *Quality of Service (QoS)* demands by applications in domains such as media streaming or virtual reality.

In the face of rising traffic and stricter application demands, a better understanding of how *Internet Service Providers (ISPs)* should manage their assets is needed. As an effort to optimize the Internet, one important concern is how applications utilize the underlying network infrastructure over which they reside. An evident issue is that most of these applications act with little regard for ISP preferences, as can be evidenced by their lack of care in achieving network proximity among neighboring peers, a feature that would be preferable by network administrators and that could also improve application performance. However, even a best-effort attempt by applications to cooperate will hardly succeed if ISP policies aren't clearly communicated to them. A system to bridge layer interests has thus much potential in helping achieve a mutually beneficial scenario.

The main focus of this thesis is the *Application-Layer Traffic Optimization (ALTO)* working group, which was formed by the *Internet Engineering Task Force (IETF)* to explore standardizations for network state retrieval. The working group devised a request-response protocol where authoritative and trustworthy entities provide guidance to applications in the form of network status information and administrative preferences, with the intent of achieving layer cooperation during normal application operations as a means to reach better Internet efficiency through the optimization of infrastructural resourcefulness and consequential minimization of its operational costs. This work aims to implement and extend upon the ideas of the ALTO working group, as well as verify the developed system's efficiency in a simulated environment.

Keywords: Application-Layer Traffic Optimization, Content Distribution Networks, Network Optimization, Peer-to-Peer, Traffic Engineering

RESUMO

Com o uso cada vez mais acrescido da Internet que acompanha o início da nova década, a necessidade de otimizar esta rede global de computadores passa a ser uma grande prioridade na esfera tecnológica, que vê o seu número de utilizadores a aumentar, assim como a exigência, por parte das aplicações, de novos padrões de Qualidade de Serviço (QoS), como se vê em domínios de stream multimédia em tempo real ou realidade virtual.

Face ao aumento de tráfego e a padrões de exigência aplicacionais mais restritos, uma melhor compreensão é necessária de como os fornecedores de serviços Internet (ISPs) devem gerir os seus recursos. Numa tentativa por otimizar a Internet, um ponto fulcral é o de perceber como as aplicações utilizam os recursos da rede sobre a qual residem. Um problema aparente é a falta de consideração que estas e outras aplicações têm pelas preferências dos ISPs durante a sua operação, como as aplicações P2P pela sua falta de esforço em obter proximidade topológica com os vizinhos na rede overlay, que caso existisse seria preferível por administradores de rede e teria potencial para melhorar o desempenho aplicacional. Todavia, uma tentativa de melhor esforço por parte das aplicações por cooperar não será bem-sucedida se tais preferências não são claramente comunicadas. Um sistema que sirva de ponte de comunicação entre as duas camadas tem portanto bastante potencial na tarefa de atingir um cenário mutuamente benéfico.

O foco principal desta tese é o grupo de trabalho ALTO, que foi formado pelo IETF para explorar estandardizações para recolha de informação do estado da rede. Este grupo de trabalho especificou um protocolo de pedido e fornecimento de recursos onde entidades autoritárias auxiliam aplicações com informação sobre estado de rede e preferências administrativas, como forma de obter cooperação entre camadas durante operação aplicacional, para melhor otimizar a Internet através de uma mais eficiente utilização de recursos infraestruturais e a consequente minimização de custos operacionais. Este trabalho pretende implementar e alargar as ideias do grupo ALTO, bem como verificar a eficiência do sistema desenvolvido num ambiente simulado.

Palavras-Chave: Application-Layer Traffic Optimization, Content Distribution networks, Engenharia de Tráfego, Otimização de rede, Peer-to-peer

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACL Access-Control List.
ADSL Asymmetric digital subscriber line.
ALTO Application-Layer Traffic Optimization.
ANE Abstract Network Element.
API Application Programming Interface.
AS Autonomous System.

BGP Border Gateway Protocol.

CAN Content Addressable Network.
CaTE Content-Aware Traffic Engineering.
CDN Content Distribution Network.
CDNI Content Distribution Network Interconnection.
CORE Common Open Research Emulator.
CPU Central Processing Unit.

DHT Distributed Hash Table.
DiffServ Differentiated services.
DNS Domain Name System.
DoH DNS over HTTPS.
DoS Denial of Service.
DPI Deep Packet Inspection.
DTO Data Transfer Object.

EGP Exterior Gateway Protocol.
EMEA Europe, the Middle East and Africa.

FCC Federal Communications Commission.

GNP Global Network Positioning.
GSLB Global Server Load Balancing.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

ID Identifier.
IDMaps Internet Distance Map Service.
IETF Internet Engineering Task Force.
IGP Interior Gateway Protocol.
IP Internet Protocol.
ipv4 Internet Protocol version 4.
ipv6 Internet Protocol version 6.
IRD Information Resource Directory.
ISP Internet Service Provider.

JSON JavaScript Object Notation.

LSPD Label Switched Path Database.

MAC Media Access Control.
MPLS Multiprotocol Label Switching.
MTR Multi-Topology Routing.
MVC Model-View-Controller.

NETCONF Network Configuration Protocol.
NetPaaS Network Platform as a Service.

OSPF Open Shortest Path First.
OSPFv3 Open Shortest Path First Version 3.

P2P Peer-to-Peer.
PaDIS Provider-Aided Distance Information System.
PC Personal Computer.
PID Provider-Defined Identifier.
PoP Points of Presence.

QoE Quality of Experience.
QoS Quality of Service.

RAM Random-Access Memory.
RBAC Role-Based Access Control.
REST Representational state transfer.
RFC Request for Comments.

RTT Round-Trip Time.

SDN Software Defined Networking.
SNMP Simple Network Management Protocol.
SQL Structured Query Language.

TCP Transmission Control Protocol.
TED Traffic Engineering Database.
TLS Transport Layer Security.

URL Uniform Resource Locator.

XMPP Extensible Messaging and Presence Protocol.

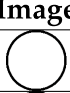
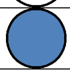
1 SYSTEM ARCHITECTURE AND DEVELOPED MECHANISMS

As the main proposed goal of this work is the implementation of a system that complies with the *Application-Layer Traffic Optimization (ALTO)* working group's devised protocol, this chapter exhibits the planned software specifications needed to implement the system as a whole, with the aforementioned protocol being a crucial part of client-server resource exchange. Initial attention is given to the general architecture on Section 1.1, with the goal of identifying key entities, their purpose, and how they interact among themselves withing the macro level. Following, Section 1.2 reviews the planned access control methods to ensure the delineation and enforcement of rules about which users can do what actions on the system's resources. The aforementioned ALTO resources can be considered the driving force behind the system, as they are what client entities seek, and what *Internet Service Providers (ISPs)* wish to provide, and an overview of their design is given on Section 1.3. Finally, Section 1.4 provides a more detailed specification of each of the main system roles. Within it, a overview of the interfaces and possible functions is given. Regarding the ALTO client, how it retrieves server insight to help its application decisions. Regarding the server, how it can be located, how it provides its data with optional query parameters, and how it can synchronize with other servers in another domain. For last, how the network state providers retrieve raw network information and gather it for administrator processing and subsequent upload into the ALTO server.

1.1 GENERAL ARCHITECTURE

Figure 1.1 presents a high-level conceptual model of how the network information flows in a given ISP. Network data originates in the topology itself, and is gathered into a network information aggregator by the appropriate means - this aggregator defines an interface through which network data can be uploaded, and entities utilize it to provide the network data they have collected. These entities will use different means to gather information, as the Internet is supported by a massive variety of protocols and standards for network and resource information querying. For example, a node could deploy a daemon listening for *Open Shortest Path First (OSPF)* protocol packets to gather path cost information, and another using *Simple Network Management Protocol (SNMP)* to gather node property information. Obviously, since the interface simply defines how raw data must be formated to be accepted by the network information aggregator, means through which the data is uploaded are left to the source itself, and because of this static data uploads that were previously collected, such as ones residing in another system's database could be used instead of dynamic retrievals. The network information aggregator serves as a hub for network administrators to process the raw network data that was collected by the previous tier, and transform it into ALTO resources ready to be accepted and distributed by the corresponding server. This task of network information processing is where ISP policies and preferences are injected via, for example, the abstraction of network entities with the aggregation of network addresses into *Provider-Defined Identifiers (PIDs)*, and the creation of cost maps which result from the transformation of network link information mixed with given ISP goals. If, say, the administrator wished to provide a cost map between network entities which aimed to reduce inter-network traffic, it would firstly aggregate endpoints into abstract entities with common properties, as an attempt not to share too much infrastructural information, and then use the previously collected network link information, attribute higher costs to undesired links, and transform it utilizing the Dijkstra's algorithm to create a shortest path map that is bound to provider preferences. Such map is then parsed as an ALTO resource - more specifically, a cost map - and afterwards uploaded into the ALTO server with the access policies the administrator sees fit.

Table 1.1: Network node entities in the conceptual ALTO system representation

Image	**Description**
	Network node
	Network node participating in a given overlay network

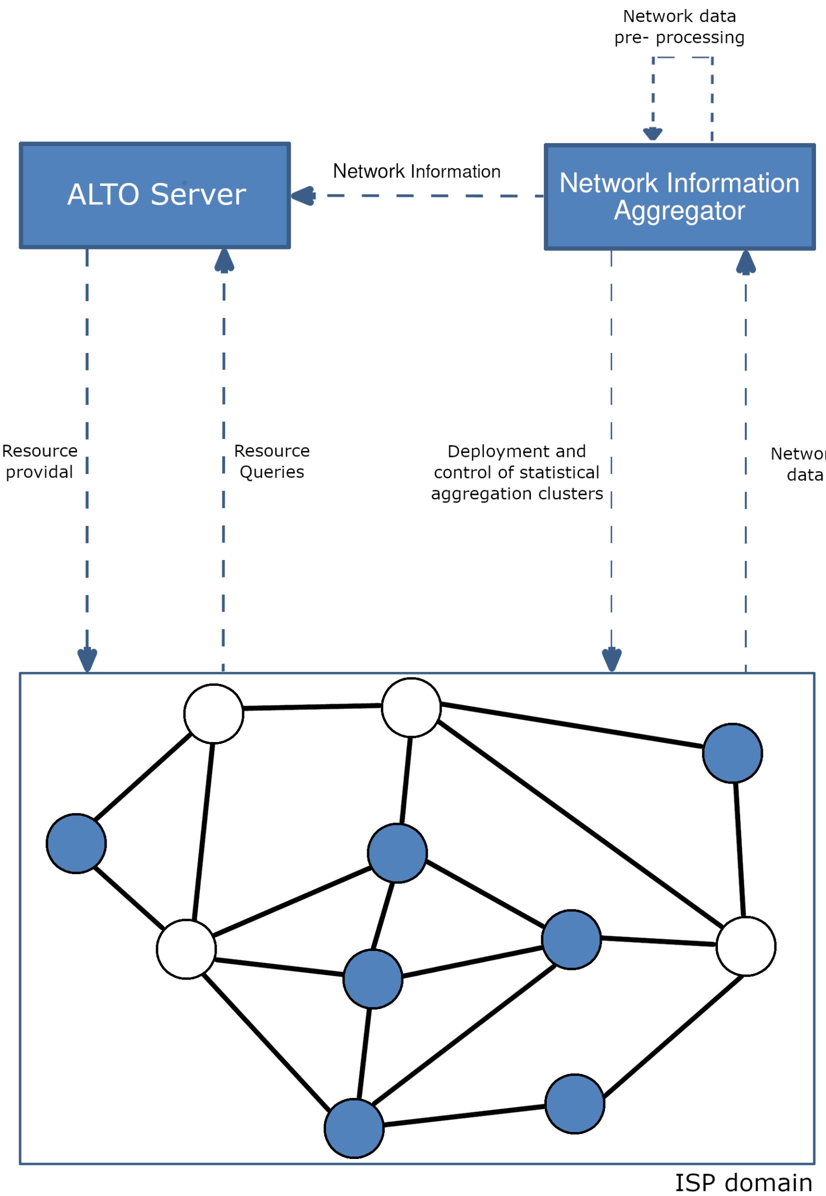


Figure 1.1: Conceptual representation of the ALTO system of a given ISP

More formally, Figure 1.2 presents the proposed system architecture. One can identify the ALTO interface, which is logically separated in its download and upload components, as a key factor of the system, since it allows to bridge three different application layers - the ALTO resource consumer, the ALTO server, and the network information aggregator, to be further specified in the following sections.

The ALTO working group has extensively specified the ALTO protocol, which regards to resource querying, and the concrete implementation of this work will aim to comply to it. However, no resource provisioning protocol was, at time of writing, specified by the working group, nor was an interface been specified to allow network data to reach the ALTO server. It has been set as a work in progress, and the topic of network information sources was briefly discussed in [71]. The working group has grouped the tasks of raw network processing and supply into the role of the ALTO server. However, as seen in the aforementioned architecture in Figure 1.2, a different approach was taken in this work, with the roles being separated and an additional protocol proposed to bridge communication between them. This was made as an attempt to adhere to the philosophy of single responsibility, making the sole task of the ALTO server the management of ALTO resources. This aims to facilitate the independent development of the different roles, and make it easier to interchange implementations, which would make it particularly useful, for example, to deploy many ALTO servers in a cascade fashion whilst utilizing only a single network information aggregator. These are, however, only conceptually separated, and an implementation could, if it is more practical, merge the server and information provider roles into a single physical entity, mimicking would then be similar to the architecture designed by the ALTO working group. A more distributed approach is then presented as an option, where multiple servers are separately deployed. To permit the data interchange and synchronization between multiple of these servers, an Inter-Domain Synchronization server is also included in the architecture as an centralized bridging entity between domains.

As most software architectures, each new communication channel represents a possible source of attack vectors and, attending to the critical security concerns posed in Section **??**, all of these channels must be secure and reliable, as signified by the padlocks on the presented architecture. This implies that data communications within it must be block being read or altered by non authorized users, and the identity of the participating parties can be trusted and made accountable. The identified communication channels must then have methods of maintaining data integrity in transit, user authentication and authorization, and communication confidentiality.

architecture-macro2.png

Figure 1.2: System architecture at a macro level

1.2 ROLE SYSTEM

As an access control measurement, the system will work with *Role-Based Access Control (RBAC)* methods which, as the name states, center their control policy logic around roles, which themselves are tags that can be attributed to users. A pre-requisite is then that users attempting to access a system employing RBAC must be authenticated as a given user, and from then a list of attributed roles can be retrieved to validate if a given action is permitted according to the set rules. The ALTO resources have associated to each of them an *Access-Control List (ACL)*, that maps, for a given set of roles, the list of user actions that are allowed to be performed to that resource, with the implicit rule that a resource's owner has full clearance. The available user actions are "read", "update" and "delete", meaning the ability to get, change the contents of, or remove the resource, respectively. This ACL must be provided by the Network Information Aggregator whenever a new resource is inserted into the ALTO server. The ISP administrator that controls the aggregator not only then designs the resource itself - adding the information that it deems important whilst not too detailed to damage privacy - but also defining access control policies on that resource, which will be then enforced by the server in future requests.

Employing access control based on roles seems appropriate for this system since roles can be applied to - and thus group - many users, and indeed that seems to be applicable on real case deployments of the ALTO system, where each given application, that consists of a great number of users, can correspond to a single group, and more private scenarios, such as a data center server cluster, can also be grouped. This facilitates permission management, as the RBAC approach allows grouping of permissions into roles, which can then be quickly manipulated to affect every user associated to it. This would contrast to an approach where permissions are set per user, which would be considerably harder to manage at scale. As a user can be granted many roles, he can naturally act on the system with a role that fits the currently queried resource, if so applies, and likewise the network administrator can give permissions per role, which in turn can group as many as millions of users, or to just a single one.

An RBAC-based access control mechanism will help mitigate security threats pertaining to the ALTO working group's architecture ,e.g. having unwanted users reading or tampering with data. However, for such mechanisms to be viable at all, authentication systems need to also be employed to help verify that the users are indeed who they are announcing to be. Authentication mechanisms are, regardless, of extreme importance, as they additionally help mitigate spoofing security threats. Data breaches are not, however, totally mitigated with authenticity and access control mechanisms. After an entity gets a resource and acts outside the system, it becomes out of its control and these mechanisms cannot be employed. This means that there are no guarantees that the resources are shared outside of the system's domain and consequentially there are no security guarantees after that point. Because of this, privilege attribution by the ISP administrators not only give clearance to do a certain action, but also imply that trust exists that these users will not be improper with the given resources, such as sharing it with users with improper clearance.

Figure 1.3 provides a high-level communication diagram of how access control is enforced. The ISP administrator that uploads the resource into the ALTO server appends to it an ACL that maps actions to the considered roles, with the implied meaning that those that weren't considered have no permitted actions. When a resource consumer requests an action, which is expected to be a "read" one, and proper authentication was performed to verify its identity, the server checks that the roles associated with that consumer have the requested action allowed in the ACL and, if indeed that is the case, the action performs as expected.

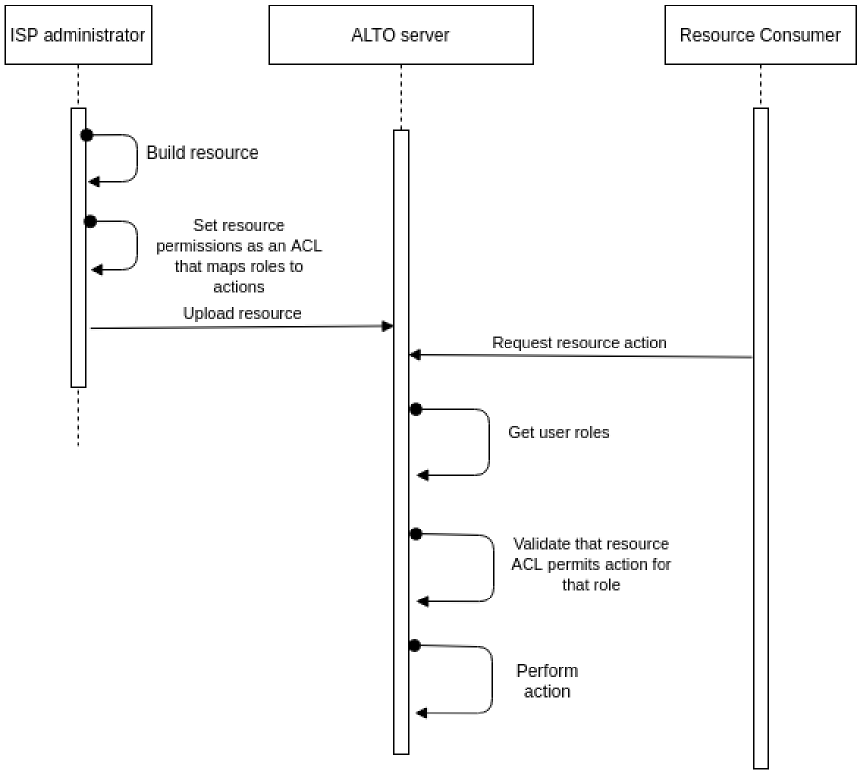


Figure 1.3: High-level communication diagram of a successful resource action request

1.3 RESOURCES

ALTO resources are pieces of network information which are provided by an ALTO server and consumed by ALTO clients that ideally would use such information to aid their application-level traffic decisions. All ALTO resources can be separated into the following components:

talk about sco

- **Meta information**: Data which regards to the resource's profile, that enable the client's ability to interpret and cross-reference the network data within. Meta information contains the resource's name, and if applicable, version, resource dependencies and cost details: enclosed cost modes, metrics, and descriptions. Finally, also belonging to the meta section of the resource's information is the resource's ACL which, to a given set of roles, specifies the allowed set of actions.

- **Network status information**: Data structures that give a characterization of the ALTO Server's vision of a network. Concretely, these can map network properties to a node - such as the connection types of their interfaces, or their geographical location - they can aggregate many network addresses to a single identifier, or they can map properties to a node link or end-to-end path, such as link or cumulative routing costs.

Meta information can be seen as a resource's header, containing data that regards to the network status and helps better handle it. Following the defined protocol [68], this field includes the resource's name for all resources, which is needed for identification and indexing, and all other fields are dependant on the type of resource: at this version of the protocol, only network maps are version-able, allowing ISPs to reference different versions of a network map as they are updated, maintaining support for previously referenced versions; cost information is, naturally, only applicable to cost maps, and gives insight on how the numeric costs are to be interpreted, i.e. what their mode and metrics are, and what description it has. Finally, extending to the protocol is the addition of an ACL as a solution to access control needs. An ACL is defined as a matrix, with each entry defining a user role and actions - discussed in Section 1.2 - as a restriction on what a given user was given clearance to do.

The network status information of a network map groups endpoint addresses into a single PID as a text literal. Akin to the working group's protocol, accepted endpoint address protocols include *Internet Protocol version 4 (ipv4)* and *Internet Protocol version 6 (ipv6)*, utilizing a 32 bit long bitmask to identify a subnetwork. Similarly, support for aggregation of *Media Access Control (MAC)* addresses was added, with a 48 bit long bitmask to identify address ranges, similar to the *Internet Protocol (IP)* variant. Additionally, generic overlay IDs can be added with the key "priv:X" - with "priv" meaning private scheme - where "X" is the qualified name - this naming scheme was adapted from the endpoint property map's specification done by the ALTO working group, for semantic consistency. As endpoint addresses utilizing this scheme aren't restricted to any type, their interpretation is also left to the client. For example, if a server defines that an endpoint addresses with "priv:my-overlay" can use regular expression to specify address ranges, a pre-agreement must exist with a client. Of course, if a given addressing scheme besides the previously mentioned ones becomes of relevant wide appeal, it could afterwards become part of the specification, but the existence of a private addressing scheme with liberal type and semantic verification gives liberties outside of the protocol for network status supply schemes that aren't supported officially. A valid network map must unambiguously map every address in the domain range to a single PID, and whenever multiple matches occur, wins the longest prefix match. As the custom addressing schemes let the network map be interpreted in an undefined way by the protocol, the server cannot properly assert to the matching validity, and thus default protocol addressing schemes for network maps should be preferred, as semantic validity in private addressing schemes is not checked. Table 1.2 provides an example network status component of a network map within the topology in Figure 1.4. Three PIDs are given, each taking portion of an ipv4, ipv6, MAC, and custom overlay address range. The private address scheme groups users in regards to their private overlay ID, and it can be seen that nodes with ID 1, 3, and 4 are grouped to a single PID, which can be seen to belong inside the ISP domain. Lastly, nodes 2 and 5 are given different PIDs as they reside outside the domain but are reachable through different peering points. The ISP could then in this case leverage the network map to logically group collections of endpoints by reachability - those local to their domain, and those reachable by one of the two possible peering points, which could be subjected to different peering agreements and as such should be treated differently in resources that reference this network map.

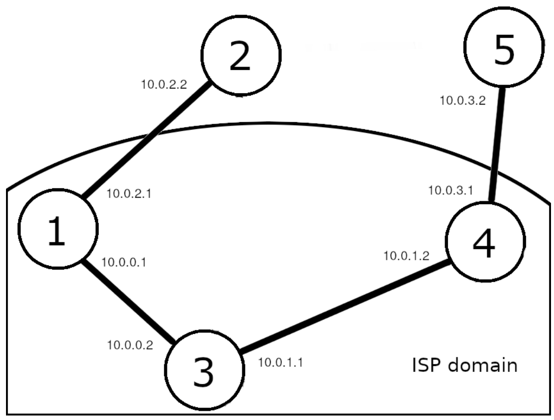


Figure 1.4: Example network topology with ISP boundary

PID	IPv4	IPv6	MAC	priv:my-overlay
1	[10.0.0.0/24,10.0.1.0/24, 10.0.2.1/32, 10.0.3.1/32]	-	D0-9F-BF-2A-00-00/32	[1, 3, 4]
2	[10.0.2.2/32]	-	D0-9F-BF-2A-FE-00/40	2
3	[10.0.3.2/32]	-	F8-BB-0B-0A-AA-AA/40	5
4	0.0.0.0/0	::/0	00-00-00-00-00-00/0	*

Table 1.2: Example network map referencing Figure 1.4

A cost map contains a list of cost map matrices, with each matrix setting pairwise values between an origin entity and a destination entity. If it is a standard cost map, these entities are represented by PIDs that can be cross-referenced from a network map which this resource depends on, whereas if it is an endpoint cost map, these entities are endpoint addresses which, similar to network maps, include ipv4, ipv6, MAC and private endpoint types.

A given matrix must specify the type of cost represented with both their cost type and cost mode, with available options being the ones specified in the ongoing ALTO group's cost metric specification [69]. Optionally, a cost matrix can specify calendar information about that matrix - similar to the current work in [67] - which signifies that besides having single-value costs, which are obligatory for any cost matrix, it also contains a time-sensitive list of costs that must be interpreted according to the calendar information provided, and give a chronological overview of what the costs will be in the future. If the ISP contains full topological knowledge of the resource it is sharing, the information that can be provided by the cost maps can be quite detailed.

Table 1.3 presents a cost table refering to the topology in Figure 1.5. One cost matrix depicts a generic "routingcost" cost matrix, depicting routing preference as a shortest path map with a Dijkstra algorithm and hop count as its link cost, and another provides a "delay-ow" cost matrix, depicting expected one-way delay in milliseconds, as the cumulative calculation of known link delays.

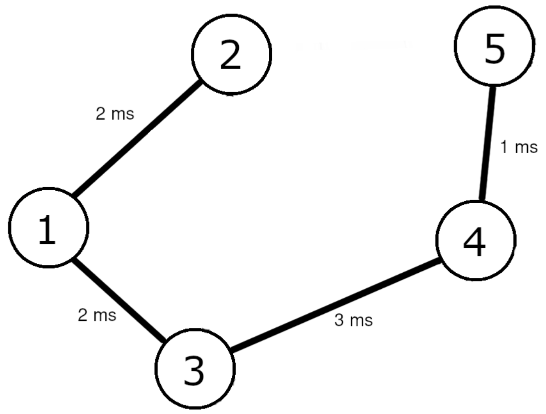


Figure 1.5: Example overlay network topology without ISP boundary

Cost Mode	routingcost				
Cost Metric	numerical				
From/To	1	2	3	4	5
1	0	1	1	2	3
2	1	0	2	3	4
3	1	2	0	1	2
4	2	3	1	0	1
5	3	4	2	1	0

(a) Routing cost cost matrix

Cost Mode	delay-ow				
Cost Metric	numerical				
From/To	1	2	3	4	5
1	0	2	2	5	6
2	2	0	4	7	8
3	2	4	0	3	4
4	5	7	3	0	1
5	6	8	4	1	0

(b) One way packet delay cost matrix

Table 1.3: Example cost map for overlay in Figure 1.4

Without either full administrative control or some multi-ALTO domain orchestration mechanism, a single ALTO server instance is restricted to the information it knows. Being bound by limited topological knowledge, however, does not necessarily mean that valuable inter-layer cooperation is not possible, and will now be subject of discussion. The network map presented previously contains in of itself important information, grouping endpoints into PIDs that represent two possible types of network borders: one local to the server, and two representing the peering relationships. This is relevant to help clients localize their traffic and would be impossible to derive without insight. Table 1.4 provides an example of cost matrices within a single cost map that consider a limited single ALTO server domain topology in Figure 1.4. Notice how the administrative domain is within scope of only three of the five network nodes. The ISP only possesses detailed network status information that regards to nodes "1", "3" and "4", which limits the amount of topological information that can be retrieved and shared to ALTO clients. However, it's still very much possible to dictate routing preferences and gaps in knowledge can be filled with probing measurements to be collected and centralized by the ALTO server to acquire historical performance metrics.

Cost Mode	routingcost				
Cost Metric	numerical				
From/To	1	2	3	4	5
1	0	10	1	2	22
2	-	-	-	-	-
3	1	11	0	1	21
4	2	22	1	0	20
5	-	-	-	-	-

(a) Routing cost cost matrix

Cost Mode	delay-ow				
Cost Metric	numerical				
From/To	1	2	3	4	5
1	2	20	1.5	3	42
2	-	-	-	-	-
3	4	24	2	2	39
4	7	27	2	2	36
5	-	-	-	-	-

(b) One way packet delay cost matrix

Cost Mode	tput				
Cost Metric	numerical				
From/To	1	2	3	4	5
1	256000	10000	256000	256000	5000
2	-	-	-	-	-
3	256000	10000	256000	256000	5000
4	256000	10000	256000	256000	5000
5	-	-	-	-	-

(c) TCP throughput cost matrix

Cost Mode	tput				
Cost Metric	ordinal				
From/To	1	2	3	4	5
1	1	2	1	1	3
2	-	-	-	-	-
3	1	2	1	1	3
4	1	2	1	1	3
5	-	-	-	-	-

(d) TCP throughput ranking cost matrix

Cost Mode	tput				
Cost Metric	numerical				
Calendar Start	Tue, 20 Sep 2020 17:00:00 GMT				
Calendar Interval size	7200				
Calendar Interval number	6				
From/To	1	2	3	4	5
1	(1,[1,1,1,2,2,1])	(2,[2,2,2,1,1,2])	(1,[1,1,1,2,2,1])	(1,[1,1,1,2,2,1])	(3,[3,3,3,2,2,1])
2	-	-	-	-	-
3	(1,[1,1,1,2,2,1])	(2,[2,2,2,1,2,1])	(1,[1,1,1,2,1,1])	(2,[2,2,2,1,1,1])	(1,[3,1,1,1,1,1])
4	(1,[1,1,2,2,1,1])	(1,[2,2,1,1,1,1])	(1,[1,1,1,1,2,1])	(2,[1,1,2,2,2,2])	(3,[3,3,3,3,2,2])
5	-	-	-	-	-

(e) TCP throughput cost matrix with calendar values

Table 1.4: Example cost map for the limited ISP domain in Figure 1.4

As can be seen in Table 1.4a, a generic "routingcost" cost matrix is presented, whose value increases with the associated costs of transferring data through that path, and constructed as the ISP best sees fit. Specifically to this case, costs within the ISP domain are minimal, whereas paths that originate locally and target "PID2" or "PID5", both requiring the utilization of peering links, are less preferable, with the former being at least twice more preferable than the latter.

A "delay-ow" cost matrix is also provided in Table 1.4b, specifying one way packet delay in milliseconds, with the ISP applying preceding probing measurements between endpoints and averaging the results as a means to fill the knowledge gap outside its domain.

Finally, a "tput" cost matrix can be seen in Table 1.4c and 1.4d, specifying expected throughput in a numerical fashion with a value of bytes per second, and in an ordinal fashion with a ranking, respectively. The ISP applied probing measurements, topological insight, as well as collected feedback of previous application connections that occurred between endpoints to deduce available bandwidth between target points in practice. The ordinal mode of displaying information serves as a way to preserve relative preference information without requiring from the the need to concretely specify network status, and instead ordering connections by relative preference, with the option of assigning equal preference to paths that differ in a given order of magnitude that the ISP sees as negligible.

Finally, the inclusion of cost calendar capabilities to the cost matrix in Table 1.4e enables users to get a chronological view of bandwidth availability at rush hours, with the single value cost being updated to the present time if a decision needs to be made only considering the current time.

In the presented example scenario, locality is correlated with more reliable communications and less operational costs from the ISP's point of view, and a concrete better choice exists regarding routing cost, delay, and throughput, between the two peering connections. That information can be part of the ALTO system as query-eligible by client applications that can now better optimize their network-related decisions in a mutually beneficial scenario.

The network status information of an endpoint property map stores the property information of a given endpoint. The ALTO working group's protocol specification [68] does not directly specify what kind of properties are pondered for this map. Following the same design pattern used for the other specified resources, the endpoint property map will have a set of defined properties with associated semantics, and all other properties can be added with the "priv" prefix to designate private properties outside of the considered domain, and thus all semantics and validation rules don't apply. Much like the other resources, an endpoint can be identified by an ipv4, ipv6, MAC or private overlay address, and the pondered properties are PID value, geographical coordinates, connection type (fiber, *Asymmetric digital subscriber line (ADSL)*, etc.), server footprint information (total *Random-Access Memory (RAM)*, *Central Processing Unit (CPU)*, and storage), and server status information (what portion of the footprint information is currently available, such as free processing power). In practice, a given property could be promoted from a private type to one pondered in the protocol and have a resulting official semantic and validation rules. Table 1.5 display an example endpoint property map, which is used to store status information relating to servers, identified by their ipv4 address, that serve the same content.

Endpoint	CPU %	RAM %	Geographic Coordinates	Connection type	priv:is-mirror
145.132.164.101	22	50	(34.28278,-82.50490)	Fiber	False
245.217.176.67	30	45	(23.24178,-53.51290)	Fiber	True
48.43.96.168	25	30	(55.33218,-12.50490)	Fiber	True
207.20.148.21	10	20	(-23.28121,-22.55530)	Fiber	True
89.140.253.77	5	0	(12.231278,75.70890)	Fiber	True

Table 1.5: Example endpoint property map for server replicas

Finally, as a means to facilitate resource divulgence from servers to clients, there is also included the specification of an *Information Resource Directory (IRD)*, that is also based from the ALTO working group's protocol specification [68]. An IRD can also be thought of as a resource, but instead of sharing network information, it serves as an index of the available resources that a given server provides. Each server must have available for query a single IRD, that lists all the available resources it provides, along with their metadata. Each resource attribute must contain the resource's ID, its *Hypertext Transfer Protocol (HTTP)* media type and, if applicable, their capabilities, accepted input media types, and resource dependencies. The capabilities identify, if existing, the cost and property types that are used. Being indexed by their unique name, this allows for these to be cross-referenced on further protocol exchanges without need to repeat information. Additionally, the resource's capabilities also serve to indicate what resource functionality extensions are enabled. These functionality extensions are currently applicable for cost maps only, and thus the capabilities serve to signal if the cost map has enabled one or more of the following functionalities: calendared costs, a protocol extension adapted from the work in progress in [67], that serves to retrieve calendar cost values; or the multi-cost extension functionality, a protocol extension adapted from [**?**], which lets multiple matrices be requested at once to save on overhead traffic that would otherwise be necessary to request many matrices.

Two additions are made to the working group's specification: firstly, a description field, which for each resource attribute gives a brief description of what it is about, as it could facilitate resource selection, since such a description could go into detail about appropriate usage guidelines of that resource and suggested use cases; secondly, the resource's ACL, letting a user know beforehand what clearances the given resource has.

A default network map entry must also exist in the IRD, as per the working group's specification, to serve as a guideline for clients that wish to use the most basic of ISP endpoint groupings.

An example IRD is provided in Table 1.6. A list of available costs and properties is shown in Table 1.6a and Table 1.6b, respectively, with their descriptive data discussed above, along with the available resources provided by that server in Table 1.6c, which contains data useful for their server cluster, as well as a broader-purpose endpoint cost map to query for path connection types and facilitate user selection. Finally, a default network map is included in Table 1.6d.

Cost ID	Cost Mode	Cost Metric	Description
routing	routingcost	numerical	Default routing preference
routing-rank	routingcost	ordinal	Routing preference by ranking
owd	delay-ow	numerical	Expected one way delay of a single packet. Based on application statistics
tput-theoretical	tput	numerical	Theoretical maximum available TCP throughput. Based on topological knowledge
tput-practical	tput	numerical	Practical expected TCP throughput. Based on application statistics

(a) Available cost types

Property ID	Property type	Description
cpu	CPU	Machine's current CPU load
ram	RAM	Machine's currently occupied RAM
coord	geographic-coordinate	Machine's geographical coordinates
connection	connection-type	Machine interface's connection type
is-mirror	priv:is-mirror	Flag stating if machine is a mirror of original server

(b) Available property types

Resource ID	URI	Media Type	Uses	Accepts	Capabilities	Description
def-nmap	resources/networkmaps/default	alto-networkmap	-	alto-networkmapfilter	-	Default
cluster-costmap	resources/costmaps/cluster	alto-costmap	def-networkmap	alto-costmapfilter	Costs: [routing, routing-rank]	For main data center cluster
cluster-endprop	resources/endpointpropmaps/cluster	alto-endpointprop	-	alto-endpointpropparams	Properties: [cpu, ram, coords]	For main data center cluster
client-endcost	resources/endpointcostmaps/	alto-endpointcost	-	alto-endpointcostparams	Costs: [routing-rank, owd, tput-practical]	For user application guidance

(c) Available resources

Resource ID
def-nmap

(d) Default Network Map

Table 1.6: Example of an ALTO server's IRD

Further formal specification is not made as it has been extensively done in the ALTO protocol [68], and the proposed system complies to it whilst extending upon the design.

1.4 ROLES

1.4.1 ALTO Client

An ALTO resource consumer is materialized in the architecture in the form of an ALTO client, which can be any entity who is able to interface with an ALTO server to query for ALTO resources. Whilst the ALTO working group was initially devised to help increase *Peer-to-Peer (P2P)*-related traffic localization via the sharing of network information, it now has an increased scope where an ideal client is any application which generates network traffic and would be able to optimize it with aid from an oracle entity with privileged network information. Thus, an ALTO client is fit to be implemented in P2P applications, and could be embedded in a P2P client itself to help with picking neighbouring and content providing nodes, or on a tracker that would accomplish the same goal on behalf of the querying peer. Likewise, nodes which are unable to optimally select between other nodes, such as *Content Distribution Network (CDN)* edge nodes or content mirrors, could also benefit from oracle guidance, and thus qualify as appropriate ALTO clients.

Figure 1.6 exemplifies how a cooperative P2P application would, acting as an ALTO client, interact with the ALTO server to retrieve relevant network resources to aid their application choice of what candidate peer to consume a service from. Firstly, a network map is retrieved to help group endpoints into groupings, and afterwards a cost map is retrieved filtering only the querying peer as source, candidate peers as destinations, and the routing cost and bandwidth cost matrices. Acting on this information, the peer chooses the candidate that gives a good balance between ISP routing cost and path bandwidth, making a decision that should ideally benefit both them and the ISP that helped provide that information.

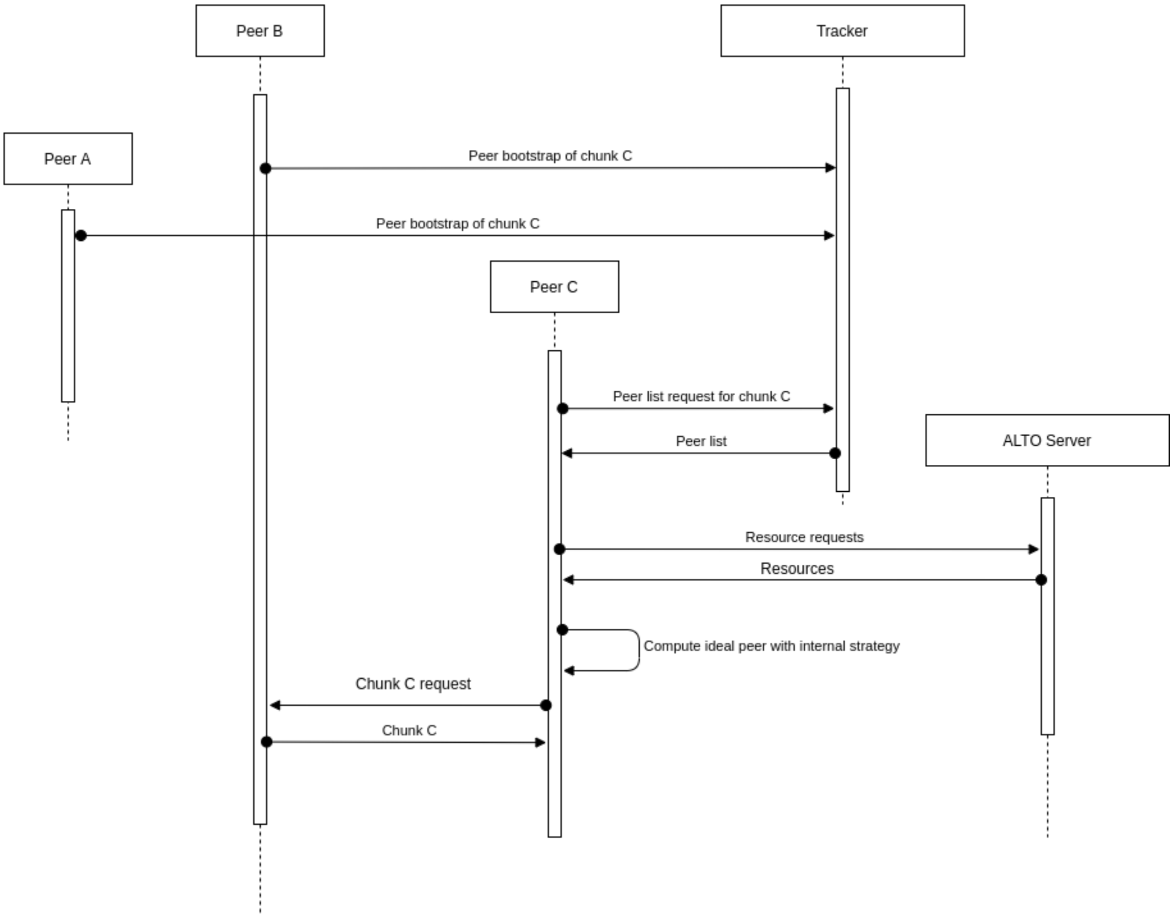


Figure 1.6: High-level communication diagram of a P2P application utilizing ALTO

Figure 1.7 is similar to the previous example, in the sense that it aids a P2P application by resorting to ALTO's guidance, but this time the application-level traffic optimization is made in a way that is transparent to the P2P client. As a choice to purely localize traffic, as this alone can bring plenty of benefits to both layers, and as a means to minimize protocol modification, it is the tracker that acts as an ALTO client. Whenever a request is made by a P2P client to retrieve peers serving a given data chunk, the tracker first consults with the ALTO server and retrieves its network map that groups peers within administrative domains: either inside the providing ISP's domain, thus the local network, and outside administrative domains, grouped by types of peering connections to different autonomous regions. The tracker could use a very simple algorithm to filter out of its candidate pool peers that reside outside of the ISP region where the requesting P2P client resides, if a local alternative exists.

After packaging a reply to the P2P client, the protocol acts normally and traffic could be successfully localized with minimal impact.

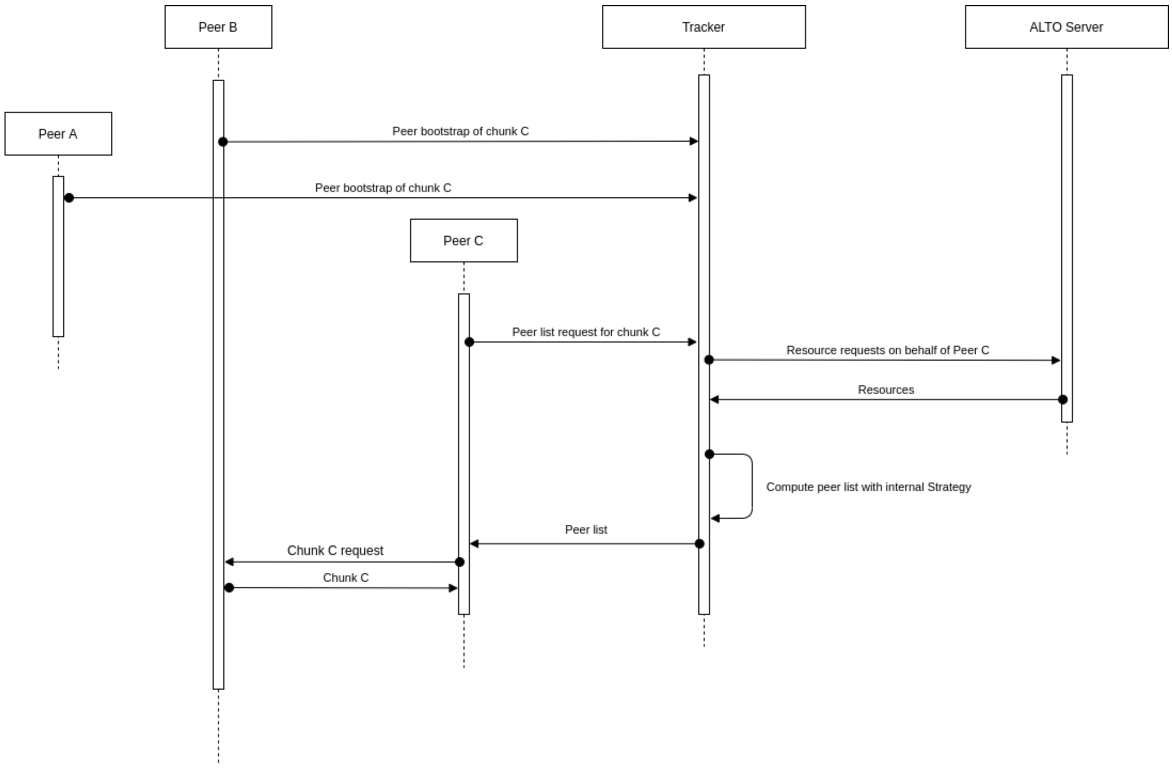


Figure 1.7: High-level communication diagram of a tracker utilizing ALTO by proxy

On the same vein, Figure 1.8 exemplifies how this time a CDN controller would use the system to better help its decision in matching CDN clients to an edge server on their system. To do this, it retrieves a property map to query for server status information, and subsequently retrieves a cost map to query for path information between the CDN client and the candidate edge servers. Having all the relevant server status information, e.g. available processing and storage resources, as well as connection properties, e.g. max possible bandwidth, latency, and packet loss, the CDN controller is in a condition to more optimally redirect his client.

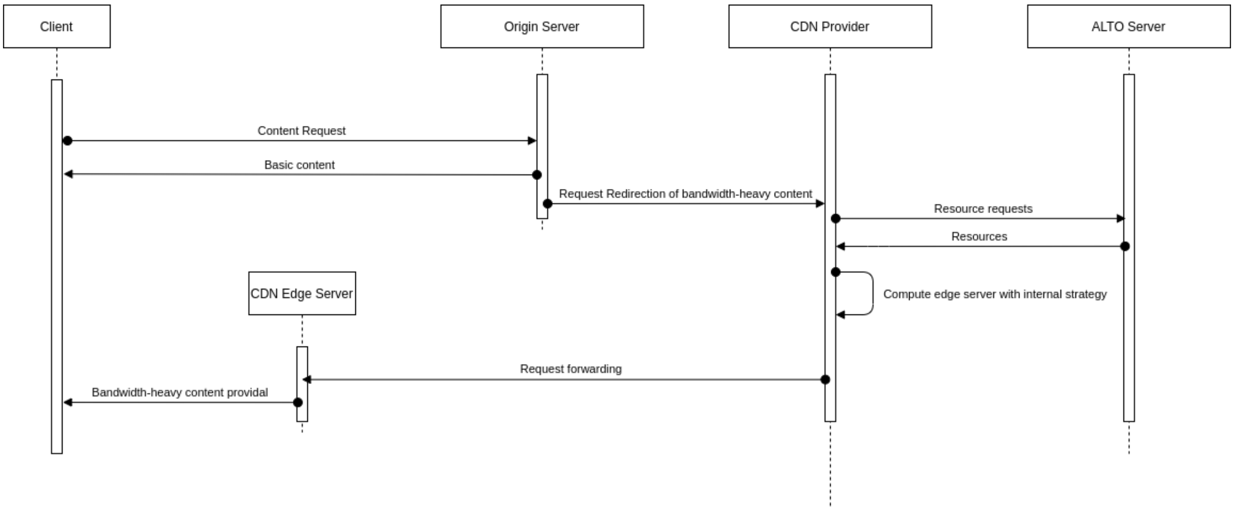


Figure 1.8: High-level communication diagram of a CDN controller utilizing ALTO

1.4.2 ALTO Server

An ALTO resource provider is the ALTO server, an entity that possesses pre-processed and authorized network information in the form of ALTO resources. Its job is to store and manage such resources so they can be provided to querying ALTO clients, with the additional responsibilities of data validation and persistence. Conceptually, the ALTO server is seen as a single entity, but considering the sensible information that could be stored within it and the influence it has on shaping network traffic, it would not be uncommon for an ALTO server to have a knowledge domain correspondent to the ISP that owns it. Physically, though, the resource provider layer could consist of many interlinked ALTO providers with an increased coverage area of network knowledge. Means through which this could occur are further specified in Section 1.4.2.3.

A listing of available HTTP endpoints of the ALTO server interface is available in Table 1.7. All resource types hierarchically descend from a "resources" path, and each unique type of resource exposes his own endpoint, with the methods to add, update, remove, and retrieve with and without filter as arguments. These methods are subjected to the access control mechanisms discussed in Section 1.2, as one should only expect reputable sources to upload and modify data, and only permitted users to query it.

HTTP Verb	Resource	Description
GET	/resources	Retrieve the Information Resource Directory for that server
GET	/resources/networkmaps	Get summary overview of all available network maps
POST	/resources/networkmaps/	Add a new Network Map
GET	/resources/networkmaps/{id}	Get the Network Map with the specified ID
POST	/resources/networkmaps/{id}	Get the Network Map with the specified ID, with the applied filter provided in body
PUT	/resources/networkmaps/{id}	Modify the contents of a Network Map with the specified ID
DELETE	/resources/networkmaps/{id}	Remove a Network Map with the specified ID
GET	/resources/costmaps	Get summary overview of all available cost maps
POST	/resources/costmaps	Add a new Cost map
GET	/resources/costmaps/{id}	Get the Cost Map with the specified ID
POST	/resources/costmaps/{id}	Get the Cost Map with the specified ID, with the applied filter provided in body
PUT	/resources/costmaps/{id}	Modify the contents of a Cost Map with the specified ID
DELETE	/resources/costmaps/{id}	Remove a Cost Map with the specified ID
GET	/resources/endpointpropmaps	Get summary overview of all available Endpoint Property Maps
POST	/resources/endpointpropmaps	Add a new Endpoint Property map
GET	/resources/endpointpropmaps/{id}	Get the Endpoint Property Map with the specified ID
POST	/resources/endpointpropmaps/{id}	Get the Endpoint Property Map with the specified ID, with the applied filter provided in body
PUT	/resources/endpointpropmaps/{id}	Modify the contents of an Endpoint Property Map with the specified ID
DELETE	/resources/endpointpropmaps/{id}	Remove an Endpoint Property Map with the specified ID
GET	/resources/endpointcostmaps	Get summary overview of all available Endpoint Cost Maps
POST	/resources/endpointcostmaps	Add a new Endpoint Cost Map
GET	/resources/endpointcostmaps/{id}	Get the Endpoint Cost Map with the specified ID
POST	/resources/endpointcostmaps/{id}	Get the Endpoint Cost Map with the specified ID, with the applied filter provided in body
PUT	/resources/endpointcostmaps/{id}	Modify the contents of an Endpoint Cost Map with the specified ID
DELETE	/resources/endpointcostmaps/{id}	Remove an Endpoint Cost Map with the specified ID

Table 1.7: ALTO server's available endpoints

1.4.2.1 *Resource Filtering*

Resource filtering is the task through which a resource consumer can pass a filter object to the resource provider that specifies the parameters that the consumer wishes to retrieve specifically. With it, there's no need to pass more information to the client than he wishes to get, thus minimizing used network bandwidth and client CPU cycles to send and process the resource, respectively. The need for filtering becomes greater at a larger system scale - with an increased number of users that query routinely, and resources that can have a massive amount of entries - specifically those that regard to network endpoints - such a mechanism becomes a necessary optimization. Keeping with the objective of implementing a fully compatible ALTO protocol as specified by the working group, so will the resource filtering specifications be equal to those already specified in the protocol design [68], thus no further specification will be necessary.

For clarification, the ALTO server must maintain an endpoint for retrieving all main specified network status resources via filtering, i.e. the server must let the client retrieve ALTO resources with parametrization that dictates what concrete fields must be delivered. The types of resource filters considered are the following:

- **Network Map filter**: List of of value PID_List, that if empty signifies the entire subset of available . All entries of value:

$$(PID, Endpoint\_List), \ PID \in PID\_List$$

must be retrieved, and all others must be filtered out.

- **Endpoint Property Map filter**: Pair of list of endpoints and list of properties that must be selected, of value $(Endpoint\_List, Property\_List)$. All entries of value

$$(Endpoint, Property), \ Endpoint \in Endpoint\_List \wedge Property \in Property\_List$$

must be retrieved, and all others must be filtered out.

- **Cost Map filter**: Tuple of list of source , list of destination , list of cost types and list of cost value conditionals of value

$(SrcPID\_LIST, DstPID\_LIST, CostType\_List, CostValueConditional\_List)$,

with $CostType\_List > 1$ assuming a multi-cost map extension and the emptiness of any of the lists signifying the entire subset of available values. All entries of value

$$\begin{gathered}(Src\_PID, Dst\_PID, Cost\_Type, Cost\_Value), \\ Src\_PID \in SrcPID\_List \wedge Dst\_PID \in DstPID\_List \wedge Cost\_Type \in \\ CostType\_List \wedge satisfies\_atleast\_one(Cost\_Value, CostValueConditional\_List)\end{gathered}$$

must be retrieved, and all other must be filtered out.

- **Endpoint Cost Map filter**: Equal to the cost map filter, but considering a list of source and destination endpoint filters instead of .

By analogy, one can consider the ALTO server to act as a remote database with an interface for clients to interact with it, and the filters act as selection statements, such as the "SELECT" method in *Structured Query Language (SQL)* databases, to retrieve specific parts of the dataset. The filters of cost maps and endpoint cost maps also include a list of premises which themselves are logical operators applied to the candidate cost values. Continuing the analogy, these allow the clients to use "WHERE" statements on the numerical cost values that are retrieved. In summary, the filter functionality could be explained in the examples shown in Table 1.8.

HTTP Verb	Resource	Body	Description

POST	resources/networkmaps/default	{
		"pids": ["PID1", "PID2"]
		}
			Retrieve an network map with id "default", filtering the entries for "PID1" and "PID2"

POST	resources/endpointpropmaps/default	{
		"endpoints": ["ipv4:10.0.0.1", "ipv4:10.0.0.2"],
		"properties": ["CPU", "RAM", "connection-type"]
		}
			Retrieve an endpoint property map with the id "default", filtering the entries for the properties "CPU", "RAM", and "connection-type" and the ipv4 endpoints "10.0.0.1" and "10.0.0.2"

POST	resources/costmaps/default	{
		"cost-type": {
		"cost-mode" : "numerical",
		"cost-metric": "routingcost"
		},
		"pids": {
		"srcs" : ["PID1"],
		"dsts": []
		}
		}
			Retrieve the cost map with the id "default", filtering the entries for the cost matrix with cost mode "numerical" and cost metric "routingcost", whose entries have source PID "PID1" and any destination

POST	resources/costmaps/default	{
		"multi-cost-types":
		[
		{
		"cost-mode" : "numerical",
		"cost-metric": "routingcost"
		},
		{
		"cost-mode": "ordinal",
		"cost-metric": "routingcost"
		}
		],
		"pids" : {
		"srcs": ["PID1", "PID2"],
		"dsts" : ["PID3"]
		}
		}
			Retrieve the cost map with the id "default" and, assuming a multi-cost protocol extension, retrieve both the "numerical" and "ordinal" variations of the "routingcost" metric, whose entries have source PID "PID1" or "PID2", and destination "PID3"

POST	resources/costmaps/default	{
		"multi-cost-types":
		[
		{
		"cost-mode" : "numerical",
		"cost-metric": "routingcost"
		},
		{
		"cost-mode": "numerical",
		"cost-metric": "delay-ow"
		}
		],
		"calendared": [false, true],
		"pids" : {
		"srcs": [],
		"dsts" : ["PID3"]
		}
		}
			Retrieve the cost map with the id "default" and, assuming a multi-cost and cost calendar protocol extensions, retrieve the numerical variants of the "routingcost" and "delay-ow" metrics, requesting a singular value and a calendar value respectively, of all entries whose destination is "PID3"

POST	resources/costmaps/default	{
		"multi-cost-types":
		[
		{
		"cost-mode" : "numerical",
		"cost-metric": "delay-ow"
		},
		{
		"cost-mode": "ordinal",
		"cost-metric": "routingcost"
		}
		],
		"pids" : {
		"srcs" : ["PID1"],
		"dsts" : ["PID2"]
		},
		"or-constraints" : {
		[
		["[0] ge 0","[0] le 20"],
		["[1] eq 1"]
		]
		}
		}
			Retrieve the cost map with the id "default" and, assuming a multi-cost protocol extension, retrieve the "numerical" mode of the "ow-delay" metric and the "ordinal" mode of the "routingcost" metric, whose entires have source "PID1" and destination "PID2", and whose cost values satisfy for a given source and destination pair that either the "ow-delay" is within 0 and 20 ms or it's the number one preferencial "routingcost" value

Table 1.8: Example ALTO queries with the filtering functionality

1.4.2.2 Server discovery

ALTO server discovery, by hand of either network information aggregators or resource consumers, must be done leveraging existing *Domain Name System (DNS)* technologies. Each server entity maintains a given domain name, and along it is included the need to also maintain domain records in the chosen DNS system to map the domain name to the server's IP address. Much like the choice to utilize HTTP as an application protocol, so do the server discovery mechanisms aim to comply with the ALTO working group's philosophy of leveraging existing proven technologies when possible as a means to facilitate development and minimize errors, with the added benefit of extending functionality with the chosen technologies since it is mature and has plenty of options. With DNS, this gives flexibility of either privately configuring domain name to IP addresses - much like happens in Linux systems with the "/etc/hosts" file - or its deployment by leveraging existing authoritative DNS servers. Additionally, by working around the existing technology and its specification, one can easily implement load balancing for performance reasons, or, among others, *DNS over HTTPS (DoH)* for preventing eavesdropping, and ensuring both data integrity and host authenticity [**?**]. A similar approach is to be taken for network information aggregator server discovery by part of the network state collectors for the same reasons explained above.

1.4.2.3 Inter-server communication

A glaring gap in the working group's base ALTO protocol is its single administrative applicability domain. Meaning, an ALTO server is managed by a single administrative entity - likely an ISP - and its knowledge domain is limited by the network topology details that the entity knows, which is a subset of the entirety of the Internet's infrastructure. In the attempt to fix the server's inability to provide network status information outside its domain, this section overviews mechanisms that enable inter-server communications as a means to expand the capabilities of each domain.

Firstly, consider how efforts for full resource synchronization could be taken. These would be similar to data synchronization mechanisms employed by popular databases to ensure consistency across several server replicas, and could increase availability as well as the serviceability of content nearby clients. However, it does not seem to fit this use case - for starters, if all data were to exist redundantly on all servers, that would defeat the purpose of having many administrative domains and thus a single server architecture would suffice; secondly, the architecture is inherently designed to work within a trust domain of selected clients and, because of it, the servers may not even be comfortable with sharing all of its information within other domains to begin with, limiting replication strategies; thirdly, accounting for the amount of users acting on the ALTO system, better scalability could be achieved with a distributed solution that limits information within set boundaries. Accounting for these reasons, an inter-server synchronization protocol was designed for servers to negotiate information exchange among themselves, as opposed to one that enabled the synchronization of a single monolithic dataset between servers.

It is very important that a multi-domain solution assures that each ISP has full sovereignty over their domain. Simply collecting data from each domain and storing it in an third-party entity from which the original source has no control fails to comply to the sovereignty requirement. Each ISP that participates in a multi-domain knowledge system must, at any time, have control over what the information is and who gets access to it, being able to retroactively change its content and the access control policies, respectively. This solution will then consider that each ALTO server belonging to the multi-domain orchestration mechanism will compute and store it's data locally,and will have an interface open for data querying from other ALTO servers, being then open to selecting and enforcing their own policies as they see fit, in regards to how and when data is calculated, and who gets access to it.

To achieve the stated requirements, a "scope" property will be added in the "meta" field of every ALTO resource. This property, like the name implies, states the scope of the resource - a "local" resource is no different from those in the base protocol, meaning that it gets calculated and distributed within a single administrative ALTO domain; a "global" resource is essentially virtual, in the sense that it is presented as if it were stored in the server, but instead is dynamically retrieved every time through inter-server communication. The decision for this property to be visible in the IRD was made so it is communicated to the client that a given property can be retrieved as a multi-domain effort - and as such is susceptible to a clash of different ISP policies and strategies - or as a locally bound resource that will be less prone to inconsistencies, as only a single domain is responsible for managing it.

To manage the synchronization of globally-scoped ALTO resources, a new entity, the domain synchronizer, was specified. This central server will too be an ALTO server, meaning that it will be used to store and manage ALTO resources, although its use is specialized for inter-server synchronization. The server maintains property maps, where each of them refers to globally and locally scoped resources, and contains the addresses of the servers that independently store that data and make it available for querying. Locally-scoped resource IDs are stored in this server with the addresses of the server's that host it, so that a given ALTO server, whenever asked for a resource which he does not posses, can contact this synchronization database and appropriately redirect a client to a server that contains that information. In contrast, globally-scoped resource IDs and their owner's addresses are also stored in this server, so that a given server can know who he must query to retrieve the needed information to locally build the resource to be delivered to the client.

Listing **??** shows example data of an endpoint property map for inter-server synchronization. Figure **??** shows the required steps taken by a given ALTO server when a client requests for data - in this case an endpoint property map and a cost map - that is globally scoped.

The required *Application Programming Interface (API)* endpoints that an ALTO server must provide for inter-server sharing of resources is no different from the ones they already must provide for clients. The only difference is that that these resources contain ACLs that dictate exclusive action access by authenticated ALTO servers.

1.4.3 Network State Provider

1.4.3.1 Network Information Aggregator

The network information aggregation layer is the layer that enables the translation of raw topological information - such as the physical attributes of network devices and connections - into processed, query-eligible network knowledge. To do so, a very important entity, perhaps the heart of the system as a whole, is the network state collector, which is the supply of network information that is injected, through a network state provisioning protocol, into a network information aggregator. This latter entity is then responsible for providing the ALTO resource provider layer with valid information after the raw topological data has been processed - this includes the calculation of optimal paths, the abstraction of network entities, or the injection of static ISP preferences. This pre-processing stage requires input from an ISP administrator, responsible for acting on the best interest of the ISP from which the raw topological data originates - by interacting with the network information aggregator, the administrator acts on this network information hub to retrieve from the database a history of retrieved network information, and afterwards manipulate this information to create ALTO resources to its liking - this is where data is transformed utilizing the algorithms the administrator deems fitting, and transforms the raw data to be publishing ready, meaning that it contains an acceptable amount of abstraction not to compromise topological privacy. Finally, the administrator defines important meta data that identifies the resource, and defines the access control list to be enforced by the ALTO server.

1.4.3.2 Network State Collector

Before ALTO resources are provided into the ALTO server by the Network Information Aggregator, the latter needs himself to be provided with raw network status information. The ALTO working group has discussed possible sources of raw topological information, including protocols like *Interior Gateway Protocol (IGP)*, *Border Gateway Protocol (BGP)*, SNMP, or *Network Configuration Protocol (NETCONF)*, or databases like the *Traffic Engineering Database (TED)* or *Label Switched Path Database (LSPD)* [71]. A protocol needs to exist to interface between the entities that collect and provide the raw topological data, and the Network Information Aggregator that processes it and provides it to the ALTO server.

The available endpoints supported by the Network Information Aggregator server are presented in Table 1.9.

HTTP Verb	Resource	Description
POST	/measurements/endpoint	Add a measured endpoint property value
PUT	/measurements/endpoint/{id}	Modify the contents of a measured endpoint property value
DELETE	/measurements/endpoint/{id}	Remove a measured endpoint property value
POST	/measurements/links	Add a measured link value
PUT	/measurements/links/{id}	Modify the contents of a measured link value
DELETE	/measurements/links/{id}	Remove a measured link value
POST	/measurements/group	Add a measured endpoint grouping
PUT	/measurements/group/{id}	Modify the contents a measured endpoint grouping
DELETE	/measurements/group/{id}	Remove a measured endpoint grouping

Table 1.9: Network Information Aggregator's available endpoints

An illustrative example on how certain Network State Collectors of given network data could use this endpoint to interface with the Network Information Aggregator is presented on Figure 1.9

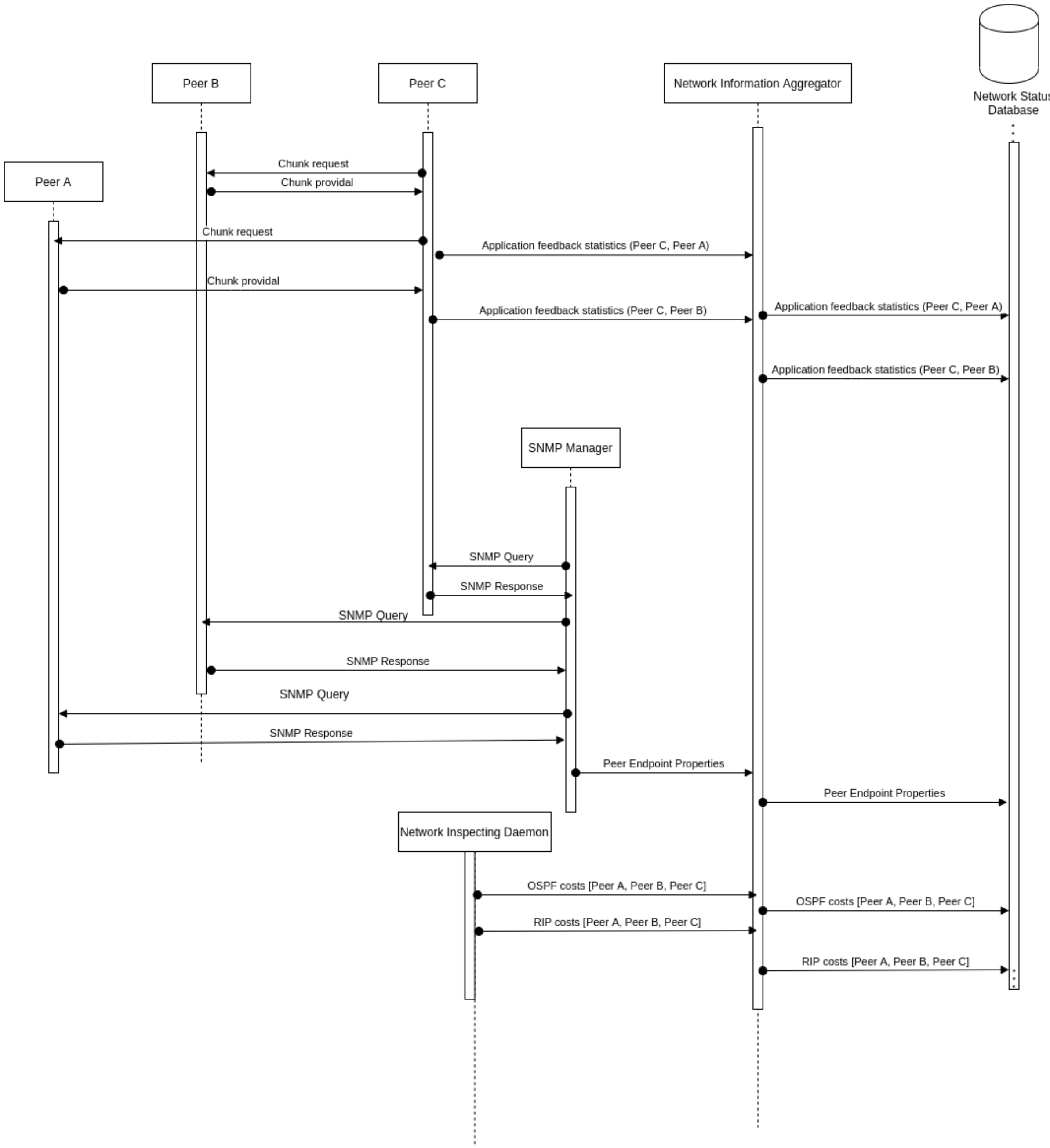


Figure 1.9: Communication diagram of how external network state providers upload information to the network information aggregator

[Network is provided raw, as is, without validation, but validation modules could be provided in the future if a need for it exists. The uploaded information firstly has meta data - which includes the source name (where the data came from, such

as an OSPF collector daemon), the time where the measurements were collected, a description, an endpoint or group of endpoints, depending on what the measurement relates to - such as an endpoint property or a cost between two properties or a grouping between N properties, and the measurement itself.]

1.4.3.3 Network Status processing

[Detail how the ISP uses the information gathered by the network state providers and pre-processes it. This includes, for example, calculating shortest path maps utilizing a Dijkstra algorithm, limiting/changing information to maintain security, adding static ISP policies, and attributing access control policies for that resource]

Creating a user-friendly network data constructor would be a good future work

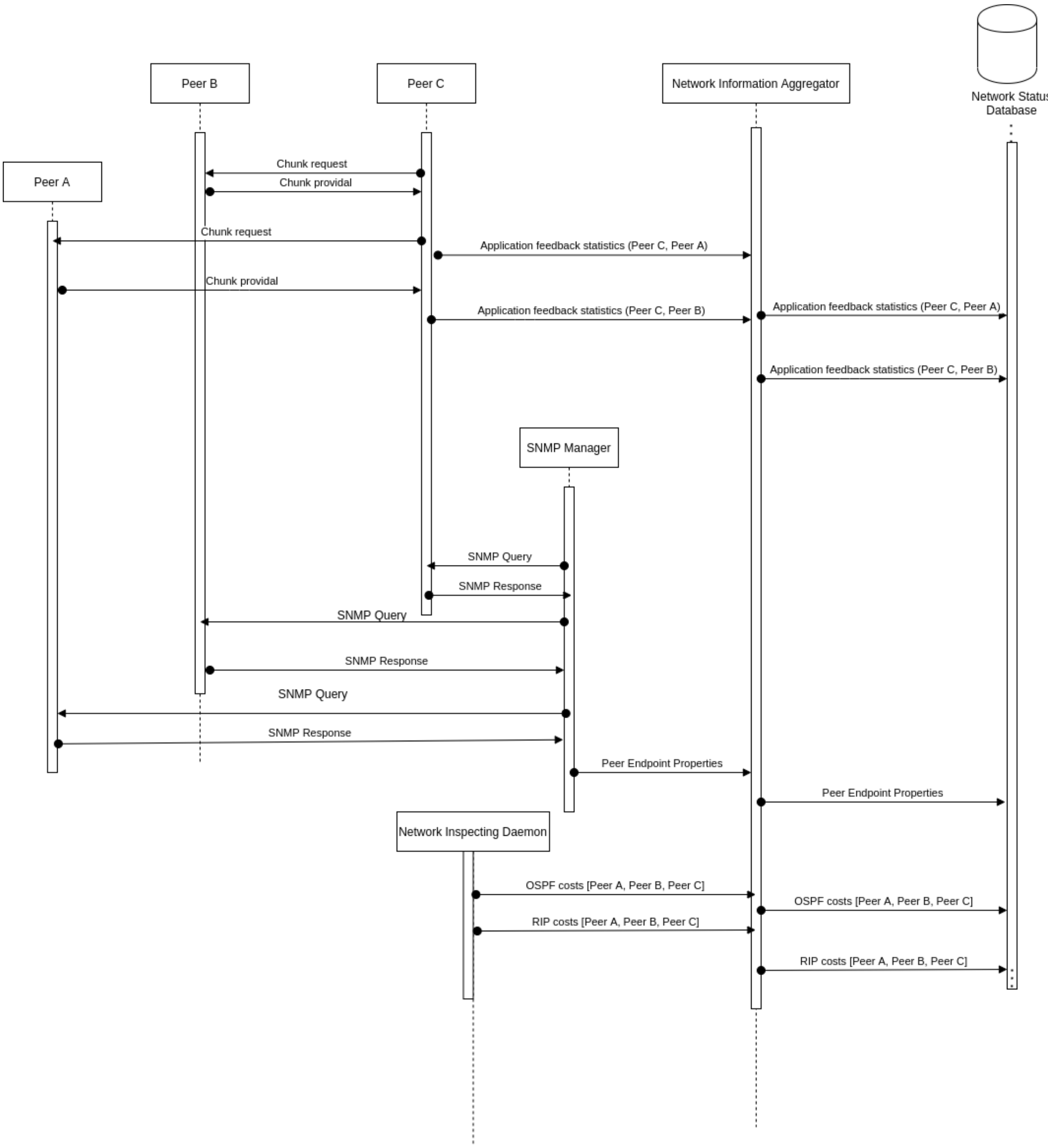


Figure 1.10: Communication diagram of how an ISP administrator pre-processes the gathered network state and uploads it to the ALTO server

BIBLIOGRAPHY

[1] Active network intelligence solutions | sandvine. https://www.sandvine.com/. Accessed: 2020-05-20.

[2] Bittorrent.org. http://bittirrent.org/beps/bep_0003.html. Accessed: 2020-05-20.

[3] The global internet phenomena report. Technical report, 9 2019.

[4] Ppstream. http://pps.tv/. Accessed: 2020-05-20.

[5] Akamai. https://www.akamai.com/. Accessed: 2020-09-20.

[6] Erik Nygren, Ramesh Sitaraman, and Jennifer Sun. The akamai network: a platform for high-performance internet applications., 01 2010.

[7] J. Liu, S. G. Rao, B. Li, and H. Zhang. Opportunities and challenges of peer-to-peer internet video broadcast. *Proceedings of the IEEE*, 96(1), 2008.

[8] Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*, 36, 12 2004.

[9] Eng Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys Tutorials, IEEE*, 7:72– 93, 04 2006.

[10] Regarding gnutella - gnu project - free software foundation. https://www.gnu.org/philosophy/gnutella.en.html. Accessed: 2020-05-20.

[11] Wikipedia Commons. The gnutella search and retrieval protocol. https://en.wikipedia.org/wiki/Gnutella#/media/File:GnutellaQuery.JPG. Accessed: 2020-05-20.

[12] Napster. https://www.napster.com/. Accessed: 2020-05-20.

[13] Freenet. https://freenetproject.org/. Accessed: 2020-05-20.

[14] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1), 2003.

[15] The free haven project. `https://www.freehaven.net/overview.html`. Accessed: 2020-05-20.

[16] Eytan Adar and Bernardo A. Huberman. Free riding on gnutella. *First Monday*, 5, 2000.

[17] Claudio Fiandrino. P2p system topology. `https://texample.net/tikz/examples/p2p-topology/`. Accessed: 2020-06-04.

[18] Q. Liao, Z. Li, and A. Striegel. Is more p2p always bad for isps? an analysis of p2p and isp business models. In *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, Aug 2014.

[19] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. An empirical evaluation of wide-area internet bottlenecks. *ACM SIGMETRICS Performance Evaluation Review*, 31, 05 2003.

[20] Bram Cohen. Incentives build robustness in bittorrent. *Workshop on Economics of PeertoPeer systems*, 6, 06 2003.

[21] F. Qin, J. Liu, L. Zheng, and L. Ge. An effective network-aware peer selection algorithm in bittorrent. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Sep. 2009.

[22] J. H. Wang, D. M. Chiu, and J. C. s. Lui. Modeling the peering and routing tussle between isps and p2p applications. In *200614th IEEE International Workshop on Quality of Service*, 2006.

[23] Thomas Karagiannis, Pablo Rodriguez, and Konstantina Papagiannaki. Should internet service providers fear peer-assisted content distribution? 01 2005.

[24] Vinay Aggarwal, Stefan Bender, Anja Feldmann, and Arne Wichmann. Methodology for estimating network distances of gnutella neighbors. 01 2004.

[25] György Dán, Tobias Hossfeld, Simon Oechsner, Piotr Cholda, Rafal Stankiewicz, Ioanna Papafili, and George Stamoulis. Interaction patterns between p2p content distribution systems and isps. *IEEE Communications Magazine*, 49, 05 2011.

[26] Al-Mukaddim Khan Pathan, Rajkumar Buyya, and Design Computing. A taxonomy and survey of content delivery networks. 2006.

[27] Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, and Dan Mackin. *UNIX and Linux System Administration Handbook (5th Edition)*. Addison-Wesley Professional, 5th edition, 2017.

[28] Netflix. `https://www.netflix.com`. Accessed: 2020-09-20.

[29] Youbtube. `https://www.youtube.com/`. Accessed: 2020-09-20.

[30] Cloudflare. `https://www.cloudflare.com/`. Accessed: 2020-09-20.

[31] Cloudfront. `https://aws.amazon.com/cloudfront/`. Accessed: 2020-09-20.

[32] M. Wichtlhuber, J. Kessler, S. Bücker, I. Poese, J. Blendin, C. Koch, and D. Hausheer. Soda: Enabling cdn-isp collaboration with software defined anycast. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, June 2017.

[33] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. Pushing cdn-isp collaboration to the limit. *SIGCOMM Comput. Commun. Rev.*, 43(3):34–44, July 2013.

[34] Raghunath Deshpande. Overview of cdn-isp collaboration strategies. 07 2014.

[35] At&t. `https://www.att.com/`. Accessed: 2020-09-20.

[36] Orange. `https://www.orange.com/`. Accessed: 2020-09-20.

[37] Swisscom. `https://www.swisscom.ch/`. Accessed: 2020-09-20.

[38] Kt. `https://corp.kt.com/`. Accessed: 2020-09-20.

[39] Lu Liu and Nick Antonopoulos. *From Client-Server to P2P Networking*, pages 71–89. Springer US, 2010.

[40] Linux mint. `https://linuxmint.com/`. Accessed: 2020-09-20.

[41] Zahra Elngomi and Khalid Khanfar. A comparative study of load balancing algorithms: A review paper. In *International Journal of Computer Science and Mobile Computing*, pages 448–458, 06 2016.

[42] Mei Chin, Chong Eng Tan, and Mohamad Bandan. Efficient load balancing for bursty demand in web based application services via domain name services. 01 2010.

[43] Xiaohui Yang. Sdn load balancing method based on k-dijkstra. *International Journal of Performability Engineering*, 14, 04 2018.

[44] Why you should switch to a different linux mint mirror today! `https://unlockforus.com/why-you-should-switch-to-a-different-linux-mint-mirror-today/`. Accessed: 2020-01-03.

[45] Pedro Sousa. *Context Aware Programmable Trackers for the Next Generation Internet*, volume 5733, page 78. 2009.

[46] Pedro Sousa. A framework for highly reconfigurable p2p trackers. *Journal of Communications Software and Systems*, 9(4):236, dec 2013.

[47] Pedro Sousa. Towards effective control of p2p traffic aggregates in network infrastructures. *Journal of Communications Software and Systems*, 11:37–47, 04 2015.

[48] D. Hughes, I. Warren, and G. Coulson. Agnus: the altruistic gnutella server. In *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*, 2003.

[49] T. N. Kim, S. Jeon, and Y. Kim. A cdn-p2p hybrid architecture with content/location awareness for live streaming service networks. In *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, June 2011.

[50] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1190–1199 vol.3, 2002.

[51] Vinay Aggarwal and Anja Feldmann. Locality-aware p2p query search with isp collaboration. *NHM*, 3, 06 2008.

[52] K. Han, Q. Guo, and J. Luo. Optimal peer selection, task assignment and rate allocation for p2p downloading. In *2009 First International Workshop on Education Technology and Computer Science*, volume 1, March 2009.

[53] M. L. Gromov and Y. P. Chebotareva. On optimal cdn node selection. In *2014 15th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, June 2014.

[54] Ben Niven-Jenkins, François Le Faucheur, and Dr. Nabil N. Bitar. Content Distribution Network Interconnection (CDNI) Problem Statement. RFC 6707, September 2012.

[55] P. Francis, S. Jamin, Cheng Jin, Yixin Jin, D. Raz, Y. Shavitt, and L. Zhang. Idmaps: a global internet host distance estimation service. *IEEE/ACM Transactions on Networking*, 9(5):525–540, 2001.

[56] T. S. E. Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 170–179 vol.1, 2002.

[57] I. Poese, B. Frank, B. Ager, G. Smaragdakis, S. Uhlig, and A. Feldmann. Improving content delivery with padis. *IEEE Internet Computing*, 16(3):46–52, 2012.

[58] Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Steve Uhlig, and Anja Feldmann. Content-aware traffic engineering. *CoRR*, abs/1202.1464, 2012.

[59] K. Mase, A. Tsuno, Y. Toyama, and N. Karasawa. A web server selection algorithm using qos measurement. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, volume 8, June 2001.

[60] M. Swain and Young-Gyun Kim. Finding an optimal mirror site. In *Proceedings. IEEE SoutheastCon, 2005.*, April 2005.

[61] André Sampaio and Pedro Sousa. An adaptable and ISP-friendly multicast overlay network. *Peer-to-Peer Networking and Applications*, 12(4):809–829, September 2018.

[62] Jan Seedorf, Sebastian Kiesel, and Martin Stiemerling. Traffic localization for p2p-applications: The alto approach. 10 2009.

[63] Application-layer traffic optimization (alto). Technical report, 11 2019.

[64] Jan Seedorf, Y. Richard Yang, Kevin J. Ma, Jon Peterson, Xiao Shawn Lin, and Jingxuan Jensen Zhang. Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO.

Internet-Draft draft-ietf-alto-cdni-request-routing-alto-08, Internet Engineering Task Force, November 2019. Work in Progress.

[65] Luis M. Contreras, Danny Alex Lachos Perez, and Christian Esteve Rothenberg. Use of ALTO for Determining Service Edge. Internet-Draft draft-contreras-alto-service-edge-01, Internet Engineering Task Force, July 2020. Work in Progress.

[66] Kai Gao, Young Lee, Sabine Randriamasy, Y. Richard Yang, and J. (Jensen) Zhang. ALTO Extension: Path Vector. Internet-Draft draft-ietf-alto-path-vector-11, Internet Engineering Task Force, July 2020. Work in Progress.

[67] Sabine Randriamasy, Y. Richard Yang, Qin Wu, Deng Lingli, and Nico Schwan. Application-Layer Traffic Optimization (ALTO) Cost Calendar. Internet-Draft draft-ietf-alto-cost-calendar-21, Internet Engineering Task Force, March 2020. Work in Progress.

[68] Sebastian Kiesel, Wendy Roome, Richard Woundy, Stefano Previdi, Stanislav Shalunov, Richard Alimi, Reinaldo Penno, and Y. Richard Yang. Application-Layer Traffic Optimization (ALTO) Protocol. RFC 7285, September 2014.

[69] Qin Wu, Y. Richard Yang, Young Lee, Dhruv Dhody, and Sabine Randriamasy. ALTO Performance Cost Metrics. Internet-Draft draft-ietf-alto-performance-metrics-08, Internet Engineering Task Force, November 2019. Work in Progress.

[70] Jan Seedorf and Eric Burger. Application-Layer Traffic Optimization (ALTO) Problem Statement. RFC 5693, October 2009.

[71] Martin Stiemerling, Sebastian Kiesel, Michael Scharf, Hans Seidel, and Stefano Previdi. Application-Layer Traffic Optimization (ALTO) Deployment Considerations. RFC 7971, October 2016.

[72] The stride threat model. `https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN`. Accessed: 2020-09-20.

[73] Barbara Van Schewick. Network neutrality and quality of service: What a non-discrimination rule should look like. Technical report, 6 2012.

[74] Federal communications commission. `https://www.fcc.gov/`. Accessed: 2020-09-20.

[75] Regulation (eu) 2015/2120 of the european parliament and of the council. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32015R2120&rid=2#d1e445-1-1. Accessed: 2020-09-20.

[76] Plusnet. https://www.plus.net/. Accessed: 2020-09-20.

[77] Nate Anderson. Deep packet inspection meets 'net neutrality'. https://arstechnica.com/gadgets/2007/07/deep-packet-inspection-meets-net-neutrality/2/. Accessed: 2020-09-20.

[78] Meo. https://www.meo.pt/. Accessed: 2020-09-20.

[79] Tarifários móveis pós-pagos unlimited. https://www.meo.pt/telemovel/tarifarios/unlimited. Accessed: 2017-12-14.

[80] Facebook. https://www.facebook.com/. Accessed: 2020-09-20.

[81] Spotify. https://www.spotify.com/. Accessed: 2020-09-20.

[82] Wikipedia zero. https://en.wikipedia.org/wiki/Wikipedia_Zero. Accessed: 2020-09-20.

[83] Wikipedia. https://en.wikipedia.org. Accessed: 2020-09-20.

[84] Lily Hay Newman. Net neutrality is already in trouble in the developing world. https://slate.com/technology/2014/01/net-neutrality-internet-access-is-already-in-trouble-in-the-developing-world.html, 1 2014. Accessed: 2020-25-10.

[85] J. Domzal, R. Wójcik, and A. Jajszczyk. Qos-aware net neutrality. In *2009 First International Conference on Evolving Internet*, pages 147–152, 2009.

[86] Danny Alex Lachos Perez, Christian Esteve Rothenberg, Qiao Xiang, Y. Richard Yang, Börje Ohlman, Sabine Randriamasy, Farni Boten, Luis M. Contreras, J. (Jensen) Zhang, and Kai Gao. Supporting Multi-domain Use Cases with ALTO. Internet-Draft draft-lachos-alto-multi-domain-use-cases-01, Internet Engineering Task Force, July 2020. Work in Progress.