



Universidade do Minho

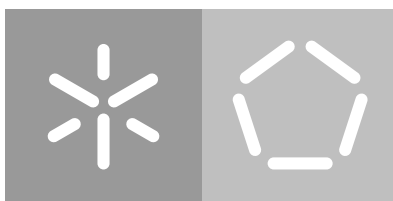
Escola de Engenharia

Departamento de Informática

Paulo Edgar Mendes Caldas

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

October 2020



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

Masters degree in Informatics Engineering

Dissertation supervised by

Pedro Nuno Miranda de Sousa

October 2020

ACKNOWLEDGEMENTS

ABSTRACT

With the ever-increasing global Internet usage that is following the start of the new decade with no intentions of stopping, the need to optimize this world-scale network of computers becomes a big priority in the technological world, as user demands increase and so do the Quality of Service (QoS) demands for applications in domains such as media streaming or virtual reality. In the topic of optimizing the Internet, one main concern regards itself to traffic that is generated at the application level. Peer-to-peer (P2P) applications are a popular example of application types which are classically unfavored by ISPs due to their unpredictable and uncontrollable nature that utilizes network resources less efficiently, resulting in higher costs and non-achievable standards of QoS. One main issue with these applications is the fact that these act with little consideration of the underlying network infrastructure on which they operate, with a popular example of this being the lack of care by P2P applications on the network proximity to the peers among whom they select for neighbouring relationships.

This thesis aims to implement and extend upon the ideas of the Application-Layer Traffic Optimization (ALTO) working group, which devised a request-response protocol where privileged and trustworthy entities provide to applications information that regard to the underlying network structure where such applications run on, with the intent of achieving layer cooperation during normal application operations as a means to achieve better Internet harmony so it can better be prepared to the needs of the present and the future whilst minimizing operational costs.

RESUMO

isto

CONTENTS

1	ACKNOWLEDGEMENTS	i
2	INTRODUCTION	1
2.1	Context and motivation	1
2.2	Objectives	3
2.3	Contributions	4
2.4	Thesis organization	4
3	STATE OF THE ART	5
3.1	Peer-to-Peer (P2P) Networks	5
3.1.1	Concepts and Applications	5
3.1.2	Architecture	7
3.1.3	Effects to the Network Infrastructure	9
3.2	Content Distribution Networks (CDNs)	13
3.2.1	Concepts and applications	13
3.2.2	Architecture	14
3.2.3	Effects to the Network Infrastructure	15
3.3	Server mirroring	16
3.3.1	Concepts and applications	16
3.3.2	Effects to network infrastructure	17
3.4	Traffic optimization by applications and layer-cooperative approaches	18
3.5	Application-Layer Traffic Optimization (ALTO) working group	23
3.5.1	Context and Motivation	23
3.5.2	Architecture	25
3.5.3	Issues and challenges	28
4	SYSTEM ARCHITECTURE AND DEVELOPED MECHANISMS	31
4.0.1	System Architecture	31
4.0.2	ALTO resources	34
4.0.3	Network intelligence provision	35
4.0.4	Network intelligence preprocessing	35
4.0.5	Multi ALTO server communication	35
5	IMPLEMENTATION	38
5.0.1	Technologies used	38

5.1	Optimizations	38
6	EXPERIMENTS	39
6.1	setup	39
6.2	Results	39
6.3	Summary	39
A	SUPPORT MATERIAL	43

LIST OF FIGURES

Figure 1	Examples of structured P2P systems that utilize DHTs	10
Figure 2	Example demonstration of an overlay network and corresponding physical layer.	11
Figure 3	high level architecture of a CDN (adapted from [1])	15
Figure 4	Linux Mint prompt to select a software repository mirror	17
Figure 5	Existing approaches to decrease tension between P2P applications and ISPs ([8])	21
Figure 6	ALTO scenario [21]	24
Figure 7	ALTO architecture (adapted from [12])	26
Figure 8	ALTO services (adapted from [12])	27
Figure 9	Conceptual representation of the ALTO system of a given ISP	32
Figure 10	System architecture at a macro level	36
Figure 11	Schematization of the pondered ALTO resources	37

LIST OF TABLES

Table 1	Types of P2P systems (Adapted from [25])	8
Table 2	Network node entities in the conceptual ALTO system representation	32

ACRONYMS

ALTO Application-Layer Traffic Optimization.

AS Autonomous System.

CAN Content Addressable Network.

CDN Content Distribution Network.

CDNI Content Distribution Network Interconnection.

DHT Distributed Hash Table.

DoS Denial of Service.

IETF Internet Engineering Task Force.

IGP Interior Gateway Protocol.

ISP Internet Service Provider.

MPLS Multiprotocol Label Switching.

MTR Multi-Topology Routing.

P2P Peer-to-peer.

PID Provider-Defined Identifier.

QoE Quality of Experience.

QoS Quality of Service.

RTT Round-Trip Time.

SDN Software Defined Networking.

SQL Structured Query Language.

INTRODUCTION

2.1 CONTEXT AND MOTIVATION

As society as a whole advances, so does seem to increase the individual's quality of life, which in turn increases the standard to be expected from the society he lives in. As such, technology itself must quickly adapt to the needs of the people it serves, whichever they may be - educational, medical, logistical, just to name a few - and consistently create or improve upon solutions that inevitably change the day-to-day living of the many that use or reap the benefits of such solutions. A particular example that is still fresh in this generation is in the relationship between people and computers - where they may have been nonexistent a century ago, reserved for industries fifty years ago and valuable household commodity a few decades ago, it is now common to see a family home with more than a dozen computers, with a variety fitting for the many needs they can solve. The increased number of computers and their expected functionalities has made it so computer networking as a whole has to be improved upon.

The internet allows computers to connect to one another in a worldwide network that applications can use to further increase their possibilities. However, when applications go unchecked they can be very difficult to manage by creating traffic that is either impossible, unviable, or too costly for Internet Service Providers (ISPs). This issue is further increased when considering the scale of the next decade, where Cisco predicts that by 2022 global internet users will make up 60 percent of the world's population, and global IP traffic will reach 396 exabytes per month [4].

Peer-to-peer (P2P) applications are an infamous example of creating an overlay network with little information or regards to the underlying network it operates on. Historically, P2P traffic was always not preferable by ISPs due to its unpredictable and hard to manage nature. Even worse, the fact that the overlay is network agnostic made it frequent that overlay traffic was inefficient, and thus costly and usually prone to

creating network congestion [8]. Indeed, if P2P applications simply keep an overlay connection between peers that does not span more than a couple of hops, whilst ignorant to them being either direct network links or spanning multiple Autonomous Systems (ASs), the generated traffic is always at risk of being inefficient and taxing on the supporting infrastructure. As file-sharing traffic currently uses around 7 exabytes per month (including P2P based file-sharing) [4], and BitTorrent alone makes up 22 percent of total upstream volume of traffic [20] it's in the best interest of both ISPs and P2P applications a way for the overlay and underlay levels to operate in synergy.

Current consumer trends suggest that media consumption will make up a considerable part of global internet traffic. In fact, Cisco predicts that, by 2022, more than 82 percent of all consumer internet traffic will be dedicated to Internet video streaming and downloads and Content Distribution Networks (CDNs) will carry 72 percent of all internet traffic [4]. Thus, it will be a challenge for both ISPs and media applications to properly adapt to an increasing audience with ever higher quality of service (QoS) demands. Much like P2P applications, media-oriented applications and ISPs could also greatly benefit from a cooperative interaction - more specifically, in tasks of client redirection, whether that be to a CDN edge server or a server mirror. These optimizations should be made by the parties which have a monetary interest in guaranteeing good performance of the overall ecosystem, i.e. those acting on the over and underlay.

In short, the issue that motivates this thesis is the lack of proper cooperation between the overlay and underlay levels in the task of traffic optimization that originates at the application level, e.g., peer selection for file retrieval in file-sharing P2P applications, neighbour selection for P2P network creation, CDN provider server or cache redirection, etc. This problem is not new to the Internet engineering task force (IETF) who, as it realized that P2P applications that select peers based on exclusive network information provided by ISPs could reduce ISP costs as well as increase application download rates, devised a working group to explore possible IETF standardization on traffic localization [21]. The result was a request-response protocol by the same name, ALTO, where clients could query authoritative and trustworthy servers on information that regards to the underlay structure where the client operates. While P2P applications were the motivation for the ALTO working group to be created, the benefits of a standardized, stable, and well provided system for network information querying could help create the vision of ISPs and applications cooperating for mutual benefit, being thus advantageous for more than P2P applications - in essence, it would be a helpful system for any situation where a decision could be optimized with the addition of proper network insight.

2.2 OBJECTIVES

The main objective of this thesis is to develop a working system that adheres and expands upon the ALTO working group's protocol and architecture. The starting point will be a preexisting software project that served as a proof of concept to the strategy of traffic optimization at the application layer, and which will now be extended in two ways: firstly, by restructuring and documenting the existing code in order to, through the compliance with the standards of object oriented programming and software development guidelines, present a solution that could be continuously maintained and modified; secondly, by further expanding on the software's functionality, e.g., adding more types of cost metrics, specifying meta-data which give data a time-specific applicability, specifying means of synchronizing data among servers, limiting user interaction to data via access control methods, etc). While expanding upon the ALTO working group's devised solution is a goal, it is also important that the developed work complies with the specifications it is based on.

With the intent of completing its main goal, this work's partial objectives were devised as follows:

- Literature review in regards to application traffic optimization and the cooperation (or lack thereof) between overlay networks and the underlay they operate on. More specifically, an understanding on the current consensus in regards to the existing issues, and an overview on currently suggested solutions.
- Complete overview of the ALTO working group's proposed solution. More specifically, an overview of both their existing RFC documents and the currently active internet drafts being developed by them at the time of writing.
- Familiarization with the existing system to be worked on and definition of both a new system architecture which complies with and extends the ALTO solution, as well as the new function modules to be added and how they should operate.
- Implementation of both the devised solution as well as a bare-bones P2P file-sharing application for testing purposes.
- Construction of a realistic network simulation scenario where the P2P file-sharing application will operate in.
- Test of the implemented solution on the simulated scenario, and its analysis in comparison to preexisting strategies.

2.3 CONTRIBUTIONS

this

2.4 THESIS ORGANIZATION

This dissertation will be organized in six chapters, as follows:

- **Introduction:** Provides context to the problem to be attempted to solve, as well as motivation to attempt to do so. Coupled to this, the dissertation's main goal is presented.
- **State of the Art:** Display of the theory related to existing and popular technologies or overall concepts that could be targeted consumers of the ALTO protocol; Discussion of existing attempts to optimize application traffic using network information with and without close underlay cooperation; Overview of the ALTO working group's proposed protocol and architecture.
- **Specification:** Presentation of the devised system's functional and non-functional requirements, as well as an overview of the planned architecture.
- **Implementation:** Details to the decisions made and steps taken in the task of implementing the specified project.
- **Testing and result analysis:** Overviews the planned simulation scenario, how it was materialized, and how the tests were run. Additionally, provides the retrieved results from such simulations.
- **Conclusion:** Presents the results of this thesis in regards to what objectives were completed. Additionally, a critical analysis on the simulation results is made and argued against the initial hypothesis, arguments are made for the product's usefulness, and future work is suggested.

STATE OF THE ART

This chapter aims at providing a literature overview on the topics that are related to the main problem that this thesis aims to solve, which is the lack of communication between applications and the infrastructure where they reside. As such, the first section focuses on discussing means and structures through which nodes are organized in networks today, from which three were selected for their current popularity and their high potential for applicational traffic optimization : peer-to-peer (P2P) networks, content distribution networks (CDNs) and server-client networks that leverage server mirroring. For each of these, a conceptual analysis is made - more specifically, the context behind them, relevant concepts, architecture, and possible use cases. Additionally, there's an examination on how applications that utilize these networks affect, positively or negatively, the physical infrastructure where they operate, and if potential exists for mutually-beneficial layer cooperation. The following section displays existing proposals for increased layer cooperation, alongside a discussion of the practical consequences of adopting such proposals. The final chapter overviews the Application-Layer Traffic Optimization (ALTO) working group's proposal, which is a subset of the proposals of the previous section that contains more in-depth information, as it is the main inspiration of this thesis.

3.1 PEER-TO-PEER (P2P) NETWORKS

3.1.1 *Concepts and Applications*

Due to the many hybrid implementations that have surfaced, the definition of a P2P network has become harder to pinpoint. Nevertheless, a P2P network is grounded on some definitions, among them that it consists on many singular computing elements, the "peers", which have among themselves similar privileges and functions (this contrasts

with the client-server architecture, where two different roles exist - the one that is to be served and the one that is to serve - with functionality and control being thus centralized). P2P networks decentralize computational resources as a means to achieve certain tasks, and such resourcefulness of the entire system as a whole gives it an interesting list of properties, among them:

- **Dynamic scaling:** As all member nodes can share their computing resources with the network, the system as a whole increases its capacity with increased users. Since the peers also act as clients to the network, scaling the service becomes less of a challenge as each new client will also act as a server. This also removes the necessity to manage how many service resources are needed - the amount of existing resources is linked to the number of existing clients, and thus there's no need to purchase and manage central resources, as the network dynamically allocates them by nature.
- **Resilience to failure:** Whereas centralized solutions are vulnerable to node and link failure, a P2P network can more easily work around such threats - as all peers can encompass the same server functionality, network services and resources are not dependant on a limited set of nodes, but instead redundantly deployed throughout.
- **Power decentralization:** As a direct consequence of computational and resource sharing, no single peer has direct control of the network, and the information is not centralized. As such, this considerably deters any attempts to overpower the network, e.g., via means of censorship or biased node favouring.

These, however, are not without their nuance - since many P2P hybrids exist, these properties can change and others can appear. For example, if we consider BitTorrent, which has Tracker servers to redirect users to a correct peer with the requested resource, whilst the network itself can still be resilient to failure, the content-retrieval service that the P2P network provides has a single point of failure and of control - the trackers themselves. Furthermore, the P2P network design, by its nature, also has some issues to consider:

- **Security:** The equal functionality property that P2P networks have give much power to peers to affect others. Without proper care, malicious peers are a security risk.

- **Management:** Since resources and services are not centralized, tasks such as event logging and resource backups become very difficult, and perhaps impossible if the peers do not abide to any orchestration protocol.

P2P applications have had, in the past decades, a mainstream image that is plagued with legality and security issues. Nonetheless, when these are overcome the P2P networking strategy possesses many interesting properties - some of them displayed above - that make it fitting for varied use cases - e.g. file sharing, media streaming, social networking and problem solving via distributed and cooperative algorithms. More recently, P2P applications have been considered a fitting solution for low-cost content delivery systems in high demand scenarios, in applications such as PPStream and PPLive in China, which offer live video streaming through P2P with great success [4].

3.1.2 Architecture

As stated previously, the term "Peer-to-peer" has become very broad and now serves as an umbrella for many different sub-types of systems. This chapter focuses on overviewing the architecture of many of these sub-types. All P2P networks are characterized by consisting of peers that know one another as to form a so-called overlay network on top of its supporting network. How peers are organized in these P2P networks and how they operate is what distinguishes the many sub-types. Table 1 groups known P2P applications in regards to their centralization and structure, as did [25] and [14], with the latter further distinguishing the protocols in regards to other metrics, e.g., security, reliability, and performance. One would expect that all P2P applications would have no centralization at all, since the P2P design sees function and routing spread throughout the network. Alas, some modifications have been made in some of these sub-types, which shift how much decentralization they have. Similarly, different levels of structure are employed that shift the network's efficiency with how much structure is set in regards to its operation. As would be expected, these sub-types of P2P networks thus possess different strengths and weaknesses, and as such could be applied to different use cases.

Early versions of Gnutella come as a famous example of a decentralized and unstructured architecture, as peers act with equal functions and privileges, and no inherent structure exists on how these peers connect, store or retrieve content. The bootstrapping method consists on users reading from a set of known Gnutella peers, which is essen-

Table 1: Types of P2P systems (Adapted from [25])

		Centralization		
		Hybrid	Partial	None
Structure	None	Bittorrent, Napster, Publius	Kazaa, Morpheus, Gnutella (extension proposals), Edutella	Gnutella, FreeHaven
	In Infrastructure			Chord, CAN, Tapestry, Pastry
	In System			Bittorrent (DHT/Trackerless), OceanStore, Mnemosyne, Scan, PAST, Kademlia, Tarzan

tially a static list of addresses obtained from a trustworthy source, and attempting to connect to each one of them until a preferred number of known neighbours is reached. The unstructured nature of this protocol makes it so there's no systematic way to efficiently retrieve content, as thus peers must flood the network with content queries until either a reply is met or the predefined TTL value is exceeded.

The author defines partially centralized architecture as similar to those which are decentralized, but with the added caveat that some peers are chosen to service a portion of the network. This is done to take use from the fact that not all network peers are alike in terms of memory, computational power, or other relevant resources. As such, more capable peers are elected as "supernodes" and are delegated with more responsibilities, noting that these self-configure in situations where such supernodes fail or willingly leave the network, and thus there is no single point of failure as there would be on a true centralized architecture.

A hybrid architecture approach in a P2P network employs some elements from the client-server architecture. With Napster as an example, whilst peers still operate as servers or clients, they must contact an intermediary - and central - server when querying for content, which will in turn redirect them to one or many peers that contain it. A similar concept applies for BitTorrent, where such intermediary servers are called trackers. Obviously, the choice to add a centralized aspect to the architecture hinders many of the advantages from a purely unstructured solution - namely its scalability, resilience to failure, and decentralization of control - as a trade-off to facilitate the

control and maintenance of the network, as well as the peers' ability to bootstrap to it and locate content.

A P2P architecture is structured if it systematically employs some non-random criteria on how the network operates, e.g., how peers organize themselves, where content is stored and how it is retrieved - for example, FreeNet uses the content's hash as a key that is used to query for it, and which is used by the peers in each subsequent hop to know where to forward the request, instead of flooding the network in attempts to blindly find it. Many of the structured P2P architectures rely on distributed hash tables (DHTs), which are structured ways to map a key to some content in the network, in such a way that the full key-space is partitioned over the peers. Two examples of structured P2P architectures that use DHTs can be seen in Figure 1 - to the left, the Chord algorithm uses a circular DHT where each peer knows the location of some peers that are their predecessors, and some that are their successors. When a peer needs to query for some content, it uses its key to firstly search for it locally and, if not, forwards the query to following peers, and the process recursively continues. To the right, the content addressable network (CAN) has the key-space mapped to a virtual two dimensional space, and its area is partitioned to peers considering their geographical location. A straight arrow from querying node to the node that has the content represents the routing path that the querying message must travel: A-B-E.

Employing a systematic way to self-organize and share content is the means to guarantee that a P2P network can be fully decentralized whilst maintaining a desirable level of performance. However, the reliance on structure means that it must be maintained, e.g. managing neighbour pointers on Chord or managing area allocations on CAN, and that can be costly or even impossible with high rates of peer churn, i.e. with a sufficiently large rate of peers entering and leaving the network.

3.1.3 *Effects to the Network Infrastructure*

Historically, ISPs have deemed P2P traffic as unideal or even undesirable. Besides the aforementioned illegality precedent that is tied to P2P applications, the overall properties of P2P networks make them unappealing to support - due to the distributed nature of these types of networks, the overall traffic is less predictable, with the higher upload traffic volume in edge networks requiring infrastructural investments, and the network-agnostic operation mode of P2P applications leads to inefficient and uncooperative network resource usage.

make own v
images

liao2014

falar de arti

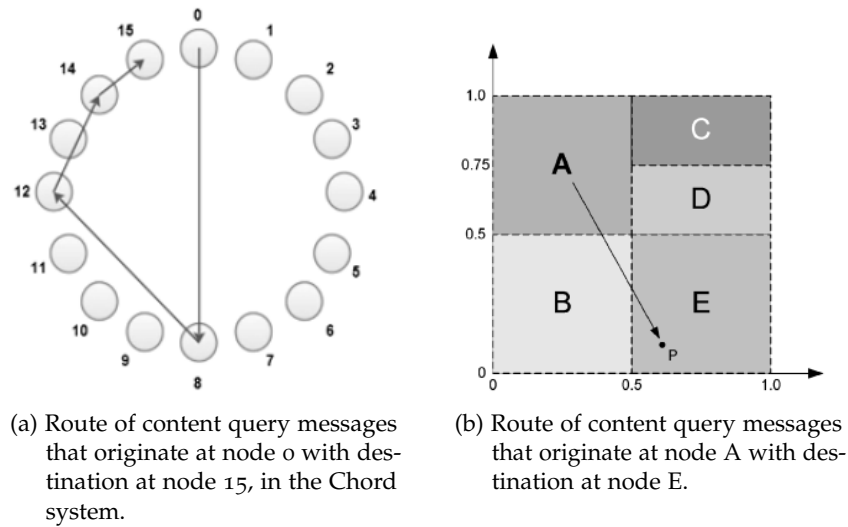


Figure 1: Examples of structured P2P systems that utilize DHTs

P2P networks who neither have structure nor a central point of control have to utilize content retrieval methods which are bound to be less efficient than their counterparts. However, architectures which fit in these categories mostly do so with a clear purpose - Gnutella's decentralized nature makes it very hard for individual nodes or external entities to regulate what can happen in the network (such as enforce legal actions), and its lack of structure simplifies the architecture and reduces the overall effort to bootstrap to the overlay, making it a good fit for applications with a high peer churn rate. Similarly, FreeHaven's architectural decisions fit a very specific use case, as it "emphasizes distributed, reliable, and anonymous storage over efficient retrieval" [?]. The lack of systematic means to efficiently locate content by these P2P architectures means that more ad-hoc methods have to be used, which are less efficient and thus incur in bigger workloads for ISPs - the usage of query flooding by Gnutella and message broadcasting by FreeHaven are examples of this.

The usage of structure by P2P networks can, as stated before, result in more efficient content and peer location algorithms. However, maintaining such structure also requires a chunk of ISP resources, as peers need to periodically update others, as well as react to peers entering and leaving the overlay. The usage of key-value mappings with DHTs is also ISP unfriendly as the hash function's purpose is to evenly distribute resources over the network - whilst such property is certainly advantageous in certain use cases, doing so removes any applicational context that exists in the content - for example,

grouping resources which belong to the same web page can't be done, as they will be individually hashed and spread out.

Regardless of the P2P system operating under structural means or using a central point of control, no classic P2P system operates in full cooperation with ISPs. The network-agnostic manner under which they operate results in overlay networks which are layered on top of the underlay where they run, as exemplified in Figure 2 - as P2P applications are network-agnostic, two neighboring peers could exist in completely different contexts on the common network layer - for example, both being physical neighbours or distancing many network providers are possible scenarios.

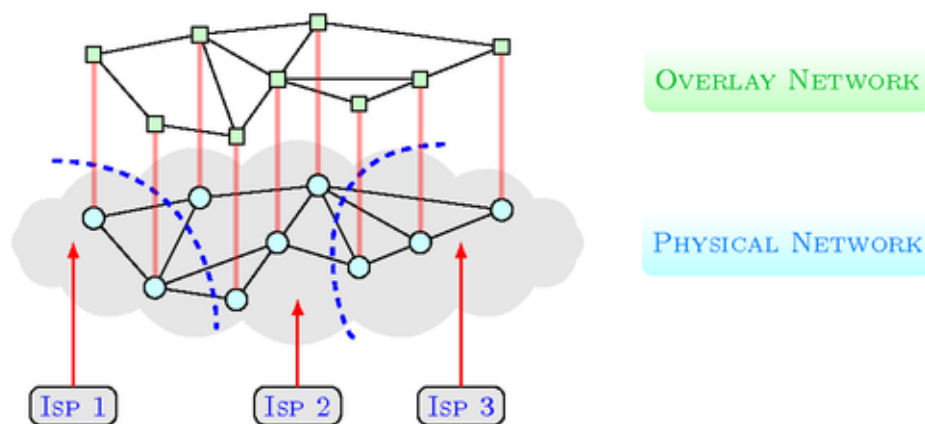


Figure 2: Example demonstration of an overlay network and corresponding physical layer.

In the theme of P2P applications not considering the underlying infrastructure emerges the issue of locality unawareness, i.e., the lack of knowledge on how much a given resource or nodes in the overlay actually distance themselves in the underlay. This issue is particularly damaging in P2P applications whenever neighbours are selected, and whenever a resource exists redundantly in the network and a decision must be made in regards to what file chunks to retrieve from what peers. If it indeed is the case that P2P applications are not locality aware, it can easily be seen how this can be an issue: for P2P applications, choosing peers which are not local to the querying peer may result in more time to retrieve the requested contents; for ISPs, bad network resources management can incur in higher operational costs, and may degrade overall network performance in many cases, e.g., overusing inter-AS links which usually are network bottlenecks (a conventional wisdom demonstrated in [7]). If the P2P application were to attempt to choose the peers that would most effectively serve the querying peer, it could depend on privileged information that the ISP has on the network's properties

Make my own
this

and current status (such as link load, scheduled tasks, or scheduled maintenance). Attempting to optimize peer selection without a co-operational channel with ISPs would be sub-optimal as not enough information is known, and could perhaps even be more damaging with the wrong techniques - consider a peer selection algorithm that chooses the peers with lowest RTT of a probing ping message, whilst having no indication on available end-to-end bandwidth. Likewise, attributing neighborhood via geographical proximity - much like the CAN architecture - whilst initially seems like a good step in location awareness, may also not be optimal - ISPs may not always prefer geographical proximity in connections, as peers could be very geographically close but residing in different ASs and thus separated by costly links. Other peer-selection techniques focus on randomly selecting nodes, which is simple and resilient to peer churn [18], but as a consequence is sub-optimal on network resource usage as no network consideration exists. It is fair to say that no P2P application can act with full ISP consideration without directly cooperating with it, and simple heuristics should be, whenever possible, traded for methods where full cooperation with the underlay is done - that is, if the needs of both layers are being considered.

Indeed, it is the case that current P2P solutions are ISP-unfriendly, resulting in large amounts of upstream and downstream traffic. To name a few examples, BitTorrent seems to employ peer selection algorithms which do not consider the underlay network, which can result in degraded download performance and increased load on ISPs [18]. [11] found that since this protocol is locality unaware, 70-90% of existing local content was found to be downloaded from external peers and suggests that locality-aware content distribution algorithms could significantly reduce the total amount of traffic generated; also, Gnutella generates traffic which is not ideal, as it may have to cross ISP network boundaries multiple times [6] due to the same fundamental issue stated before - an application layer that operates in disregard to the network underlay it runs on.

As [8] describes, the ongoing friction between the overlay and underlay layers has made it to the point where ISPs have chosen to throttle the bandwidth of P2P traffic, or even outright blocking it. In return, P2P applications have tried to mask their presence to bypass such restrictions via tunnelling or using non-standard and random port numbers. This is an unsustainable system that is bound to hurt both ISP profit and application functionality, and a strategy of cooperation between the overlay and underlay layers is crucial to guarantee that the requirements of both parties are met.

3.2 CONTENT DISTRIBUTION NETWORKS (CDNS)

3.2.1 *Concepts and applications*

A content distribution network (CDN), as the name implies, is a network specifically designed with its main focus on distributing content to a set of end users. Its design allows for the alleviation of performance bottlenecks on the Internet generated by client requests, and has been recently been considered a powerful tool as a response to the existing high demand for media-like content consumption that takes a huge share of the global Internet traffic of today. The current focus of CDNs is thus to provide content, e.g., web pages, documents, photos, videos, or even media-related streams, with high availability and performance. The strategy used by CDNs to guarantee a satisfying quality of experience (QoE) on a global scale is the deployment of content close to the end-user - a CDN contains many nodes which are geographically spread throughout the globe and close to the users they wish to serve, and whenever such users request for content, they are routed to the node which is closest to them ([16]). Data replication to servers which are strategically placed closest to end-users, coupled with good means to properly redirect such users to the most attractive edge server, is what allows content to be available more often and more quickly, which are undoubtedly attractive features in the world of e-commerce, where user QoE can dictate much of the profit - for example, Akamai, one of the leaders in CDN-related services, ran a research concluding, among other things ([?]):

- A 100 millisecond slower webpage loading speed can result with a 7% drop in sales
- A 2 seconds slower webpage loading speed can almost double the number of visitors who end up abandoning their carts
- 53% of users who use smartphones to visit web stores won't make the sale if the webpage takes more than 3 seconds to fully load
- 28% of users won't return to the same web store if they think it takes too long to load
- A 250 millisecond faster loading time proved to keep users from visiting a competitor web store

It should then come to no surprise that streaming services such as Netflix and Youtube, who now reach a global scale and whose utility is highly dependant on their high availability and low transmission delay, routinely use CDN solutions [16]. Indeed, companies that wish to provide some service in the web and who wish to have global presence routinely partner with companies whose focus is providing content delivery services, with popular examples being Akamai, CloudFlare, or Amazon Cloudfront. Coupled with the promise of highly available and quick content retrieval, these companies also couple other attractive services, such as firewalls and DDoS protection. The Internet's currently most targeted use for media consumption has made it so CDNs and their providers have an important role in dictating a very considerable percentage of flow of traffic in ISP-owned infrastructures, and as such their study and improvement is quite important, as are the efforts to increase harmonious behaviours between content delivery applications and service providers of the networks where the CDNs are deployed, with the goal in mind being network resource efficiency to guarantee that ISPs can remain operational and applications can provide a satisfiable user experience.

3.2.2 *Architecture*

Whilst implementations may vary, Figure 3 shows a high level representation of most CDN solutions. Delivery nodes possess the data which is to be requested by end-users, and are usually deployed as close to the demand points as possible, as discussed before. Such content is provided to the delivery nodes by the storage nodes which in turn are retrieved by the content's source. Integral to the operation are the control nodes, which are responsible for managing, routing and monitoring the CDN, being the role responsible for dictating how it operates. Content can be provided to delivery nodes before any request for it occurs in areas where its request can be predicted to exist, via push operations, or requested at the same time as it is also done by the client, via a pull operations, and subsequently cached. Additionally, a continuous request for data can be made by such edge nodes whenever a data stream is requested [?].

Request routing is the mechanism through which clients are located and mapped to a given CDN edge server. The prevalent approaches are, according to [27], the following:

- **DNS request routing:** The user must first resolve a domain name to retrieve a content. The CDN's DNS processes such request and, utilizing the user's IP

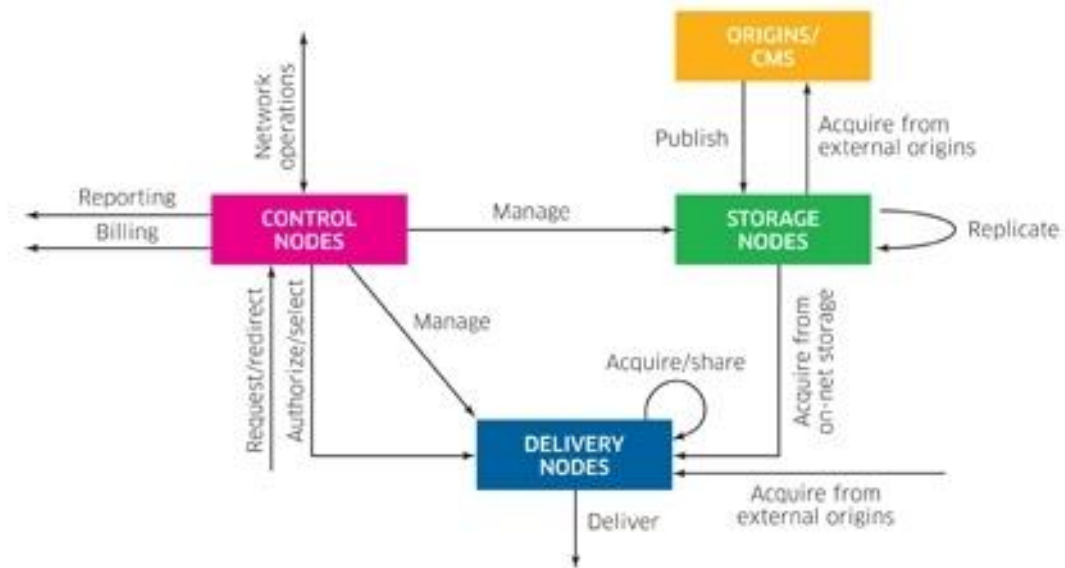


Figure 3: high level architecture of a CDN (adapted from [1])

address, historical measurement information and current server loads, responds with the edge server that seems most fitting to provide such content.

- **HTTP request routing:** Content is firstly requested to a nearby proxy server, which in turn answers with an HTTP redirect to be resolved by the client in order to find the content. The HTTP requests can occur in subsequent rounds and can also use DNS knowledge when the redirection domain must be resolved.
- **Anycast request routing:** The CDN provider announces an anycast prefix to the network. Whenever a router receives multiple announcements to the same prefix incoming from different locations, it chooses one considering some custom criteria, usually being AS hop count and Interior Gateway Protocol (IGP) weight.

3.2.3 Effects to the Network Infrastructure

As previously discussed, CDNs came as a tool to strategically position content in such a way that it can more quickly and more reliably be retrieved by an end-user. The usage of CDNs are of great interest by ISPs as their mode of operation, if made properly, can be very appealing not only to them but also to end-users. The usage of a single content-providing server (or a limited set of them) which is far away from the content supply that can have large number of points that are scattered throughout the globe, is

prone to overloading such server and path congestion if a big enough scale is achieved. The usage of data caches is a classic solution for network inefficiency problems which is also used by CDNs as a means to replicate content to strategic locations to better serve users, with the added benefit for ISPs that their network resources are efficiently used, reducing the total amount of used bandwidth needed for a service to operate, as data travels a shortened total amount of network hops from data source to points of data demand. It can thus be stated that the relationship between CDNs and ISPs are a win-win situation because efficient network usage has consequentially better service quality. However, attributing edge servers to end-users entirely on geographical data was previously discussed as being a non-optimal way of assessing node selection at the application layer. Whilst it may be intuitive that the best edge server to serve an end-user would be the one most geographically close to it, that is not always necessarily the case, much like was discussed in similar strategies used in P2P systems. Again, much like in the scenario of peer selection in P2P systems, the usage of network measurements made by the CDN itself to better pick the appropriate end-server, while it could potentially be beneficial, it could certainly be improved if it used additional, hard to retrieve data that only ISPs or other privileged entities could possess, and which are at a position to guide applications in the infrastructure with whom they have detailed knowledge. Acting with regards to the underlying network structure would be beneficial to better optimize server selection in a way that is mutually beneficial to the overlay and underlay. Whilst the problem of server allocation becomes easier with a smaller number of deployed end-nodes - e.g., choosing an edge server may be trivial if there's one per continent - as the number of these increases with user demand, it becomes more important to optimize server selection in a way that is dynamic and finely tuned to the network where it operates.

3.3 SERVER MIRRORING

3.3.1 *Concepts and applications*

Server mirroring is the act of continuously replicating a server into another, essentially creating an exact copy of it that is now accessible as if it were the original. Whilst CDNs aim at replicating chunks of contents wherever it may be necessary, the act of server mirroring performs an integral copy of a server which is self sufficient at serving a given client, as long as it periodically checks up with the primary server for

pros and cons
request routing

find examples
misbehavior
at the very least
optimize network

synchronization . It is a standard business strategy that uses redundancy as a means to increase reliability, availability and performance. The existence of many servers that perform the same task means that these can be strategically chosen to serve a client in a given situation, e.g., by selecting the one that has small end-to-end message delay and little current server load. Figure 4 shows an application of this, where multiple server mirrors exist to deliver software packages to the Linux Mint distribution. The user has the choice to manually select one of these mirrors, and ideally chooses the one that is most fitting to them.

cite

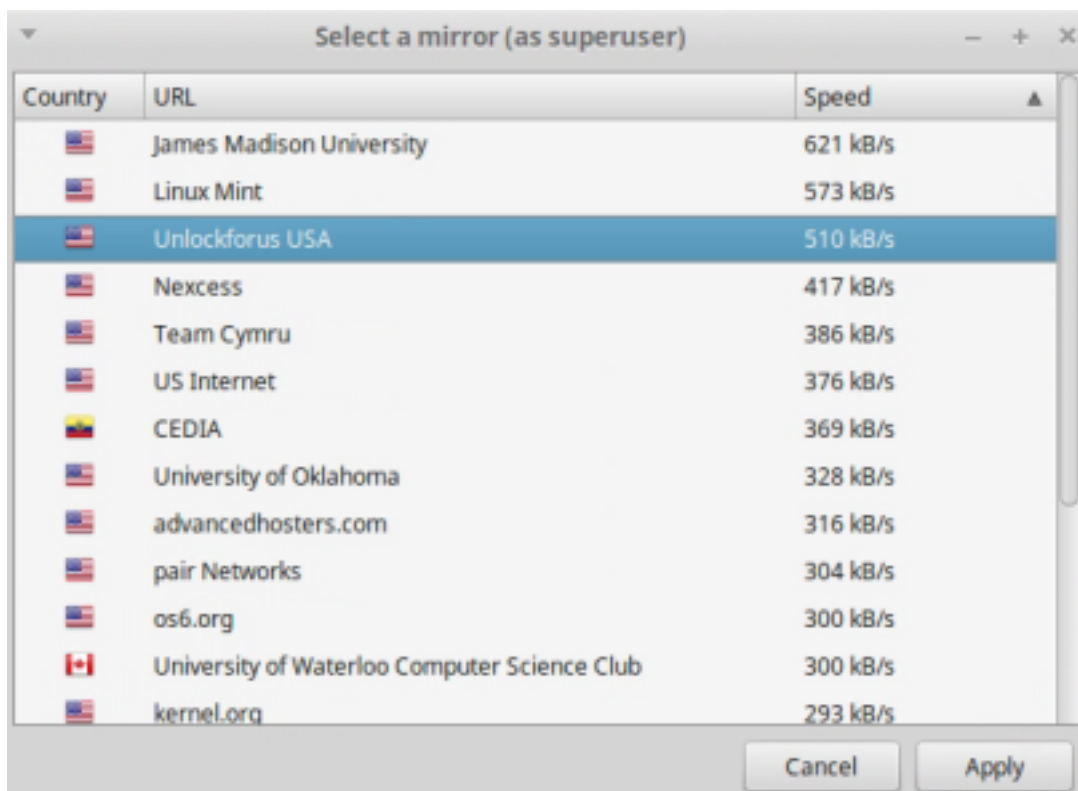


Figure 4: Linux Mint prompt to select a software repository mirror

3.3.2 Effects to network infrastructure

Much like the content replication utilized in CDNs, an integral replication of a main server proves itself as an advantageous tool capable of delivering services more closely to users, and as such allows the reduction of total amount of bandwidth used to serve all clients. Much like all previous use cases discussed before, optimizing application traffic

is crucial to guarantee good network resource usage, and in case of server mirroring it comes down to good strategical deployment and dynamic, intelligent algorithms to properly attribute mirrors to requesting end-users. Giving end-users the choice to manually select the serving mirror seems problematic, as application-generated traffic is not optimized. In fact, considering the Linux Mint software package distribution discussed above, despite currently existing seventy mirrors deployed throughout the globe to fit this role, a large number of these remains mostly unused whilst the main and default server is constantly prone to overworking [2]. It can be stated that end-users both don't care enough to optimize traffic nor do they have enough information to properly do so even if they did. Deployment of server mirrors is a great tool that brings with it the issue of optimizing server selection, and much like all examples given so far, traffic generated by applications can be firstly optimized by the applications themselves if they consider static and dynamic information of the network they operate on.

3.4 TRAFFIC OPTIMIZATION BY APPLICATIONS AND LAYER-COOPERATIVE APPROACHES

This section serves to display the proposed solutions and existing implementations that have been made in the attempt to optimize application traffic utilizing network information. Given the increasing scale of the Internet as a near ubiquitous system, and increasing tension between service providers and applications, it comes as no surprise that the area of layer cooperation has been through exhaustive work. Many solutions have been devised for specific use cases, with varying degrees of power to each one of the layers.

Many different mechanisms have been developed with the goal of decreasing tensions between ISPs and P2P applications, which is a subset of the general layer cooperation problem. Figure 5 represents a grouping proposed by [8] where such mechanisms are ordered in agreement with how much involvement the P2P systems and ISPs have. These classes are as follows:

- **Class 1:** There is not much interference in the overlay by ISPs nor are P2P systems cooperative. Instead, ISPs apply traffic engineering methods to selectively favour types of traffic. This is usually done to guarantee certain QoS levels to some classes of traffic, which are then to be treated favourably at the forwarding and routing levels. Examples of such techniques are DiffServ, Multi-Topology Routing (MTR) and MultiProtocol Label Switching (MPLS). These classes of methods do

not fix the underlying problem, but are instead used to control preexisting traffic. As such, the peers' routing decisions are not affected and P2P traffic still remains non localized.

- **Class 2:** There is ISP intervention in the overlay in such a way that peers continue normal operation without realizing that such interventions occur. This can be reached via the use of proxies that can effect the control plane with the redirection of content requests to local peers, or at the data plane with content caches which act as normal peers and are strategically placed in the network. These methods are advantageous because they do not require any changes to P2P protocols, because the ISP has an active role in molding to the overlay, intercept traffic, and either help or guide it in a way that favours them. Indeed, these techniques can be proven to work, as concluded in [8], and put into practice, for example, in [23] and [24] via the specification of a BitTorrent tracker that is programmable to allow for P2P qualitative differentiation and ISP-cooperative traffic engineering that could help reduce inter-domain traffic significantly. However, this class of mechanisms are not without their challenges - firstly, it involves much effort by ISPs, as it requires structural upgrades and constant adaptiveness to new and changing P2P protocols. Perhaps worse, even considering proper budget and maintenance, such methods can prove themselves to be not possible at all - for legal reasons, as data caches could possibly contain illegal content; and for technical reasons, since the packet inspection required by ISPs to detect and steer P2P traffic may be blocked due to the peer's attempts to mask its traffic as non-P2P related.
- **Class 3:** Relative to previous classes, the active role is switched, and it is the P2P system itself that acts in regards to the underlay it operates on, but without ISP involvement, which would require change from classic P2P protocols. Peers probe the neighboring network elements as a way to get more familiar to connection properties, and act on these probings during operation, e.g., when choosing neighbours to construct the overlay network on when choosing where to retrieve a given resource. Whilst these methods can be advantageous for both applications and ISPs, it can't be assumed that to always be the case. As peers have no ISP input, they cannot have a full scope on the network and ISP needs, and as such these application optimizations can end up being more hurtful than helpful. For example, this paper mentions a simple scenario where a peer uses RTT measurements to choose between two candidate peers, but the one

that is geographically closest to it belongs to another AS, and his selection would incur in more costs. The paper describes this class as a "win-not-lose" situation, meaning that while the P2P system can, in the right circumstances, improve its performance via measurement, the ability to act beneficially to the underlay without any feedback from ISPs cannot be guaranteed. such a example of improved performance could be seen at [18], which improved BitTorrent's download performance and even managed to reduce ISPs' backbone and cross-ISP traffic. The technique consisted in having peers send traceroute measurements to the tracker, which in turn grouped them into local, intra-ISP and inter-ISP groups, with the assumption that inter-ISP links generally have much more latency than the rest. As peers would later query the tracker for content, the returned peer list would be biased in such a way that promotes traffic locality.

- **Class 4:** Full and active cooperation exists between the ISPs and P2P systems. The role of the ISPs is to provide information, and P2P systems let that information dictate its decisions. It is the methodology that most comes close to a mutually advantageous scenario for both parties, given that they both keep the entire group's needs in mind. For example, [5] proposes an oracle that receives as input from querying peers a list of candidate peers, and ranks them in order of proximity to the querying node; such method was tested in simulation and proven to decrease negotiation traffic and improve scalability of P2P networks. Another example of this is [13], which devised a CDN-P2P hybrid where peers utilize RTT measurements to group themselves by separate orders of geographical proximity with the same intent of the previous example, which is to localize traffic whenever possible. This technique also proved itself to be advantageous, as the solution was more efficient in terms of total service disruption time when compared to a previous iteration of the hybrid architecture which used random peer selection to look up available target peers. The functional intent is that the oracle possesses privileged network information and acts on it to provide guidance to querying applications, and thus has the liberty to impose policies and optimizations, e.g., pair peers which are the least number of network hops apart via a Dijkstra algorithm . Another more complex approach that could be used by the oracle proposed by [10], which contains algorithms to dictate peer selection, task assignment and rate allocation. The method requires the full network topology as input - including link capacities and peer service costs - to minimize file downloading time and cost. The oracle would also be free to enforce

citar nosso?

ISP biases as it desires by modifying such algorithms as to, for example, minimize usage of costly links (such as inter-AS ones). The ALTO working group - whose work this thesis attempts to materialize into a working library and further extend its features - was formed to standardize the oracle-user scenario so it could be properly used in many situations at the scale of the Internet as a whole.

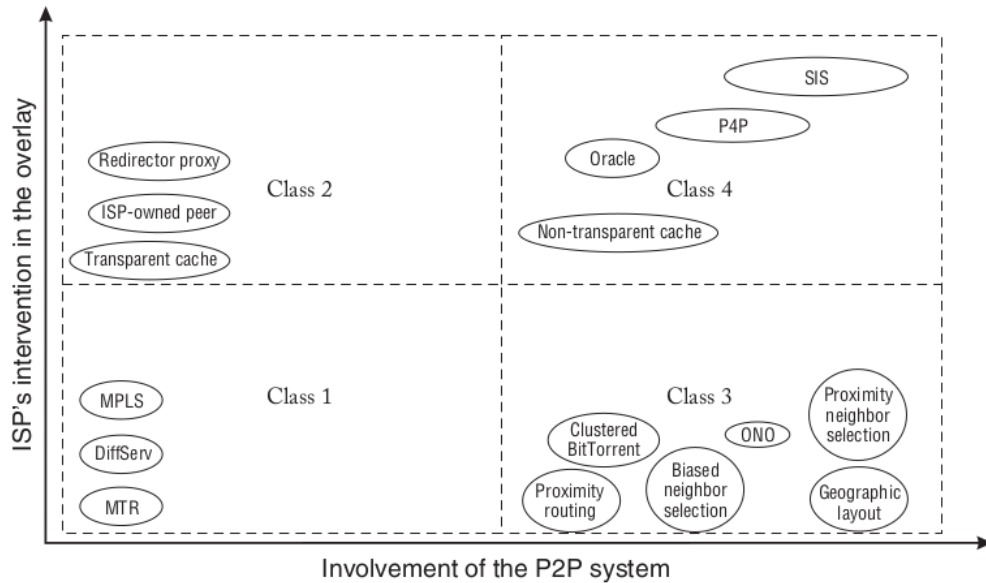


Figure 5: Existing approaches to decrease tension between P2P applications and ISPs ([8])

Given the current share that CDNs have on global Internet traffic of today, coupled with the demand for a good QoE by end-users, it should come to no surprise that this applicational domain has also been through efforts to optimize its traffic. One such way to do so is to optimize client query redirection, i.e., better choose which edge server should be attributed to an end-user when a name resolution is requested for some content. [9] considers a CDN built to deliver video data where some given set of content exists redundantly in many edge-servers, and argues for an algorithm where the choice is made to optimize client download time, which in turn has to consider the network parameters at time of request, as well as current server load. In another angle, IETF devised a problem statement in regards to Content Distribution Network Interconnection (CDNI) [17], which outlines the efforts needed to specify a set of interfaces that allow for the interconnections of many CDNs, with the added benefits that a multi-CDN system will have better availability, coverage, and capabilities than the

single CDNs by themselves. The four devised interfaces (CDNI Control interface, CDNI Request Routing interface, CDNI Metadata interface, and CDNI Logging interface) are all control plane interfaces to be operated at the application layer, and the group states that no new application protocol needs to be devised, and instead existing ones could be leveraged, e.g., HTTP, Atom publishing protocol, XMPP, and in particular to the CDNI Request Routing interface, the ALTO protocol who's the focus of this work. Still in the topic of optimizing CDN's edge server selection, [27] suggests a way of optimizing anycast request routing, which differs from the DNS-oriented request routing techniques, which, while very light in terms of network engineering and infrastructural overhead when compared to existing alternatives whilst maintaining a close to optimum network path, it sacrifices flexibility to do so, as it is agnostic to the network's status and not much network engineering can thus be done. As such, the work proposes anycast request routing utilizing software defined networking (SDN), where load balancing is made at the ISP network with help of CDN-provided additional information. This example of layer cooperation can allow for many optimization opportunities that leverage an existing and low-maintenance mean of request routing with the flexibility achieved with SDN solutions.

Attempting to optimize web server selection, [15] argues that DNS-oriented solutions, which select the nearest server but also employing load balancing, may not be the best at optimizing server-client QoS levels. Instead, it is proposed that selection is based on QoS measurements, from which three types are distinguished: a static method, such as choice based on least number of hops to server (which is unlikely to change); a dynamic method, consisting on dynamic instantaneous probing of the network to monitor, for example, round-trip time (RTT) delay to the servers; and statistical methods, which decide based on a larger set of measurements made in various points in time. Utilizing the latter method, RTT measurements and web-related request benchmarking is made, such as time to establish TCP connection, elapsed time from GET HTTP method to first packet received, time to retrieve data fully, etc, every five minutes and spanning several weeks. The work concluded that statistical methods used to select between multiple equal web servers had high correlation with download time from the selected server, but optimizations should be evaluated in regards to computer workload and the amount of probing traffic. Tackling a similar challenge, [26] proposes a method of server mirror selection which is better optimized than the more popular approach of giving the user the selecting choice. Thus, the proposed solution's architecture consists of two types of agents: a client agent, which monitors the mirror server it was deployed in and stores static information, e.g., geographical location of server and maximum

capacity, and dynamic information, e.g., current load and bandwidth. This information is then sent to the other role of the architecture, the server agent, which compiles it and acts as an oracle that is queried by users whenever mirror selection is needed, replying with a ranking of candidate servers based on bayesian networks.

Congruent to the task of optimizing network traffic with layer cooperation, [19] proposes a reconfigurable and adaptable overlay multicast system, further optimizing the multicast strategy - used for group communication as a means to reduce redundant traffic - and leveraging collaborative efforts between it and the ISPs to construct multicast distribution trees whilst integrating traffic engineering mechanisms for the task of network usage optimization.

3.5 APPLICATION-LAYER TRAFFIC OPTIMIZATION (ALTO) WORKING GROUP

3.5.1 *Context and Motivation*

Following research indications that improved peer selection algorithms based on ISP-provided information could help reduce ISP costs and increase P2P application performance, the Internet Engineering Task Force (IETF) devised working groups to explore IETF standardization in the area of layer-cooperation [21]. Among them is the Application-Layer Traffic Optimization (ALTO) working group, whose domain is traffic localization.

The ALTO working group designed an HTTP-based application protocol whose function is to allow hosts to query privileged servers on network information. The envisioned scenario of the service provided by the ALTO architecture, as can be seen on Figure 6, considers both the physical and application levels. The ALTO service is provided by some oracle, which needs to be himself informed on network information that can take many forms - topological structure, routing costs considering many metrics, static provider policies, etc - and, most importantly, such data is to be fed by an ISP, or another entity that contains truthful and relevant network information that the oracle could deem useful in aiding its clients. Consider that "Peer 2" has to choose, among "Peer 1" and "Peer 3", which peer to retrieve a resource from (the information of what peers contain the resource in question could've been retrieved from a tracker or via a flooding query). Instead of resorting to classical strategies, such as random choice or probing measurements, "Peer 2" is to use the ALTO service, querying the oracle on information pertaining to the candidate peers, and in regards to metrics that better fit

alto for dete
edge (2x)

the needs of the application (because different applications could have different QoS metric priorities in mind, such as a media stream with low delay needs or a file sharing application with focus on bandwidth). Ideally, it would make sense that the querying peer would end up choosing "Peer 3", as they reside on the same network. As could be deduced from this and similar scenarios, an architecture containing one or more servers that are knowledgeable on the network they reside on could be an important tool to make P2P applications locality-aware.

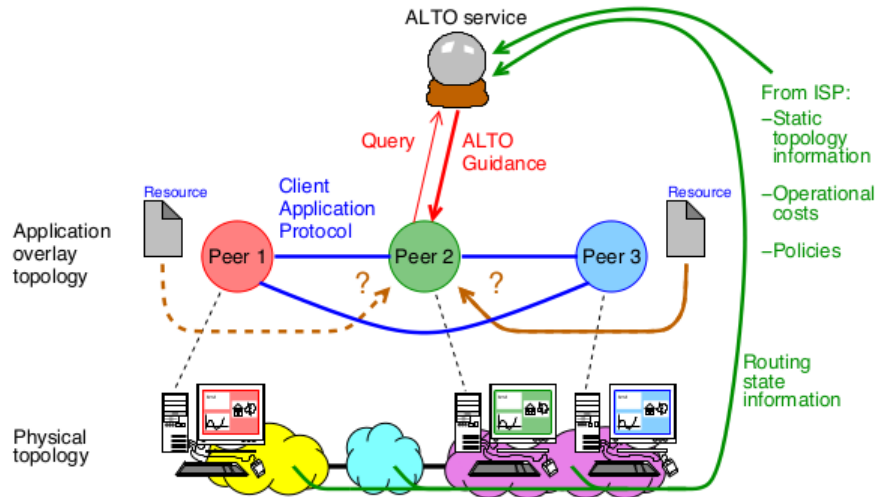


Figure 6: ALTO scenario [21]

Despite the origins of the ALTO protocol lying in P2P applications, it is now being considered in other fields, such as data-center networks and content distribution networks (CDNs) [3]. With interest in the latter example, ALTO protocol extensions are being developed as a means to create functionality that could be useful for services employing CDNs, most specifically the implementation of the CDNI Request Routing Footprint & Capabilities Advertisement interface [22], which is a subset of the CDNI standard [17] that aims to allow upstream CDNs to query known downstream CDNs if they are able and willing to accept the content request. In particular, one of the main functionalities of the CDNI request routing interface is the ability for upstream CDNs to retrieve static or dynamic information on downstream CDNs (resources, footprint, load), which they provide themselves, and that allow upstream CDNs to better choose the appropriate edge server that could serve a given end-user. ALTO serves as a good protocol to implement such functionality because it fits its use-case: some node (in the upstream CDN, where the content query originates) wishes to improve its routing (in regards to resolving content requests) by using information, which is hard to deduce

by itself, to properly choose the most efficient node (the downstream CDNs where the content resides). At a more abstract level, this is similar to the use case fulfilled to P2P applications to help them better select peer connections.

A mode of operation where applications no longer operate in disregard to the network infrastructure they run on, but instead in deep consideration of it, could help significantly alleviate the issues emerging from the tension between the underlay and overlay levels, and is of mutual interest - improving application performance and reducing infrastructural costs. Enabling a communication channel can thus allow for many different co-operational use cases besides the aforementioned ones - for example, redirecting users to nearby data caches or warning them of server maintenance ahead of time. The existence of an all-encompassing oracle could also prove beneficial for applications which utilize periodic network probing to guide their choices, as such information could be made by a select few nodes in the network and applicable to all nodes which are close-by to such node in ways that the ISP seems advantageous (such as belonging to the same AS or geographically near), thus minimizing the amount of probing traffic used and giving it to an entity that could better reason with it to help the querying user.

Finally, standardizing an architecture and related protocols that could help a large subset of problems could also be of great value as it would facilitate the integration of solutions into other ones which already follow the specified standard, thus leveraging the ALTO protocol to their needs, not requiring further cycles of development. Also, in a more abstract level, the attempt to standardize is helpful as it joins forces from many different domains which share common patterns (many exemplified previously), and as such could result in a better and more proven product that could be used at the scale of the Internet itself.

3.5.2 *Architecture*

The general ALTO architecture can be seen on Figure 7. Central to the operation is the ALTO server, which stores network information and provides it to querying clients. Such network information is provided by trustworthy entities, with some enumerated in the same figure. Two protocols can be seen as part of the general architecture: the provisioning protocol, not contemplated by the ALTO working group, should specify how information is provided to the ALTO Server; the ALTO protocol, main focus of this working group, specifies server-client interactions mainly in the form of data querying

adicionar js
em apendice

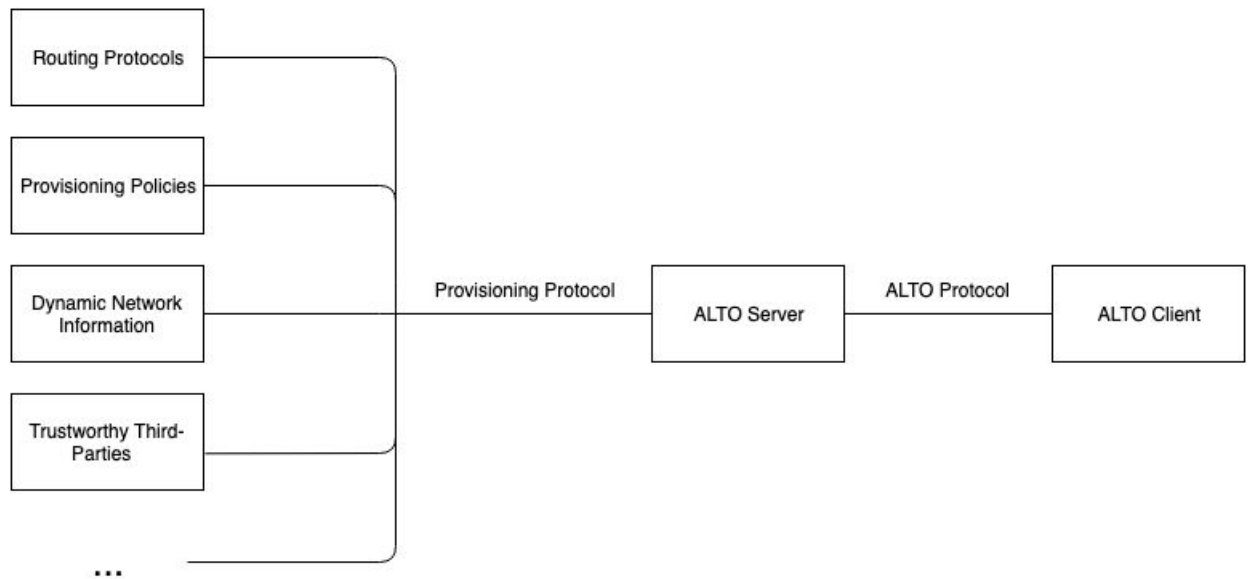


Figure 7: ALTO architecture (adapted from [12])

and delivering. The ALTO client is the main consumer of the ALTO service as a whole, and it queries the ALTO server on network information whenever it deems such data as necessary to what it's doing at a given moment. An ALTO client could be seen as any entity which is able to interface with the ALTO protocol as the role of a client, and as such is not tied to a specific implementation - in the example of P2P file sharing, a peer can act as an ALTO client (like the example scenario in Figure 6), but instead a tracker could enhance its role in assisting peer communication by having an embedded ALTO client which would then act on behalf of querying peers as to provide them with an optimal response.

The ALTO services contemplated by the corresponding working group can be visualized in Figure 8.

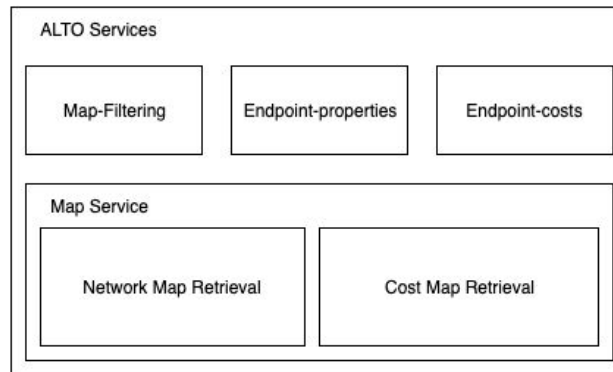


Figure 8: ALTO services (adapted from [12])

The ALTO server stores and provides special mappings in the form of network and cost maps. A network map provides network location grouping identifiers and the corresponding aggregated endpoints. It utilizes Provider-Defined Identifiers (PIDs) as a key, and the mapping itself is left to the responsibility of the providers - it is thus a way of indicating that many endpoints should be handled similarly. A provider can then aggregate endpoints by geographical proximity, one or many subnets, one or many Autonomous Systems (ASs), etc., and attribute properties to the aggregate, instead of the endpoint. The other resource type provided is a cost map, which can be defined as a matrix M , where M_{ij} - with i and j being the source and destination PIDs, respectively - is the associated path cost between the two indexes. The cost has two components: its metric and mode. The ALTO base protocol only defines a single, generic, cost metric called "routingcost". However, the current draft [28] is, as time of writing, currently specifying more concrete metrics, with many associated with Quality of Service (QoS) evaluation, e.g. one way and round trip delay, packet loss and throughput. The other cost property, cost mode, can either specify that the metric is to be interpreted as a numerical value or as an ordinal ranking among all other costs in that cost map - this is useful in cases where too much network information is not deemed reasonable to share, and a simple order of preference that doesn't expose too much infrastructural detail can suffice. The decision to separate network and cost map information into two types of resources comes from the reasoning that network mappings are unlikely to change, whereas cost mappings could be periodically updated. As such, it alleviates client applications from the need to retrieve redundant information, and the ability to only retrieve a subset of it - this ability is further expanded in the map filtering

service, which allows an ALTO client to further specify which regions of the requesting maps it wishes to retrieve (much like a "SELECT" statement from the Structured Query Language (SQL)), and only these are transmitted to it.

Finally, the last two services focus on mappings that regard to specific endpoints, instead of abstract mappings that utilize PIDs. An endpoint is identified by one of the following: IP address, MAC address, or overlay ID. The endpoint property service maps to an endpoint a set of properties, e.g., geographical location or connectivity type, and the endpoint cost map has the same meaning of a cost map, but mapping to particular endpoints and not abstract collections.

As could be seen, the ALTO project specifies an architecture for trading of network-related information, with well defined roles and a request-response protocol to fulfill interactions between them. It also attempts to standardize such interactions in the form of data structures with well defined attributes which are then to be manipulated for each use case. This could then serve as a useful service for any application that wishes to retrieve network information that could improve its decision making at the application level. It is important to note that there are restrictions to what kinds of information are contemplated by the ALTO protocol - for example, transport-level congestion is beyond its scope, and thus should not replace conventional mechanisms. The type of data which is valid to consider, according to [12], should not be easily obtainable by the clients themselves (such as end-to-end delay), and should be variable on a longer timescale than the instantaneous kinds that are seen on congestion control, as the resulting intelligence gathering traffic generated would be counterproductive to the task of traffic optimization. Potentially valuable information that is in the ALTO scope would then have to be harder to obtain without aid of this service, and not highly mutable through time - for example, routing costs, geographical locations, network proximity, operator's policies, scheduled down-times, etc.

This project is, at time of writing, still on-working, with many drafts being published and updated. These are, however, relating to service extension and deployment, as the main architecture, protocol design, implementation guidelines and security analysis are fully published into their respective RFC documents.

3.5.3 *Issues and challenges*

Given the nature of this architecture - the trading of sensitive network structure information that can alter application behavior - it is quite apparent that its implemen-

ALTO serve
tistical in ho
peers

the voluntar
system mak

pg 8 de dan

tation is not without challenges from a security perspective. As such, [12], besides over-viewing the protocol, also does a threat analysis of the ALTO architecture. Utilizing the "STRIDE" threat model, the main threats to the ALTO architecture are as follows:

- Spoofing of a legitimate ALTO server that would mislead clients with wrong information - this could give the malicious party the ability to change traffic to its will. Spoofing of the clients themselves can also occur, and could allow a malicious party to retrieve sensitive network data they aren't authorized to.
- Tampering of data to mislead either ALTO servers or clients. If some unauthorized and malicious party can retrieve data that is in transit and tamper with it, clients would act on information that they assume is trustworthy but in fact has been modified. As such, clients could be redirected to wrong addresses, or receive incomplete or incorrect data that results in bad decision making. On the other hand, data tampering that occurs between data providers and the ALTO server would give it, a seemingly trustworthy party, untrustworthy data. This would result in the same issues pointed previously.
- Repudiation of being the source of some network information, whether it be by a third party or the ALTO server itself, would make it difficult to neutralize and attribute culpability to incorrect or malicious sources.
- Information disclosure in the form of ALTO resources to parties that were not in the circle of authorized parties, or via the invasion of privacy of a querying client in data such as usage patterns
- Denial of service (DoS) of the ALTO server itself via many requests. Additionally, another DoS threat exists via the manipulation of ALTO resources themselves - if a cost map is manipulated to highly favor a specific subset of servers, these could be favourably picked by clients in a disproportionate matter, and highly affect these servers' availability.
- Elevation of privileges that lead to obtaining or modifying more information than initially permitted.

Many of these threats are standard and could be solved with state of the art solutions which are well proven and tested, as indeed states [12]. However, threats of information disclosure - whilst it can be guaranteed in-transit via encryption, what is done with this

information the moment it reaches the client is hard to control. Situations may arise when a privileged client shares, intentionally or not, sensitive network information it retrieved from an ALTO server to an unprivileged client. Furthermore, another situation that could arise is one where many differently privileged clients share information amongst them to get a complete view of the network structure. As such, it is firstly important for the ISP or third party that provides such information to carefully plan out the information it wishes to share. Possible solutions to minimize these threats are as follows:

- Reduce the variety of the provided data, with the consequence of less precise ALTO guidance
- Provide information that refers to abstraction instead of concrete data. One example is the usage of network groupings by PIDs instead of information mapping to concrete endpoints. Another example is the usage of ordinal cost types that indicate relative preference instead of concrete cost values. This strategy suffers from the same downsides as the previous one
- Work only with a small set of trustworthy ALTO clients that are to act on behalf of a larger subset of less trustworthy clients. For example, via the deployment of certified trackers that choose on behalf on P2P clients by giving them customized responses. This is still, however, worthy of a threat analysis as many relative information could be derived from the clients via carefully crafted requests.
- Utilize terms of agreements that are to be enforced on every querying client. This would work as a dissuasion method but could be infeasible and impractical if other threats are not neutralized (such as spoofing). Furthermore, such enforcement could be seen as unappealing by some users as it could violate user privacy.

As a conclusion, it is important to understand that the ALTO project, for better and for worse, attempts to achieve layer cooperation. Whilst it can be an advantageous partnership for both parties, it is to be expected that an ISP would be reluctant to share infrastructural detail to a pool of potentially malicious parties. It is thus important that a compromise is reached that could allow for a mutually beneficial exchange.

SYSTEM ARCHITECTURE AND DEVELOPED MECHANISMS

As the main proposed goal of this work is the implementation of a system that complies with the ALTO working group's devised protocol, this chapter exhibits the planned software specifications needed to implement the system as a whole.



Initial attention is given to the general architecture on the first section, with the goal of identifying key entities, their purpose, and how they interact among themselves. The following section will target the specification of ALTO resources, which can be considered the driving force behind the system, as they are what the client entities seek, and likewise what the ISPs wish to provide. The next section will focus on specifying the task of network intelligence provision to an appropriate ALTO server, in such a way that a common interface exists among all entities that are able to increase the server's knowledge of the network's physical topology. Specified the way that network information is provided, the next section details how a given actor, such as an ISP administrator, can manipulate such information before it is forwarded to the server - which includes the insertion of static ISP preferences or the abstraction of network entities as a means to dilute concrete topology details without forfeiting the usefulness the clients can retrieve from the processed resources. Finally, the task of inter-ISP ALTO server synchronization and communication is specified in the form of required protocol extensions and needed mechanisms that allows the increase of a single ALTO server's knowledge domain.

4.0.1 *System Architecture*

Figure 9 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 intelligence aggregator by any given means, means as direct as importing a file that contains all the needed data, or as complex as a cluster of nodes that inspect the

network for routing protocol packet messages and query nodes for physical properties and inject the resulting data into a central aggregator. Such intelligence is then parsed and uploaded to the ALTO server.

Table 2: Network node entities in the conceptual ALTO system representation

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

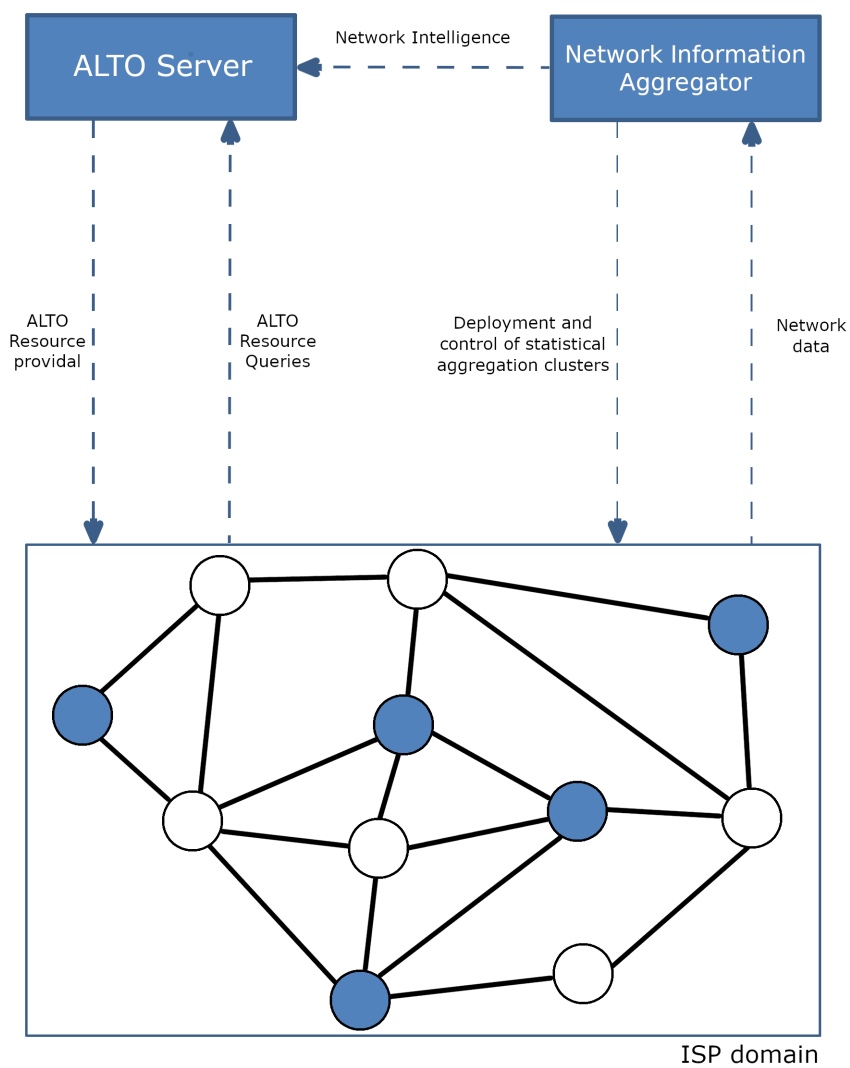


Figure 9: Conceptual representation of the ALTO system of a given ISP

More formally, Figure 10 presents the proposed system architecture. One can identify the ALTO interface as a key component of the system, as it allows to bridge three different application layers - the ALTO resource consumer, the ALTO resource provider, and the network intelligence aggregator, to be further specified in the following sections. The ALTO working group has extensively specified the ALTO protocol in regards to resource query, and the concrete implementation of this work will aim to fully 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. With this in mind, this work specifies a protocol extension that enables a provisioning party to interact with an ALTO server with the intent to provide it with ALTO resources.

ALTO Resource Consumer

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 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 content which resides redundantly on many other nodes, such as CDN edge nodes or content mirrors, could also benefit from oracle guidance, and thus qualify as appropriate ALTO clients.

ALTO Resource Provider

An ALTO resource provider is materialized in the form of an ALTO server, an entity that possesses pre-processed network information in the form of ALTO resources. Its job is to store and manage such resources, and provide it to querying ALTO clients. Additionally, data validation and persistence are responsibilities that belong to the resource provider layer. 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 4.0.5.

Network Intelligence aggregation

The network intelligence aggregation layer is the layer that enables the translation of raw topological information - such as the physical attributes of network devices and connections - and processed, query-eligible network knowledge. To do so, a very important entity, perhaps the heart of the system as a whole, is the network intelligence provider, which is responsible for retrieving network information and sending it, through a well defined network intelligence provisioning protocol, to a network intelligence 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.

4.0.2 *ALTO 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 applicational decisions. All ALTO resources must have the following:

- **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. Following suit to the defined protocol, meta information contains the resource's name, version and, if applicable, resource dependencies and cost details - its mode, metric name, and description.
- **Network 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 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, respectively).

Further formal specification is not made as it has been extensively done in the ALTO protocol [12]. Figure 11 summarizes the alto resource specification and lists the concrete resource types they can be materialized into, each with their well defined structure that is expected to be known by entities that interact with the ALTO interface.

Add network

4.0.3 *Network intelligence provision*

4.0.4 *Network intelligence preprocessing*

4.0.5 *Multi ALTO server communication*

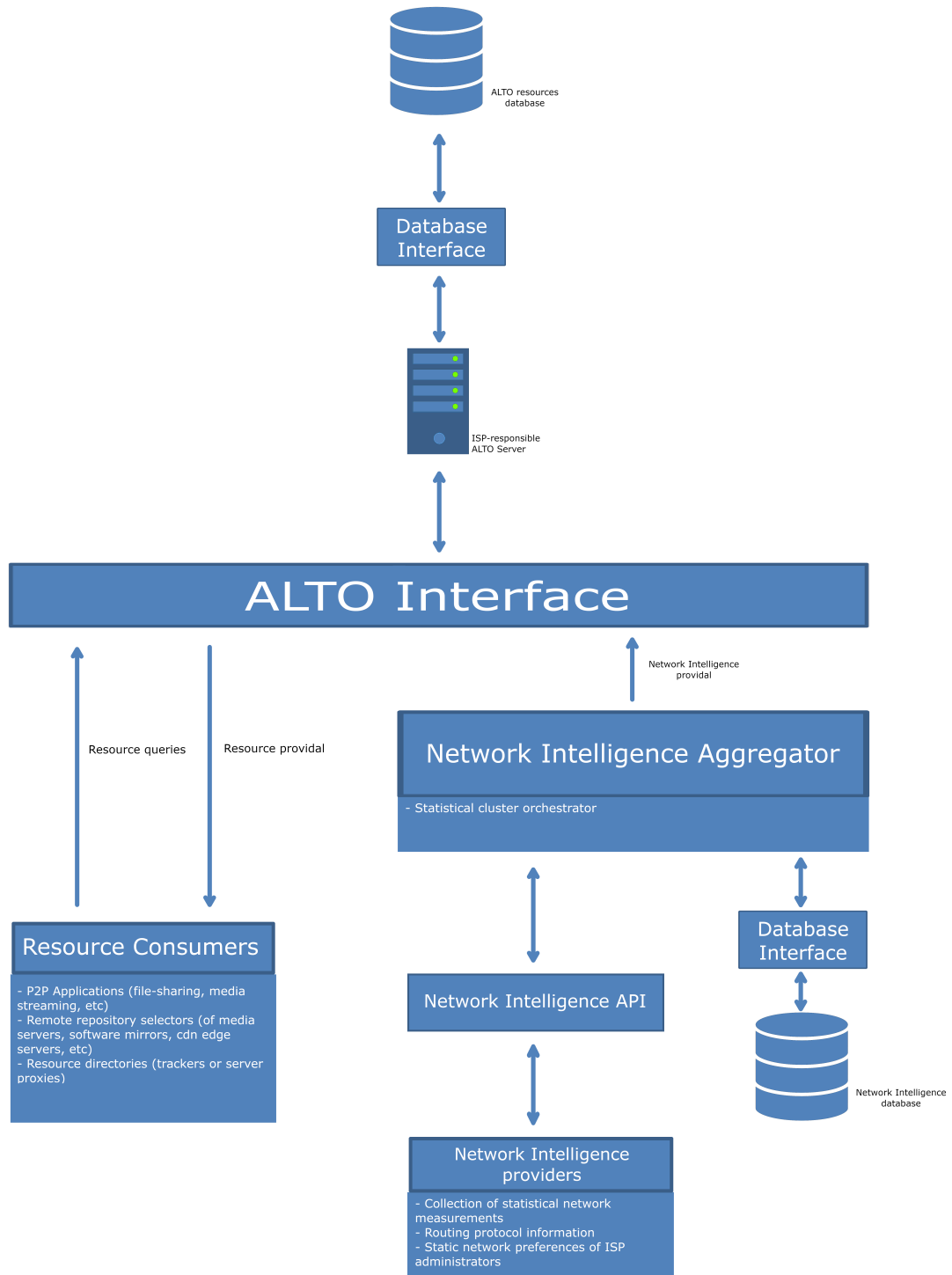


Figure 10: System architecture at a macro level

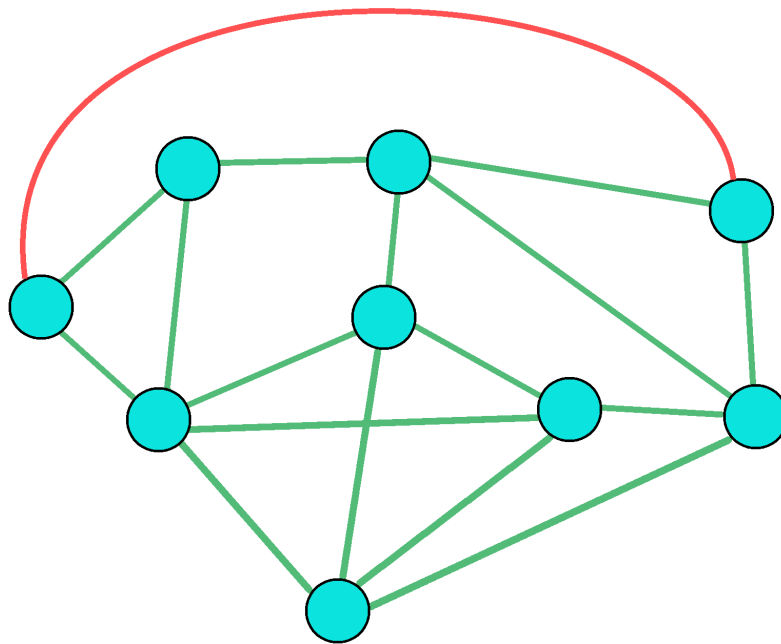
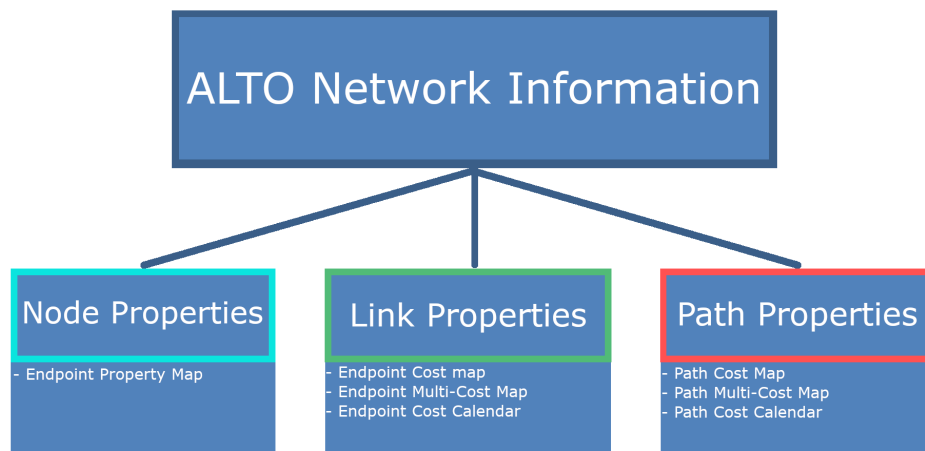


Figure 11: Schematization of the pondered ALTO resources

5

IMPLEMENTATION

5.0.1 *Technologies used*

5.1 OPTIMIZATIONS

EXPERIMENTS

6.1 SETUP

6.2 RESULTS

6.3 SUMMARY

BIBLIOGRAPHY

- [] The free haven project. <https://www.freehaven.net/overview.html>. Accessed: 2020-05-20.
- [1] What is a cdn? content delivery network explained by global-dots. <https://www.globaldots.com/content-delivery-network-explained#content-delivery-networks-for-specific-verticals>. Accessed: 2020-01-03.
- [2] 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.
- [3] Charter for working group. Technical report, 11 2019. URL <https://datatracker.ietf.org/wg/alto/about/>.
- [4] Cisco visual networking index: Forecast and trends. Technical report, 2 2019. URL <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>.
- [5] V. Aggarwal and A. Feldmann. Locality-aware p2p query search with isp collaboration. *NHM*, 3, 06 2008. doi: 10.3934/nhm.2008.3.251.
- [6] V. Aggarwal, S. Bender, A. Feldmann, and A. Wichmann. Methodology for estimating network distances of gnutella neighbors. 01 2004.
- [7] A. Akella, S. Seshan, and A. Shaikh. An empirical evaluation of wide-area internet bottlenecks. *ACM SIGMETRICS Performance Evaluation Review*, 31, 05 2003. doi: 10.1145/885651.781075.
- [8] G. Dán, T. Hossfeld, S. Oechsner, P. Cholda, R. Stankiewicz, I. Papafili, and G. Stamoulis. Interaction patterns between p2p content distribution systems and isps. *IEEE Communications Magazine*, 49, 05 2011. doi: 10.1109/MCOM.2011.5762821.

- [9] 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. doi: 10.1109/EDM.2014.6882495.
- [10] 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. doi: 10.1109/ETCS.2009.248.
- [11] T. Karagiannis, P. Rodriguez, and K. Papagiannaki. Should internet service providers fear peer-assisted content distribution? 01 2005. doi: 10.1145/1330107.1330115.
- [12] S. Kiesel, W. Roome, R. Woundy, S. Previdi, S. Shalunov, R. Alimi, R. Penno, and Y. R. Yang. Application-Layer Traffic Optimization (ALTO) Protocol. RFC 7285, Sept. 2014. URL <https://rfc-editor.org/rfc/rfc7285.txt>.
- [13] 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. doi: 10.1109/ISCE.2011.5973865.
- [14] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys Tutorials, IEEE*, 7: 72– 93, 04 2006. doi: 10.1109/COMST.2005.1610546.
- [15] 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. doi: 10.1109/ICC.2001.936549.
- [16] E. Nemeth, G. Snyder, T. R. Hein, B. Whaley, and D. Mackin. *UNIX and Linux System Administration Handbook (5th Edition)*. Addison-Wesley Professional, 5th edition, 2017. ISBN 0134277554.
- [17] B. Niven-Jenkins, F. L. Faucheur, and D. N. N. Bitar. Content Distribution Network Interconnection (CDNI) Problem Statement. RFC 6707, Sept. 2012. URL <https://rfc-editor.org/rfc/rfc6707.txt>.

- [18] 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. doi: 10.1109/IIH-MSP.2009.90.
- [19] A. Sampaio and P. Sousa. An adaptable and ISP-friendly multicast overlay network. *Peer-to-Peer Networking and Applications*, 12(4):809–829, Sept. 2018. doi: 10.1007/s12083-018-0680-y. URL <https://doi.org/10.1007/s12083-018-0680-y>.
- [20] Sandvine. The global internet phenomena report. Technical report, 10 2018.
- [21] J. Seedorf, S. Kiesel, and M. Stiernerling. Traffic localization for p2p-applications: The alto approach. 10 2009. doi: 10.1109/P2P.2009.5284511.
- [22] J. Seedorf, Y. R. Yang, K. J. Ma, J. Peterson, X. S. Lin, and J. J. 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, Nov. 2019. URL <https://datatracker.ietf.org/doc/html/draft-ietf-alto-cdni-request-routing-alto-08>. Work in Progress.
- [23] P. Sousa. *Context Aware Programmable Trackers for the Next Generation Internet*, volume 5733, page 78. 2009. doi: 10.1007/978-3-642-03700-9_9.
- [24] P. Sousa. A framework for highly reconfigurable p2p trackers. *Journal of Communications Software and Systems*, 9(4):236, dec 2013. doi: 10.24138/jcomss.v9i4.144. URL <https://doi.org/10.24138%2Fjcomss.v9i4.144>.
- [25] D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*, 36, 12 2004. doi: 10.1145/1041680.1041681.
- [26] M. Swain and Young-Gyun Kim. Finding an optimal mirror site. In *Proceedings. IEEE SoutheastCon, 2005.*, April 2005. doi: 10.1109/SECON.2005.1423293.
- [27] M. Wichtlhuber, J. Kessler, S. Bucker, I. Poesse, 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*, 2017.
- [28] Q. Wu, Y. R. Yang, Y. Lee, D. Dhody, and S. Randriamasy. ALTO Performance Cost Metrics. Internet-Draft draft-ietf-alto-performance-metrics-08, Internet Engineering Task Force, Nov. 2019. URL <https://datatracker.ietf.org/doc/html/draft-ietf-alto-performance-metrics-08>. Work in Progress.



SUPPORT MATERIAL

NB: place here information about funding, FCT project, etc in which the work is framed. Leave empty otherwise.