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

AUTHOR COPYRIGHTS AND TERMS OF USAGE BY THIRD PARTIES

This is an academic work which can be utilized by third parties given the compliance of the rules and good practices regarding author and related copyrights, which are internationally accepted.

Therefore, the present work can be utilized according to the terms provided in the license shown below.

If the user needs permission to use the work in conditions not foreseen by the licensing indicated, the user should contact the author, through the RepositóriUM of University of Minho.

License provided to the users of this work

Attribution-NonCommercial-ShareAlike
CC BY-NC-SA
https://creativecommons.org/licenses/by-nc-sa/4.0/

[This license allows others to remix, transform and build upon this work, to non-commercial purposes, as long as appropriate credit is given, and the new contributions are licensed under the same license as the original]

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 STATE OF THE ART

This chapter aims to provide a literature overview of the topics that relate to the main problem that this thesis aims to help solve, which is the lack of cooperation between applications and the service providers of the infrastructure where such applications reside. As such, focus is given on discussing structural network patterns utilized by applications to give them particular properties which are helpful to achieve their use case. Among these patterns, particular interest will be given to three of them for the following reasons: firstly, due to their popularity in the current network paradigm; secondly, due to their potential to optimize traffic that is generated at the application level. Considering this, the patterns that will be discussed are the distributed approach of *Peer-to-Peer (P2P)* architectures on Section 1.1, the quite recently popular *Content Distribution Networks (CDNs)* on Section 1.2, and the classic client-server model on Section 1.3. For each of these, a conceptual analysis is made - more specifically, contextual background, the architecture itself, advantages and disadvantages, and possible use cases. Additionally, there's an examination on how applications that utilize these patterns affect, positively or negatively, the physical infrastructure where they operate on, and where does potential reside for mutually-beneficial cooperative behaviour between these two layers. Following this, Section 1.4 displays existing proposals for increased layer cooperation, and alongside it a discussion on the practical consequences of adopting them. Finally, Section 1.5 gives special attention to the *Application-Layer Traffic Optimization (ALTO)* working group's proposal for a layer-cooperative system, as it is the baseline for this thesis's work.

1.1 PEER-TO-PEER (P2P) NETWORKS

1.1.1 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 properties, among them that it consists of many singular computing elements, the "peers", which have between themselves similar privileges and functions (this contrasts with the client-server architecture, where two different roles exist - the one that provides a service and the one that can consume it - with functionality and control being thus centralized). P2P networks decentralize computational resources as a means to achieve a given task in a way that is inherently different from a centralized counterpart. This decentralized architecture 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 increases its capacity with an increase in its 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 then 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 much more vulnerable to node and link failures, a decentralized one can more easily work around such threats - as all peers can encompass the same server functionality, network services and resources are not provided on a limited set on nodes, but instead can be redundantly deployed throughout as needed.
- **Power decentralization**: As a consequence to sharing server roles and resources, 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 nuances - since many P2P hybrids exist, these properties are not immutable. 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 brings, by its nature, alongside their potentially advantageous properties, also some some potentially disadvantageous ones to consider:

- **Security hazards:** The equal functionality property that P2P networks have give peers much power to influence 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 by any proper orchestration strategies.

P2P applications have had, in the past decades, a mainstream image that is plagued with legality and security issues. Nonetheless, its design 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 with distributed and cooperative algorithms.

Either way, the influence of P2P applications is undeniable: Sandvine [1] published a global Internet phenomena report concluded that BitTorrent [2] alone had in 2019 a global application total traffic share of 2.4%, and perhaps most importantly over 27% of total upstream volume of traffic, and over 44% in *Europe, the Middle East and Africa (EMEA)* alone [3]. Beyond file sharing purposes alone, P2P applications have been recently considered a fitting solution for low-cost content delivery systems in high demand scenarios - for example, in applications such as PPStream [4] in China, which provide television content over IP to large audiences. Similarly, Akamai [5] recognizes the potential of P2P technologies to provide a highly distributed option for serving static content over the network, although it being currently lacking in management and control features [6]. Indeed, P2P Internet video broadcast services - and worldwide static content delivery services for that matter - seem attractive as they are cost-effective and easy to deploy, and are fitting to large scale demands, and thus have the potential to become a more mainstream solution [7].

Concluding, the P2P network architecture has many fitting use cases, and their rather different strategy, compared to the client-server architecture, to achieve its goals gives it many potentially interesting properties for users and *Internet Service Providers (ISPs)*. Considering its large global traffic share, particularly in upstream traffic, and its potential adoption towards the large scale demands of the future, P2P applications are likely to persist and will be in the minds of ISP administrators for the near future.

1.1.2 Architecture

As stated previously, the term "Peer-to-peer" has become very broad and now serves as an umbrella for many different variations of the core decentralized architecture design. Thus, this section focuses not on giving an overview of a single conceptual architecture of what defines a P2P network, but instead of the many existing variations and how they differ among themselves. 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 underlay network. How peers are organized in these P2P networks and how they operate is what distinguishes the many sub-types. Table 1.1 groups known P2P systems in regards to their centralization and structure, similarly to the groupings of [8] and [9], with the latter further distinguishing the protocols in regards to other parameters, e.g., security, reliability, and performance. The rest of this section follows the survey made by the former.

One would expect that all P2P applications would have no centralization at all, since the P2P design ponders functionality spread throughout the network. However, some modifications have been made in some of these sub-types, which shift how much decentralization they really have in practice. Similarly, different strategies exist that dictate the structural hierarchy that resides within the member peers. As would be expected, these sub-types of P2P networks thus possess different strengths and weaknesses, and these can be leveraged to the most appropriate use cases.

Table 1.1: Types of P2P systems (Adapted from [8])

		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

Early versions of Gnutella [10] come as a famous example of a decentralized and unstructured architecture, as peers act with equal functions and privileges, and no inherent structure exists for peers to connect to each other, nor does it for storing or retrieving content on the overlay network. The bootstrapping phase consists of users reading from list containing a set of Gnutella peers - a list that is retrieved 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, and thus a technique of flooding the network with content queries is used until either a reply is met or the predefined TTL value is exceeded, as is exemplified in Figure 1.1.

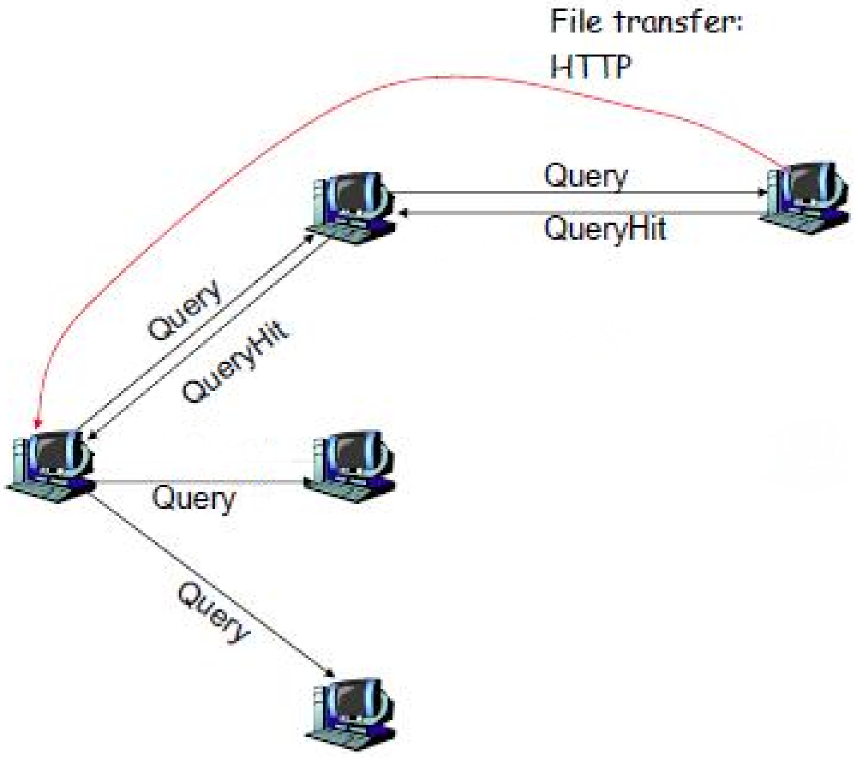


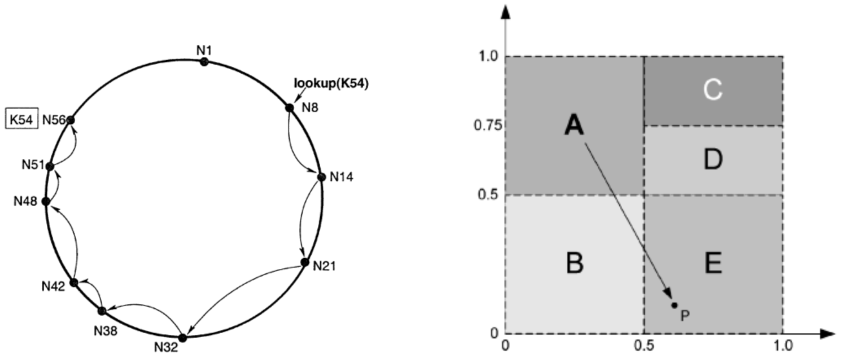
Figure 1.1: Demonstration of Gnutella's file searching mechanism [11]

Partially centralized architectures were defined 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 the network self-configures in situations where such elected nodes either fail or willingly leave the network, and as such 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 [12] 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 - serving 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 employs some non-random and systematic criteria on how the network operates, e.g., how peers organize themselves and where content is stored and how it it retrieved. For example, FreeNet [13] uses the content's hash as a key that is used to query for it, and which in turn 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 like Gnutella does. Many of the structured P2P architectures rely on *Distributed Hash Tables (DHTs)*, which act as a decentralized map structure that binds a given 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 1.2. Specifically, Figure 1.2a displays how 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 it doesn't exist, it forwards the query to their successors, and the process recursively continues. On the other hand, Figure 1.2b display how *Content Addressable Network (CAN)* has the key-space mapped to a virtual two dimensional grid, and its area is partitioned to peers considering a deterministic function, which in turn is used by querying peers to figure out where a given content is stored. A straight arrow from querying node to providing node 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.



(a) Chord file query mechanism [14]

(b) CAN file query mechanisms [8]

Figure 1.2: Examples of structured P2P query mechanisms that utilize DHTs

1.1.3 Effects to the Network Infrastructure

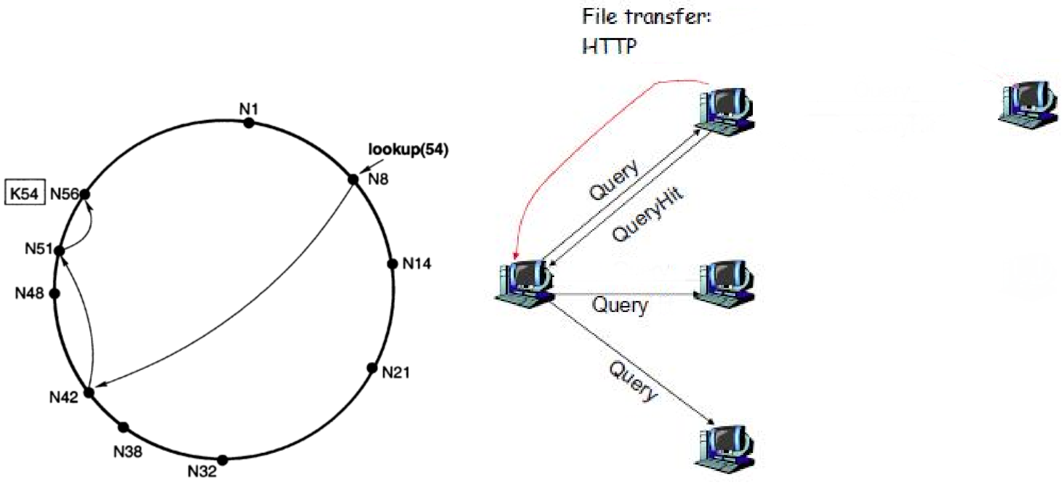
Historically, ISPs have deemed P2P traffic as unideal or even undesirable. Besides the aforementioned illegality precedent that is tied to P2P applications, the overall properties of P2P networks make them unappealing to support - due to the distributed nature of these types of networks, the overall 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.

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 implementation 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, an also unstructured and decentralized P2P protocol, has its architectural decisions fit a very specific use case, as it "emphasizes distributed, reliable, and anonymous storage over efficient retrieval" [15]. 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 Free-Haven 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 other neighbouring peers, as well as react them abruptly entering and leaving the overlay. The usage of key-value mappings with DHTs can also have the potential to be ISP unfriendly, as the hash function can be designed to evenly distribute resources over the peer pool, and thus over the network - whilst such property is certainly advantageous in certain use cases, doing so removes any application context that exists in the content - for example, grouping resources which belong to the same web page with such a method isn't efficient, as they will be individually hashed and spread throughout the network, despite the fact that they'll likely be requested together for each page access.

A first point of improvement is optimizations made in the applications themselves to less degrade network resources. An example of these can be visualized in Figure 1.3. Specifically, regarding Figure 1.3a, a point of optimization in the Chord system would try to reduce the number of query messages per resource by increasing the number of successors a given node knows. That way, the querying node can instead query not for the single successor it knows, but instead by querying for the one who's ID immediately precedes the content's, thus insuring a reduced number of hops to retrieve the message. This consequentially also reduces the total amount of traffic on the network, and improves application times. Regarding Gnutella in Figure 1.3b, a point of optimization would try to tackle the usage of query flooding to locate data, as such flooding grows exponentially and thus intakes a massive toll on network resources. A query flooding system would not be as prejudicial if content was equally scattered throughout the overlay and each given content was a minimal amount of hops away. However, as concluded by extensive analysis of user traffic on Gnutella during its heightened use, nearly 70% of users shared no files and nearly 50% of all responses were returned by the top 1% of sharing hosts [16].

Regardless of the many ways through which P2P systems can operate, e.g., in regards to structural mechanisms and centralization, and even disregarding potential application optimizations, no classic P2P system operates in full understanding of the underlying network topology, nor with a cooperative behavior towards ISPs. The network-agnostic manner under which they operate results in overlay networks which



(a) Chord file query mechanism with more neighbours known per peer [14]

(b) Gnutella file query mechanism assuming reduced resource-to-peer hop distance

Figure 1.3: Examples of P2P query mechanisms when application optimizations are in place

are layered on top of the underlay where they run, as exemplified in Figure 1.4 - as P2P applications are network-agnostic, two neighboring peers could exist in completely different contexts on the common network layer - for example, they could either be connected by a single data link or be multiple network provider domains away from each other.

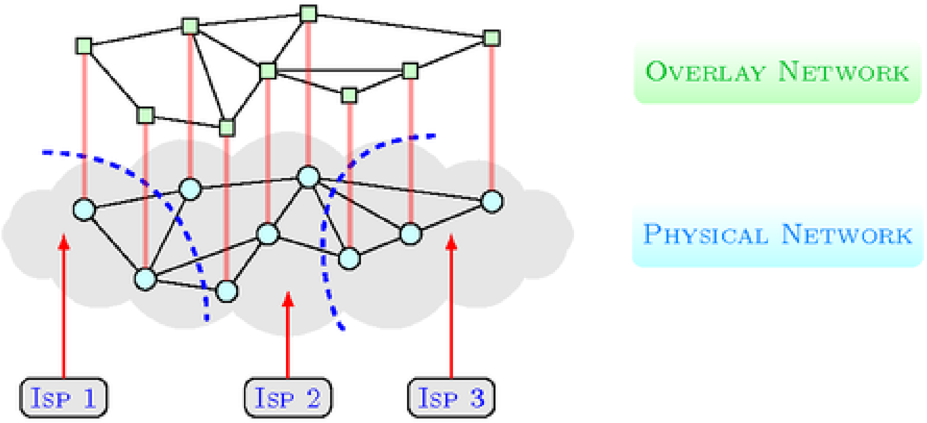


Figure 1.4: Example demonstration of an overlay network and corresponding physical layer [17]

The inability of P2P applications to localize traffic is seen as a big problem - as concluded by [18], ISPs face bottleneck bandwidth pressure in the large scale Internet of the future, with it being in particular from P2P applications. However, an argument is made that an increase in users from such applications is not necessarily harmful, and perhaps helpful, if traffic locality can be boosted. Thus, there is a necessity to ensuring that peers prefer to generate traffic within network locality over traffic that crosses network borders.

A first step towards more traffic locality would be increasing the availability of content inside network boundaries, through means such as cache injections or peer content serving incentives. Given that content resides locally, a second step would be that P2P applications have a network-aware vision of the overlay, which does not happen with classic P2P protocols. This issue is particularly damaging in P2P applications whenever neighbours are selected, and whenever a service exists redundantly in the network and a decision must be made in regards to what peer or collection of peers to consume the service from.

If it is the case that P2P applications are not locality aware, it can easily be seen how this can be an issue. In one hand, for the applications, choosing peers which are not local to the querying peer may result in more time to retrieve the requested contents. In the other hand, for the ISPs, bad network resource utilization can incur in higher operational costs, and may degrade overall network performance in many cases, e.g., by not being able to stop applications from overusing inter-AS links, which usually are network bottlenecks (a conventional wisdom demonstrated in [19]) and which, due to peering agreements with other ISPs, are less desirable to be overused due to peering costs. 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 and current status, such as the inherent network topology, link properties, or scheduled server maintenance.

Attempting to optimize peer selection without a co-operational channel with ISPs would be sub-optimal, as not enough information can be derived with proving alone, and could perhaps even be more damaging with the wrong techniques - consider a peer selection algorithm that chooses the peers with lowest *Round-Trip Time (RTT)* of a probing ping message, whilst having no indication on available end-to-end bandwidth, existing network bottlenecks, or peak traffic hours. Likewise, attributing neighbour pairings via geographical proximity, 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 *Autonomous Systems (ASs)* and thus separated by costly links. Other peer-selection techniques focus on randomly selecting nodes, such as the means through which a BitTorrent tracker selects between redundant peers serving the same file chunk [20], which is simple and resilient to peer churn [21], but as a consequence is sub-optimal on network resource usage as no network consideration exists. It is reasonable to assert 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 met.

Indeed, it is the case that current P2P solutions are ISP-unfriendly. More concretely, [22] shares the view that P2P applications and ISPs are in a tussle, since P2P applications generate traffic which favours the application's needs whilst ignoring the ISP's, which in turn upsets the latter's business model. 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 [21]. [23] 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. Likewise, Gnutella generates traffic which is not ideal, as it may have to cross ISP network boundaries multiple times [24] due to the same fundamental issue stated before - an application layer that operates in disregard to the network underlay it runs on.

As [25] 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 block 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, specially in the face of the increased technical and scale-related challenges of the future.

1.2 CONTENT DISTRIBUTION NETWORKS (CDNS)

1.2.1 Concepts and applications

A 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 content, which has a huge share of the global Internet traffic of today.

Functionalities of CDNs include [26]:

- **Content outsourcing and distribution:** Replicate content throughout the network into edge servers, which are deployed nearby end users. This allows CDN clients to pay for their content to be hosted on these edge servers, and in doing so guaranteeing that it is quickly accessible by end users.
- **Request redirection:** Direct a content request to the most appropriate edge server at a given time. This redirection is done considering relevant network properties, such as geographical locations and current server loads.
- **Content negotiation** Manage the network's properties to fit the needs of its clients through negotiation.
- **Management:** Manage the distribution network, which includes accounting, monitoring, statistical analysis on content consumption, etc. A close management of the distribution network is important for its business model, as, besides being needed for a billing system, allows for a better understanding of the networks' usage patterns, which is helpful information for better engineering the system to most optimally serve content with increased revenue and decreased costs.

The current focus of CDNs is thus to provide content, e.g., web pages, documents, photos, videos, or media-related streams, with high availability and performance. The strategy used by them 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 ([27]). 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. These 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 ([5]):

- A 100 millisecond slower web page loading speed can result with a 7% drop in sales
- A 2 seconds slower web page 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 web page 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 as no surprise that streaming services such as Netflix [28] and Youtube [29], who now reach a global scale and whose utility is highly dependant on their high availability and low transmission delay, routinely use CDN solutions. More broadly, typical CDN customers include media and Internet advertisement companies, data centers, ISPs, online music retailers, mobile operators, etc [26]. Indeed, companies that wish to provide a given service in the web at scale routinely partner with companies whose focus is providing content delivery services, with popular examples being Akamai, CloudFlare [30], or Amazon Cloudfront [31]. Coupled with the promise of highly available and quick content retrieval, these companies also provide other attractive services, such as firewalls and *Denial of Service (DoS)* protection.

The Internet's currently most targeted use being 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. With their great influence over global traffic, active efforts should exist to increase harmonious behaviours between applications that utilize their services and the ISPs, with the goal in mind being network resource efficiency to guarantee that the infrastructure can remain operational and applications can provide a satisfiable user experience.

1.2.2 Architecture

Figure 1.5 represents a high-level conceptual architecture of a CDN. The true power of a CDN comes from its strategically deployed cluster of replicated servers - at a global scale, this implies having them geographically dispersed and located inside or nearby networks with large content demand. The origin server possesses the content that is to be served, and the bootstrapping process has it uploaded into the network, which is afterwards replicated into these edge servers.

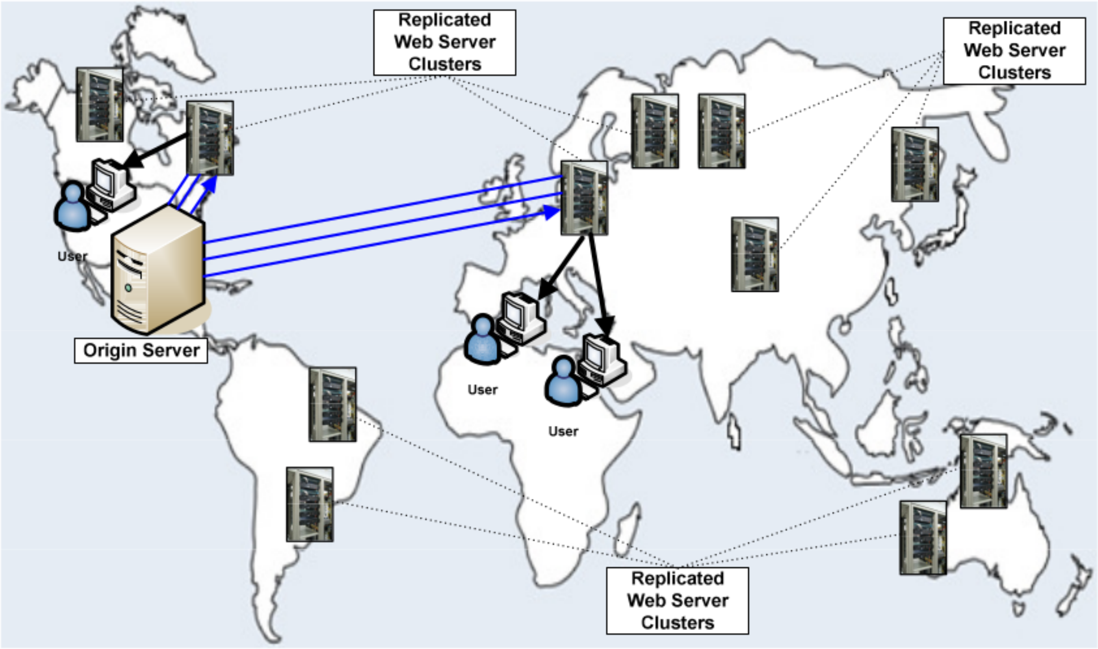

Replicated Web Server Clusters
Replicated Web Server Clusters
User
Origin Server
User
User
Replicated Web Server Clusters

Figure 1.5: Conceptual architecture of a CDN [26]

Figure 1.6 displays how, conceptually, the request routing functionality of a CDN. As can be seen, the request is firstly directed to the origin server, and serves only light, basic content. In the situation where large static content is requested, the origin server redirects the request to the CDN provider, which utilizes a selection algorithm to elect the most appropriate edge server to serve the content to the end user.

The request routing mechanism is the one that redirects a given user to a given edge server. The prevalent approaches are, according to [32], the following:

- ***Domain Name System (DNS)* request routing**: The user must first resolve a domain name to retrieve a content. The CDN's DNS processes the request and,

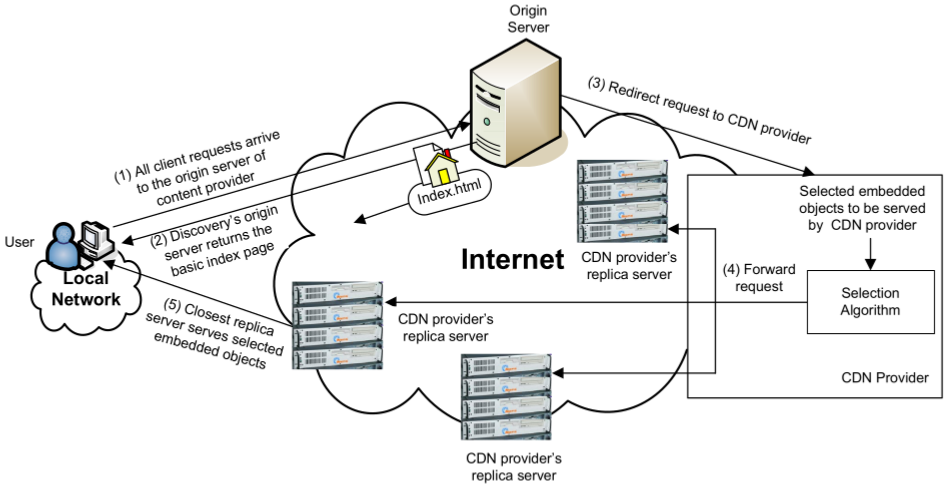

Origin Server
(3) Redirect request to CDN provider
(1) All client requests arrive to the origin server of content provider
Index.html
(2) Discovery's origin server returns the basic index page
User
Local Network
Internet
CDN provider's replica server
(4) Forward request
Selected embedded objects to be served by CDN provider
Selection Algorithm
CDN Provider
(5) Closest replica server serves selected embedded objects
CDN provider's replica server
CDN provider's replica server

Figure 1.6: Request routing functionality of a CDN [26]

utilizing the user's *Internet Protocol (IP)* address, historical measurement information and current server loads, responds with the address of the edge server that seems most fitting to provide such content.

- ***Hypertext Transfer Protocol (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. Thus, different routers can have a given anycast address mapped to different hosts, meaning the ones that are most suitable for that particular router.

These mechanisms are also discussed in [26], but also add:

- ***Global Server Load Balancing (GSLB)*:** Service nodes, consisting of edge servers and GSLB-enabled web switches, are interconnected in a network. Individual nodes possess global awareness of the network, meaning the status of each individual server. With edge servers having more information on the health and status of all other candidate edge servers, and the web switches acting as authoritative DNS servers, the network can enable the support of a global-wide server load balancing mechanism that intakes dynamic server information.

- ***Uniform Resource Locator (URL)* rewriting:** The origin server redirects the end user via dynamically altering the host pages' URL links.
- **CDN peering:** An extension of a single CDN network to multiple, interconnected CDNs which serve content on behalf of others when appropriate. This is helpful to extend the domain reachability of a single CDN, increase fault tolerance, and ability to achieve better performance with more candidate servers to choose and balance loads from.

It is vital that a CDN possesses a clear view of the network performance inside its domain. [26] lists important metrics which are used to measure CDN performance:

- Geographical proximity of end users
- Path latency
- Path packet loss
- Path Bandwidth
- Path startup time
- Path frame rate
- Server load

Means through which these metrics are retrieved by the network include traffic monitoring of end user to surrogate server communications, and surrogate server feedback retrieval via application requests or measurement probings [26] [6].

Having a clear understanding of CDN performance is important for system management in two fronts - firstly, through performance evaluation, by providing the billing and monitoring modules the verification of how the network is faring at its task of caching and delivering content on behalf of clients; secondly, through performance optimization, by providing the logical layer an updated view on network status, serving as contextual input that the system uses to better reason on how to act - for example, in regards to the caching and request routing mechanisms.

1.2.3 Effects to the Network Infrastructure

As previously discussed, CDNs came as a tool to strategically position content on the network in such a way that it can more quickly and more reliably be retrieved by an end-user. These CDN systems are then inherently a means of optimizing network resource utilization in the application layer, and thus are of great interest for ISPs and, if done properly, can be very appealing not only to them but also to end-users that use applications leveraging these networks.

The usage of a single content-providing server - or a limited set of them - which is far away from the content demand, that in turn has a large geographical distribution, is prone to server overloading and path congestion problems if a big enough scale is achieved. The usage of data caches is a classic solution to network inefficiency problems, which is 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, with the ability of 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 - and reducing congestion of inter-domain links - as data caches will reside locally and redistribute traffic away from highly shared network links. It can thus be stated that the relationship between CDNs and ISPs allows for a win-win scenario because efficient network usage has consequentially better service quality, benefiting service providers and applications, respectively.

However, CDNs seem to lack cohesion with the underlying network infrastructure during normal application operations. As stated by Akamai, a leader in CDN services, in their report [6], the large scale and complexity of the Internet, where it takes well over 650 networks to reach 90% of all access traffic, adds to it many challenges to the CDN's role of content delivery. In particular, whilst good investment seems to exist at the first and last mile of Internet access (website hosting and end users, respectively), there seems to be little economic incentive to invest in the middle mile, composed of peering points shared among networks. These peering then become network bottlenecks that are susceptible to increased traffic packet loss and latency, making inter-network communications unreliable, and loose coordination between autonomous networks with internal biases is pointed as a main cause. Due to this, even a well provisioned data center will be at the mercy of the various inter-network bottlenecks that may arrive, and performance takes a hit. In fact, the paper suggests a clean-slate redesign of the Internet as a potential solution to its many problems - besides the peering point congestion mentioned above, inefficient routing protocols, unreliable networks, inefficient communications protocols, and application limitations also add to the problematic - but such a redesign to a massive and highly invested global infrastructure doesn't seem plausible.

In alternative to proper network infrastructural insight, CDNs have to rely on network probing, traffic monitoring, and server feedback, as discussed on Section 1.2.2. Even assuming that these are sufficient, the usage of probing techniques will incur in overhead traffic on the network, with this overhead being exacerbated if many other overlay networks or applications also apply this technique. Similarly, traffic monitoring to extract end-to-end path metrics to end users requires resources and takes time, and may also incur in redundant operations being applied on the network. Such advantage goes outside of the CDN realm, being useful for any overlay-residing application that wants to utilize network probing to be more underlay-aware.

Advantageous as network probing and traffic monitoring mechanisms can be for CDNs to properly conduct request routing and caching decisions, a case must be made for proper application-infrastructure synergy during decision making in the overlay. Much like P2P systems, to infer on network status by measuring it is insufficient when compared to receiving input from trustworthy and authoritative entities that possess privileged network information, such as ISP administrators. Attributing node pairings entirely on geographical data was previously discussed as being a non-optimal way of assessing node selection at the application layer in Section 1.1.3, and the same would apply in the case of CDNs, where edge servers are paired with end-users. 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 potentially beneficial, can certainly be improved upon if it used additional, hard to retrieve data that only ISPs or other privileged infrastructural entities could possess, and which are at position to guide applications in the infrastructure they deeply possess knowledge on.

There indeed seems to be a consequential coupling between overlay decisions in the CDN systems and the underlying infrastructure. If the CDNs were not to take ISP input when redirecting clients, suboptimal choices would be made that would be prone for bottleneck congestion, and if, in the other hand, ISPs were to employ only their own biases in content request redirection, user-level application *Quality of Service (QoS)* agreements might not be met. [33] states that this lack of awareness to network status is indeed problematic for CDN systems, listing end-user mismatching to edge servers based on dubious DNS-based location binning and resource consuming, non exact methods to detect bottlenecks, as well as lack of agility in server deployment in ideal locations. This is a view shared by [34], which adds that these problems reside in a shared medium that raises the opportunity for cooperative behavior that would enable better application performance and optimized ISP resource utilization. In fact, Akamai themselves have formed content delivery strategic alliances with major ISPs, with AT&T [35], Orange [36], Swisscom [37] and KT [38] among them [33], which hints at this type of partnership being the norm for content distribution technologies.

ISP input permits applications to act in a more network-aware fashion than without it - whereas pairing overlay nodes or deploying edge servers in terms of geographic distance, RTT distance, or any other metric, may give further decision power than a purely network-agnostic overlay system, proper guidance by ISPs could help pairing based on more complex metrics that, besides the aforementioned ones, also consider ISP objectives, e.g., to minimize network distance, avoid bottlenecks, locate content caches, etc. A more efficient Internet can be obtained if many other overlays coordinate their efforts with the ISP, which in turn can now orchestrate its traffic in a way that would previously be generated with no prior guidance, and that can now be engineered to maximize network resource utilization and, consequently, application performance.

1.3 CLIENT-SERVER MODEL

1.3.1 Concepts and applications

The Client-Server model is a classical way of attributing roles in the network. Whereas the P2P architecture blends server and client roles into each node, this architecture delineates two distinct ones - the client and server - and their purpose on the network, as Figure 1.7 shows. With it, the service layer is centralized onto the server nodes and the only role expected of client nodes is to request services from them. Operating with opposite philosophies from the P2P architecture, it is to be expected that the advantages and disadvantages should too be contrasting - by centralizing the services, maintenance and general administration of serving nodes become attainable, and by limiting the number of servers to a select and trustworthy few, instead of scattering that functionality to all nodes in the network, greatly minimizes security risks. On the other hand, the drawbacks are also apparent and mirror a distributed solution - issues of scalability, resilience to failure, and resilience to monopolization of power arise from the decision to engineer an application that creates clear boundaries between client and server.

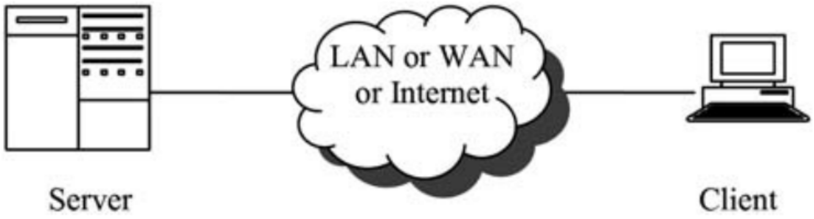
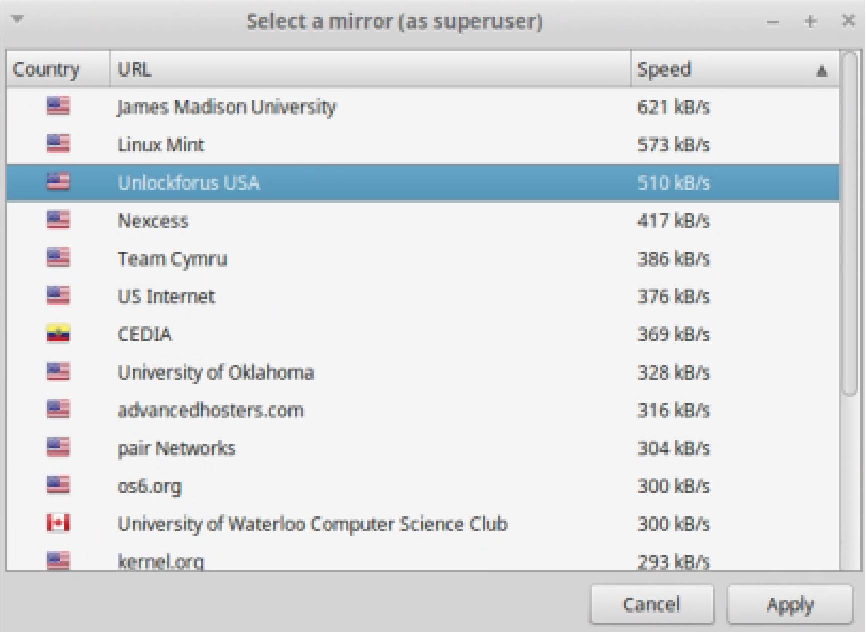


Figure 1.7: Client-Server architecture [39]

This classic architecture has seen a lot of adoption by applications, with use cases that include serving content via HTTP or *File Transfer Protocol (FTP)*, or enabling e-mail communications [39], to name a few. Adding to this, the client-server approach also serves as a direct alternative to the P2P one in many fields, e.g., file transfer or media streaming, with their respective advantages and disadvantages needing to be weighted out for a proper choice to be made.

As an attempt by service providers to still operate a client-server model whilst reducing its associated downsides, given techniques were created and employed. One of which, by the name of CDN, is of particular importance in this work and as such has its specific overview in Section 1.4.2. Some others will be discussed below.

Server mirroring, one famous technique, is the act of continuously synchronizing a server into its replica, or mirror, 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 servicing a given client, as long as it periodically checks up with the primary server for 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 reduced load and smaller end-to-end message delay to the user. Figure 1.8 shows an application of this, where multiple server mirrors exist to deliver software packages to the Linux Mint [40] distribution. The user has the choice to manually select one of these mirrors, and ideally chooses the one that is most fitting to them.



Country	URL	Speed
🇺🇸	James Madison University	621 kB/s
🇺🇸	Linux Mint	573 kB/s
🇺🇸	Unlockforus USA	510 kB/s
🇺🇸	Nexcess	417 kB/s
🇺🇸	Team Cymru	386 kB/s
🇺🇸	US Internet	376 kB/s
🇪🇨	CEDIA	369 kB/s
🇺🇸	University of Oklahoma	328 kB/s
🇺🇸	advancedhosters.com	316 kB/s
🇺🇸	pair Networks	304 kB/s
🇺🇸	os6.org	300 kB/s
🇨🇦	University of Waterloo Computer Science Club	300 kB/s
🇺🇸	kernel.org	293 kB/s

Figure 1.8: Linux Mint prompt to select a software repository mirror

Server load balancing is another popular technique to reduce server-client downsides, and it refers to the act of distributing tasks over a set of computing nodes as a means to make the service more efficient, since a server with little processing load is bound to respond quicker than one with more, and situations of overloading become less common as it is now spread out over multiple processing units. Load balancing algorithms can be categorized, according to [41], in two categories:

- **Static**: Does not take current system status into account, instead acting upon static rules that define the load type, in regards to, for example, processing power and memory requirements. A round-robin or random approach of load attribution are part of a static approach.
- **Dynamic**: Load is distributed at run time, considering dynamic information about the system that is continuously being collected.

A static approach can be compelling due to its simple nature that makes it easy to implement and much faster in comparison to run. On the other hand, a dynamic approach is more complex by definition as it requires the additional infrastructure required to collect and compute upon the system statistics. As it might be expected though, a load balancing scheme that attempts to optimize resource utilization considering real-time information about system status and performance is bound to have bigger potential at effectively selecting which server should handle a given request since more factors are known to better aid the decision, including the current server loads in the server cluster, their available resources, network conditions, etc.

Several methods exist through which to implement load balancing. For example, DNS-based approaches perform the load balancing at the name resolving stage, where the IP address response is given after selecting an optimal server, such as the proposal in [42]. *Software Defined Networking (SDN)* solutions, as another example, leverage a control pane that is responsible for intercepting server access calls and activate the load balancing mechanisms that redirect to the selected server, as happens in the proposal in [43].

1.3.2 Effects to network infrastructure

As a base method, the utilization of the client-server architecture is not at all recent to ISPs. Quite the contrary, it is the norm throughout the decades to provide a service by using a set of single-purpose server machines. Intuitively, using a limited number of servers to respond to clients needs will have problem scaling, as increased demand will overload servers and create path congestion, hence the need to employ some of the strategies discussed above. These strategies come from the necessity to fulfill both application and ISP requirements, i.e., to achieve proper QoS levels and infrastructure resourcefulness that helps the general good function of the network, respectively.

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. 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, as commonly happens when downloading software, for example, seems problematic, as application-generated traffic is not optimized. In fact, when considering the Linux Mint software package distribution discussed above, despite there 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 [44]. 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.

Load balancing is too a point of great interest in the task of application-layer traffic optimization. The task of attributing a current request to a pool of available redundancy workers is a common strategy to help scalability, and how it is done has concrete network consequences as it has the power to shape great amounts of traffic in ways that can be more or less preferable for ISPs. It could then be deduced that a load balancing strategy has better chances of being efficient if it fully aware of both server status and network conditions. Doing so with ISP cooperation would reveal insight in some network status information and preferences that could not be retrieved otherwise, and, much like what was argued for in section 1.2.3, could severely reduce overhead traffic created by probing and measurement strategies if such data was centralized in an entity with full network knowledge.

1.4 TRAFFIC OPTIMIZATION BY APPLICATIONS AND LAYER-COOPERATIVE APPROACHES

This section serves to display 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 the increasing tension between applications and service providers, it comes with 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 given to each one of the layers, and different levels of cooperation. Along with their description, their review is made in regards to what the proposals state their impact is to applications and the infrastructure, as well as the accompanied advantages and disadvantages.

1.4.1 Peer-to-peer applications

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 1.9 represents a grouping proposed by [25] where such mechanisms are ordered in agreement with how much involvement the P2P systems and the ISPs have.

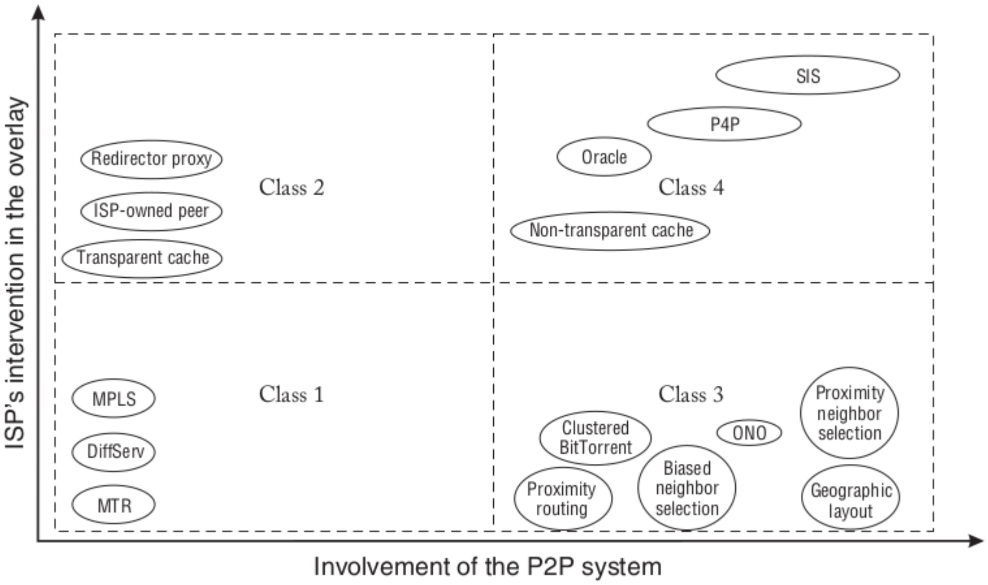


Figure 1.9: Approaches to decrease tension between P2P applications and ISPs grouped by their involvement [25]

In a more detailed fashion, these classes are as follows:

- **Class 1**: There is not much - or any - 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 *Differentiated services (DiffServ)*, *Multi-Topology Routing (MTR)* and *Multiprotocol Label Switching (MPLS)*. These classes of methods do not fix the underlying application behaviour, 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 affect 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 the 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 haven been proven to work , as concluded in [25], and put into practice, for example, in [45], [46], and [47], 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. Additionally, in [48], with the injection of special nodes on the Gnutella overlay which interface with the base protocol nodes but with the added caching and load-balancing mechanisms, in the attempts to alleviate the great amount of "free riding" that exists in Gnutella applications - as discussed in Section 1.1.3 - by minimizing the total amount of query floods and more evenly distributing content throughout the network for increased infrastructure resourcefulness. However, this class of mechanisms is not without its 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 when 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. It's also important to note that since no application-layer input exists, this approach could be one-sided in the sense that only ISP needs are favored without directly considering application needs.

- **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. Peers probe the neighboring network elements as a way to get more familiar with connection properties, and act on these probings during operation, e.g., when choosing neighbours to construct the overlay network with, or when choosing from whom to request 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 knowledge scope of the network and its needs, and as such these application optimizations can end up being more hurtful than helpful. For example, consider a 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 preference for it to supply the service would incur in more infrastructural costs. The paper describes this class as a "win-not-lose" situation, meaning that while the P2P system can, in the right circumstances, improve their performance via measurement-oriented strategies, the ability to act in a way that positively affects the underlay, without any feedback from ISPs, cannot be guaranteed. Such an example of class 3 mechanisms could be seen in [21], which improved BitTorrent's download performance and even managed to reduce ISPs' backbone and cross-ISP traffic. The technique consisted of having peers send traceroute measurements to the tracker, which in turn grouped the peers 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. Another example of this is [49], 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 reduced 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. As a final example, [50] proposed a node binning scheme that groups nodes of similar orders of magnitude of RTT values to pre-defined landmarks, and utilized such scheme for topology-aware overlay construction mechanisms in some unstructured and structured P2P overlays. Results allowed to conclude that even surface-levels topological information is advantageous and can significantly improve application performance.

- **Class 4**: Full and active cooperation exists between the ISPs and P2P systems. The role of the ISPs is to provide information and guidance, and P2P systems let themselves be influenced during operation. 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, [51] proposes an oracle that receives as input a list of candidate peers that the querying peer is considering connecting to, and ranks them by client connection proximity. Such method was tested in a simulated environment and proven to decrease negotiation traffic and improve scalability of P2P networks. Similarly, [46] proposes a framework for programmable trackers in a BitTorrent scenario, providing an interface to directly define tracker behavior which, given ISP input, can provide a collaborative environment between infrastructure administrators and the BitTorrent users in an attempt to, among other things, localize traffic. The functional intent of the oracle pattern is that he possesses privileged network information and acts on it to provide guidance to querying applications, and thus has the power to impose policies and optimizations unto applications, e.g., pair peers which are the least number of network hops apart via a Dijkstra algorithm using link costs derived from network-related ISP insight. Another approach that could be used by the oracle proposed in [52] 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 ISP biases as its preferences by modifying such algorithms to, for example, minimize usage of costly links (such as inter-AS ones, and subject to peering agreements). The ALTO working group - whose work this thesis attempts to materialize into a working system 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.

1.4.2 Content Distribution Networks

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 as no surprise that this application 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.

[53] considers a CDN built to deliver video data where some given set of content exists redundantly in many edge-servers, and presents 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.

Some simple, flexible and scalable techniques exist that utilize no ISP input. For example, [50], mentioned in Section 1.4.1 for its P2P overlay construction with a binning technique based on RTT measurements to landmarks, also utilized such binning technique for improved server selection. Similarly, *Internet Engineering Task Force (IETF)* tackled application traffic optimization via multi-CDN cooperation, and devised a problem statement in regards to *Content Distribution Network Interconnection (CDNI)* [54], which outlines the efforts required to specify a set of interfaces that allow for the interconnections of many CDNs, with the added benefits that a multi-CDN system, over an individualistic one, will have better properties, e.g., in regards to availability, coverage, and supported capabilities, as well as better QoE for the end user, and reduced delivery costs for the service providers. The four devised interfaces - CDNI Control interface, CDNI Request Routing interface, CDNI Metadata interface, and CDNI Logging interface - are all to be operated at the application layer, and the group states that no new application protocol needs to be devised. Instead, existing protocols could be leveraged, e.g., HTTP, Atom publishing protocol, *Extensible Messaging and Presence Protocol (XMPP)*, and in particular to the CDNI Request Routing interface, the ALTO protocol could enable CDN server footprint retrieval.

Centralized network measurement repositories for wide consumption were tackled in project such as *Internet Distance Map Service (IDMaps)* [55] and *Global Network Positioning (GNP)* [56], that describe architectures for a global distance estimation service, leveraging measurements made by specialized nodes that retrieve raw network data, and heuristic provide scalable and functionally reliable path costs in metric such as bandwidth and latency. These consist of systems that centralize and share network probing results to querying entities, thus minimizing overhead traffic on the network. Such advantage goes outside of the CDN realm, being useful for any overlay-residing application that wants to utilize network probing to be more underlay-aware for application optimization.

[34] argues for the advantage of CDN-ISP cooperative interactions and overviews three possible strategies that will be now discussed briefly: *Provider-Aided Distance Information System (PaDIS)* [57] is a system deployed and controlled by ISPs that monitors the network by listening to *Exterior Gateway Protocol (EGP)* and IGP messages and contains a privileged view of the topology and its status. It provides a service that ranks host-client pairs in regards to, for example, delay, bandwidth, or hop count, and experimental testings concluded that the download times of content provided by CDNs that utilize PaDIS could be improved up to a factor of four, and generally gives much flexibility for ISPs to engineer traffic; *Content-Aware Traffic Engineering (CaTE)* [58] is designed in a similar manner to PaDIS but requires no client-side configuration, and experimental results concluded that network wide traffic was reduced by 15%, link utilization was reduced by 40%, and user-server performance generally increased;

Network Platform as a Service (NetPaaS) [33] was devised to fulfill two key enablers in a fruitful CDN-ISP collaboration - user-server assignment, as it was tackled in the previous two examples, and server allocation, i.e., where should a CDN deploy its servers and its contents. This service, besides having the advantages of increased application performance and better ISP traffic control that were also mentioned in the previous two solutions, also facilitates the task of server allocation for CDNs, reinforcing the discussed advantages and optimizing CDN operations. Still in the topic of CDN's edge server selection, [32] suggests an SDN-oriented solution that combines the performance of DNS load balancing with the low management overhead of IP anycast. Load balancing is performed at the SDN control layer by applying collaborative efforts between the CDN and the ISP. 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 of SDN solutions.

1.4.3 Server-client applications

Attempting to optimize web server selection, [59] 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 proposes a selection based on QoS measurements, from which three types are distinguished: a static method, such as choices based on least number of hops to server (which is unlikely to change); a dynamic method, consisting of network probing to assert, for example, RTT values to the servers; and statistical methods, which decide based on a larger set of measurements previously made in various points in time. Utilizing the latter method, RTT measurements and web-related request benchmarking is made, such as time to establish a *Transmission Control Protocol (TCP)* connection, elapsed time from an HTTP GET 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, [60] proposes a method of server mirror selection which is better optimized than the more popular approach of giving the user the selecting choice. 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, [61] 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.

1.4.4 Summary

Concluding, application traffic optimization does indeed to be a common concern for P2P, CDN, and Server-Client systems, as it improves application performance. Indeed, potential to improve exists if attempts are made to better comprehend current network status to aid application decisions, and so is made by realizing more about the underlay, whether by probing it, or retrieving that information from - or delegating decisions to - authoritative and generally trustworthy sources that keep both interests in mind. A fully mutual cooperative scenario seems much more efficient than one sided approaches. Considering ISPs, the ability to directly impact application behavior lets them engineer traffic at a more fine-grained level, that would be impossible without it. Considering applications, at worst, resorting to ISP guidance would help minimize the amount of redundant network overhead generated from all kinds of applications monitoring the underlay for their status, and at best will allow better insight that only the ISP itself could provide, with its privileged intimate knowledge of the infrastructure and centralized monitoring structure that gives information about historical traffic patterns. Thus, a win-win scenario is theoretically possible, a result much valuable assuming an existing cooperative infrastructure and voluntary participation from both layers.

1.5 APPLICATION-LAYER TRAFFIC OPTIMIZATION (ALTO) WORKING GROUP

1.5.1 Context and Motivation

Acting on research indications that improved peer selection algorithms based on ISP-provided information could help reduce infrastructural costs and increase P2P application performance, the IETF devised working groups to explore possible standardization in the area of layer-cooperation [62]. Among them is the ALTO working group, whose domain is traffic localization.

The ALTO [63] working group designed an HTTP-based protocol whose function is to allow hosts to query privileged servers on network information. The IETF-devised working group's project has gathered much academic interest, e.g., [62], [57], [61], [18], and [25], as well being suggested as an appropriate framework to help fix various problems, e.g., [58], [57], [32], and [54], and these form a subset of a larger current preoccupation with the underlay-overlay tussle and the attempt to find means of collaborative layer effort. The envisioned scenario of the service provided by the ALTO architecture, as can be seen in Figure 1.10, considers both the physical and application domains - the underlay and overlay, respectively. The ALTO service is provided by some oracle, which in turn needs to be supplied with network information that can take many forms - topological structure, routing costs, static policies, etc. - and, most importantly, such data is to be fed by an ISP or such other authoritative entity that contains truthful and relevant network information that the oracle could deem useful in aiding its clients. Using Figure 1.10 as an example, consider that "Peer 2" wishes to retrieve a given resource with the application, and after probing the overlay network - via querying a tracker, using a flood of peer pings, or some of the means utilized by structured P2P networks - the peer locates "Peer 1" and "Peer 3" as possible candidates to serve the content it wants to retrieve. Aware of the fact that choosing whom to consume a service from has impacts on both application performance and network resource utilization, "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 the needs of the application - because different applications could have different QoS metric priorities in mind, like a media stream with low delay needs or a file sharing application with focus on high bandwidth availability. The ISP is then be in full control of engineering how the traffic from this resource transfer will flow, and can steer "Peer 2" in favoring "Peer 3" - since they reside in the same physical network domain, this would improve infrastructure resourcefulness and there would be no need to make use of peering links that interface with external regions. 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, a common goal for the underlay and overlay parties since it is a win-win scenario.

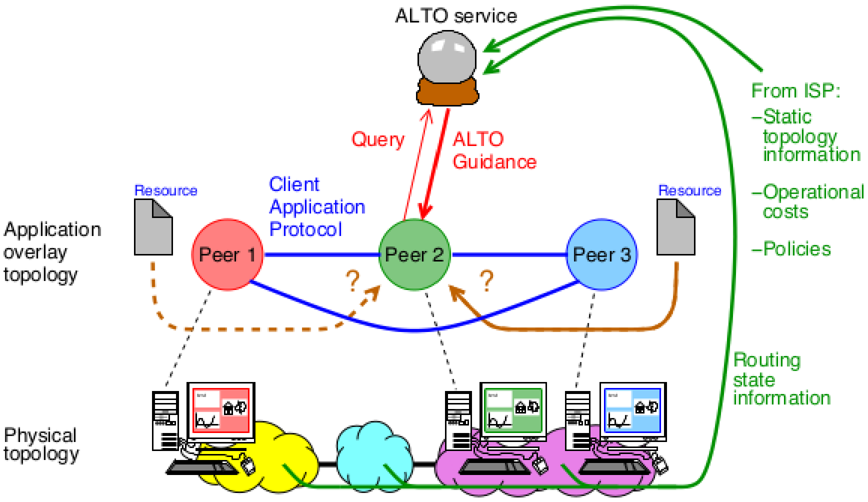


Figure 1.10: ALTO scenario [62]

Despite its origins lying in the efforts to localize traffic in P2P applications, the ALTO protocol and its encompassing system is now being considered in other fields, to be now further discussed.

A first area of interest is CDNs, most specifically the on-going works in extending the base protocol to implement the CDNI Request Routing Footprint & Capabilities Advertisement interface [64], which is a subset of the CDNI standard [54] that aims to allow upstream CDNs to query known downstream cdn for their willingness to accept content requests on their behalf. 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 download CDNs (resources, footprint, load), which they provide themselves, and that allow the 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 wishes to improve its routing decisions to better decide on which other node to select, by using information that is hard, or nearly impossible, to independently retrieve. Regarding CDNs, the querying node is an upstream CDN server that wishes to resolve their content requests by attributing the client to the most optimal downstream CDN servers, where the content resides. At a more abstract level, this is similar to the use case discussed above where P2P required assistance to more optimally select peer connections.

Edge computing, similarly to CDNs, uses a paradigm of flexible service distribution that enables deployment closer to the end user for better performance, and thus is inherently effected by network status. Current work is being made on how ALTO can be leveraged to aid the task of deployment of functions or applications in the network edge [65]. Much like the previous example, ALTO is being used to guide an application in a decision with impacts on both layers - by querying the ALTO server, the client can retrieve information that regards to *Points of Presence (PoP)* where functions/applications can be deployed, such as cloud computing provider's available resources, e.g., *Central Processing Unit (CPU)*, *Random-Access Memory (RAM)*, or storage, but also network information that pertains to the outside of the PoP, mainly network connectivity metrics, e.g., end-to-end bandwidth and delay, and routing costs. The utilization of the ALTO protocol in this context would allow edge service clients and providers, as well as ISPs, the ability to combine efforts as a means to optimize edge computing deployment that considers the current network status, and doing so would thus result in benefits for both end-users and infrastructure maintainers.

More broadly, current work is also being done in specifying abstract network entities path arrays between points [66] and time-sensitive cost values [67], both of which share higher insight of the network, at the discretion of the ISP, as a means to provide even more context to applications about the infrastructure, such as identifying potential path bottlenecks and high traffic time peaks, respectively, and thus improve the ability of applications to optimally generate application traffic on the network.

A mode of operation where applications no longer operate in disregard of 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, 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 measured 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 deems advantageous, such as belonging to the same AS or geographically close by, thus minimizing the amount of otherwise redundant probing required by all application entities that wanted some network status information. ISP-guided counseling, however, does more than centralize measurements that could be done by the application itself. Besides containing measurements that could only be retrieved by the ISP itself due to its privileged access to the network, such as IGP packet inspections or secure *Simple Network Management Protocol (SNMP)* queries, by handing over the decision-making process to the service provider entity, it is given power to better steer traffic in a way that favors internal policies and strategies, such as peering agreements, current traffic flow of other applications, known bottlenecks, etc., that could not be deduced by the applications alone. Thus, in the decision of how to generate application traffic, the responsibility should reside in both the application and the infrastructure as a way to benefit all relevant parties, i.e., the end users, the application stakeholders, and the service providers. ALTO protocol serves an enabler of a mutually cooperative layer interaction system that, by becoming the standard, would aid towards a sustainable life-cycle of the Internet.

Finally, standardizing an architecture and related protocols for a clear problem domain could help a large subset of similar issues, since a well defined and tested specification would exist, thus allowing many applications to leverage the ALTO protocol's functionalities to their needs, not requiring further cycles of development for a specification when one already exists. Also, the attempt to standardize the oracle pattern is helpful as it joins forces from many different domains which share common problems - many of which were exemplified previously - into a single specification. A widely accepted and used solution can evolve from a combined effort, and would target issues such as security and scalability, creating a single point of convergence that is mature enough to be adopted with confidence, accelerating the transformation of the Internet as individual players would not need to develop their own specifications.

1.5.2 Architecture

The high-level conceptual ALTO architecture can be seen in Figure 1.11. 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 and relevant entities, as could be derived by routing protocols, ISP-specific policies, historic measurements, and feedback provided by third parties regarding application performance on the network. Two protocols can be seen as part of the general architecture:

the provisioning protocol, which is not currently contemplated by the ALTO working group, should specify how information is provided to the ALTO server; the ALTO protocol, which is the main focus of the working group with the same name, specifies server-client interactions as a request-response interface for retrieval of network attributes. The ALTO client is the main consumer of the ALTO service, 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, with some potential use cases discussed previously. An ALTO client could be seen as any entity which is able to interface with the ALTO protocol with 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 1.10), but a tracker could instead take that role, enhancing its ability of assisting peer communication by having an embedded ALTO client that would then act on behalf of peers when querying for ISP insight as to provide an optimal peer pairing. Using the tracker-oriented ALTO client approach would minimize needed P2P client protocol modifications and thus facilitate integration with currently existing applications.

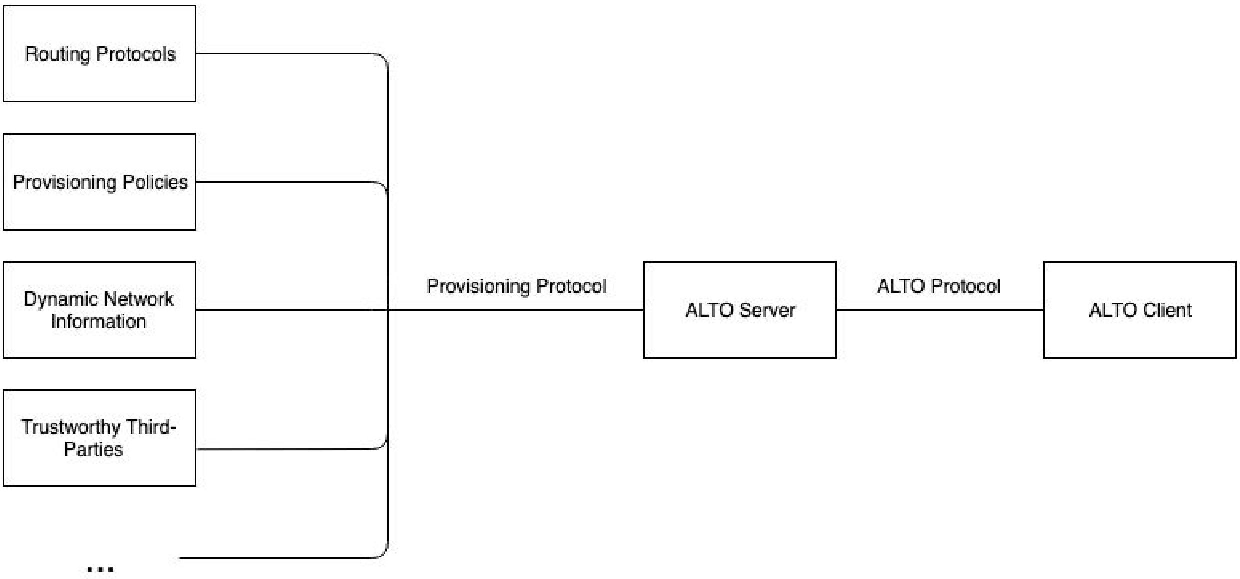


Figure 1.11: ALTO architecture (adapted from [68])

The ALTO services contemplated by the working group can by visualized in Figure 1.12.

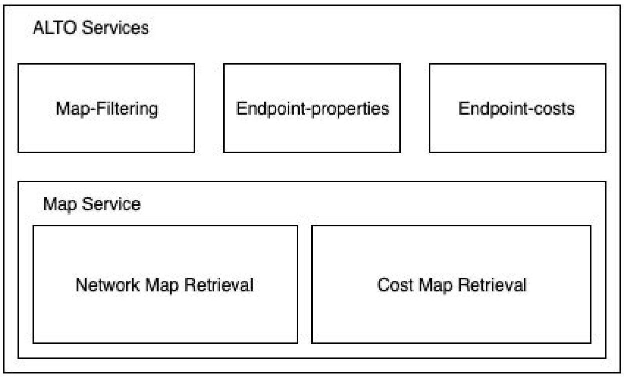


Figure 1.12: ALTO services (adapted from [68])

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 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 ASs, etc., and attribute properties to the aggregate, instead of the endpoint. This is advantageous not only for scalability reasons - since it can compress information - but also because it allows ISPs to abstract network endpoints into groups, thus ensuring privacy of network topology details whilst maintaining useful network guidance, as the ISP has full control of how endpoints are aggregated, and consequentially how traffic is engineered since this changes how clients interpret resources.

A second 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, [69] is currently specifying more concrete metrics, with many associated with 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 excessive infrastructural details 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 an *Structured Query Language (SQL)* database), and only these are transmitted.

Finally, the last two services focus on mappings that regard to specific endpoints, instead of abstract mappings that utilize PIDs. An endpoint is currently identified by one of the following: IP address, *Media Access Control (MAC)* address, or generic 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 addresses and not abstract collections. The ISP has thus the ability to work with abstract aggregates or specific endpoints, showing as little or as much network information as it deems fit.

As could be seen, the ALTO project specifies an architecture for sharing 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 as a means to 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 [70], should not be easily obtainable by the clients themselves - such as immediate end-to-end delay - and should be variable on a longer timescale than the instantaneous kinds that are seen on, for example, congestion control mechanisms, as the frequently resulting querying traffic 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, historical application feedback, etc.

This project is, at time of writing, still on-working, with many drafts being created and updated as the ALTO project matures and increases its domain applicability. These are, however, relating to service extension and deployment, since the main architecture, protocol design, implementation guidelines and security analysis are fully published into their respective *Request for Comments (RFC)* documents, serving as pillars for this work, and the ongoing efforts will serve as inspirations for potential extensibility.

1.5.3 Viability

1.5.3.1 Security

Given the nature of this system - the trading of sensitive network structure information that can alter application behavior - it is quite apparent that its design and implementation are not without challenges from a security perspective. Indeed, the working group published discussions regarding security preoccupations at the development and deployment stages of the ALTO system [70] [68] [71].

Utilizing the "STRIDE" threat model [72], the main threats to the ALTO architecture can be summarized 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 outside their permission. Finally, spoofing of a provider of network status that could feed information into the server to be spread to applications, possibly misleading them in the same way an ALTO server spoofing could, but by proxy.

- Tampering of data to mislead either ALTO servers or clients. If some unauthorized and malicious party can retrieve data that is in transit or storage 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 the latter, from a seemingly trustworthy party, untrustworthy data. This would result in the same issues that could arrive from spoofing threats. Tampering could also occur in input forms in the server-client or server-provider interface with potential to inject malicious code execution.

- Repudiation of being the source of some network information, whether it be by a third party that volunteered the data or the ALTO server itself, which would make it difficult to neutralize and attribute culpability to incorrect or malicious sources, jeopardizing the legitimacy of the provided network information.

- Information disclosure in the form of ALTO resources being made available to entities that were not contemplated to access it. These resources could give malicious parties insight on network topology status as well as the ability to derive clients' network usage patterns by observing what kinds of resources they attempted to retrieve at a given moment.
- DoS of the ALTO server through request flooding beyond its capability, which would severely hinder - or even negate - its ability to serve legitimate users. By proxy, service denial of external entities can also happen through the manipulation of ALTO resources themselves - leveraging the system's potential to guide traffic, if a given resource is manipulated in such a way that unreasonably favors the preference of a specific subset of servers, these could be selected 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, resulting in the previous threats being heightened.

Many of these threats are standard preoccupations for most computing systems and could be solved with state of the art solutions which are well proven and tested, as indeed states [68]. However, regarding threats of information disclosure - whilst they can be negated with in-transit encryption, what is done with this information the moment it reaches the client is hard to control - situations may arise when a client with proper resource permissions shares, intentionally or not, sensitive network information with other users who may or may not have proper clearance, in interactions outside the ALTO architecture. Furthermore, many authenticated clients with different permissions could share information among themselves which they retrieved legitimately, to get a illegitimate complete view of the network structure. Thus, individual clients could internally collaborate outside the system to bypass access control measures applied inside it. As such, it is firstly important for the ISP or third parties to carefully plan on what information they are comfortable with sharing, knowing that it may be susceptible to future disclosure outside the secure domain. Possible solutions to minimize these threats include:

- Reduce the granularity of the provided data. Intuitively, the less granular and precise the shared information by the ALTO server is, the less valuable the resulting application guidance will be, and thus a balance would have to be found between layer cooperation and ISP privacy. One example is the usage of network groupings by PIDs instead of mapping information to concrete endpoints, working with network status around abstract entities. Another possible mean to reduce information granularity would be to utilize ordinal cost values, which instead of specifying a concrete metric, e.g., bandwidth in bits per second or packet loss in percentage, the server would give a relative preference rating with lower costs meaning lower preference. In both examples, the granularity of network information transmitted to the client is several levels higher in abstraction than the actual physical layer, and this could reduce the flexibility of applications to optimize traffic. However, they can still provide acceptable flexibility without impacting ISP privacy, acting as a much needed compromise.

- 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, network status resources could only be provided to authorized cooperation-oriented trackers in the BitTorrent protocol, which would in turn use this information to provide customized replies to clients without the need to change the base protocol. Similarly, information relevant for user-server assignment could only be provided to authenticated CDN control nodes, who'd use among themselves a private virtual domain to share information about user-server connectivity and server status that would otherwise be inappropriate for any other type of user to retrieve. This is still, however, worthy of further threat analysis as restricted information could still leak outside of the system - beyond the means of spoofing discussed previously, seeing how a system behaves with ALTO guidance can give - albeit limited - insight into SP bias. To see this, consider how a BitTorrent peer could continuously query a tracker with carefully crafted parameters - such as source address and candidate peers - and attempt to derive information from the resulting action, or similarly how a end-user could utilize similar parameter modification to observe the edge server selection mechanism in action.

- Utilize terms of agreements that are to be enforced on every querying client, stating that network status information does not get used beyond its original purpose, prohibiting sharing. Although a potentially helpful mechanism to dissuade malicious users, it can be deemed impractical to apply, especially considering the scale at which this information could be shared. Thus, utilizing such means should be applied at a case-by-case situation and it should not replace ISP discretion and server resource maintenance to ensure a given standard.

1.5.3.2 Privacy

Privacy concerns are also very prevalent in the ALTO system, being an ubiquitous talking point in most of the working group's problem statements and protocol specifications. When an ALTO client queries a server for one or more network status resources - in the attempt to optimize the application traffic it will generate in the near future - certain parameters can be passed to the server that can make the response be more personalized and contain more granular information. For example, a real-time P2P media-streaming application seeking ALTO guidance to help choose among a list of candidate streaming peers may wish to include in its query helpful parameters such as the peer list itself, the desired QoS, and the network position of the querying client itself. Indeed, these and more patterns will help increase the effectiveness of the ALTO server's guidance in helping the client application achieve its goal, but such happens at the expense of potentially allowing an ALTO server to infer on user pattern statistics. Even assuming that the previously discussed information disclosure threats are nonexistent in the ALTO system, privacy concerns can arrive from client applications because the resource queries they need to produce can contain information about what the client either will or wants to do. This is recognized by the ALTO working group as a possible concern [68] [70]. In response, they state that the clients should firstly be cognizant about the potential tracking risk that is associated with the usage of the system and, as an attempt to make tracking harder, they could disable HTTP cookies and/or opt for more vague query parameters, e.g. by randomizing some bits on endpoint addresses or simply using more broad addresses, whilst being aware that the helpfulness of query results may vary with increased parameter obfuscation.

Very much like client privacy, ISP-related privacy is also considered by the working group. PIDs were created as a means for ISPs to abstract network components as a collection of single network endpoints with similar properties, helping them not to disseminate network information that is too sensitive, and in turn also allows clients to make queries based on these identifiers and maintain a higher level of privacy. An ongoing proposal for protocol extension includes path vectors [66], that aim to represent information on the intermediary hops from a given source-destination pair, and each of these hops is represented as an *Abstract Network Element (ANE)* that, similar to PIDs, give ISPs the ability to under or over-abstract the topological representation that gets published to clients, giving more options to balance guidance usefulness with provider privacy. Other solutions could also be considered depending on the needs of the clients and the direction of the project as a whole. For example, the servers themselves could operate on a secure communications channel and maintain a clear agreement on what can and cannot be made with the collected information. Alternatively, clients that wish not to impose much trust on the server's claims not to track them could make bulk queries (or use proxies to do so for them) and privately filter out the relevant information, heavily restricting on the ability to retrieve user activity patterns.

1.5.3.3 Incentivisation

Incentivization relates to creating and divulging, to both layers, incentives to a fully cooperative layer relationship that is inherent in the oracle pattern adopted by the ALTO system. It is quite the challenge to fundamentally change how applications behave on the internet, as indeed is to ask of ISPs to launch a view of their infrastructure to the outside world. [25] notes incentivisation as one of the key challenges in overlay-underlay cooperation in regards to P2P applications, stating that incentive mechanisms need to exist to ensure that both layers agree to participate in, and maintain, a cooperative relationship. According to the ALTO problem statement [70], the incentives for both parties to act on the system are the advantages that derive from using it - clients are to expect better application performance by leveraging ALTO guidance, and similarly ISPs should expect that their internal goals, such as an optimization of infrastructure utilization, can be met with the increased traffic engineering ability that results from their oracle role. If the overlay consuming ALTO guidance has a manageable number of accountable entities - such as a single CDN or data center that the ISP agrees to partner with - it is realistic to maintain a cooperative agreement that can be solidified with feedback and service agreements. However, if the overlay utilizing the ALTO system makes it hard to pinpoint accountability, such as a large P2P application with many users, it will naturally be harder to ensure that the power dynamic between layers doesn't shift beyond an equilibrium. In these cases, policies could be created and enforced to give insurance to both parties that a cooperative relationship is maintained.

The ISPs could too, like mentioned above, lack proper cooperation, as they are found in a new power dynamic that would leverage their application traffic engineering capabilities to steer traffic in a way that is advantageous to only them, or at least in a disproportionate manner. Again, much like the lack of cooperation by clients, it is difficult to guarantee an equilibrium in the power dynamic between layers, but by guaranteeing improved QoS levels for applications that utilize ALTO cooperation, ISPs become responsible for guaranteeing that these improvements are met, fearing client abandonment otherwise. Giving freedom to both layers on how they act ensures that the system evolves to a common ground that benefits both sides, at least enough to justify them remaining there.

Finally, if the application-ISP tussle becomes harsher and unmaintainable - a point that was argued for in the network infrastructure effects portion in Section 1 - a cooperative system such as ALTO may become necessary - and thus beyond preferable - meaning that ISPs may be forced to block or throttle traffic that it cannot route properly, as it historically happened. Thus, acting with ALTO could go beyond a voluntary action into a symbiotic necessity, meaning that both parties have to act cooperatively to maintain network sustainability. Regardless, the best approach seems to be that the system must in of itself be self-justifiable, meaning that the advantages that it brings should be enough to convince both parties to act on it. ISPs are nevertheless free to deploy their own incentive mechanisms to facilitate early application adoption, that could include monetary rewards or routing privileges, but doing so could damage network neutrality.

1.5.3.4 Network Neutrality

As stated by [73]: "According to most network neutrality proponents, network neutrality rules are intended to preserve the Internet's ability to serve as an open, general-purpose infrastructure that provides value to society over time in various economic and non-economic ways. In particular, network neutrality rules aim to foster innovation in applications, protect users' ability to choose how they want to use the network, without interference from network providers, and preserve the Internet's ability to improve democratic discourse, facilitate political organization and action and to provide a decentralized environment for social, cultural and political interaction in which anyone can participate.". Network neutrality has been a popular point of discussion as society grows around the Internet, sparking debates around the world on what the best course of action should be - for example, regulations were introduced by the *Federal Communications Commission (FCC)* [74] in the United States to police network neutrality [74], and the European Union has a framework for net neutrality laid down in Article 3 [75]. However, potential violators of the spirit of a network neutrality exist, such as British Plusnet's [76] usage of *Deep Packet Inspection (DPI)* to implement limits and differential charges for different traffic [77], or Portuguese MEO's [78] smartphone contracts which include zero rating programs for a given set of services [79] that bundle applications such as Facebook [80] or Spotify [81].

Network neutrality advocates are concerned with guaranteeing that ISPs keep Internet communications free and do not discriminate based on the traffic's specifics, such as platforms, applications, or source and destination addresses. On the other hand, opponents of net neutrality, among them the ISPs themselves, broadband and telecom companies, and hardware manufactures, argue against net neutrality - they claim that it would would reduce incentive to invest, as investments would be harder to insure without the ability to charge higher rates for better infrastructure capabilities. Zero rating programs, such as Wikipedia Zero [82], which provide Wikipedia [83] pages with no charge to a select group of low income regions, are popular in developing countries [84] , provide to select regions Internet content they could not otherwise get, but in the form of a non-neutral view to the network. Additionally, with net neutrality, the ISP's ability to route traffic could itself be at jeopardy - as [85] states in its solution to compromise net neutrality with QoS demands via service differentiation, the Internet is growing at an astonishing rate, as are the demands of applications, and operating the infrastructure on a purely best-effort basis will not be sufficient without a constant provisioning of such infrastructure to keep up with demand, and this too may not be viable nor even possible. Thus, discriminating by traffic services may be needed to guarantee that, say, real-time medical information gets priority over real-time media streaming, which in turn gets priority over e-mail or file sharing.

Considering that the ALTO system behaves in an oracle pattern of cooperation where a single entity - the ISP - is able to heavily influence the traffic patterns of the applications it aids, on the promise of a cooperative network underlay-overlay relationship, such system could violate the principles of net neutrality. In particular, this could happen if the oracle either blocks, or at the very least provides different guidance to different clients, depending on where the query originated from - e.g., which application, which source address, or other defining characteristics. A possible consequence of such a system guiding the Internet could be that given applications can consistently have better QoS measurements not on the basis of the application's implementation, but on the ISP personal biases. Oracle systems such as ALTO do not seem to be analogous to other traffic engineering strategies, such as the usage of MPLS, DiffServ, nor to other means of ISP intervention on overlays, such as the deployment of data caches and redirector proxies - this is because the oracle system, in contrast to the previously mentioned strategies, is one of mutual voluntary and cooperative nature between ISPs and applications. However, it could be argued that if the ALTO system offered guidance to applications in such a way that consistently resulted in better application performance, such applications would be pressured to use such guidance as a means to remain competitively viable, and the ISPs would then have a platform to influence a considerable amount of traffic to their will, being in a position to, depending on how they treat guidance requests, break network neutrality. This neutrality concern can be alleviated if application guidance operated on classes of traffic, e.g. real-time communication or file sharing, thus operating on traffic aggregates to insure QoS levels needed to given application types, but never discriminating beyond such given classes.

As the protocol is defined [68], the provided network status information is truthful and guidance is optional, and neutrality can then still remain outside of the system, since no routing measurements exist within it. If particular implementations of the ALTO system give guidance in such a way to guide traffic in a discriminatory fashion, and if such guidance has advantages that much outweigh any alternative - thus rendering it beyond optional - a case can be made for how ALTO as a concept can break network neutrality - for all its advantages and disadvantages discussed below - as the ISP can utilize discriminatory behaviour to treat applications on their infrastructure differently.

1.5.3.5 Multi-Domain orchestration

The Internet as we know it today spans the entire globe and is rather complex in nature. According to [25], the classic vision of the Internet consisting of a network of transit and stub ASs no longer seems accurate, as it now is much more complex - for starters, the role of network owner and service provider are separating, and Internet access is provided by numerous competing ISPs. As a demonstration of such complexity, Figure 1.13 displays how the Internet is structured into many tiers of different service providers. It can thus be inferred that the act of layer cooperation can get harder when the influence domain increases and potentially spans many different ISP regions which will inevitably act differently as they can have different technologies, biases, policies, and overall goals. These per-ISP biases can make it difficult to guarantee that traffic optimization spanning multiple administrative domains is actually useful and achieves the cooperative nature in mind - for example, an ISP may not be comfortable categorising end-point costs of a given metric, thus making path calculations that pass through that ISP domain not viable. Regardless, per-domain ISP guidance has nonetheless plenty of potential - for example, the ability to localize traffic, as entities outside of domain can be identified by ISP, and similarly per-domain optimization of resources can still be useful when such domain is large, and can be applied to high-volume operations such as those in a data center. The ALTO server within a given domain can also leverage probing measurements and feedback statistics to derive information in areas whose topological detail is unknown, giving a partial network view that contains topological insight and also information derivation that, whilst not being as good as a complete topological insight, may nonetheless power a cooperative effort within a given domain with good results. Some data may, whoever, be both not shared by an external domain nor derivable - endpoint property information, such as network connection types, or server footprints - e.g., available CPU, RAM and storage - can only be retrieved by authoritative entities in a given domain and probing solutions may not be available, thus considerably limiting the applicability of a single ALTO domain.

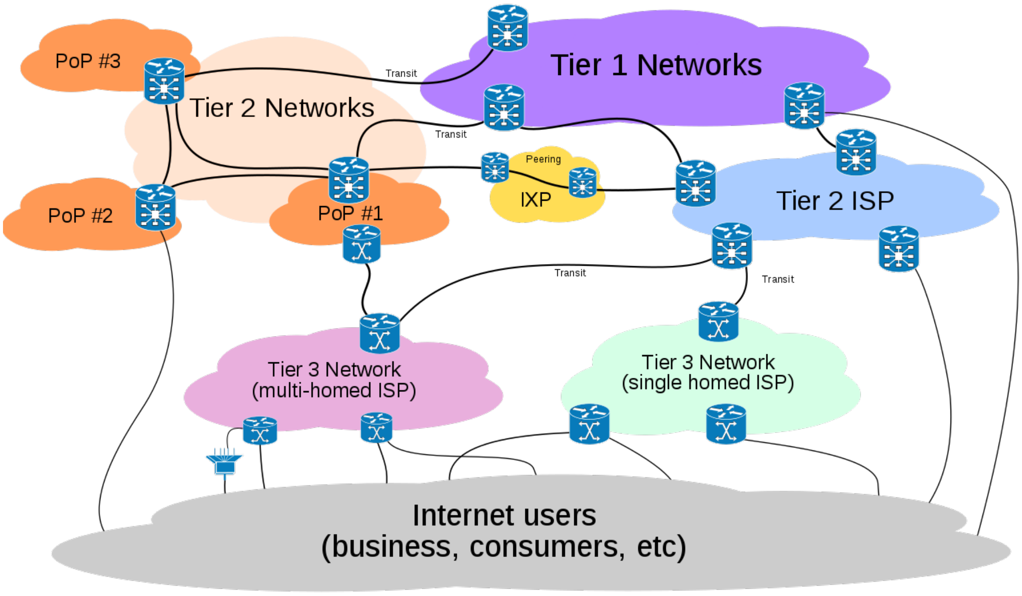


Figure 1.13: Conceptual representation of ISP diversity on the Internet

Even assuming that all ISPs are comfortable with sharing sufficient information, ambiguity may arise - considering a cost map with the generic "routingcost" cost metric, ISPs could internally calculate routing costs differently, and prioritizing different goals, e.g., reducing overall link usage versus reducing inter-AS traffic first and foremost. The base ALTO protocol specification states that each network region can provide its ALTO services, which in turn convey network information from their perspective. A network region, per the protocol specification, consists of a given administrative domain, such as an AS, an ISP, or a given set of agreeing ISPs [68], thus implying that if multiple ISPs share an ALTO server they must reach a consensus on what network status is available for query from the outside. Furthermost, the ALTO working group's deployment considerations [71] document states that an ALTO client can query a single server for one or many metrics, or he can additionally query multiple server instances on different networks [71]. It is explicitly stated in the document that each server could give guidance for only a given network partition, and such guidance may wildly differ between them due to the fact that different algorithms and objectives may have been applied. The document also states that, in regards to extending the reachability of a single server, three different strategies could be applied:

- **Authoritative Servers**: A given set of servers can provide guidance for all kinds of destinations to all ALTO clients.
- **Cascaded Servers**: An ALTO server can possess an embedded ALTO client and query other neighbouring servers if it cannot serve the original request, acting as a middleman between the client and the more appropriate servers.
- **Inter-server Synchronization**: Different ALTO servers communicate among themselves to expand the knowledge space.

The last strategy is still being subject to development and standardization by the working group as part of a bigger attempt to link different network regions and technologies into a single, homogeneous abstraction of the Internet. Current efforts in multi domain orchestration and relevant use case examples are summarized in the ongoing work of [86].

1.6 SUMMARY

In the taken case studies seen in this chapter, it can be clearly seen that there is room for improvement in application-layer traffic generation that can benefit both the applications themselves and the infrastructure administrators that support them. Applications struggle to achieve optimal network resource utilization, whether that be in the task of matching peers in overlay networks, deploying and attributing edge server to media clients, selecting mirror servers, etc., and solutions are being continuously proposed and created that attempt to optimize these decisions. Traffic optimization solutions vary between them with the range of control that is given to the overlay and underlay parties. It seems to be the case that one-sided solutions can hurt the other layer in a worse case scenario, and be lacking maximum efficiency at the best. ALTO's proposal seems to bridge the best of both types of proposals that are either underlay or overlay-centric, standardizing a system and associated protocol whose purpose is to achieve layer cooperation so proper network utilization is possible. Despite many of its associated challenges - namely in the regions of security, privacy, and incentivisation - the project certainly has the potential for a more resourceful Internet that can be more sustainable.

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.