UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

THE ROLE OF TOPOLOGY AND CONTRACTS IN INTERNET CONTENT DELIVERY

Autor:
Syed Anwar Ul Hasan

Director:
Dr. Sergey Gorinsky, IMDEA Networks Institute

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés (Madrid), Abril de 2016

UNIVERSIDAD CARLOS III DE MADRID

Ph.D. Thesis

THE ROLE OF TOPOLOGY AND CONTRACTS IN INTERNET CONTENT DELIVERY

Author:
Syed Anwar Ul Hasan

Director:
Dr. Sergey Gorinsky, IMDEA Networks Institute

DEPARTMENT OF TELEMATICS ENGINEERING

Leganés (Madrid), 2016

The Role of Topology and Contracts in Internet Content Delivery

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Prepared by
Syed Anwar Ul Hasan

Under the advice of
Dr. Sergey Gorinsky, IMDEA Networks Institute

Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid

Date: April, 2016

Web/contact: syed.anwar@imdea.org

This work has been supported by IMDEA Networks Institute.

TESIS DOCTORAL

The Role of Topology and Contracts in Internet Content Delivery

Autor: Syed Anwar Ul Hasan
Director: Dr. Sergey Gorinsky, IMDEA Networks Institute

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Dedication

To my family, for their support and encouragement.

Acknowledgments

I would like to express my sincere and heartfelt thanks to all the people who has made this thesis possible. I take this opportunity to thank them.

First and foremost, I would like to thank my advisor Sergey Gorinsky for his guidance, support, patience and kindness over the years. I feel very fortunate to work with him. His interests in pursuing impactful research, hard work, and attention to detail continue to amaze me. He has provided his help even on basic things so that I am able to understand the research process and show progress. I owe a lot to him for his belief in my abilities and always motivating me to overcome my shortcomings. I can sincerely say that everything I have learned about research has been from him. I also take this opportunity to thank IMDEA Networks for providing financial support during my Ph.D. studies.

I am extremely thankful to my mentors and wonderful collaborators during my internship at Georgia Tech - Prof. Constantine Dovrolis (Georgia Tech) and Prof. Ramesh Sitaraman (University of Massachusetts, Amherst). I thank Constantine for giving me an opportunity to work with him and was amazed by his quality of putting all the effort to the problem formulation and modeling with so much clarity. Because of his guidance, support, patience and motivation, my stay in the US was pleasant and inspiring. I owe a lot of gratitude to Ramesh for his inspiration and kindness, and I will always remember his teaching, advocating to be fearless in research for solving challenging problems. Moreover, I am in awe of Ramesh's clarity and knowledge on interesting research problems.

During the course of my Ph.D. studies at IMDEA Networks Institute, I met some of the nicest, smartest, and most hardworking colleagues. I owe a lot of thanks to my fellow Ph.D. students - Pradeep Bangera, Kshitiz Verma, Foivos Michelinakis, Ignacio Castro, Christian Vitale, Qing Wang, Shahzad Ali, Juan Camillo Cardona, Jordi Arjona, Arash Asadi, Allyson Sim, Ubaid Ur Rahman, Tu Pham, Shailendra Natraj and Lin Wang. I am also thankful to Jose Felix Kukielka, Vincenzo Mancuso, Rade Stanojevic, Joerg Widmer, Arturo Azcorra, Antonio Fernandez, late Prof. Nicolas Georganas, Pierre Francois, Rebecca de Miguel, Brian Dunne, Ana Gonzalez, Flora Quintans, and Marisol Velasco from IMDEA, and my sincere thanks to professors Alberto Garcia, Carmen Guerrero, Ruben Cuevas, Jaime Garcia, Carlos Garcia Rubio from Carlos III University of Madrid. I am also thankful to Aemen Lodhi, Demetris Antoniades, Bilal Anwer, Yogesh Mundada, Abhinav Narain, Saamer Akhshabi and Ahmad Haider for their endearing support during my stay in Georgia Tech. I would like to thank Amogh Dhamdhere (CAIDA) and Manish Jain (Akamai) for their valuable inputs on my research.

I met some amazing people outside of my workspace at IMDEA - Andres Segura Aragones(UC3M), Bilal Sardar, Jose Maria Perez, Isabel Cervantes, Jose Ramon Segura Aragones, Jose Carlos Alva, Fernando (tennis coach); IMDEA building staff (Antonio, Cristina, Salvador, Jose, Pedro, Nani) and thankful to them for their presence in my life.

Finally, I am immensely grateful to my family for their endearing support, patience, and believe, in particular my late father, mother, grandfather, and sisters. If not for their their constant encouragement and support over the years, I might never have had the chance to write this. I dedicate this thesis to them.

Abstract

The Internet depends on economic relationships between ASes (Autonomous Systems), which come in different shapes and sizes - transit, content, and access networks. CDNs (Content delivery networks) are also a pivotal part of the Internet ecosystem and construct their overlays for faster content delivery. With the evolving Internet topology and traffic growth, there is a need to study the cache deployments of CDNs to optimize cost while meeting performance requirements. The bilateral contracts enforce the routing of traffic between neighbouring ASes and are applied recursively: traffic that an AS sends to its neighbour is then controlled by the contracts of that neighbour. The lack of routing flexibility, little control over the quality of the end-to-end path are some of the limitations with the existing bilateral model, and they need to be overcome for achieving end-to-end performance guarantees. Furthermore, due to general reluctance of ASes to disclose their interconnection agreements, inference of inter-AS economic relationships depend on routing and forwarding data from measurements. Since the inferences are imperfect, this necessitates building robust algorithmic strategies to characterize ASes with a significantly higher accuracy.

In this thesis, we first study the problem of optimizing multi-AS deployments of CDN caches in the Internet core. Our work is of significant practical relevance since it formalizes the planning process that all CDN operators must follow to reduce the operational cost of their overlay networks, while meeting the performance requirements of their end users. Next, we focus on developing a temporal cone (TC) algorithm that detects PFS (Provider-free ASes). By delivering a significant portion of Internet traffic, PFS is highly relevant to the overall resilience of the Internet. We detect PFS from public datasets of inter-AS economic relationships, utilizing topological statistics (customer cones of ASes) and temporal diversity. Finally, we focus on a multilateral contractual arrangement and develop algorithms for optimizing the cost of transit and access ASes. In particular, we implement Bertsekas auction algorithm for the optimal cost assignment of access ASes to transit ASes. Furthermore, we implement an epsilon-greedy bandit algorithm for optimizing the price of transit ASes and show its learning potential.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet ecosystem consists of thousands of autonomous systems (ASes), interconnected with each other for providing end-to-end reachability to the end users. Border Gateway Protocol (BGP) [1] is the routing protocol through which traffic is routed between ASes. BGP allows each AS to choose its own administrative policy in selecting routes and propagating reachability information to others [2]. Routing policies depend on contractual agreements or economic relationships between ASes. The most common economic relationships are transit and settlement-free peering. Under transit, an AS pays to its upstream transit AS for routing its traffic in both directions. On the other hand, in a settlement-free peering relationship, there is no payment between participating ASes. Due to the general reluctance of ASes to disclose their business agreements, researchers infer inter-AS economic relationships [2] from BGP route advertisements or actual IP (Internet Protocol) [3] forwarding routes. The inferred AS graph classifies AS relationships into customer-provider (transit), peering, and sibling relationships. Thus, economic relationships between ASes matter for Internet routing. For example, it is financially more attractive for an AS to route traffic through a peering link than a transit connection of the AS to its provider.

Transit, access and content-provider ASes are different types of ASes. Access ASes focus on providing last-mile Internet access to end users. Transit ASes in the Internet core provide traffic-delivery services to access ASes. Content-provider ASes provide reachability to the Internet for content providers such as Facebook and Youtube. An ISP (Internet service provider) is the organization that owns ASes. Many organizations often use multiple ASes, either to implement different routing policies, or as legacies from mergers and acquisitions [4]. Despite a trend towards flattening [5] and significant presence of remote peering at IXPs (Internet eXchange Points) [6], the Internet routing ecosystem is essentially hierarchical [2, 5, 7, 8]. A vast majority of ASes are relatively small and route traffic either as customers of transit links or by peering with local ASes of a similar stature. There exists only a small set of *provider-free ASes* (PFS) that reach the entire Internet without paying anyone for the traffic delivery.

PFS clearly plays a key role as the transit core of the Internet ecosystem and it has many applications and implications. First, economic disputes between provider-free ASes can endanger the universal connectivity of Internet users. Second, while humans prefer to think in discrete categories, designation of an autonomous system as provider-free can have tangible marketplace implications. Third, some algorithms for inter-AS relationship inference use PFS as an input [8, 9] and hence need to know PFS accurately.

There is a recent trend on Internet traffic-delivery economics. Bangera et al. demonstrate that transit providers can derive substantial financial benefits from attracting customer traffic via BGP-based prefix deaggregation [10] and IP prefix hijacking [11], respectively. They suggest based on the observed trade-offs that increasing financial pressure on IP transit business might prompt such actions.

With traffic growth, ASes participating in peering relationships get into a tussle when mutually agreed traffic ratios are violated. For example, AS p agrees to peer with AS q if traffic $T_{p,q}$ that AS p sends to AS q is approximately equal to traffic $T_{q,p}$ that AS q sends to AS p. The peering relationship is sustained when $T_{q,p}/T_{p,q} < t_p$, where t_p is AS p's traffic ratio threshold [12]. A violation of this condition can lead to de-peering where peering ASes disconnect. Due to the de-peering, the users of the de-peered ASes cannot reach major portions of the Internet. Such instances have necessitated regulatory intervention and emergence of paid peering, partial transit [13], and other private economic arrangements as alternatives. Under a partial-transit relationship, the access is for only a fraction of the global Internet address space. With paid peering, there is a payment involved, with one of the peering ASes paying the other peer AS for exchanging their customer traffic.

With bilateral contracts, an AS only directly controls the next hop in the AS-level path and delegates handling of the rest of the path to other ASes. Therefore, the end-to-end performance that a user perceives is the result of contracts that are applied recursively. Though, such arrangements facilitate management of network operations, routing decisions are globally suboptimal due to limited prefix visibility [14]. Furthermore, ASes do not optimize end-to-end performance in a coordinated manner. In particular, performance such as QoS (Quality of Service) or QoE (Quality of Experience) cannot be guaranteed at all times. The end users of the access AS suffer from performance degradation if a transit AS on the end-to-end path has performance bottlenecks. This calls for exploring a multilateral contractual model, where access ASes form a contract with multiple transit providers for end-to-end performance guarantees, potentially with contracts for short time intervals. Such contractual arrangement compliments existing bilateral ones.

The first attempts to look at the multilateral contract model from different perspectives has been explored [15–17] in recent years. A trusted centralized entity (web service, broker, cryptographic public ledger) provides the infrastructure and economic medium for such an arrangement between access and transit ASes. The payments from access ASes can be paid by the centralized entity on their behalf to transit ASes. Moreover, technologies like MPLS (Multiprotocol label switching) at transit ASes along with special routers for transport layer upgrades like Serval [18], PacketShader [19] router show the viability of routing and forwarding of traffic under such multilateral arrangement.

The bandit algorithms have been applied for optimizing ad placement on websites and content experiments [20] run by Google for analytics purposes. The reward metric in those applications is the users' click-through rate (CTR). Under a multilateral arrangement, there is a problem of pricing for transit ASes under uncertainty. The bandit-based learning is applicable to modeling such a problem. We sketch the use of exploration and exploitation trade-offs from learning literature. For example, in order to maximize the short-term revenue, the transit AS should exploit its current prices and choose prices that access ASes are likely to accept. On the other hand, to maximize the long-term revenue, the transit AS needs to explore, i.e., identify which prices have the largest selection probability of access ASes. This kind of exploration necessitates choosing prices whose current probability estimates are low, which thus leads to choosing prices with low probabilities in the short term.

Content delivery networks (CDNs), which were originally designed to deliver web content, video content, and file downloads, currently serve a much broader family of applications, including social networks, e-commerce sites, CRM (customer relationship management) portals, and web-based SaaS (software as a service). CDNs play an important role in the economic structure of the Internet ecosystem where they optimize content delivery by enhancing routing performance. Also, they serve content faster than the default Internet routing based on BGP. CDNs like Akamai deploy caches close to the end users where they cache content of content providers like Facebook and news sites, thereby offering faster delivery to end users. CDNs have their own set of client mapping, load balancing, and caching algorithms. There is an evolving trend of optimizing cache deployments by CDNs as the Internet traffic is growing at an exponential rate, and end users are becoming highly sensitive to performance. Some of the reasons for this trend are the increasing use of smartphones and consumption of video-streaming content.

1.1 Motivation

CDNs succeed by deploying widely in a large number of ASes and creating an overlay network that is capable of delivering better end-to-end performance than the Internet underlay can. While the very existence of CDNs hinges on their ability to improve upon the direct Internet delivery, a better delivery requires a larger and more costly footprint of the distributed caches. The question arises in which ASes the CDNs should deploy their caches. CDNs need to deploy widely in all types of ASes. In particular, a CDN needs to deploy in both core and access ASes to meet performance needs of all downstream end users. Access ASes economically benefit from CDNs because delivery of the content from a local CDN cache reduces the transit traffic and thus transit expenses of the access AS. Consequently, access ASes eagerly host CDN caches and even incentivize their deployment, e.g., by not charging the CDN for the cache colocation space and on-net traffic (i.e., the CDN traffic that stays within the access AS). On the other hand, CDN deployments in a core AS might decrease the transit revenues of the AS and make it reluctant to offer any kind of deployment incentives. For these reasons, planning the cache deployment in access ASes is a simpler matter and impacts only a smaller portion of the total deployment cost.

Cache placement has been studied in models that do not account for the economic structure of the Internet [21–23]. Because the economic considerations are crucial for CDN planning, we are motivated to look beyond the single-network perspective and examine cache deployment in realistic AS-level Internet topologies. There is no prior study of multi-AS deployment optimization with realistic cost and performance constraints, such as the ones examined in our work.

The cache deployment optimization (CaDeOp) relies on knowledge of Internet core topology. Hence, this thesis also studies topology inference for detection of PFS. Over the past decade, numerous research efforts have tried but failed to infer inter-AS economic relationships precisely. We explore this failure in the specific context of provider-free ASes. Our interest in PFS arises due to a number of reasons. First, the provider-free ASes clearly play a key role as the transit core of the Internet ecosystem. By delivering a significant portion of Internet traffic, PFS is highly relevant to the overall resilience of the Internet to accidental failures and intentional disruptions.

After our studies on CDN cache optimization and PFS detection, we look at the multilateral contractual model for the Internet since it offers a greater flexibility in routing, cost and performance improvements than existing bilateral contracts. The economic aspects of a multilateral contractual model for transit and access ASes have not been studied in sufficient detail. Thus, we explore them in this thesis. Moreover, the algorithmic and learning models are fundamental to study access-transit ASes interaction, economics, and optimization of multilateral contracts. We look at a couple of representative examples. First, we explore market-based solutions for obtaining an end-to-end Internet path in a fair and competitive manner. Second, we also know how online machine learning has been applied to the shortest-path problem, termed as the on-line shortest path problem [24], where a weighted directed acyclic graph is given such that edge weights can change in an arbitrary or adversarial manner, and the decision maker has to pick in each round a path between two vertices, so that the chosen path has a minimal weight. We can think of access ASes utilizing a learning strategy as in the on-line shortest path problem to pick cheaper end-to-end paths offered by transit ASes under dynamic price changes in time-varying conditions.

1.2 Scope

In this thesis, we first formulate and solve the problem of optimizing multi-AS deployments of CDN caches in the Internet core. The work formalizes the planning process that CDN operators must follow to reduce the operational cost of their overlay networks, while meeting the performance requirements of their end users. CDN operators can extend our modeling efforts to assess trade-offs and make informed decisions on their evolving deployment strategies. We do not consider the real-time aspects of the CDN operation but rather focus on planning the CDN deployment in the longer term. We investigate the problem in AS-level Internet topologies derived from real measurements of BGP paths and end-user traffic demands seeded with real data.

Consequently, our work focuses on optimizing CDN deployments in the core of the Internet where the problem is both more complex and more expensive. We study cache deployment optimization (CaDeOp) for any real-life CDN where deployment decisions are taken on the time scale of months or quarters. The objective is to minimize the deployment cost incurred by the CDN, subject to meeting the end-user performance requirements. The primary output of the optimization is the set of cache ASes and the amount of server, bandwidth, and energy resources that the CDN has to deploy in each cache AS. We strive for practical relevance of our assumptions by leveraging realistic data. Another aspect worth noting is that a CDN typically needs to optimize the cache deployment incrementally because of already having a deployment and needing to modify it to meet new traffic, performance, and cost requirements. Hence, we consider an incremental version of the problem as well.

Our second work focuses on Internet AS-level topologies and goes deeper in the domain of topology inference. We have a specific goal of detecting PFS. We develop an algorithm that detects PFS from public datasets of inter-AS economic relationships. Finally, we focus our attention towards a multilateral contractual model that compliments the existing bilateral contracts model. In particular, we develop an algorithmic approach for optimizing the cost objectives of transit and access ASes participating in a multilateral contract for end-to-end reachability. We develop and implement an epsilon-greedy bandit learning algorithm for optimizing the price of transit ASes, and also propose an auction algorithm for cost-optimal assignment of access ASes to transit ASes.

1.3 Contributions

In Chapter 2, we study the trade-offs in optimizing the cache deployment of CDN caches in the Internet core. We model the server costs as linear and realistically represent bandwidth and energy costs as non-linear functions. In particular, we consider bandwidth-cost functions that are sensitive to the geographic location. We approximate the non-linear energy and bandwidth costs and transform them to piecewise-linear so that our cache deployment optimization problem (CaDeOp) is a MILP (Mixed Integer Linear Programming) problem, for which solutions can be obtained. We evaluate our CaDeOp solutions in realistic settings by using AS-level Internet topologies, traffic demand distributions seeded with Akamai traffic data, C-BGP routing solver to compute AS paths from cache ASes to other ASes. We also explore the incremental cache deployment optimization (InCaDeOp) which is related to upgrading cache deployments. We present an InCaDeOp formulation to be solved and evaluated in future work.

Our main conclusions are as follows. When the end-user performance requirements become more stringent, the CDN footprint expands rapidly, requiring cache deployments in additional ASes and geographical regions. With higher performance requirements, the CDN cost also rises by several times. The costs of energy and bandwidth grow because the CDN loses some of the economies of scale in procuring these resources. The traffic distribution among the cache ASes stays relatively even, with the top 20% of the cache ASes serving around 30% of the overall CDN traffic. It is notable that the Pareto principle does not apply to CDN deployments, in part due to the highly distributed nature of the Internet traffic [25]. The work in this chapter is based on the following publication.

- Syed Hasan, Sergey Gorinsky, Constantine Dovrolis, Ramesh K. Sitaraman. “Trade-offs in Optimizing the Cache Deployments of CDNs”. The 33rd IEEE International Conference on Computer Communications (IEEE INFOCOM 2014), 27 April 2014

- 2 May 2014, Toronto, Canada [26].

In Chapter 3, we focus on PFS (provider-free ASes). We show that straightforward extraction of PFS from the public datasets yields poor results. We then develop a TC (temporal cone) algorithm that detects PFS from public datasets of inter-AS economic relationships. Our algorithm utilizes topological statistics (customer cones of ASes) and temporal dataset diversity to infer PFS with significantly higher accuracy.

The TC algorithm is useful because it enables accurate inferences of PFS in the future even if PFS insights from the non-verifiable sources become unavailable. While the knowledge of PFS is highly valuable, validation of PFS inference results constitutes a major challenge because the ground truth lies outside the public domain. To tackle the validation challenge, we utilize trustworthy but non-verifiable sources such as Wikipedia. Whereas it seems practically impossible to obtain the complete ground truth from network operators, the non-verifiable source insights form the best available baseline for result validation in this important domain. The work in this chapter corresponds to the following publication.

- Syed Hasan, Sergey Gorinsky. “Obscure Giants: Detecting the Provider-free ASes”. 11th International IFIP TC 6 Networking Conference, May 21-25, 2012, Prague, Czech Republic [27].

In Chapter 4, we focus on the transit and access ASes optimization objectives under a multilateral interdomain contract arrangement. First, we consider the access-AS cost optimization problem by setting up a market, with access ASes submitting bids for individual path segments to transit ASes for an end-to-end path. We formulate the problem as an optimal linear assignment problem for a path segment, which extends to a sequence of linear assignment problems for the end-to-end path. We propose and implement Bertsekas auction algorithm for the optimal cost assignment of access ASes to transit ASes. We perform a sensitivity analysis by considering graphs of different sizes. We assess the convergence time of access ASes and also the mapping of submitted bids by access ASes until their final assignment. Next, we formulate the pricing of transit ASes as a multi-armed bandit problem and utilize the most commonly used epsilon-greedy bandit algorithm as the solution strategy. We implement the epsilon-greedy algorithm and evaluate it with pricing data to assess the exploration-exploitation trade-offs. The material in this chapter is a basis for a currently prepared submission.

- Syed Hasan, Sergey Gorinsky. “Optimizing the Cost of Multilateral Interdomain Contracts”. Under preparation for submission.

We conclude by outlining the contributions of this thesis and ideas for future work in Chapter 5.

Chapter 2

Trade-offs in Optimizing the Cache Deployments of CDNs

2.1 Introduction

CDNs (Content Delivery Networks) have revolutionized Internet data dissemination by storing content in their geographically distributed caches and thereby improving the experience of end users [25, 28]. The proximity of the CDN caches to the end users provides low-latency low-loss paths between the caches and the users, improving the performance. Originally designed to deliver web content, video content, and file downloads, CDNs currently serve a much broader family of applications, including social networks, e-commerce sites, CRM (customer relationship management) portals, and web-based SaaS (software as a service).

CDNs play an important role in the economic structure of the Internet ecosystem. Content providers, such as the New York Times, Netflix, or Facebook, pay CDNs for delivering their content to end users with greater reliability, performance, and scalability than what is possible directly over the Internet. The Internet is a best-effort network and does not provide performance guarantees or a globally differentiated delivery service. Part of the reason is that the Internet infrastructure consists of tens of thousands of independent ASes (Autonomous Systems) that are owned by separate business entities who do not cooperatively optimize end-to-end performance for the users [29]. CDNs succeed by deploying widely in a large number of ASes and creating an overlay network that is capable of delivering better end-to-end performance than the Internet underlay can.

The CDN delivery quality is subject to trade-offs with the CDN cost. While the very existence of CDNs hinges on their ability to improve upon the direct Internet delivery, a better delivery requires a larger and more costly footprint of the distributed caches.

Furthermore, the optimal trade-offs between footprint/cost and performance change over time. Real-world commercial CDNs have caches in thousands of ASes and keep expanding their AS presence to provide greater performance by being closer to the end users [30].

There are many types of ASes. Access ASes focus on providing last-mile Internet access to end users. Transit ASes in the Internet core provide traffic-delivery services to access ASes. CDNs need to deploy widely in all types of ASes. In particular, a CDN needs to deploy in both core and access ASes to meet performance needs of all downstream end users. Access ASes economically benefit from CDNs because delivery of the content from a local CDN cache reduces the transit traffic and thus transit expenses of the access AS. Consequently, access ASes eagerly host CDN caches and even incentivize their deployment, e.g., by not charging the CDN for the cache colocation space and on-net traffic (i.e., the CDN traffic that stays within the access AS). On the other hand, CDN deployments in a core AS might decrease the transit revenues of the AS and make it reluctant to offer any kind of deployment incentives. For these reasons, planning the cache deployment in access ASes is a simpler matter and impacts only a smaller portion of the total deployment cost. Consequently, our work focuses on optimizing CDN deployments in the core of Internet where the problem is both more complex and more expensive.

2.1.1 CaDeOp: cache deployment optimization

Our work is the first formal study of the cache deployment optimization (CaDeOp) problem that is an important operational component of any real-life CDN. CaDeOp is an offline planning problem where the CDN operator makes deployment decisions on the time scale of months or quarters. The CaDeOp objective is to minimize the deployment cost incurred by the CDN, subject to meeting the end-user performance requirements. We call an AS where the CDN deploys its caches a *cache AS* to differentiate such an AS from the ASes where the CDN has no cache deployments. The primary output of CaDeOp is the set of cache ASes and the amount of server, bandwidth, and energy resources that the CDN has to deploy in each cache AS.

To optimize the deployment of a CDN, one must consider the traffic of the end users and examine which ASes can satisfy these end-user traffic needs with acceptable performance. Therefore, CaDeOp also produces a tentative assignment of end users to cache ASes. While these assignments are guidelines for which cache ASes can serve which users, the CDN may choose to alter these assignments in real time when the Internet performance [25] or cost characteristics [31, 32] change. In this work, we do not consider the real-time aspects of the CDN operation but rather focus on planning the CDN deployment in the longer term. Another aspect worth noting is that a CDN typically needs to optimize the cache deployment incrementally because of already having a deployment and needing to modify it to meet new traffic, performance, and cost requirements. While CaDeOp takes the clean-slate approach of computing a full optimal deployment from scratch, our CaDeOp problem formulation can be easily extended, without affecting the solution methodology, to optimize the incremental update of an existing deployment. Our subsequent work will also examine a strategy of potentially sacrificing the current optimality to enable updates that optimize the deployment under expected future conditions.

In studying the CaDeOp problem, we strive for practical relevance of our assumptions. Unlike previous cache-deployment studies that assume a highly hypothetical network topology such as a line or ring [21], we investigate CaDeOp in AS-level Internet topologies derived from real measurements. Using C-BGP [33], we compute realistic BGP (Border Gateway Protocol) [1] paths from cache ASes to the other ASes in these topologies. We model the traffic demands of the end users by adopting a realistic Zipf distribution seeded with data from Akamai Technologies. To evaluate the sensitivity of the results to parameter settings, we consider 3 topologies from 2 different sources as well as 2 opposing vectors of traffic demands. We model the cost of the CDN as a combination of bandwidth, energy, and server costs. While the server costs are linear, we realistically represent bandwidth and energy costs as non-linear functions. In particular, we consider bandwidth-cost functions that are sensitive to the geographic location and configured according to data from TeleGeography [34].

2.1.2 Our contributions

This work is the first to formulate and solve the CaDeOp problem of optimizing multi-AS deployments of CDN caches in the Internet core. Our work is of significant practical relevance since it formalizes the planning process that all real-life CDN operators must follow to reduce the operational cost of their overlay networks, while meeting the performance requirements of their end users.

We evaluate our CaDeOp solutions in realistic settings, examine the sensitivity of the results to our parametric assumptions, and reach the following main conclusions:

1. When the end-user performance requirements become more stringent, the CDN footprint expands rapidly, requiring cache deployments in additional ASes and geographical regions.

2. With higher performance requirements, the CDN cost also rises by several times. While the server costs remain about the same, the costs of energy and bandwidth

grow because the CDN loses some of the economies of scale in procuring these resources. Consequently, the cost balance in CDNs with higher performance shifts toward bandwidth and energy costs. As the end-user performance requirements become more and more stringent over time, our work suggests that adoption of schemes for energy-usage reduction [31] and bandwidth optimization [35] will become even more important for CDNs of the future.

3. The traffic distribution among the cache ASes stays relatively even, with the top 20% of the cache ASes serving around 30% of the overall CDN traffic. It is notable that the Pareto principle, which applies in many related domains, does not apply to CDN deployments, in part due to the highly distributed nature of the Internet traffic [25].

The rest of this chapter has the following structure. Section 2.2 formulates the CaDeOp problem. Section 2.3 presents our solution methodology. Section 2.4 reports the results. Section 2.5 presents the future extensions by looking into incremental CaDeOp. Section 2.6 discusses related work. Section 2.7 sums up our chapter.

2.2 Formulating the CaDeOp problem

We model the Internet core as a connected directed graph G where the nodes denote ASes and form set N. The graph edges represent inter-AS economic relationships annotated as transit or peering.

In the given model, *cache ASes* refer to the ASes where the CDN deploys caches. One of the CaDeOp goals is to identify set H of the cache ASes. Bit I_i identifies whether AS i is a cache AS: $I_i = 1$ for AS $i \in H$, and $I_i = 0$ for AS $i \notin H$.

Any AS can have end users of the CDN. Measured in Mbps, traffic demand T_j denotes the overall rate of the content traffic from the CDN caches to the end users in AS j. To satisfy the traffic demands, the cache ASes transmit the content traffic along the paths computed according to the BGP protocol. Distance $D_{i,j}$ represents the number of hops on the AS-level path from cache AS i to AS j.

Traffic split $m_{i,j}$ is another CaDeOp output and denotes the fraction of traffic demand T_j satisfied by cache AS i. The traffic splits are subject to the following constraints:

$$0 \leq m_{i,j} \leq I_i \qquad \forall i \in N, \forall j \in N, \tag{2.1}$$

$$\sum_{i \in H} m_{i,j} = 1 \qquad \forall j \in N. \tag{2.2}$$

Inequalities 2.1 imply that only the cache ASes serve the content, i.e., $m_{i,j} = 0$ if AS $i \notin H$. Equalities 2.2 ensure that the cache ASes fully satisfy the traffic demand of each AS j.

With the traffic splits determined, we establish the overall rate V_i of the content traffic transmitted by each cache AS i:

$$V_i = \sum_{j \in N} (m_{i,j} \cdot T_j) \,. \tag{2.3}$$

Some of this traffic might be on-net, i.e., sent to local end users. The rate of the on-net traffic for cache AS i can be determined as $F_i = m_{i,i} \cdot T_i$. Then, the rate of the off-net traffic (i.e., the overall traffic from AS i to the other ASes) is $J_i = V_i - F_i$.

To characterize the delivery quality provided by the CDN to the end users, the CaDeOp model incorporates the following performance constraint d:

$$\frac{\sum\limits_{j \in N} (D_{i,j} \cdot m_{i,j} \cdot T_j)}{V_i} \leq d \qquad \forall i \in H \tag{2.4}$$

where the hop distance from cache AS i to AS j is weighed with the fraction of the V_i traffic that the cache AS sends along the delivery path. This metric reflects the delay in AS hops acceptable for the end users. For example, $d = 0$ requires delivery from a cache in the same AS. Meeting the $d = 1$ constraint corresponds, on average, to delivery from an adjacent cache AS.

To operate each cache AS i, the CDN incurs server cost S_i, energy cost E_i, and bandwidth cost B_i. These costs are computed as

$$S_i = a \cdot V_i, \; E_i = e \cdot V_i^h, \text{ and } B_i = b_i \cdot V_i^g \tag{2.5}$$

where a, e, and b_i are positive constant factors, and h and g are constant exponents with values between 0 and 1. The non-linear E_i and B_i functions capture the economies of scale in energy and bandwidth consumption [36–38]. While pricing may vary with geography, we model such variations for bandwidth pricing only, by allowing different cache ASes i to differ in their factors b_i (factors e and a are fixed for all ASes).

We use C to denote the CDN cost that combines the individual costs of the CDN in all cache ASes:

$$C = \sum_{i \in H} (S_i + E_i + B_i) \,. \tag{2.6}$$

The objective in CaDeOp is to minimize C. With all the notation summarized in table 2.1,

Notation	Semantics
G	AS-level topology of the Internet core
N	set of ASes in G
T_j	traffic demand of the end users in AS j
H	set of cache ASes
I_i	bit indicating whether AS $i \in H$
$D_{i,j}$	AS-hop distance from cache AS i to AS j
m_{ij}	fraction of T_j satisfied by cache AS i
V_i	overall traffic of cache AS i
F_i	on-net traffic of cache AS i
J_i	off-net traffic of cache AS i
d	performance constraint
S_i	server cost of cache AS i
E_i	energy cost of cache AS i
B_i	bandwidth cost of cache AS i
a, e, b_i	cost-function factors
h, g	cost-function exponents
C	CDN cost

Table 2.1: Notation in the CaDeOp problem formulation.

we formulate the CaDeOp problem as follows:

- *Inputs:* topology G, AS-level paths in G, traffic-demand vector T, cost-function parameters a, e, b_i, h, g, and performance constraint d;
- *Objective:* minimize CDN cost C;
- *Constraints:* equalities 2.2 and inequalities 2.1 and 2.4;
- *Outputs:* set H of cache ASes, traffic-split matrix m, overall-traffic vector V, on-net traffic vector F, off-net traffic vector J, server-cost vector S, energy-cost vector E, bandwidth-cost vector B, and CDN cost C.

2.3 Solution methodology

2.3.1 Approximation of non-linear costs

While the energy and bandwidth costs in the formulated CaDeOp problem are non-linear, we apply the classical convex combination method [39], which uses special ordered sets of type 2 (SOS2), to approximate these costs with piecewise-linear functions. The approximating functions employ between 6 and 8 linear segments.

2.3.2 Solving the CaDeOp problem

With the energy and bandwidth costs transformed to piecewise-linear, CaDeOp becomes a MIP (Mixed Integer Programming) problem. We express the MIP problem in AMPL [40] and solve it in Gurobi Optimizer 5.0, a commercial optimization solver [41], with the maximum optimality gap of 5%. Table 2.2 reports the Gurobi parameter values in our solutions.

Gurobi parameter	Setting	Description
lpmethod	3	concurrent algorithm (dual simplex + barrier) is used to solve for the root node of the MIP model
presparsify	1	setting this option for significant reduction in the problem size
threads	2	number of threads used for solving the MIP model
mipstart	0	setting mipstart to 0, we do not employ initial guesses to solve the MIP problem with integer variables
mipgap	0.05	maximum relative MIP optimality gap is set to 5%

Table 2.2: Gurobi parameter settings for solving our MIP model instances

2.3.3 Parameter settings

Topology: We solve the CaDeOp problem for the Internet core. Our iterative algorithm, which detects and resolves customer-provider cycles, extracts an Internet core topology from an Internet-wide AS-level topology by peeling off the current edge ASes, i.e., ASes that do not provide transit for another AS in the current topology. We apply the extraction algorithm to the Internet-wide topologies from UCLA [42] and CAIDA [43] to derive 3 Internet core topologies, to which we refer as UCLA, CAIDA1, and CAIDA2 through the rest of this chapter. Table 2.3 reports statistical properties of these Internet core topologies.

To understand the structure of the Internet core topologies, we classify the ASes according to their centrality defined with respect to their customer cone. The customer cone of an AS consists of ASes that are either direct or indirect transit customers of that AS in the core topology, i.e., those ASes reachable from the AS through a sequence of provider-to-customer transit links [44]. We consider 4 centrality classes: large cone, medium cone, small cone, and tiny cone. The customer cone of a tiny-cone AS contains at most 5 ASes. Note that the centrality classification is for the Internet core topology only and that even a tiny-cone AS without a customer AS in the core topology can have

Topology	ASes	Transit	Peering	node degree(max)	Diameter	Avg AS-hop dist
UCLA	320	1308	3535	167	16	2.69
CAIDA1	302	1074	3345	167	9	2.54
CAIDA2	490	1726	5335	253	10	2.69

Table 2.3: Statistical properties of the considered Internet core topologies.

many customer ASes in the original Internet-wide AS-level topology. The customer cone of a small-cone AS includes between 6 and 50 ASes. Medium-cone ASes have at least 51 ASes in their customer cones. We create the large-cone category by extracting from the medium-cone categories those ASes that are tier-1 according to Wikipedia [27,45] (3 of such ASes are not in the large-cone or medium-cone categories for the CAIDA1 topology). Table 2.4 shows the split of the ASes according to their centrality in the Internet core topologies.

Centrality type	UCLA	CAIDA1	CAIDA2
large-cone ASes	15	12	15
medium-cone ASes	30	5	11
small-cone ASes	167	74	99
tiny-cone ASes	108	211	365

Table 2.4: Split of the ASes according to their centrality in the UCLA, CAIDA1, and CAIDA2 topologies.

AS-level paths: AS-level paths in the Internet core topology constitute another input for the CaDeOp problem. We use the C-BGP tool [33] to compute realistic AS-level paths in the UCLA, CAIDA1, and CAIDA2 topologies.

Traffic demands: To set the traffic demands of ASes, we utilize data from Akamai caches in Indiana, California, Sweden, and Switzerland. The datasets report average monthly rates of content traffic served by the caches. We present the data in Table 2.5. We scale up the actual traffic rates of these Akamai caches to estimate the overall monthly traffic demand for all ASes in the Internet core topology. It is worth noting that, with our focus on optimizing CDN deployments in the Internet core, we do not consider access ASes and their traffic demands. We distribute the overall core demand between the individual core ASes by assigning traffic-demand rates to the ASes according to the Zipf distribution [46, 47] where the maximum traffic demand of an AS and skew parameter are set to 5 Gbps and 0.8 respectively.

While we do not have access to data for the content consumption by specific ASes, we consider 2 opposing assignments of the traffic-demand shares to individual core ASes: (T1) Traffic-demand vector T1 sets the traffic-demand shares of ASes in the order of the

Akamai cache location	Akamai response traffic (Mbps)				Akamai request traffic (Mbps)			
	average traffic		max traffic		average traffic		max traffic	
	month	year	month	year	month	year	month	year
Indiana GigaPoP, US [48]	315	530	1220	2902	127	188	445	557
SUNET, EU [49]	285	390	1070	1950	80	108	383	1240
CERN, EU [50]	41	60	165	154	20	25	110	56
Santa clara NOC, US [51]	9	15	455	477	4	6	192	192

Table 2.5: CDN cache traffic data: average and maximum for a *month* between 10th June to 10th July of 2013, and a *year* from 10th July, 2012 to 10th July, 2013, obtained at four Akamai cache locations in the US and Europe.

node degrees of the ASes; this traffic allocation roughly corresponds to the centrality classification of the ASes in the Internet core topology and places larger traffic demands toward the topological center; (T2) Traffic-demand vector T2 assigns the traffic-demand shares to the ASes in the reverse order and allocates larger traffic demands at the edges of the Internet core topology. While the Zipf profile of the traffic-share distribution is realistic [46, 47], extremes T1 and T2 of the broad traffic-demand vector range enable us to evaluate the sensitivity of the results to traffic demands. Figure 2.1 plots the T1 and T2 traffic assignments as functions of the node-degree of the ASes.

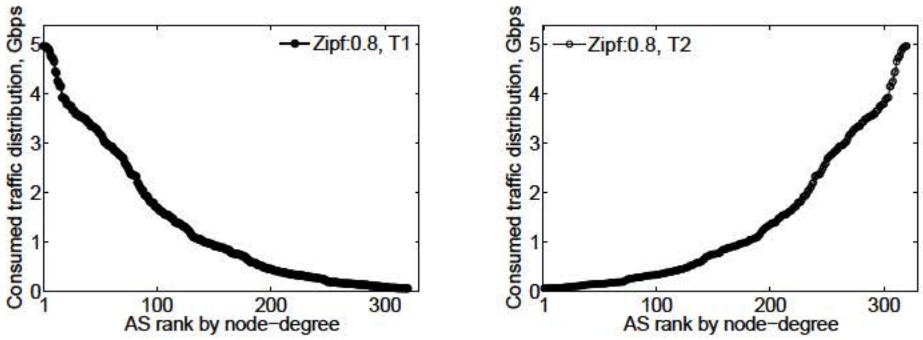


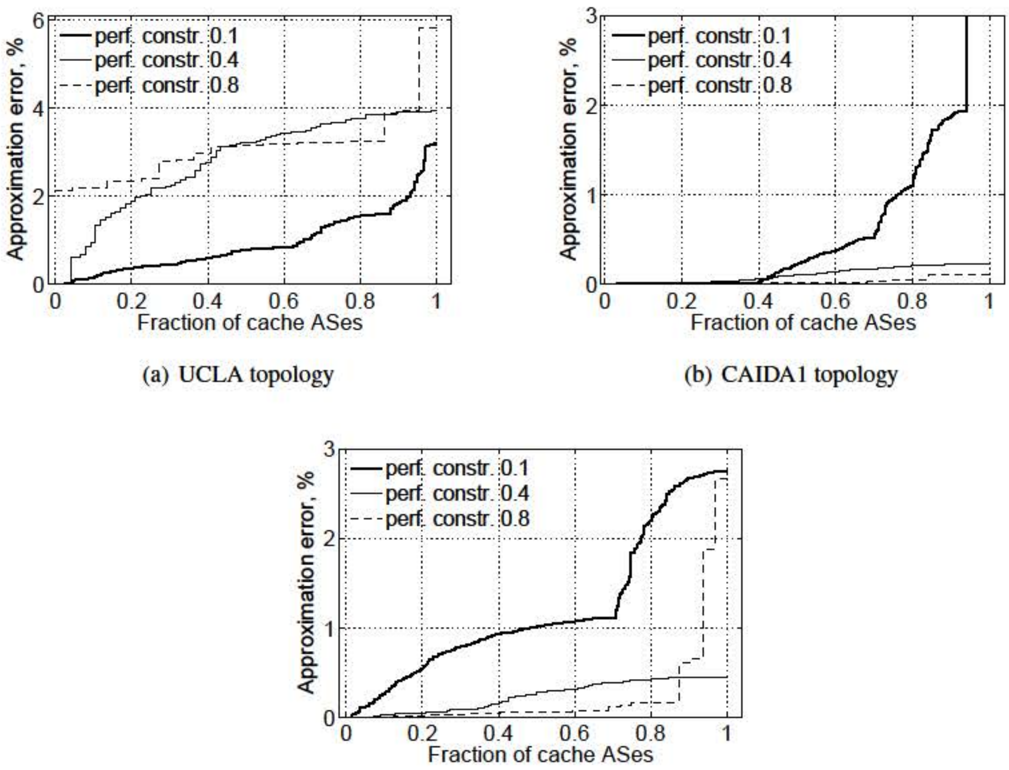
Figure 2.1: T1 and T2 traffic assignments for the ASes in the UCLA topology

Cost functions: Based on data from public sources and TeleGeography [34], our default parameter settings for the server, energy, and bandwidth cost functions are as follows: $a = 0.88$, $e = 20$, $b_i = 70$, $h = g = 0.75$ (all measurement units are such that expressions 2.5 compute monthly costs in U.S. dollars). We also evaluate location-aware bandwidth pricing where the value of b_i depends on the geographical region of AS i. Again guided by the TeleGeography data, we set the location-aware b_i value to 51, 71, 215, 264, and 270 for Europe, North America, Asia-Pacific, South America, and Oceania respectively.

2.4 Evaluation results

2.4.1 Accuracy of approximating the non-linear costs

While our method for solving the CaDeOp problem approximates the non-linear energy and bandwidth costs with piecewise-linear functions, we quantify how accurately the piecewise-linear functions represent the non-linear costs for each cache AS. Keeping the approximation accuracy high is important in order to avoid error cascades and preserve the high precision in the problem solutions [52].



(a) UCLA topology

(b) CAIDA1 topology

(c) CAIDA2 topology

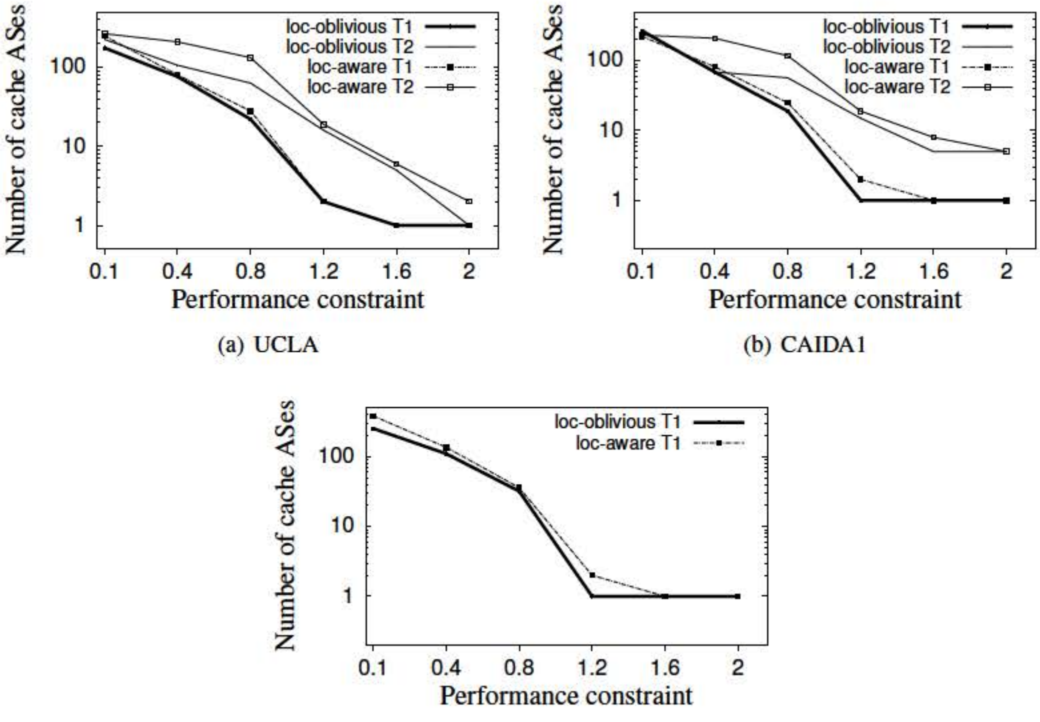
Figure 2.2: Approximation error for the energy and bandwidth costs with traffic-demand vector T1 and location-oblivious bandwidth pricing.

Figure 2.2 plots the approximation error for the energy and bandwidth costs of every cache AS with traffic-demand vector T1 and location-oblivious pricing. For the UCLA topology, Figure 2.2a shows that the error is less than 4% for all but one cache AS when performance constraint d is at most 0.8. The error stays at 0.6% when $d = 1.2$. With $d = 1.6$ or $d = 2$, the approximation does not introduce any error into the energy and bandwidth costs of cache ASes.

For the CAIDA topologies, the approximation method provides the exact costs with $d \geq 1.2$. With $d \leq 0.8$, Figure 2.2c shows that the error consistently stays under 3% for the CAIDA2 topology. For the CAIDA1 topology with $d = 0.8$ or $d = 0.4$, Figure 2.2b shows the error values that are always significantly below 1%. On the other hand, with $d = 0.1$, the error stays under 3% for only 94% of the cache ASes and grows to almost 15% over the remaining 6% of the cache ASes. While the cost-approximation accuracy deteriorates for some cache ASes in CAIDA1, we use UCLA as the baseline topology in our evaluation.

2.4.2 Deployment footprint

We start evaluating our CaDeOp solutions by examining how performance constraint d affects the footprint of the optimal CDN deployment. Figure 2.3a plots the number of cache ASes in the optimal deployment for both traffic-demand vectors in the UCLA topology with location-oblivious and location-aware bandwidth pricing. When the required delivery quality is low, the optimal deployment involves only a few cache ASes.



(a) UCLA

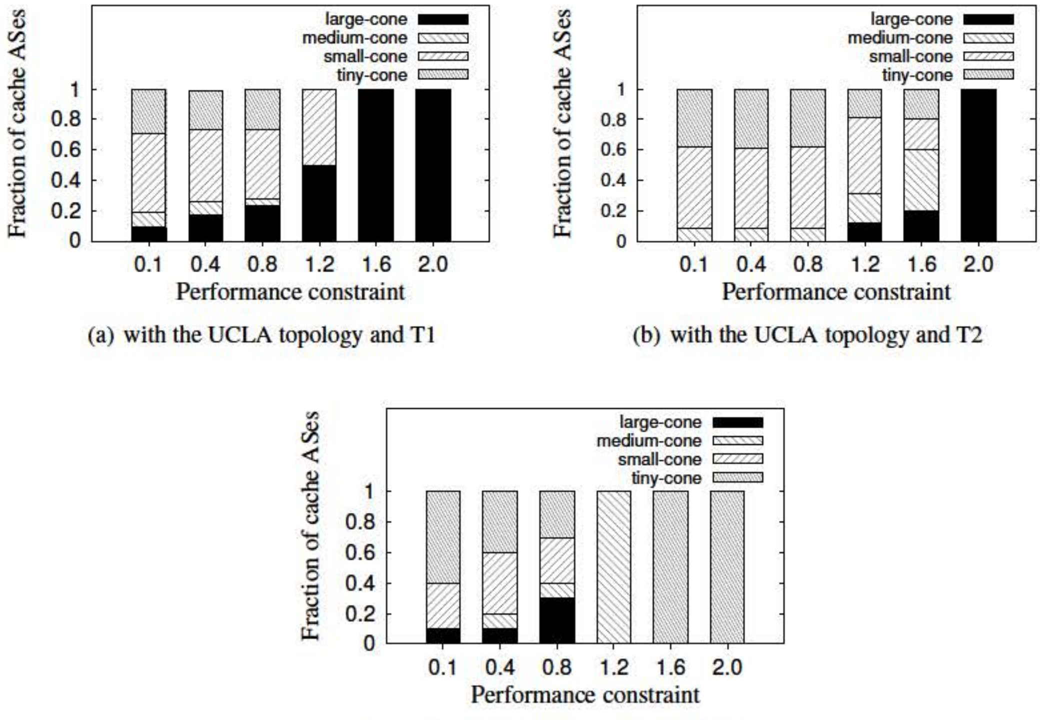
(b) CAIDA1

(c) CAIDA2

Figure 2.3: Trade-offs between the CDN footprint and performance in the UCLA, CAIDA1, and CAIDA2 topologies.

For $d = 2$, a single-AS deployment provides the required delivery quality and minimizes the CDN cost in 3 out of the 4 plotted settings (and 2 cache ASes are needed in the 4th setting). When the performance constraint becomes more stringent, the footprint of the optimal CDN deployment consistently expands by employing more cache ASes. For each of the 4 traffic/pricing settings, the footprint expansion is roughly exponential in the number of cache ASes.

Assessing the sensitivity of the performance-footprint trade-offs to the topology and traffic demands, Figures 2.3a, 2.3b, and 2.3c report similar trade-off profiles in the UCLA, CAIDA1, and CAIDA2 topologies with traffic-demand vector T1. However, the trade-off profiles with traffic-demand vector T2, which shifts large traffic demands towards the edges of the topology, are quite different: the optimal CDN deployment tends to involve a larger number of cache ASes than with vector T1, which places large traffic demands toward the topological center. Hence, CaDeOp solutions are more sensitive to traffic demands than topology.



(a) with the UCLA topology and T1

(b) with the UCLA topology and T2

(c) with the CAIDA2 topology and T1

Figure 2.4: Split of the cache ASes according to their centrality when the bandwidth pricing is location-oblivious.

To understand which ASes are chosen for the cache deployment, we classify the cache ASes according to their centrality types. Figure 2.4a plots the distribution of the cache

ASes as per their centrality with traffic-demand vector T1 and location-oblivious bandwidth pricing.

For $d = 2$ or $d = 1.6$, the only cache AS of the optimal deployment belongs to the large-cone type, i.e., the center of the topology. When the performance constraint tightens, the fraction of large-cone ASes in the CDN footprint steadily declines, and the CDN spreads its cache ASes through the topology. For $d = 0.1$, the fractions of the large-cone, medium-cone, small-cone, tiny-cone ASes in the optimal CDN deployment are respectively 9%, 10%, 52%, 29% which closely approach the corresponding 5%, 9%, 52%, 34% shares of these AS types in the overall AS population of the UCLA topology.

Figure 2.4b depicts counterpart results for traffic-demand vector T2 and shows similar footprint-performance trade-offs: while the optimal deployment with $d = 2$ consists of one large-cone AS, the cache ASes spread through the topology away from its center when the delivery quality requirements become more stringent. With traffic-demand vector T2, the fraction of large-cone ASes has a steeper decline, and the optimal deployment shifts the cache ASes towards the edges of the topology more aggressively, because this vector places large traffic demands away from the topological center.

Switching the topology from UCLA to CAIDA2, Figure 2.4c shows the split of the cache ASes according to their centrality type with traffic-demand vector T1 and location-oblivious bandwidth pricing. When the performance constraint tightens from 0.8 to 0.1, the footprint expansion is qualitatively the same as in the UCLA topology: the CDN spreads its cache ASes away from the topological center. On the other hand, when the performance constraint is loose, the qualitative picture differs from the UCLA case: the CDN locates its consolidated footprint in tiny-cone or small-cone ASes of CAIDA2.

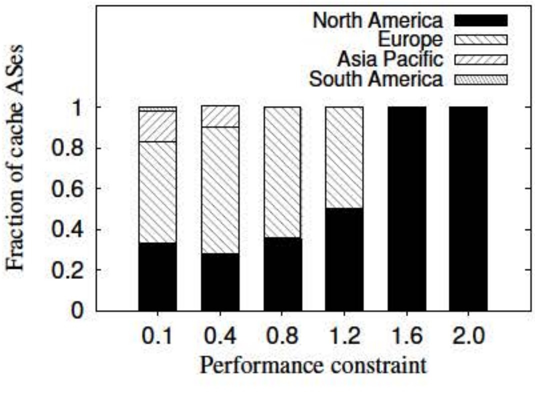


Figure 2.5: Geographic distribution of cache ASes in the optimal deployment for the UCLA topology, traffic-demand vector T1, and location-oblivious bandwidth pricing.

Turning our attention to the geography of the optimal CDN deployment, Figure 2.5

depicts the geographic profile of the cache ASes for the UCLA topology, traffic-demand vector T1, and location-oblivious pricing. For $d = 2$ or $d = 1.6$, the only cache AS of the optimal footprint is based in North America. When the delivery quality requirements become stricter, the optimal footprint expands first to Europe, then to Asia-Pacific, and eventually to South America. This geographic perspective confirms the more general observation that content delivery at a high quality necessitates an extensive CDN footprint throughout the topology.

The above evaluation can be summed up as follows: *When the delivery-quality needs become more stringent, the CDN footprint expands rapidly, requiring cache deployments in additional ASes and geographical regions.*

2.4.3 Traffic patterns

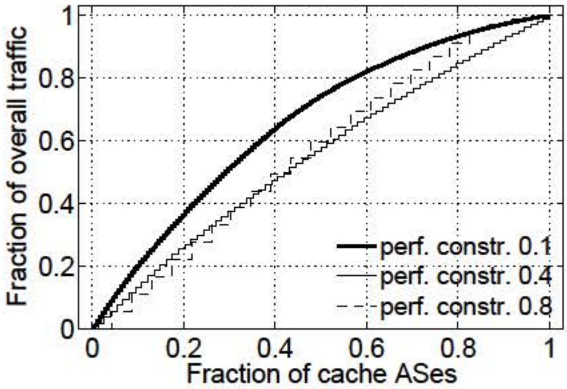
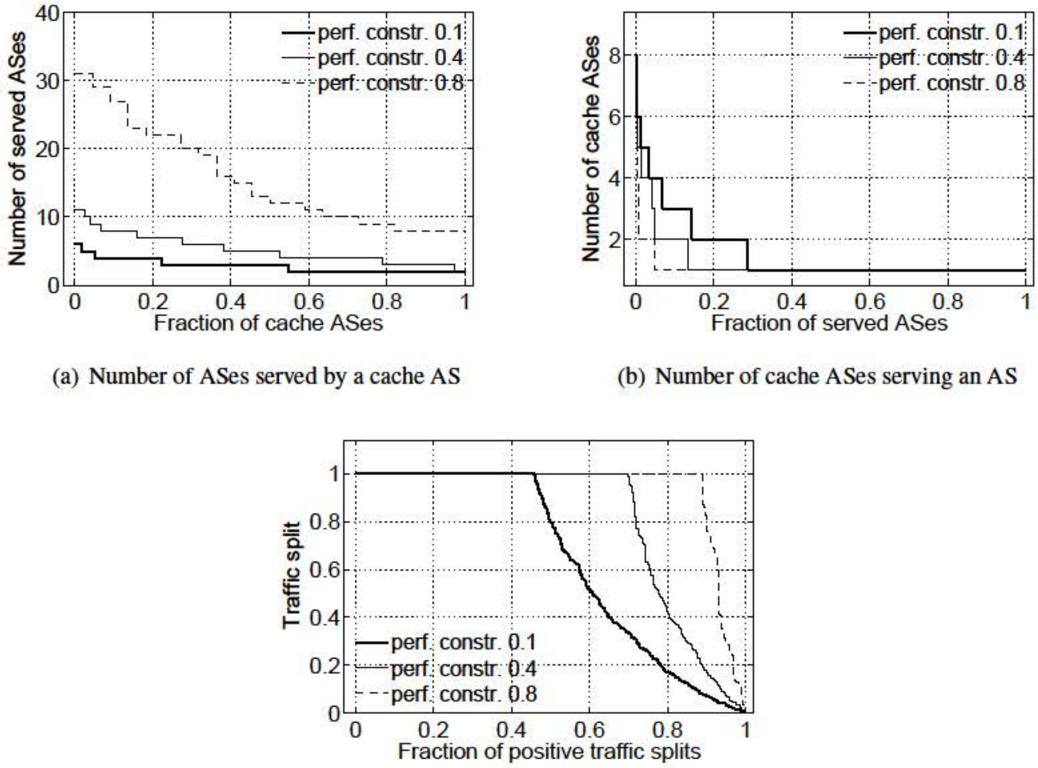


Figure 2.6: Distribution of the overall traffic among cache ASes with the UCLA topology, traffic-demand vector T1, and location-oblivious pricing.

We now examine how the optimal footprint satisfies the traffic demands of end users. Figure 2.6 depicts overall-traffic vector V, in the decreasing order of the V_i values, for different delivery-quality needs with the UCLA topology, traffic-demand vector T1, and location-oblivious pricing. For $d = 0.8$ when the optimal footprint involves 22 cache ASes, the overall traffic is spread among the cache ASes quite uniformly, with the exception of a few cache ASes that have the lowest load. In the 3 plotted distributions for the performance constraints between 0.8 and 0.1, the top 20% of the cache ASes serve around 30% of the overall traffic, and the top 50% of the cache ASes serve around 65% of the overall traffic. Hence, the distribution of the overall traffic among cache ASes stays relatively even.

Figure 2.7 maps served ASes to cache ASes. With $d = 0.8$, Figure 2.7a shows that the individual cache ASes serve between 31 and 8 ASes each. With $d = 0.4$ and $d = 0.1$, the maximum number of ASes served by the same cache AS reduces to 11 and 6 respectively, while the minimum number is 2 ASes which include the cache AS itself. Figure 2.7b shows that most of ASes receive content from only one cache AS. With $d = 0.1$, only 29% of all ASes are served by multiple cache ASes, and the maximum number of cache ASes serving the same AS is 8. For the looser performance constraints, the fraction of ASes served by multiple cache ASes shrinks even further.



(a) Number of ASes served by a cache AS

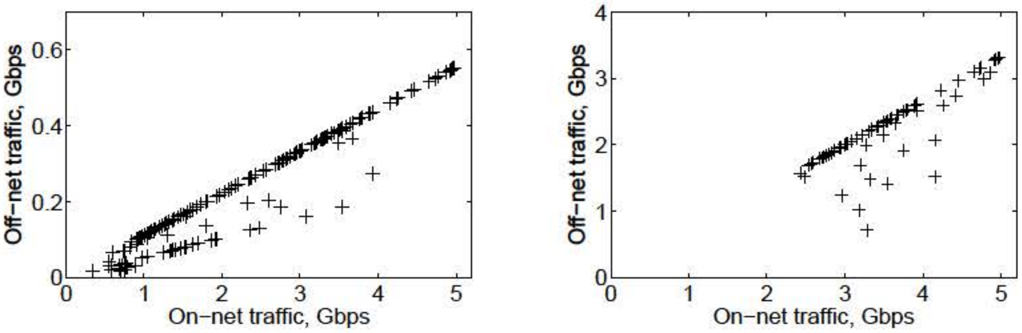
(b) Number of cache ASes serving an AS

(c) Distribution of traffic splits $m_{i,j}$

Figure 2.7: Mapping of served ASes to cache ASes for the UCLA topology, traffic-demand vector T1, and location-oblivious pricing.

Focusing on the ASes served by multiple cache ASes, Figure 2.7c plots the cumulative distribution of positive traffic splits m_{ij} and reveals that the traffic splits are spread relatively smoothly between 1 and 0. Thus, tightening the performance constraint decreases the fraction of ASes served by multiple cache ASes but the traffic of such served ASes remains distributed smoothly among their multiple cache ASes.

To expose the traffic patterns in more detail, Figure 2.8 presents scatter plots of the on-net vs. off-net traffic of the cache ASes. When the performance constraint becomes more stringent and increases the number of cache ASes, the off-net traffic of individual cache ASes expectedly decreases. Figure 2.8 also reveals that the off-net traffic of a cache AS is bounded from above by a linear function of the on-net traffic with a slope of $d/(1-d)$. The bound is a consequence of inequalities 2.4 because serving the off-net traffic of a cache AS at a rate that is larger than $d/(1-d)$ times the on-net traffic rate would violate the performance constraint.



(a) performance constraint of 0.1

(b) performance constraint of 0.4

Figure 2.8: Scatter plots of the on-net vs. off-net traffic of the cache ASes for the UCLA topology, traffic-demand vector T1, and location-oblivious bandwidth pricing.

Perhaps more surprising is that a large number of points coincide with the linear bound. This happens when the performance constraint is dominant, and inequalities 2.4 are satisfied with equalities for many cache ASes in the optimal deployment. Each of these cache ASes serves the off-net traffic at the maximum rate that does not violate the performance constraint. In the above examination of CDN traffic patterns, we can highlight the following observations: *The traffic distribution among the cache ASes stays relatively even, with the top 20% of the cache ASes serving around 30% of the overall traffic. For a large fraction of the cache ASes, the off-net traffic is proportional to the on-net traffic of the AS.*

For the UCLA topology, traffic-demand vector T1, and location-oblivious bandwidth pricing, Figures 2.4a and 2.9 compare the cache ASes and their overall traffic. The comparison shows that the cache ASes and overall traffic have qualitatively similar distributions when split according to the centrality type of the cache ASes.

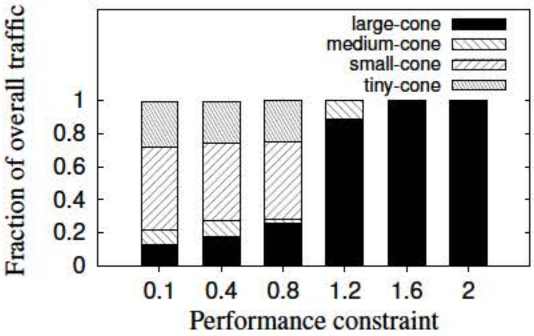
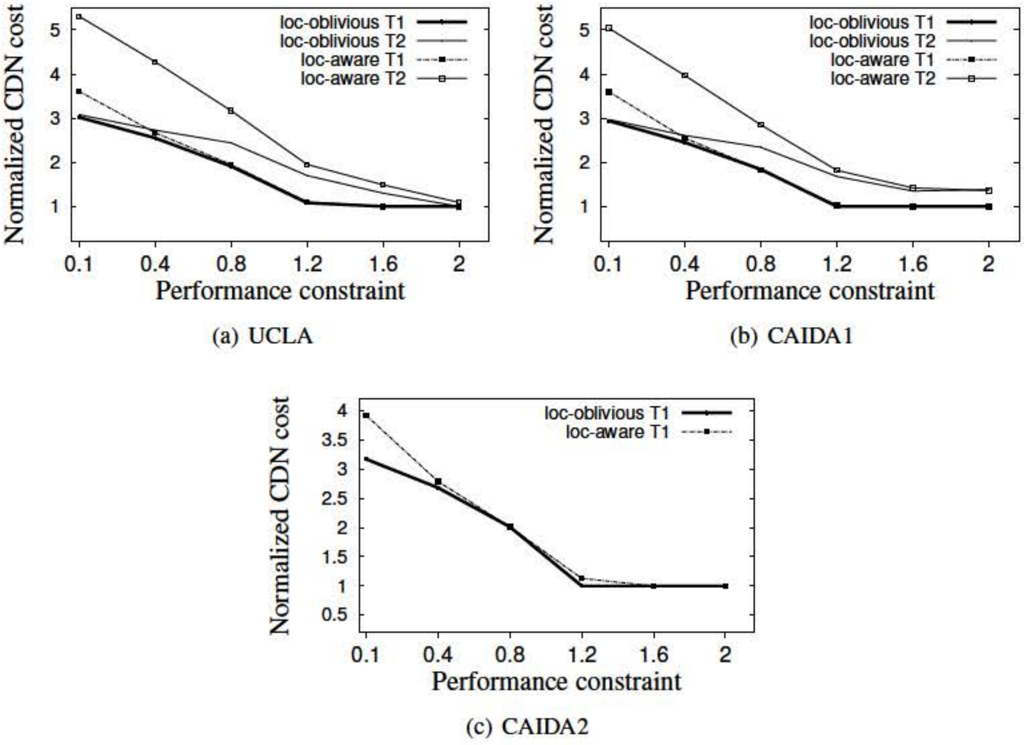


Figure 2.9: Split of the overall traffic according to the centrality of the cache ASes with the UCLA topology, traffic-demand vector T1, and location-oblivious bandwidth pricing.

2.4.4 CDN cost

While the above evaluation focuses on the CDN footprint and traffic patterns, we now examine the cost of the optimal deployment. Figure 2.10 plots the normalized CDN cost in the UCLA, CAIDA1, and CAIDA2 topologies.



(a) UCLA

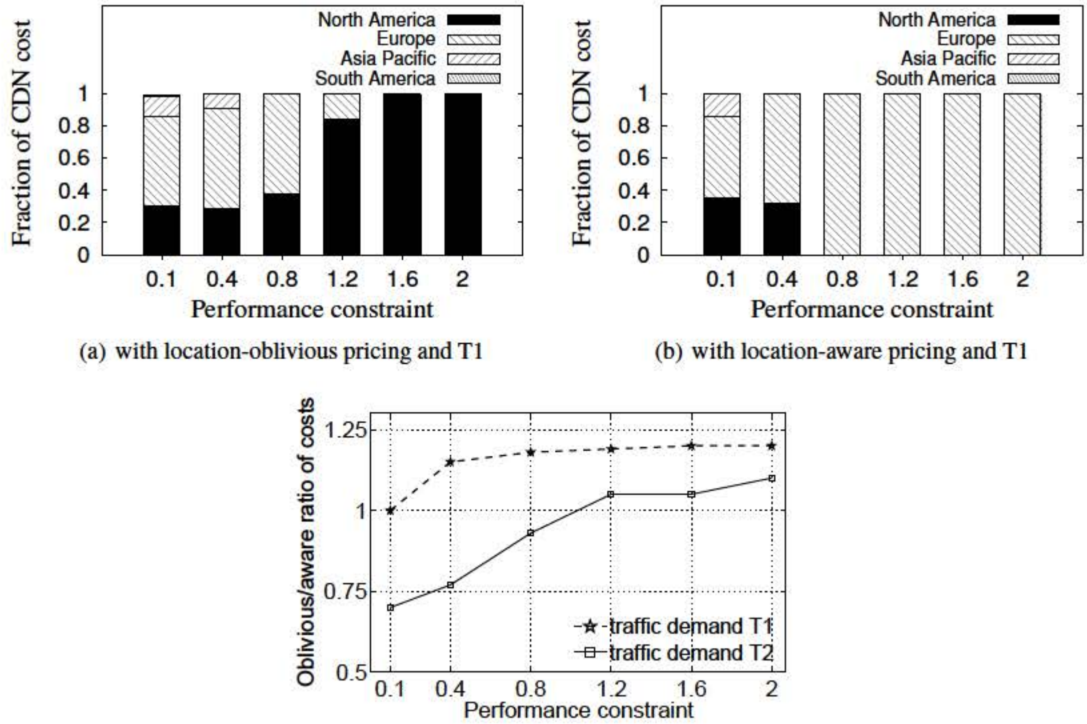
(b) CAIDA1

(c) CAIDA2

Figure 2.10: Trade-offs between the CDN cost and performance in the UCLA, CAIDA1, and CAIDA2 topologies.

The normalization is done with respect to the CDN cost with no constraint on the performance, i.e., when the optimal deployment involves only one cache AS. The plot shows that tightening the performance constraint increases not only footprint but also cost. For $d = 0.1$, the CDN cost is several times larger than in the baseline single-AS deployment.

We now examine how the CDN cost depends on the awareness of location-specific bandwidth prices. For the UCLA topology and traffic-demand vector T1, Figures 2.11a and 2.11b depict the geographic distributions of the CDN cost with location-oblivious and location-aware bandwidth prices respectively. When the performance constraint is loose, the awareness shifts the entire deployment from North America to Europe due to the lower prices in the latter.



(a) with location-oblivious pricing and T1

(b) with location-aware pricing and T1

(c) ratio of the CDN costs

Figure 2.11: The CDN cost for the UCLA topology with location-oblivious vs. location-aware bandwidth pricing.

When d tightens, the geographic distribution of the CDN cost involves additional regions and converges to similar regional splits for both pricing models because the caches in the expanding footprint are deployed closer to the end users. Nevertheless, the location-specific pricing affects the CDN cost even with tight performance constraints.

For traffic-demand vectors T1 and T2, Figure 2.11c shows that the ratio of the CDN costs with location-oblivious and location-aware prices varies from 1.2 (with $d = 2$ and T1) to 0.7 (with $d = 0.1$ and T2). For tighter d values, the oblivious/aware ratio of the CDN costs consistently diminishes because the CDN deploys caches in new geographical regions that have higher prices. Hence, the awareness of location-specific prices has profound effects.

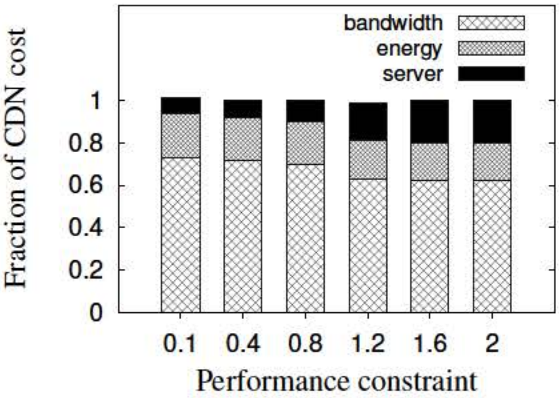


Figure 2.12: Split of the CDN cost among the bandwidth, energy, and server categories for the UCLA topology, traffic-demand vector T1, and location-oblivious pricing.

Figure 2.12 tracks the split of the CDN cost among the bandwidth, energy, and server categories. When the delivery-quality requirements become more stringent and expand the footprint with new cache ASes, the CDN ability to benefit from the economies of scale in energy and bandwidth costs diminishes, and the fractions of these costs in the CDN cost increase. On the other hand, while the server costs do not change, the relative share of the server costs in the increasing CDN cost declines. Thus, when the performance constraint tightens, the cost balance shifts toward the bandwidth and energy costs.

The following are the main insights from this last portion of our evaluation: *When the delivery-quality requirements become more stringent, the CDN cost rises several times, and the cost balance shifts toward bandwidth and energy costs.*

2.5 Future Extensions

2.5.1 InCaDeOp: Incremental Cache Deployment Optimization

To address emerging traffic, performance, and cost challenges, a CDN needs to adjust its cache deployment over time. Furthermore, beyond the time horizon of the next deployment upgrade, the new requirements become, to a large extent, unpredictable. This unpredictability is costly in the long term, e.g., because relocation of caches to accommodate the new requirements has a cost. Hence, after a series of incremental upgrades, a realistic CDN cache deployment is likely to be less optimal than in the hypothetical scenario of accommodating the current requirements from scratch. Thus, we explore the problem of upgrading the CDN cache deployment to satisfy the performance requirements of end users, while minimizing the cache deployment cost. We refer this problem as Incremental Cache Deployment Optimization (InCaDeOp) and present its formulation below. The robust evaluation of the InCaDeOp model has been left for future work.

2.5.2 Formulating InCaDeOp

We consider a time instance when the CDN plans to upgrade its current cache deployment. A connected directed graph G represents ASes and their interconnections at the upgrade time. ASes are the nodes of the graph and form set N. The edges denote inter-AS economic relationships classified as transit or peering. End users of the CDN are potentially located in any of the N ASes. Traffic demand T_j is measured in Mbps and refers to the overall rate of the content traffic that the CDN needs to deliver to the end users in AS j.

To satisfy the performance requirements of the end users, the CDN upgrades its current cache deployment. The current deployment is hosted by set X of ASes, which might differ from set H of the cache ASes in the upgraded deployment. P_k refers to the maximum rate at which the CDN can serve content from its current caches in AS k, with $k \in X$. Because the current cache deployment might be suboptimal, the current service might underutilize the available capacity at AS k and transmit content at a smaller rate than P_k.

The upgrade might be accomplished by purchasing additional servers or by relocating those servers that are no longer needed in their current location. Let $R_{k,i}$ refer to the traffic rate of the servers relocated from AS k to AS i:

$$R_{k,i} \geq 0 \qquad \forall k \in X \; \forall i \in H. \tag{2.7}$$

Servers are not relocated within the same AS because this would not change the traffic capability of the AS:

$$R_{i,i} = 0 \qquad \forall i \in H. \tag{2.8}$$

Also, the ability to relocate servers from AS k is limited by their availability:

$$\sum_{i \in H} R_{k,i} \leq P_k \qquad \forall k \in X. \tag{2.9}$$

With A_i denoting the traffic rate supported by the additionally purchased servers, the following inequality assures that each AS in the upgraded deployment has sufficient traffic capability to satisfy the performance requirements of the end users:

$$V_i \leq P_i - \sum_{j \in H} R_{i,j} + \sum_{k \in X} R_{k,i} + A_i \qquad \forall i \in N. \tag{2.10}$$

Notation	Semantics
G	AS-level topology of the Internet core
N	set of ASes in G
T_j	traffic demand of the end users in AS j
H	set of cache ASes
l_i	bit indicating whether AS $i \in H$
$D_{i,j}$	AS-hop distance from cache AS i to AS j
m_{ij}	fraction of T_j satisfied by cache AS i
V_i	overall traffic of cache AS i
O_i	on-net traffic of cache AS i
F_i	off-net traffic of cache AS i
q	performance constraint
X	set of ASes in the current cache deployment
P_k	current traffic capability in AS k
$R_{k,i}$	traffic capability relocated from AS k to AS i
A_i	traffic capability purchased for AS i
Y_i	relocation cost for AS i
Z_i	purchase cost for AS i
S_i	server cost for cache AS i
E_i	energy cost for cache AS i
B_i	bandwidth cost for cache AS i
y, z, s, e, b_i	cost-function factors
u, w	cost-function exponents
C	CDN cost

Table 2.6: Notation in the InCaDeOp problem formulation.

The relocation of servers to AS i and purchase of additional servers for AS i have the following costs respectively:

$$Y_i = y \cdot \sum_{k \in X} R_{k,i} \text{ and } Z_i = z \cdot A_i \qquad \forall i \in N \tag{2.11}$$

where y and z are positive constant factors.

When upgrading the deployment, the CDN strives to minimize its costs. These include the costs of the upgrade itself and of operating the upgraded deployment:

$$C = \sum_{i \in H} (Y_i + Z_i + S_i + E_i + B_i) \,. \tag{2.12}$$

The objective in InCaDeOp is to minimize C. With all the notation summarized in Table 2.6, we formulate the InCaDeOp problem as follows:

- *Inputs:* topology G, AS-level paths in G, vector T of traffic demands, set X of ASes in the current cache deployment, vector P of current traffic capabilities, cost-function parameters y, z, s, e, b_i, u, w, and performance constraint q;
- *Outputs:* set H of cache ASes, traffic-split matrix m, overall-traffic vector V, on-net traffic vector F, off-net traffic vector J, capability-relocation matrix R, capability-purchase vector A, relocation-cost vector Y, purchase-cost vector Z, server-cost vector S, energy-cost vector E, bandwidth-cost vector B, and CDN cost C;
- *Constraints:* equalities 2.2 and 2.8 and inequalities 2.1, 2.4, 2.7, 2.9, and 2.10;
- *Objective:* minimize CDN cost C.

2.6 Related work

Cache placement has been studied in models that do not account for the economic structure of the Internet [21–23]. Because the economic considerations are crucial for CDN planning, we go beyond the single-network perspective and examines cache deployment in realistic AS-level Internet topologies.

Complementary to the planning problem of CaDeOp that we study, there is much prior work on the real-time operations of the CDN in areas such as dynamic content management and load balancing [22,23,31,53–55].

There exists recent work on data-center placement and upgrade [56, 57]. The data-center optimizations are based on physical resources, e.g., water, energy, and land. While CDN caches are deployed in data centers, the performance and cost considerations for deploying CDNs are drastically different from those relevant for data-center operators.

The problem of optimizing the set of upstream ASes for a multihomed network [58] has similarities with the CaDeOp subproblem of choosing the set of cache ASes. However, CDN specifics necessitate a different formulation and solution methodology for CaDeOp.

Optimization techniques have been used to study telco CDNs and IXP-colocated CDNs [59, 60]. However, there is no prior study of multi-AS deployment optimization with realistic cost and performance constraints, such as the ones we examined.

Reducing the energy cost of a CDN has been a focus in earlier work [32, 61]. Our investigation takes a more comprehensive view by considering all major sources of the CDN cost, including energy, bandwidth, and server costs, in the joint CaDeOp problem.

2.7 Conclusion

Our work is the first to formulate and solve the CaDeOp (Cache Deployment Optimization) problem of determining an optimal set of cache ASes in the Internet core and provisioning server, energy, and bandwidth resources in each cache AS. CaDeOp strives to minimize the CDN cost while satisfying the end-user performance requirements. Our evaluation of the CaDeOp solutions exposed trade-offs in CDN deployment for realistic AS-level topologies, Internet routing, traffic demands, and non-linear costs for energy and bandwidth. We also evaluated the sensitivity of the conclusions to our parametric assumptions. When the delivery-quality needs become more stringent, the CDN footprint expands rapidly, requiring cache deployments in additional ASes and geographical regions. Also, the CDN cost increases several times, with the cost balance shifting toward bandwidth and energy costs. On the other hand, the traffic distribution among the cache ASes stays relatively even, with the top 20% of the cache ASes serving around 30% of the overall traffic.

Chapter 3

Obscure Giants: Detecting the Provider-Free ASes

The previous chapter showed the trade-offs in optimizing cache deployments of CDNs in the Internet core composed of transit ASes. While Chapter 2 relied on knowledge of the Internet core topology, we now shift the focus onto Internet topology inference and its economics implications. This chapter proposes an algorithm to detect the set of provider-free ASes (PFS). PFS is learnt by the application of topology inference and valuable for understanding Internet resilience and economics of the Internet core.

Economic relationships between ASes (Autonomous Systems) matter for Internet routing. For example, it is financially more attractive for an AS to route traffic through a peering link than a transit connection of the AS to its provider. Despite a trend towards flattening [5], the Internet routing ecosystem is essentially hierarchical [2, 5, 7, 8]. A vast majority of ASes are relatively small and route traffic either as customers of transit links or by peering with local ASes of a similar stature. There exists only a handful of *provider-free ASes* that reach the entire Internet without paying anyone for the traffic delivery. While a tier-1 network is a more common name for a provider-free AS, we use the latter term because prior attempts to redefine AS tiers make network tiering an ambiguous notion. The *set of the provider-free ASes*, to which we refer as *PFS*, contains only large networks. Nevertheless, the real difference between them and another large network can be subtle. For example, if a network is not a provider-free AS because it pays for less than 1% of its inter-domain traffic, the lack of the provider-free status can be obscure to outsiders, especially if the disqualifying payments are for a paid peering relationship which is subject to a non-disclosure agreement.

Due to the general reluctance of ASes to disclose their business agreements, researchers infer the inter-AS economic relationships from BGP (Border Gateway Protocol) [1] route advertisements or actual IP (Internet Protocol) [3] forwarding routes. Such inferences are imperfect, e.g., a router misconfiguration can trigger an inference of an invalid relationship. Also, the inference algorithms are heuristic and can cause additional deviations from the reality. Finally, the economic relationships are dynamic: while it takes time to collect a comprehensive set of measurements, changes in the relationships can decrease the inference accuracy. Over the past decade, numerous research efforts have tried but failed to infer inter-AS economic relationships precisely. We explore this failure in the specific context of provider-free ASes.

Our interest in PFS arises due to a number of reasons. First, the provider-free ASes clearly play a key role as the transit core of the Internet ecosystem. By delivering a significant portion of Internet traffic, PFS is highly relevant to the overall resilience of the Internet to accidental failures and intentional disruptions. In particular, economic disputes between provider-free ASes can endanger the universal connectivity of Internet users. Second, while humans prefer to think in discrete categories, designation of an autonomous system as provider-free can have tangible marketplace implications. Third, some algorithms for inter-AS relationship inference use PFS as an input [8, 9] and hence need to know PFS accurately.

In this chapter, we contribute by developing an algorithm that detects PFS from public datasets of inter-AS economic relationships. We show that straightforward extraction of PFS from the public datasets yields poor results. Our alternative algorithm utilizes topological statistics (customer cones of ASes) and temporal dataset diversity. The more sophisticated algorithm infers PFS with a significantly higher accuracy. Although a lot of related studies deal with the more general problem of inter-AS relationship inference, our algorithm succeeds by focusing on the more specific problem of PFS detection. Another group of related work redefines tier-1 networks as per a new classification of Internet ASes, e.g., based on their graph-theoretic topological properties. In contrast, our study detects provider-free ASes in accordance to the traditional tier-1 definition. The two main contributions are in deriving:

- *PFS insights from mostly trustworthy but non-verifiable sources*; we carefully filter out occasional spurious answers;
- *TC (Temporal Cone) algorithm that detects PFS based on public datasets of inter-AS economic relationships*; the derived TC algorithm is useful because it enables accurate inferences of PFS in the future even if PFS insights from the non-verifiable sources become unavailable.

While the knowledge of PFS is highly valuable, validation of PFS inference results constitutes a major challenge because the ground truth lies outside the public domain. To tackle the validation challenge, we utilize trustworthy but non-verifiable sources such as Wikipedia [45]. These sources do not disclose their data and methods. Thus, their conclusions are not purely scientific. Nevertheless, our conversations with network operators indicate that the non-verifiable sources reflect the reality accurately. Whereas it seems practically impossible to obtain the complete ground truth from network operators, the non-verifiable source insights form the best available baseline for result validation in this important domain. As a midpoint between traditional science and citizen science [62,63], our PFS detection method expands the scope of knowledge but softens the benchmark for validation.

We structure the rest of the chapter as follows. Section 3.1 reports PFS insights from the non-verifiable sources. Section 3.2 describes the public datasets in our study. Section 3.3 considers a straightforward PFS detection method. After analyzing the failures of this straightforward method, Section 3.4 develops the more sophisticated TC algorithm. Section 3.5 evaluates the TC algorithm. Section 3.6 comments on related work. Section 3.7 concludes the chapter by summing up its contributions.

3.1 Non-verifiable sources

While the obscure inter-AS economic relationships do not reveal the ground truth about PFS, a number of non-verifiable sources offer insights into this set. We consider three such non-verifiable sources: Wikipedia, Renesys, and Hurricane Electric.

Wikipedia maintains an article about provider-free ASes [45]. Our primary interest is in the Wikipedia perspectives throughout 2009 because the development of our TC algorithm relies on public datasets collected during that year. According to Wikipedia, PFS consisted of 8 members on 1/1/2009: AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Verizon, and Savvis [64]. The article has seen frequent revisions and expanded its PFS with Telia on 28/1/2009 [65]. The addition of Tata on 25/3/2009 resulted in the following PFS [66]:

$W_1 = \{$AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Verizon, Savvis, Telia, Tata$\}$.

Except for few incidents in June and October when spurious modifications disappeared shortly after being made, PFS preserved this 10-member composition until the end of 2009. In 2010 and 2011, Wikipedia continued the trend of the PFS expansion and typically recognized Tinet as the 11th member of the PFS, e.g., in the 10/2/2011 revision [67]:

$W_2 = \{$AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Verizon, Savvis, Telia, Tata, Tinet$\}$.

Whereas Wikipedia is an online encyclopedia that anyone may edit, some short-lived revisions of this particular article certainly distorted the reality [68]. Nevertheless, experts think that on the whole the Wikipedia perspective reflects PFS accurately [8].

Renesys is a private company that sells Internet business information. In 1/2009, Renesys announced a 12-member set of commercial default-free ASes [69], i.e., ASes that can route traffic to any Internet destination without relying on a default route. Default-free ASes are either provider-free or reaching the entire Internet by paying for peering but not for transit. The Renesys set subsumes W_1 and includes two more ASes: XO and AboveNet. Interestingly, the Wikipedia article explicitly stated in all its revisions that XO and AboveNet were not provider-free due to paid peering [64–68]. Thus, the Renesys perspective is consistent with limiting PFS to W_1.

Hurricane Electric is an Internet service provider that offers an online tool for ranking the peers of an autonomous system [70]. The specific criteria for the ranking are not clear but seem to rely on the number of active BGP connections for the AS or the percentage of BGP paths transiting the AS. For each AS in W_1, all the other ASes in W_1 are among highly ranked peers of this AS according to the Hurricane Electric tool. Thus, the ASes of W_1 do form a close-knit peering community as expected for provider-free ASes.

Based on the above considerations, we subsequently treats W_1 as the primary PFS answer from the non-verifiable sources for 2009.

3.2 Public datasets

PFS insights in Section 3.1 came from the non-verifiable sources that did not disclose their data and methods. The rest of our study explores datasets from two public sources: UCLA (University of California, Los Angeles) [42] and CAIDA (Cooperative Association for Internet Data Analysis) [43]. The datasets from both public sources characterize the economic relationships between Internet ASes. UCLA classifies inter-AS links as transit or peering. CAIDA uses an additional category for sibling relationships: a sibling link connects two ASes belonging to the same Internet service provider.

Both UCLA and CAIDA leverage BGP measurements but employ different methods to infer the economic relationships from the BGP data. The UCLA method utilizes the Route Views [71] and RIPE RIS (Réseaux IP Européens Routing Information

Service) [72] measurement infrastructures where route collectors engage via BGP with routers in strategic Internet locations to collect AS-level path announcements. The routers that supply the announcements are called BGP monitors. UCLA collects the announcements from BGP monitors located in provider-free autonomous systems (as identified by Wikipedia). The UCLA method categorizes the collected inter-AS relationships into peering and transit based on the valley-free routing conditions [73] and depending on how consistent the views from different monitors are.

CAIDA looks up the IRR (Internet Routing Registry) database [74] to detect sibling links: if two linked ASes belong to the same organization as per the database, the CAIDA method classifies the relationship between the autonomous systems as a sibling link. To infer peering and transit links, CAIDA relies on BGP measurements from Route Views. The CAIDA heuristic for identifying and directing the transit links strikes a balance between maximizing the following two metrics: (1) number of BGP paths that are valid according to the valley-free routing rules and (2) number of links where the provider node of the link has a higher degree than the customer node of the link (the degree of a node refers to the number of links between this node and other ASes). With the CAIDA method, peering relationships are links between nodes with similar degrees.

While the UCLA datasets are available starting from 10/2008, CAIDA reports its datasets infrequently for 2009 and only twice after 2009. During the development of our PFS detection algorithm, it would be desirable to have similar time series for the two sources. Hence, our Sections 3.3 and 3.4 focus on the 12 months of 2009. Guided by the CAIDA dataset availability and picking one day per month, we select the following days for both sources: 22/1, 20/2, 11/3, 29/4, 20/5, 15/6, 20/7, 30/8, 20/9, 20/10, 20/11, and 15/12. June is the only exception: because the number of links in the UCLA 15/6 dataset is extremely low, we use 16/6 instead for UCLA.

Figure 3.1 depicts the inter-AS economic relationships in the UCLA and CAIDA datasets during 2009. For either source, the total number of links tends to grow with time, and the few down-and-up swings are most likely due to imperfect measurements rather than actual fluctuations in the number of economic relationships. The sibling relationships in the CAIDA datasets constitute a negligible fraction of the overall link population. While the number of peering links is much higher for UCLA than for CAIDA, the number of transit links is rather similar for the two sources. The transit-link profiles are mostly consistent but do have some aberrations such as the dip for CAIDA in 4/2009. The numbers of transit links for the two sources remain most stable and close to each other between 5/2009 and 7/2009.

When evaluating our TC algorithm in Section 3.5, we utilize the UCLA datasets for

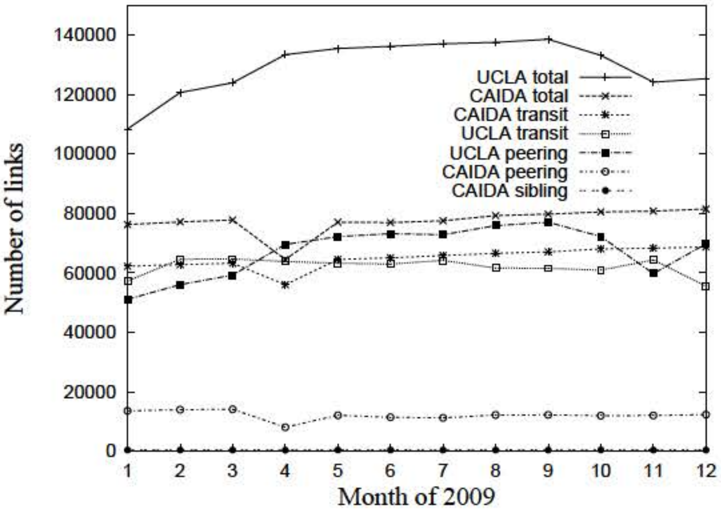


Figure 3.1: Inter-AS economic relationships in the UCLA and CAIDA datasets during 2009.

all 32 months of their availability from 10/2008 to 5/2011. We select the 20th day for all 20 additional months except 5/2011, for which we use 10/5/2011 as the last day of our data gathering.

3.3 Straightforward inference

Given a dataset of inter-AS economic relationships, one might hope to infer PFS using the following *straightforward method: compose PFS from all such ASes in the dataset that have no transit provider*. We apply this straightforward method to the UCLA and CAIDA datasets of Section 3.2. Table 3.1 sums up the generally disappointing results for all 12 months of 2009. Throughout the year, the straightforward method includes into its PFS up to 23 non-W_1 ASes and excludes up to all 10 ASes of W_1. For the UCLA and CAIDA datasets from 6/2009 (when the numbers of transit links for the two sources remain most stable and close to each other), PFS contains respectively 17 and 27 ASes, with respectively 9 and 7 of these ASes belonging to W_1.

For the UCLA 6/2009 dataset, the straightforward method excludes Tata from PFS because NTT and GIT Telecom (a Cypriot AS) are transit providers for this missing member of W_1 according to the dataset. Among the 8 non-W_1 members of PFS, Sunkist Growers (a not-for-profit cooperative of citrus growers in California and Arizona), Open Peering Initiative (a public peering IXP in Amsterdam), and Siemens seem highly unlikely to be genuine provider-free ASes. These 3 ASes do have providers in the CAIDA

1	2	3	4	5	6	7	8	9	10	11	12
8 (1)	6 (0)	7 (0)	17 (9)	16 (9)	17 (9)	15 (9)	19 (9)	19 (9)	16 (10)	17 (9)	18 (9)
23 (6)	26 (6)	26 (6)	29 (7)	30 (7)	27 (7)	28 (7)	29 (7)	25 (6)	26 (6)	27 (6)	27 (6)

Table 3.1: PFS size according to the straightforward method for the datasets of UCLA (row 2) and CAIDA (row 3) and (in parentheses) number of ASes from W_1 in this PFS monthly for year 2009.

dataset from the same month.

For the CAIDA 6/2009 dataset, the straightforward method omits NTT, Savvis, and Tata from PFS because these 3 members of W_1 have transit providers. Specifically, NTT has 3 providers: Verizon, Telia, and Easynet. Savvis has 5 providers: Telia, Tata, Tinet, XO, and Deutsche Telekom. Although Tata is a transit provider for Savvis, the straightforward method does not recognize Tata as a provider-free AS either: Tata appears as a customer of NTT, Telia, and Tinet. On the other hand, PFS of the straightforward method includes 20 non-W_1 ASes such as the University of Texas System, NASA, and New Zealand Research Network, which do have providers in the UCLA 6/2009 dataset.

Link misclassification in the datasets is the most common source of errors for the straightforward method. The UCLA and CAIDA datasets are typical in this regard. We applied the straightforward method to another dataset inferred with Gao's algorithm, and the respective results suffer from the link misclassification as well.

3.4 TC algorithm

Section 3.3 demonstrates that the straightforward inference yields disappointing PFS results with respect to both false positives and false negatives. Two factors undermine the straightforward method. First, while the UCLA and CAIDA datasets do not classify the inter-AS links fully and correctly, even a single error in the input dataset can mislead the straightforward method. The method can exclude a genuine provider-free AS (e.g., Tata in the UCLA 6/2009 dataset) from PFS because the dataset mistakenly reports a provider for this AS. Also, the method can wrongly include an AS (e.g., Sunkist Growers) into PFS because the dataset misses the transit link between this AS and its provider. Second, the straightforward method implicitly assumes that having no provider implies the ability to reach the entire Internet. In reality, some ASes in the Internet ecosystem do not strive for the universal reachability. For example, the main goal of an IXP (Internet eXchange Point) [5, 75] is to serve as a peering infrastructure that enables other ASes to exchange their local traffic. The straightforward method can incorrectly classify an IXP (e.g., Open

Peering Initiative) as a provider-free AS.

Thus, we develop a more sophisticated TC (Temporal Cone) algorithm for detecting PFS. Sections 3.4.1, 3.4.2, and 3.4.3 discuss the three important components of our algorithm: its use of topological statistics to deal with the noisy data, setting the PFS size, and exploiting the temporal diversity of the datasets to improve the accuracy of the PFS detection further.

3.4.1 Customer-cone ranking

Topological statistics represent a promising basis for accurate PFS detection because of their potential resilience to individual errors caused by the link misclassification. While the datasets of inferred inter-AS relationships clearly contain numerous errors, our approach relies on the premise that the datasets are also rich in correct information and that looking at the datasets from a right perspective can reveal PFS accurately.

After examining a number of options, we choose the *customer cone* as the topological parameter for the TC algorithm: the customer cone of an AS includes the AS itself as well as all direct and indirect customers of the AS, i.e., every customer reachable from the AS through a sequence of provider-to-customer transit links [44]. We expect the customer cones of the provider-free ASes to be among the largest because the customer cone of an AS is strictly larger than the customer cone of any of its customers. This expectation is certainly a heuristic (in principle, a provider-free AS can have a smaller customer cone than a network that lies outside this customer cone and has a provider) but our results confirm its effectiveness. Due to multihoming [76] which is common throughout the Internet ecosystem, the customer cones of two ASes can overlap. We compute the customer cone of each AS using a recursive algorithm that takes the overlaps of the customer cones into account.

To illustrate the potential of the customer cone for PFS detection, let us revisit the false negatives and false positives of the straightforward method for the 6/2009 datasets in Section 3.3. For the UCLA 6/2009 dataset, the straightforward method computes the PFS that incorrectly excludes Tata and wrongly includes Sunkist Growers, Open Peering Initiative, and Siemens. The customer cones of Tata, Sunkist Growers, Open Peering Initiative, and Siemens are 26014, 69, 75, and 8 ASes respectively. While the customer cone of 26014 ASes is the 13th largest among all networks in the dataset, the customer-cone perspective leaves Tata as a plausible candidate for PFS. On the other hand, the small customer cones of Sunkist Growers, Open Peering Initiative, and Siemens clearly suggest that these 3 networks are not provider-free ASes. Similarly, for the CAIDA 6/2009 dataset, the 3 false negatives of the straightforward method are NTT, Savvis, and Tata which have very large customer cones of 24473, 23769, and 23788 ASes respectively. The University of Texas System, NASA, and New Zealand Research Network are false positives of the straightforward method, and their small corresponding customer cones of 19, 11, and 232 ASes strongly indicate that these 3 networks are not provider-free. The above examples confirm that the customer-cone metric is more robust to the link misclassification than the simple inspection of the link types as with the straightforward method.

Among alternative topological parameters that we considered as a basis for the TC algorithm, the customer count of an AS is easier to compute than the customer cone and refers to the number of direct customers of the AS. A very large value of the customer count has some correlation with the provider-free status. However, the correlation is weaker than for the customer cone: even if a network does not belong to PFS due to being a direct customer of a provider-free AS, this network can have a very large number of own direct customers.

While the PFS members peer with each other, another potential approach to detecting PFS is to search for close-knit peering communities, e.g., to examine the number of peering links of each AS. However, our preliminary analyses for peering-based and other alternative parameters did not yield encouraging results. Consequently, the customer cone serves as the topological basis for our PFS detection algorithm.

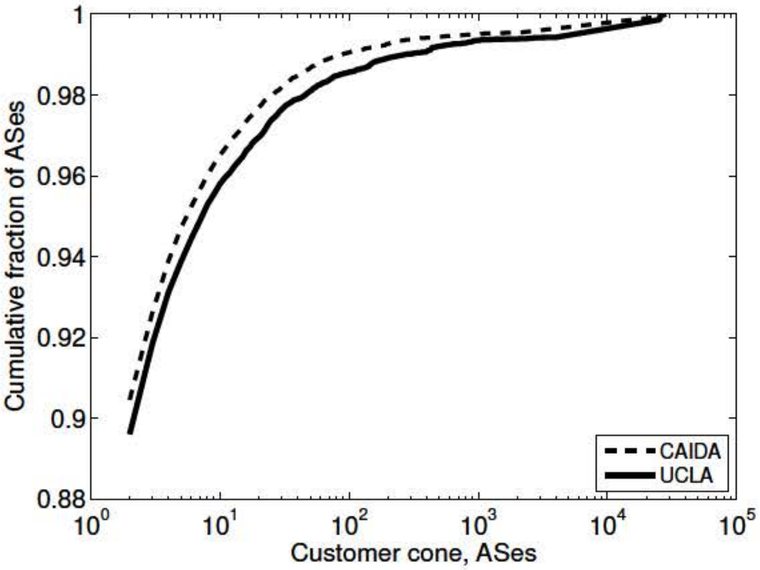


Figure 3.2: 6/2009 distributions of the AS customer cones.

Figure 3.2 plots the distributions of the AS customer cones in the UCLA and CAIDA datasets for 6/2009 and shows that only a tiny fraction of all ASes have a really large customer cone. Table 3.2 zooms in on the tail of the UCLA 6/2009 distribution. The tail covers set W_1 quite tightly: all 10 members of W_1 appear among the top 13 ASes ranked by the customer cone; this is an improvement over the straightforward method which includes only 9 members of W_1 into its 17-member PFS for 6/2009.

Rank	AS name (AS number)	Customer cone, ASes	In W_1?
1	Sprint (1239)	28478	✓(1)
2	Level3 (3356)	28168	✓(2)
3	NTT (2914)	27650	✓(3)
4	AT&T (7018)	27613	✓(4)
5	Global Crossing (3549)	27236	✓(5)
6	Verizon (701)	27121	✓(6)
7	Telia (1299)	26833	✓(7)
8	Qwest (209)	26764	✓(8)
9	Deutsche Telekom (3320)	26263	–
10	Ipercast (34763)	26127	–
11	Savvis (3561)	26082	✓(9)
12	GIT Telecom (38925)	26015	–
13	Tata (6453)	26014	✓(10)

Table 3.2: UCLA customer-cone ranks of ASes for 6/2009.

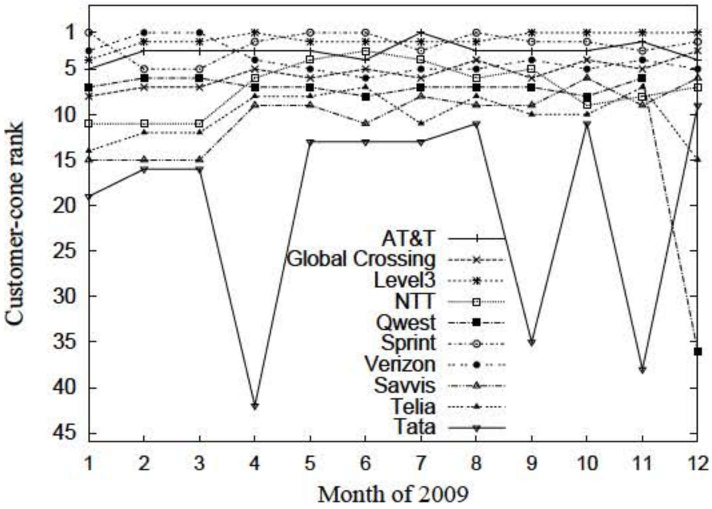


Figure 3.3: UCLA customer-cone ranks of the ASes in W_1.

Figure 3.3 tracks the UCLA customer-cone ranks of all ASes in W_1 throughout 2009. The ranks remain close to the top 10 with few exceptions such as three dramatic dips for Tata. While Figure 3.3 corroborates the promising potential of the customer-cone statistics for PFS detection, the results also suggest that our algorithm needs additional features for overcoming the noise in the datasets. Figure 3.4 depicts the CAIDA customer-cone ranks of all ASes in W_1 during 2009. In agreement with Table 3.1, the customer-cone results in Figures 3.3 and 3.4 imply that the UCLA datasets are less noisy and thus more suitable for PFS detection than the CAIDA datasets.

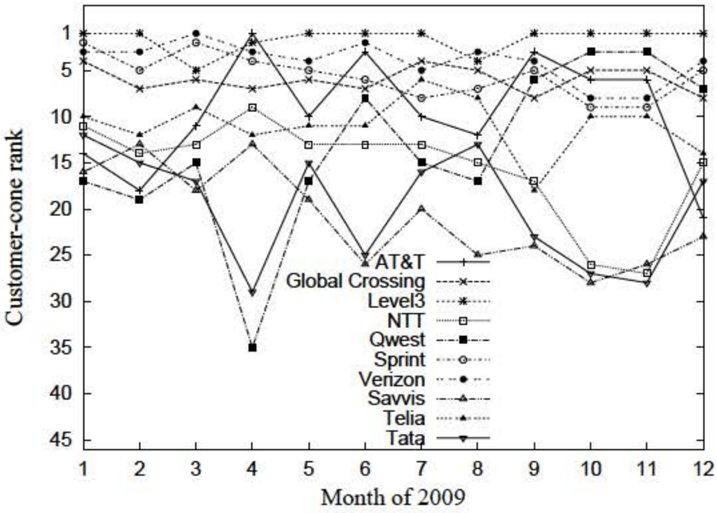


Figure 3.4: CAIDA customer-cone ranks of the ASes in W_1.

Figure 3.5 presents the 2009 customer-count ranks of all ASes in W_1 for the UCLA datasets. Compared to Figure 3.3 for the customer-cone ranks, Figure 3.5 shows that the customer-count ranks are less effective in capturing W_1.

3.4.2 PFS size

To detect PFS, the TC algorithm has to size this set. Whereas the Internet is growing, our hypothesis is that the set of provider-free ASes scales up proportionally with the overall population of Internet ASes. More specifically, we set size S_m of PFS at time m to:

$$S_m = \lfloor k \cdot P_m \rfloor \tag{3.1}$$

where P_m represents the total number of Internet ASes at time m, and k is a fixed factor.

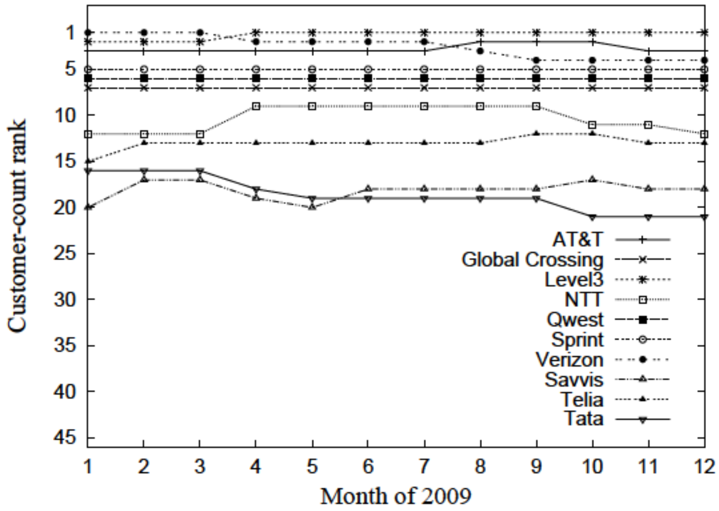


Figure 3.5: UCLA customer-count ranks of the ASes in W_1.

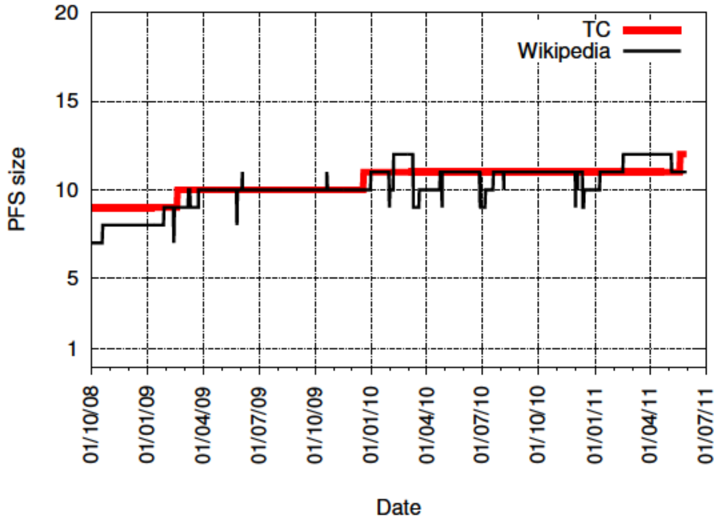


Figure 3.6: PFS size according to Wikipedia and TC algorithm for the UCLA datasets.

To validate the hypothesis and select the value of k, we explore how PFS evolved from 10/2008 to 5/2011 according to the Wikipedia perspective. During this time interval, the article has been revised on 113 days, and multiple revisions on a single day were common. Figure 3.6 depicts the PFS size according to Wikipedia, with short-lived spikes representing spurious revisions. For every day throughout the 32-month interval, Figure 3.6 also plots the PFS size as per Equation 3.1 with the value of k set to 0.00032, which corresponds to 1 in about 3000 Internet ASes being provider-free.

This equation-based prediction is aligned well with the PFS growth trend in the depicted Wikipedia data. Whereas the amount of the available data is too limited to recommend strongly the specific value of k or even to defend confidently the proportionality of PFS to the overall population of Internet ASes, the available data do suggest that Equation 3.1 offers a reasonable approximation for the PFS size.

3.4.3 Temporal dimension

With the PFS size selected, the algorithm still needs to identify the ASes of the set. We utilize the temporal dimension of the datasets to tackle the noise remaining in the customer-cone statistics. Our intuition is that the membership of an AS in PFS is relatively stable. While a new AS can join PFS and subsequently lose the provider-free status again, such transitions are infrequent, caused by rare mergers/acquisitions and guarded against by long-term business contracts. Therefore, to decide whether an AS is provider-free for month m, our algorithm looks w months back and ahead from month m and includes the AS into PFS for month m only if the AS belongs to the set according to the customer-cone ranks for at least n out of these $2w + 1$ months.

For an input with M months in the time series, our algorithm outputs PFS for each month except for the first w and last w months, i.e., the algorithm computes PFS for the $M - 2w$ middle months. While one-year contracts between ASes are common, we recommend $w = 6$ months and $n = 5$ months as default values for the w and n parameters of the algorithm, i.e., inclusion of an AS into PFS requires from the customer-cone ranks to endorse the AS for at least 5 out of 13 months. These settings enable our algorithm to recognize a genuine one-year PFS membership in spite of multiple months of erroneous disqualifications by the customer-cone ranks.

These settings also allow the algorithm to exclude a non-provider-free AS from PFS despite multiple months of mistaken customer-cone endorsements. In Section 3.5, we study sensitivity of the TC algorithm to the w and n parameters and show that $w = 6$ months and $n = 5$ months are reasonable settings.

We refer to the developed PFS detection algorithm as TC (Temporal Cone). Table 3.3 explains the notation used in the TC algorithm 1 in detail.

3.5 Evaluation

According to Sections 3.2 through 3.4, the datasets from UCLA are available for more months and less noisy than the CAIDA datasets. To evaluate the developed TC algorithm,

Notation	Semantics
m or i	month
M	number of months in the time series
C_m	list of the Internet ASes ordered by their customer-cone ranks for month m
L_m	ordered list of PFS candidates for month m
S_m	size of PFS for month m
w	lookback/lookahead window
F_m	PFS for month m
a	AS
b_a	counter of months when AS a belongs to PFS as per the customer-cone rankings
$r_{a,m}$	rank of a in L_m
n	PFS membership threshold

Table 3.3: Notation for our Temporal Cone (TC) algorithm as shown in Algorithm 1 (below)

Algorithm 1: TC (Temporal Cone) algorithm for PFS detection

```
for $m = 1, \ldots, M$ do
    compute $C_m$;
    $L_m \leftarrow C_m$;
    calculate $S_m$ according to Equation 3.1;
for $m = M - w, \ldots, w + 1$ do
    $F_m \leftarrow \emptyset$;
    $a \leftarrow$ first AS in $L_m$;
    while $|F_m| < S_m$ and $a \neq null$ do
        $b_a \leftarrow 0$;
        for $i = m - w, \ldots, m + w$ do
            if $r_{a,i} \leq S_i$ then
                $b_a \leftarrow b_a + 1$;
        if $b_a \geq n$ then
            $F_m \leftarrow F_m \cup \{a\}$
        else remove $a$ from $L_m$; $r_{a,m} \leftarrow \infty$;
        $a \leftarrow$ next AS in $L_m$
```

Section 3.5.1 relies on the UCLA datasets for the 32 months from 10/2008 to 5/2011 and – following the recommendations from the previous section – sets the PFS sizing factor, lookback/lookahead window, and PFS membership threshold to $k = 0.00032$, $w = 6$ months, and $n = 5$ months respectively. Then, Section 3.5.2 examines the sensitivity of the TC algorithm performance to the w and n parameters.

3.5.1 TC results

During its first iterative stage, the TC algorithm determines the AS customer-cone ranks and PFS sizes for all $M = 32$ months. Figure 3.7 plots the customer-cone ranks of the ASes in set W_2, i.e., all W_1 members and Tinet which became the 11th member of PFS according to Wikipedia after 2009. All 11 members of W_2 consistently appear among the top 11 ASes ranked by the customer cone in 2010 and 2011, indicating a higher accuracy of the more recent UCLA datasets.

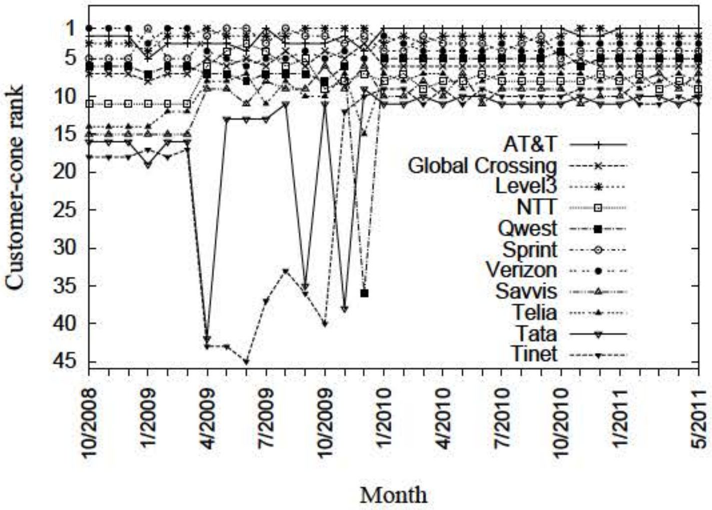


Figure 3.7: UCLA customer-cone ranks of the ASes in W_2 (i.e., the W_1 members and Tinet) from 10/2008 to 5/2011.

As shown in Figure 3.6, the TC algorithm sizes PFS to 9 ASes between 10/2008 and 1/2009, 10 ASes between 2/2009 and 12/2009, and 11 ASes from 1/2010 to 5/2011. This expansion is consistent with the PFS insights from the trustworthy but non-verifiable sources. With $w = 6$ months to look back and ahead, the TC algorithm executes its second stage to compute PFS for the $M - 2w = 20$ middle months from 4/2009 to 11/2010. Among the 9 months of 2009 (when the PFS size is 10 ASes), PFS perfectly matches W_1 for one month, omits only Qwest for another month, and excludes only Tata for the other 7 months. For all 11 months of 2010 (when the PFS size is equal to 11 ASes), PFS matches W_2 exactly.

Table 3.4 sums up the performance of the TC algorithm. A quick comparison of these results with Table 3.1 reveals that the TC algorithm detects PFS significantly better

Year	2009		2010
Month	4-8, 10-12	9	1-11
UCLA	10 (9)	10 (10)	11 (11)

Table 3.4: Size of PFS according to the TC algorithm for the UCLA datasets and (in parentheses) number of ASes in this PFS that match the Wikipedia insights (W_1 for 2009 and W_2 for 2010).

than the straightforward method. While the TC algorithm agrees with the Wikipedia perspective on the PFS size, the false positives of the algorithm are equal in number to its false negatives. Hence, we further quantify the performance of the TC algorithm with the following 2 metrics:

- *Accuracy* A_m of the PFS detection for month m is the fraction of ASes in the computed PFS that are provider-free during month m according to Wikipedia;
- *Average accuracy* of the PFS detection is the average of monthly accuracies A_m over all the $M - 2w$ middle months in the input time series.

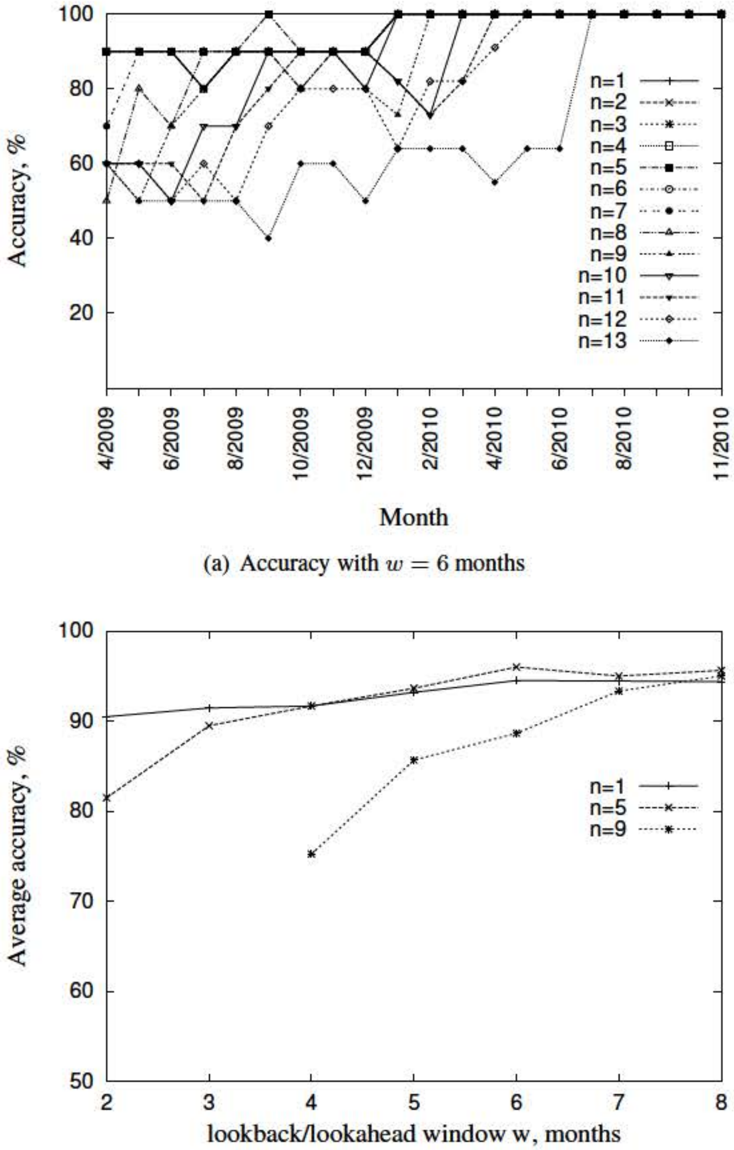
For the TC results in Table 3.4, the accuracy of the PFS detection is 90$\%$ for 8 months and perfect 100$\%$ for the other 12 months. Thus, the corresponding average accuracy of the PFS detection is 96$\%$.

3.5.2 Parameter sensitivity

Whereas our TC algorithm relies on parameters w and n, this section studies the sensitivity of the algorithm performance to these 2 parameters. We conduct such study for not only UCLA but also CAIDA. Throughout the study, we use $k = 0.00032$ as discussed in Section 3.4.2.

For the UCLA datasets, Figure 3.8a depicts the sensitivity of the TC algorithm accuracy to PFS membership threshold n with $w = 6$ lookback/lookahead months. Any of the examined n values delivers 100$\%$ accuracy for the last few months. For earlier months, the accuracy is lower and varies from one value of n to another. With $n = 5$ months, the accuracy is most stable and remains at least 90$\%$. With $n = 13$ months, the accuracy is only 40$\%$ for 9/2009. In general, the results indicate that values of n in the lower portion of its range are more attractive than values in the upper portion.

By averaging the accuracy over the individual months, Figure 3.8b exposes more clearly the trend revealed in Figure 3.8a. With $w = 6$ months, the average accuracy of the TC algorithm declines steadily and dramatically as PFS membership threshold n grows beyond 5 months. When n decreases from 5 months to 1 month, the average



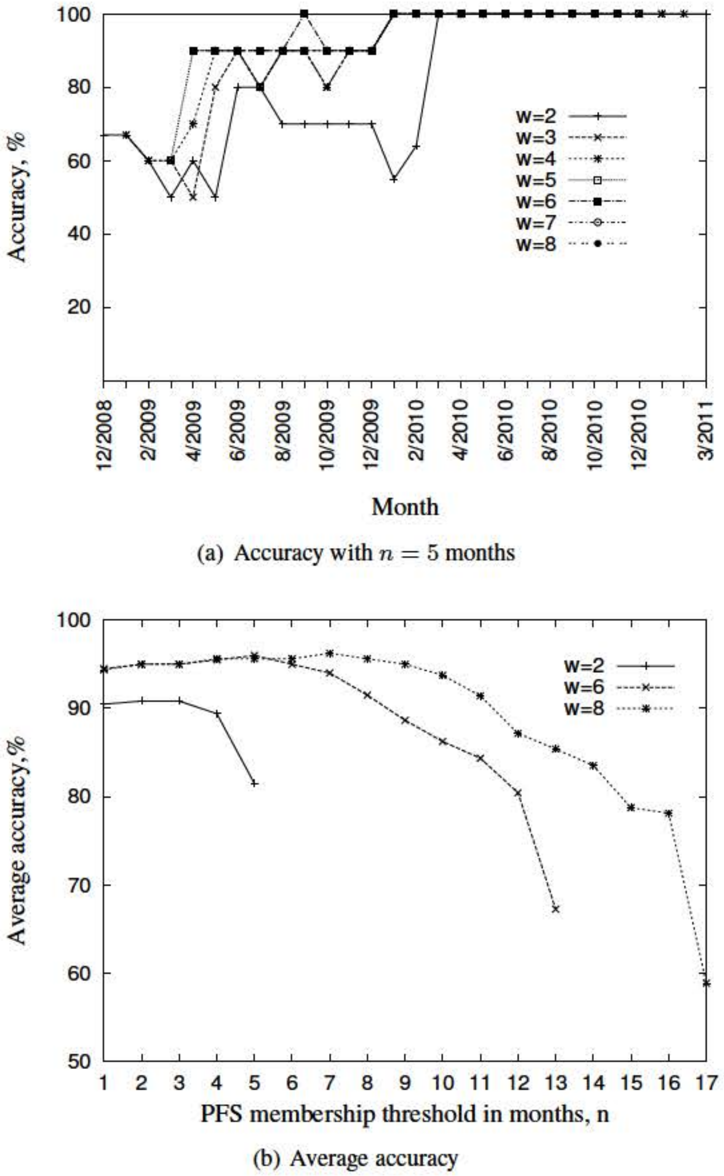
(a) Accuracy with $w = 6$ months

(b) Average accuracy

Figure 3.8: Sensitivity of the TC algorithm accuracy to PFS membership threshold n for the UCLA datasets.

accuracy declines slightly. Hence, for $w = 6$ months, the average accuracy attains its peak of 96% when n is set to 5 months.

Figure 3.8b also plots the average accuracy for $w = 2$ months and $w = 8$ months, with the profile of the accuracy sensitivity to n remaining qualitatively the same. The average accuracy is stable for smaller values of the PFS membership threshold but decreases consistently and significantly after n grows beyond a tipping point.



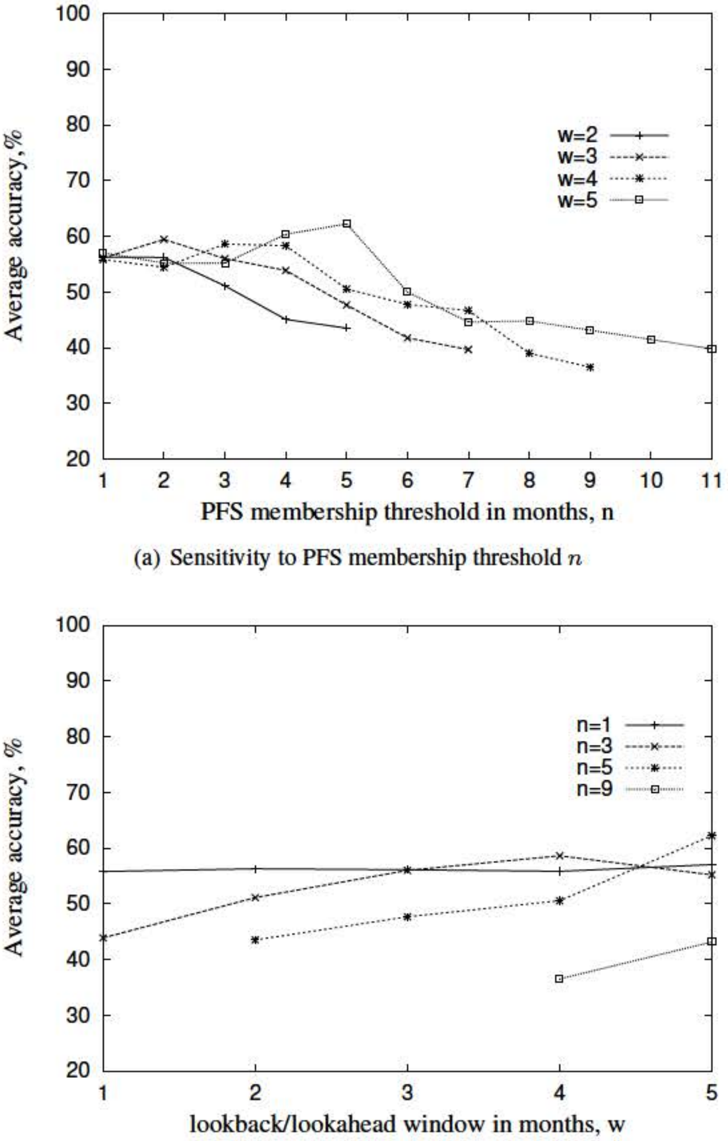
(a) Accuracy with $n = 5$ months

(b) Average accuracy

Figure 3.9: Sensitivity of the TC algorithm accuracy to lookback/lookahead window w for the UCLA datasets.

Figure 3.9 shows the sensitivity of the TC algorithm accuracy to lookback/lookahead window w for the UCLA datasets. For $n = 5$ months, Figure 3.9a presents the accuracy for individual months and suggests that larger values of w are generally beneficial. Figure 3.9b reveals this dependence more clearly. As w grows, the average accuracy increases first but then tends to flatten out. With $n = 5$ months, the average accuracy reaches the maximum of 96% when w is set to 6 months. $w = 7$ months and $w = 8$ months yield similarly high values of the average accuracy. Based on the above observations, we conclude that $w = 6$ months and $n = 5$ months constitute reasonable settings of the two parameters for the UCLA datasets.

For the CAIDA datasets, we conduct a similar sensitivity study and report the results in Figure 3.10. The study relies on data for the 16 months from 10/2008 to 1/2010. The shorter duration of the CAIDA time series reduces the meaningful value range for w to be up to 7 months. Figure 3.10a plots the average accuracy of the PFS detection as a function of n.



(a) Sensitivity to PFS membership threshold n

(b) Sensitivity to lookback/lookahead window w

Figure 3.10: Sensitivity of the TC algorithm accuracy to PFS membership threshold n and lookback/lookahead window w for the CAIDA datasets.

The dependence is qualitatively the same as with the UCLA datasets. When the the PFS membership threshold increases, the average accuracy remains rather stable first but then declines steadily and substantially after a tipping point. The average accuracy peaks at 62% with $w = 5$ months and $n = 5$ months. The qualitative profile for the sensitivity of the TC algorithm accuracy to lookback/lookahead window w is also similar to the pattern observed for UCLA. As w increases, the average accuracy improves first but then stays mostly stable. Note that the best CAIDA settings of $w = 5$ months and $n = 5$ months are close to the settings recommended above for the UCLA data source.

Although the sensitivity of the PFS detection accuracy to the w and n parameters has a qualitatively similar profile for the UCLA and CAIDA datasets, quantitatively the TC algorithm performs very differently with the 2 sources. In particular, the average accuracy peaks at 96% and 62% for UCLA and CAIDA respectively. The performance differences could be partly attributed to the differences in the UCLA and CAIDA inference methodologies. For example, UCLA might be yielding the more accurate results because of considering the RIPE RIS BGP measurements in addition to the Route Views BGP measurements. Whereas the CAIDA method accounts for node degrees, the UCLA methods disregards them to focus on valley-free routing. Our results suggest that PFS detection might benefit from disregarding the node degrees. Finally, while UCLA collects its data from BGP monitors located in provider-free autonomous systems as identified by Wikipedia, CAIDA gathers its data from a more diverse group of BGP monitors. The higher precision of the UCLA datasets might be due to utilizing, at least indirectly, the provider-free AS knowledge taken from the non-verifiable source.

3.6 Related work

The TC algorithm derives PFS from inter-AS economic relationships. Since the pioneering work by Gao [2], the problem of inter-AS relationship inference has attracted a variety of other heuristic solutions [8, 9, 44, 77–82]. While our work is the first to focus on detecting PFS, previous works used PFS as an input to their inter-AS relationship inference algorithms [8, 9]. PFS also served as a basis for studies of backbone networks and resilience of routing to failures [83–85].

Derivation of PFS from public inter-AS relationship datasets is challenging because missing or misclassified links make the datasets noisy. Addressing the problem of hidden links [86–90] has a potential for making the results of our TC algorithm even better.

While the TC algorithm exploits the temporal diversity of the inter-AS relationship datasets, prior works explored the temporal dimension for studying other problems such as network graph evolution [91,92].

In general, Internet AS-level graphs have been studied from numerous perspectives. For example, [93,94] studied the structure of the AS-level graphs using the k-dense and k-clique community detection algorithms. The work by Subramanian et al. [78] is the closest in spirit to ours. Among its other contributions, that paper proposed a new hierarchical taxonomy for Internet ASes and developed an algorithm that uses AS customer counts to detect the top-tier ASes of the newly proposed hierarchy. While similar in spirit, our work is very different in its specific goals and methods. In particular, we strive to detect PFS in accordance to the traditional definition of provider-free ASes.

3.7 Conclusion

PFS, or the set of provider-free ASes, is important for the Internet resilience and economics. Albeit the ground truth about PFS is not publicly available, there is a significant interest in knowing PFS. For example, the Wikipedia article on provider-free ASes has been viewed about half a million times during the previous three years. In this work, we sought to supplement the non-verifiable sources, such as the Wikipedia article, with scientific insights from public datasets of inferred inter-AS economic relationships. In particular, we developed the TC algorithm that sized PFS to a fraction of the overall AS population and determined the PFS members by means of AS customer-cone ranking and temporal dataset diversity. In comparison to the straightforward method for extracting PFS, our TC algorithm detected PFS with a substantially higher precision. We also assessed the sensitivity of the TC algorithm to its parameters. The derived TC algorithm is useful because it enables accurate inferences of PFS in the future even if PFS insights from the non-verifiable sources become unavailable.

Whereas the current insights from the non-verifiable sources appeared trustworthy and were corroborated through conversations with network operators, we used the Wikipedia insights to validate the accuracy of our TC algorithm. Although clearly imperfect, this validation method seemed the best option available currently for scientific studies of PFS. One could see our work as a middle point between traditional science and citizen science: our PFS detection method expanded the scope of knowledge but softened the benchmark for validation. Choosing such trade-off is not a novel feature of our methodology: even the discussed UCLA relationship inference method exhibits this property because of its reliance on the insights from Wikipedia. In spite of utilizing the non-verifiable information, this trade-off is useful for networking practice due to the scientific component that rises the knowledge above the state-of-the-art level of pure beliefs.

Chapter 4

Optimizing the Cost of Multilateral Interdomain Contracts

The previous chapters look at solutions for optimizing CDN cache deployments and algorithmic approach for detecting the provider-free ASes, respectively. Both works deal with AS-level Internet topologies derived from BGP routing based on bilateral contractual model for interconnections. In this chapter, we consider a multilateral contractual model for Internet interconnections that offers a potential to overcome limitations of the existing bilateral model.

The problem of trust originates from end-to-end traffic delivery that requires coordination among multiple ASes. By offering bilateral contracts between neighbouring ASes, the trust concern is resolved, however, at the cost of routing flexibility [15]. Access networks purchase upstream transit connectivity but have little control over the quality of the overall end-to-end path. For example, the access ASes and end users do not get end-to-end QoS or QoE guarantees since a transit AS in the path might not support the needed performance. Congested links, route instability and BGP oscillations also negatively affect end-to-end performance. For the same reason, many users and content providers utilize CDN services for efficient content delivery with performance guarantees. CDNs like Akamai use their intelligent overlay routing and load-balancing algorithms to deliver content from their servers to end users.

With BGP routing, ASes do not have any mechanism to either publicize or exchange performance information about unused capacity, thereby creating inefficiencies for the network that may have both excess supply and unrealised demand [95]. To overcome issues with bilateral contracts and BGP routing, multilateral contract models have been proposed where end users (or access ASes) can obtain end-to-end (or edge-to-edge) paths by forming contracts with multiple transit providers providing reachability to destinations. Access networks can purchase end-to-end paths that satisfy performance and cost considerations, and get more direct control over their end-to-end routing. A path is formed by composing path fragments (pathlets) leading from a source to a destination. Pathlets were first proposed in pathlet routing [96], a clean-slate routing architecture designed for flexibility.

Bilateral contracts might be negotiated for days or weeks, including discussions at network operators meetups, and are applied recursively: traffic that an AS sends to its neighbor is then controlled by the contracts of that neighbour. However, under a multilateral arrangement, there is a single contract between access and transit ASes for an end-to-end path. One such example is MINT, a connectivity market for exchange of end-to-end paths between independent networks [95].

The first attempts to consider such multilateral contract model have been reported recently [15–17]. A trusted centralized entity (web service, broker, cryptographic public ledger) provides an infrastructure and economic medium for the connectivity market between access and transit ASes. The payments from access ASes to transit ASes can be paid by this entity on behalf of the access ASes. Moreover, technologies like MPLS in transit ASes along with special routers for transport-layer upgrades, such as Serval [18] and PacketShader [19], show the possibility and viability of realizing the routing and forwarding of traffic under such multilateral arrangement.

In this work, we study economic aspects of access ASes and transit ASes operating in the multilateral arrangement. We leverage the background of auctions and machine learning for help in modeling. We sketch the use of exploration/exploitation trade-offs from learning literature in our work. For example, in order to maximize the short-term revenue, the transit AS should exploit its current prices and choose prices that access ASes are likely to accept. On the other hand, to maximize the long-term revenue, the transit AS needs to explore, i.e., identify which prices have the largest probability of being selected by access ASes. This kind of exploration necessitates choosing prices that currently have a low probability to be selected, which leads to opting for a low revenue in the short term.

First, we set out to find the optimal-cost assignment of access ASes to transit ASes for path segments in the end-to-end path. This models the access AS's objective of maximizing revenue for the preferred path. We investigate the following questions: How to simulate a realistic market model with access ASes and transit ASes? How much time does it would take for the algorithm to perform an optimal assignment with a considerable input size? Also, we look at pricing under uncertainty for transit ASes for their revenue maximization objective. We model it as an inter-temporal bandit decision problem and leverage *learning* tools such as exploration/exploitation trade-offs.

Main contributions of our work are as follows: First, we propose Bertsekas auction algorithm as the solution strategy for the optimal cost assignment of access ASes to transit ASes for path segments in the end-to-end Internet path. We implement the algorithm on bipartite graphs of different sizes to simulate the bidding and assignment phases. In particular, we observe the convergence time (in seconds) for the final assignment grows at an approximately exponential rate, with 0.46 seconds and 9.8 seconds for graphs with 200 x 200 and 1000 x 1000 ASes, respectively. We also observe that having a larger positive increment (ζ) value results in a smaller number of bids submitted by access ASes until their final assignment, thereby requiring a smaller number of reassignments. Besides, we develop and implement an epsilon-greedy bandit algorithm as the solution strategy for pricing of transit ASes under uncertainty and observe the learning ability of the algorithm.

4.1 Background

4.1.1 Auction algorithms

Consider a problem where n persons and n objects have to match on a one-to-one basis through a market mechanism. There is benefit a_{ij} for matching person i with object j, and the assignment of persons to objects is performed so as to maximize the total benefit of persons [97]. We want to find a one-to-one assignment, a set of person-object pairs $(1, j_1), \ldots, (n, j_n)$, such that objects $j_1, \ldots, j_n$ are all distinct, and the total benefit $\sum_{i=1}^{n} a_{i,j_i}$ is maximized.

From Bertsekas et al. [97], we adopt an auction algorithm described below. Each person as an *economic agent* acts in his own best interest. Suppose that object j has price p_j and that the person who receives the object must pay price p_j. Then, the net value of object j for person i is $a_{ij} - p_j$, and each person i would logically want to be assigned to object j_i with the maximal value, i.e., with

$$a_{ij_i} - p_{j_i} = \max_{j=1\ldots,n} \{a_{ij} - p_j\} \tag{4.1}$$

We say that person i is happy if this condition holds and that the assignment and set of prices are at an equilibrium when all persons are happy. The naive auction algorithm proceeds in rounds starting with any assignment and any set of prices. There is an assignment and a set of prices at the beginning of each round, and if all persons are happy with these, the process terminates. Otherwise some person who is not happy is selected.

This person i finds an object j_i which offers the maximal value, i.e.,

$$j_i \in \arg \max_{j=1\ldots,n} \{a_{ij} - p_j\} \tag{4.2}$$

Then, person i:

- Exchanges objects with the person assigned to j_i at the start of the round,
- Sets the price of the best object j_i to the value at which he is indifferent between j_i and the second best object, i.e., sets p_{j_i} to $p_{j_i} + \gamma_i$

The bidding increment is $\gamma_i = v_i - w_i$, where v_i is the best object value, and w_i is the second best object value, respectively.

$$v_{ij} = \max_{j} \{a_{ij} - p_j\} \tag{4.3}$$

$$w_{ij} = \max_{j \neq j_i} \{a_{ij} - p_j\} \tag{4.4}$$

γ_i is the largest increment by which the best object price p_{j_i} can be increased, with j_i still being the best object for person i. As in other auctions, bidding increments and price increases encourage competition by making the bidder's own preferred object less attractive to other potential bidders. At each round in the auction process, bidder i raises the price of his preferred object by bidding increment γ_i. Each bid for an object must raise its price by a minimum positive increment. Person i is almost happy with an assignment and set of prices if the value of its assigned object j_i is within ζ [1] of the maximal one, i.e.,

$$a_{ij_i} - p_{j_i} \geq \max_{j=1\ldots,n} \{a_{ij} - p_j\} - \zeta \tag{4.5}$$

By reformulation of the naive auction process so that the bidding increment is always at least equal to ζ. Thus, bidding increment γ_i is expressed as

$$\gamma_{ij} = v_{ij} - w_{ij} + \zeta \tag{4.6}$$

With this choice, the bidder of a round is almost happy at the end of the round. The auction process terminates in a finite number of rounds, necessarily with an assignment and set of prices that are almost at an equilibrium. Once an object receives a bid for the

[1] We employ ζ to represent the minimum positive increment instead of ϵ as in Bertsekas auction literature since we use ϵ to represent the probability in the bandit algorithm discussed in next subsection of this chapter

first time, the person assigned to the object at every subsequent round is almost happy. This is because a person is almost happy just after acquiring an object through a bid. Thus, the people who are not almost happy must be assigned to objects that have never received a bid. Specifically, once each object receives at least one bid, the algorithm must terminate. If an object receives a bid in m rounds, its price must exceed its initial price by at least $m\zeta$ [98]. Thus, for sufficiently large m, the object will become expensive enough to be judged inferior to some object that has not received a bid so far.

4.1.2 Bandit-based learning

A multi-armed bandit problem is a sequential decision making problem under uncertainty defined by a set of actions. At each time step, a unit resource is allocated to an action and some payoff is obtained. The goal is to maximize the total payoff obtained in a sequence of allocations [99]. In many real-world scenarios, decisions are taken to maximize some expected numerical reward. However, decisions or actions not only bring more reward but also can help to discover new knowledge or information that can be used to enhance future decisions [100]. In a casino, a sequential allocation problem is obtained when the gambler faces many slot machines at once (multi-armed bandit), and he must repeatedly choose where to insert the next coin. Each machine provides a random reward from a distribution specific to it. Initially, the gambler has no knowledge about the machines, but through repeated trials, he can focus on the most rewarding ones. The gambler plays iteratively one machine at each round and observes the associated reward. The gambler's objective is to maximize the sum of the rewards earned through a sequence of machine pulls over the considered time period.

Multi-armed bandit problems are considered an abstraction for decision problems incorporating an exploration-versus-exploitation trade-off. This trade-off can be seen clearly in the gambling scenario described above. Once the gambler has discovered a slot machine that has a fairly good average payoff, there is a pressure to choose between continuing to play this slot machine (exploitation) versus trying other alternatives that have never been tested or that have only been tested infrequently (exploration). Such a trade-off is ubiquitous in on-line decision problems, thus many applications of multi-armed bandit algorithms are found. Some of those [101] are:

- **Internet ad placement on websites and search engines:**
  A website owner buys advertising space, and each time a user visits the owner must choose to display one of n possible ads. The payoff of displaying an ad to a user is 1 if the user clicks on the advertisement, 0 otherwise. This is a multi-armed bandit problem where the levers are the set of ads which the site can display. For search

engines, a high revenue is achieved by displaying ads that have high bids along with high likelihood of being chosen by users. The objective of the search engines is to select ads that maximizes their total daily revenue [102].
The click-through rate (CTR) $c_{i,j}$ of ads $a_{i,j}$ for the target query phrase Q_j denotes the probability of a user to click on ad $a_{i,j}$ given that the ad was displayed to the user. The expected revenue $(c_{i,j} \cdot b_{i,j})$ is also a factor in the selection of ads. To maximize the short-term revenue, the search engine should exploit its current CTR estimates by displaying ads whose estimated CTRs are large. On the other front, in order to maximize the long-term revenue, the search engine needs to explore, i.e., identify which ads have the largest CTRs. This kind of exploration necessitates displaying ads that have current CTR estimated with a low confidence, which unavoidably leads to displaying some low-CTR ads in the short term.

- **Server selection in networks:**
  The process by which clients choose one server from a set of servers supplying a specified service. For example, a DNS resolver looks up a hostname in a given domain by selecting one DNS server from the list of authoritative name servers for the queried domain. The objective is to minimize the server's response time. There will be some servers located closer to the client and will have consistent faster response time. This is a multi-armed bandit problem where the set of authoritative name servers constitutes the levers, and the response time of each name server at time t constitutes the reward.

4.2 Problem overview

4.2.1 Optimal cost assignment of access ASes to transit ASes for path segments

We consider a multilateral contract environment where the access ASes obtain end-to-end paths by contracting with multiple independent transit providers that offer Internet path segments. The access ASes buy path segments from transit providers for the paths they are interested in. There is a market environment with the access ASes as buyers and transit ASes as sellers. There are individual path segments offered by transit providers for various destinations. For each path segment that is a part of those end-to-end paths, there exists competition among access ASes. Let A denote the set of access ASes and H denote the set of transit ASes. The individual path segments $p_1, \ldots, p_q$ provided by transit ASes constitute end-to-end path m, and there are many such paths indexed by m, where $m \in |\mathcal{M}|$ for a set of destinations M. Path segment p_i for a path m is offered by a set of transit ASes N, where $N \subseteq H$. The objective is to maximize the net benefit of access ASes.

A total of $|q|$ linear assignment problems need to be solved either sequentially or in parallel to find a preferred end-to-end path among those offered by the transit ASes. For our problem, we restrict the attention to the key subproblem of finding an optimal cost matching of access ASes for a path segment and not for the entire path. We also assume the symmetric case and do not consider the asymmetric problem where the number of transit ASes offering path segments are larger than the number of access ASes. Moreover, market mechanisms exist for tackling the asymmetric case, one such example is reverse auctions [103].

4.2.2 Dynamic pricing under uncertainty for transit ASes

The transit ASes offering path segments can learn what the most profitable prices are by price experimentations. They have to make a tradeoff between charging the most profitable price according to their current information, i.e., to exploit their information and inquiring on the profitability of the other prices that is to explore the profitability of the other prices. The pricing strategy under uncertainty is thus an inter-temporal decision problem and is modeled as a multi-armed bandit problem for transit ASes. Our bandit-based price optimization of a transit AS is modeled similar to a webstore pricing its objects [104]. The bandit design components of a machine such as levers, total rounds (T), horizon (H), exploration-exploitation trade-offs, and rewards are generic and relate to many optimization problems. Horizon H denotes the rounds remaining to be played. The levers represent the set of prices or cost functions. The product of the unit percentile-based, or volume-based, bandwidth price and routed user traffic constitute the *revenue*, or the reward function, of the provider.

4.3 Auction algorithm for optimal cost assignment of access to transit ASes

We evaluate Bertsekas auction algorithm [98] as a solution strategy for the assignment of access ASes to transit ASes offering individual path segments. The algorithm proceeds in iterations, and in each iteration there are two phases: bidding phase and assignment phase.

Bertsekas Auction algorithm:
Input: benefit matrix a_{ij} for bipartite graph $G(A \cup N, E)$ captures preferences and budget

set by access AS i for transit AS j, initial price p_j, minimum positive increment ζ.

Repeat the following phases, until a complete assignment with a perfect match is found.

- **Bidding Phase:** Each unassigned access AS $i \in A$ finds transit AS $j^* \in N$ that has the highest net profit v_{ij^*} as in equation 4.3, and the second best transit AS $j^{'}$ with its net profit $w_{ij^{'}}$ as in equation 4.4. AS i raises the price of preferred transit AS j^* by bidding increment γ_{ij^*} as in equation 4.6 and sends its bid b_{ij^*} to j^*:

$$\begin{aligned} b_{ij^*} &= p_{j^*} + \gamma_{ij^*} \\ &= p_{j^*} + v_{ij^*} - w_{ij^{'}} + \zeta \\ &= a_{ij^*} - w_{ij^{'}} + \zeta \end{aligned} \tag{4.7}$$

- **Assignment Phase:** Transit AS j that receives the highest bid from the AS i^* assigns itself to AS i^*, i.e., the connection (i^*, j) is added to the current assignment. Transit AS j also updates its own price to the received bid from access AS i^*, i.e., $p_j = b_{ij^*}$.

Output: Assignment of transit to access ASes $\mathcal{A}$, $p_j^{'}$ final prices of each transit AS j.

4.3.1 Experiment methodology and details on the input

We implement the auction algorithm in MATLAB (R2009a, 32-bit) on an Intel machine with the CPU of 2.00GHz Core2 Duo, RAM of 1GB, and Ubuntu OS. We run the algorithm on bipartite graphs of different sizes with associated benefit matrix inputs. As the first step, we consider a toy example with a small bipartite graph connecting 4 ASes. Afterwards, we apply the algorithm on graph instances with 100, 200, 400, 600, 800 and 1000 ASes for a sensitivity analysis.

We generate benefit matrix a_{ij} based on uniformly distributed random values for each of the graphs. We consider $p_j = 1$ as the initial price of a transit AS j and set minimum positive increment ζ using ζ-scaling [105]. Both ζ and maximum absolute benefit $\mathbb{C}$, i.e., $max(abs(a_{ij}(:)))$, decide how much work is needed before the auction algorithm gets to terminate. By means of ζ-scaling, the algorithm is applied several times, starting with a large value of ζ and successively reducing ζ up to an ultimate value. Typical ζ-reduction factors after each scaling phase are of the order of 4 to 10 [105].

Based on these observations, we set the ζ initial value as $\mathbb{C}/20$ and ζ decrease factor as 0.2. We run the auction algorithm iteratively for the top 4 values of ζ (in their decreasing

Bipartite graph: access & transit ASes	Benefit matrix value		Minimum positive increment, ζ			
AS size	maximum	minimum	ζ_1	ζ_2	ζ_3	ζ_4
100	26613	206	1330	266	53	10
200	47792	346	2389	477	95	19
400	103382	238	5169	1033	206	41
600	159890	251	7994	1598	319	63
800	222191	305	11109	2221	444	88
1000	276559	92	13827	2765	553	110

Table 4.1: Details on the benefit matrix and ζ values used in the experiment with varying graph size.

order). We provide details in Table 4.1 on the maximum and minimum value (rounded to nearest integer) of the benefit matrix for each graph size and also on the ζ values.

4.3.2 Experiment results

A. Toy example

By running the auction algorithm with the given inputs: a_{ij} of size (4 x 4), $\zeta = 110$, we simulate the bidding and assignment processes and present the final assignment results in Table 4.2.

$$a_{ij} = \begin{bmatrix} 1000 & 1150 & 1800 & 2250 \\ 1400 & 1450 & 1250 & 2550 \\ 1050 & 1700 & 1850 & 2400 \\ 1550 & 1700 & 2150 & 2750 \end{bmatrix}$$

Bidding Phase			Assignment Phase
Access AS i	Transit AS j	Bid value b_ij	Assignments (i,j) pair; Prices p_j
1 2 3 4	4 4 4 4	561 1211 661 711	**(2,4); 1211**
1 3 4	3 3 3	761 261 561	**(1,3); 761**
3 4	2 2	621 261	**(3,2); 621**
4	1	121	**(4,1); 121**

Table 4.2: Bertsekas auction for the assignment of access AS i to transit AS j, with given a_{ij}, $\zeta = 110$, initial price $p_j = 1$, completed in four phases (bidding and assignment)

We observe that the assignment with auctions was completed in four iterations. In the first iteration, all the four access ASes bid for transit AS 4's path segment, and the access AS 2 obtains its assignment by winning the bidding competition. In subsequent iterations, assignments continue to be based on the winning bidder, and there are no reassignments.

B. Convergence time with varying graph sizes

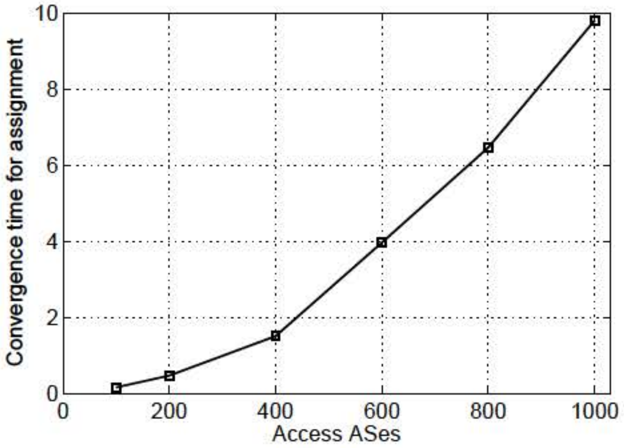


Figure 4.1: Average convergence time (in seconds) for the auction assignment with varying graph sizes.

We obtain the convergence time (in seconds) for the final assignment with the auction algorithm. We run the algorithm iteratively for the four values of ζ and repeat the experiment 10 times. The convergence time is measured as the average over the ζ values in all experimental runs. Figure 4.1 depicts that the convergence time grows at a nearly exponential rate with the increase in the AS size. For instance, for a graph of size 200 x 200 ASes, the convergence time is 0.46 seconds; however, its value grows to 9.8 seconds for a graph of size 1000 x 1000 ASes.

C. Difference in total benefit between auction and random assignments

We have the final assignment for the auction algorithm that maps access ASes to their optimal transit ASes. We compute the total assignment benefit by summing up the assignment benefit values of individual mapped pairs. We compute this value for different graph sizes. Next, we apply a random assignment method on the same input benefit matrix to compute the random assignment and then obtain its associated total benefit.

We employ MATLAB's randperm function to first obtain a random permutation of

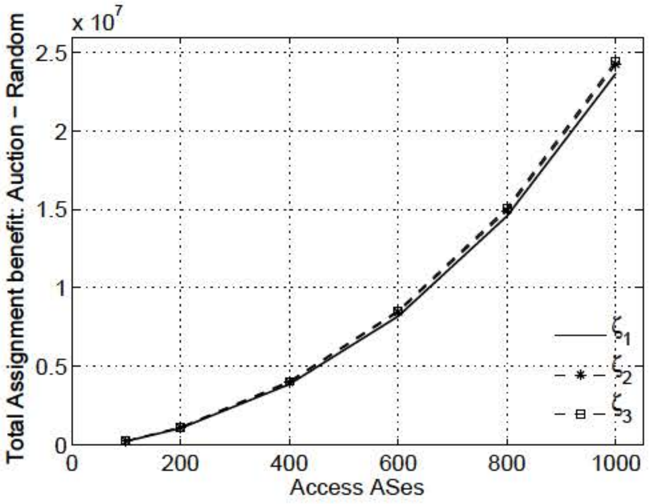


Figure 4.2: Total assignment benefit difference with Bertsekas auction and random assignment.

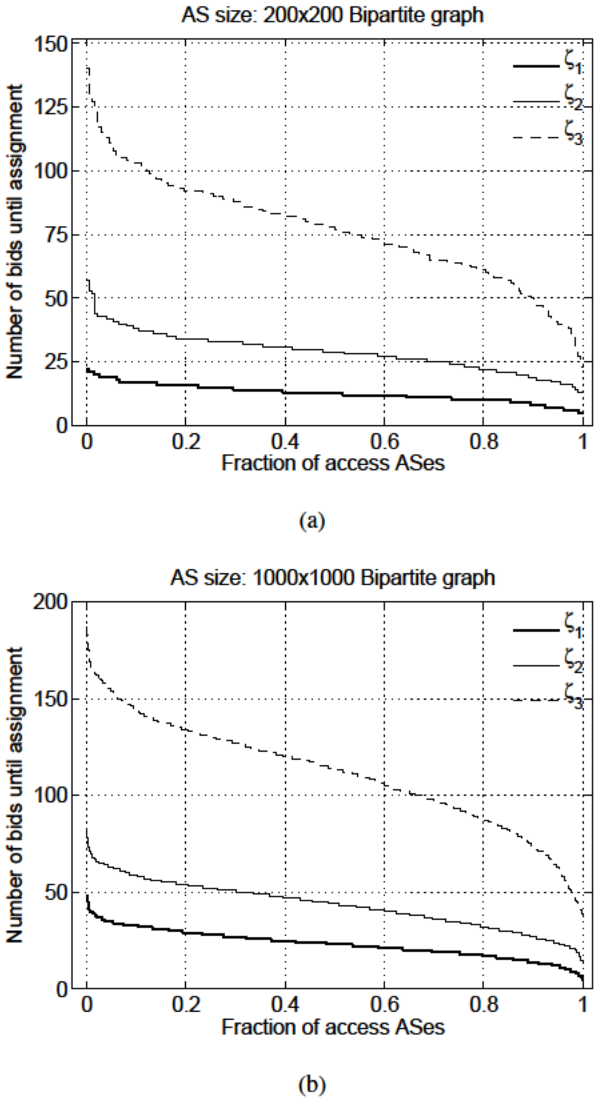
the entire benefit matrix and then perform a random assignment to map the access ASes to transit ASes. We compute the difference in the total assignment benefits with respect to the auction algorithm and the random method. Figure 4.2 plots the total assignment difference to the access ASes, for three decreasing values of ζ: ζ_1, ζ_2, ζ_3. We observe the benefit difference to grow exponentially with increase in AS graph size and it stays relatively similar with ζ values.

D. Access ASes and total bids for final assignment

We compute the number of bids submitted by each access AS until a final assignment by the auction algorithm, where each AS is said to be finally happy. We plot the distribution of access ASes versus total bids submitted until their assignment. The results are for bipartite graphs of two different sizes, 200 x 200 and 1000 x 1000 ASes respectively, and with varying ζ values that depend on the AS size.

For ζ_3, the smallest increment, with the graph of size 200 x 200, Figure 4.3(a) shows that the individual access ASes submit between 140 and 23 bids each until the final assignment is done. With the higher values ζ_2 and ζ_1, the maximum number of bids submitted by the same access AS reduces to 57 and 22 respectively, while the minimum number of bids equals 13 and 5 respectively.

Figure 4.3(b) represents the distribution results for a graph of size 1000 x 1000 ASes. With ζ_3, the plot reveals that the individual access ASes submit between 187 and 36 bids each. Following the same pattern as before, with ζ_2 and ζ_1, the maximum number of bids



(a)

(b)

Figure 4.3: Mapping of submitted bids by access ASes in Bertsekas auction with varying ζ values.

submitted by the same access AS reduces to 83 and 48 respectively, and the minimum number of bids equals 13 and 5 respectively. Thus, having a larger increment results in a smaller number of bids and reassignments.

4.4 Bandit-based approach for dynamic pricing of transit ASes

We apply the multi-armed bandit algorithm [100] as a solution strategy for the individual transit ASes to maximize their revenue. The iterative epsilon-greedy algorithm is the commonly used bandit algorithm. Its goal is to minimize the regret for the revenue.

This is possible by finding optimal prices more often during the successive iterations. The number of total rounds T represents the period during which the transit AS as a seller has offered its services to an access AS for a particular price.

We consider m prices. Price a is the chosen one with r_a being its associated mean price reward. We present the epsilon-greedy algorithm in Algorithm 2 below. The goal is to maximize the sum of the price rewards of the chosen prices over all the rounds.

Algorithm 2: Epsilon-greedy algorithm at a transit AS for revenue maximization

```
Input  : $m$; $\epsilon$; $T$; $r_m$; $\mu^*$
Output: $a$; $r_a$; $n_a$; $\varphi$
for every iteration $t$ do
    Exploration: With probability $\epsilon$, select price $a$ uniformly at random over the space of
      $m$ different prices
    Exploitation: Else, with probability (1-$\epsilon$), select price $a$ that has the highest average
      price reward estimate $R'_a$ among other prices over previous iterations. Then, map the
      chosen price $a$ to its mean reward value $r_a$
Regret after T rounds $\varphi$ = $T$ $\cdot\mu^*$ - $\sum_{t=1}^{T}$r$_a$ [Difference between optimal average price and
  the sum of mean reward values over all rounds]
```

The exploitation phase based on the knowledge already acquired is the greedy choice. The exploration phase chooses a new price with a uniform probability over the set of prices. The probability ϵ is a tuning parameter. ϵ = 0.1, means that for 10% of the rounds, a random price is to be chosen. The optimal average price reward is given by $\mu^* = max_m\{r_m\}$, and we denote the number of times each price a is chosen by n_a. Horizon H is an exploration control parameter, e.g., for H = 1, the optimal strategy is reduced to pure exploitation, that is to choosing the price with the highest revenue.

It is to be noted that in the epsilon-greedy algorithm and its variants, the amount of exploration is fixed a priori. Though, it is possible to have pure exploration first for $\epsilon \cdot T$ rounds, followed by exploitation phase for $(1-\epsilon)\cdot T$ rounds, which happens in the ϵ-first strategy.

4.4.1 Evaluating the potential of the bandit algorithm

We evaluate and implement the epsilon-greedy bandit algorithm described in Algorithm 2. The experiment settings are: m=500; T=1000 (total rounds); S=1000 (experiment repetitions); ϵ = 0.1 and 0.4; uniformly distributed random values r_m from interval (0,1). We derive reward payoffs uniquely for price a selected at different iteration t. If price a is selected for a total of n_a times, we generate expected reward values r_a^t each time by drawing from the normal distribution as suggested in [106], with the mean equal to

r_a and standard deviation (σ) parameter. This leads to improving the bandit algorithm's performance with hundreds of prices.

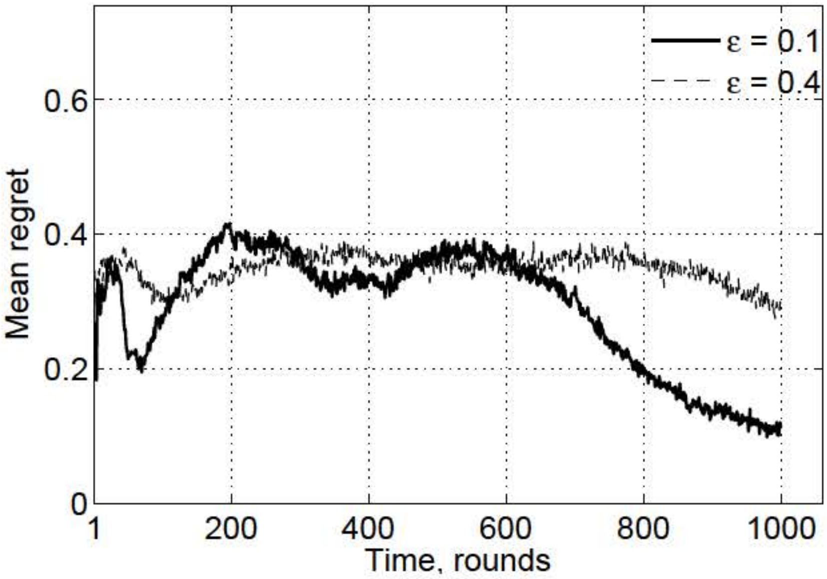


Figure 4.4: Mean regret over time with experiment settings: M=500, T=1000, and S=1000, and uniformly distributed random prices.

We kept $\sigma = 0.3$ in order to generate values close to the input mean reward (r_a). Then, the expected average price reward R_a of choosing price a is equal to the fraction of the sum of the payoffs r_a^t to the number of times n_a it has been chosen. We present the results of our experiment in Figure 4.4 above, where we observe the mean regret at each round by averaging over experimental run S. It is evident that the mean regret stays close for all epsilon values from rounds 250 to 650. The explorative behavior intensifies with $\epsilon = 0.4$, yielding higher regret from rounds 700 to 1000, in comparison to the lowest value of 0.1. The key insight is with the lower epsilon value and higher T and S values, we can achieve lower regrets even with random pricing data. For instance with $\epsilon = 0.1$, the overall mean regret is 0.292. The mean regret signifies how far the chosen prices are from the optimal average price and exhibits exploration/exploitation trade-off. By devising an appropriate price reward estimate mechanism, there is a possibility of decreasing the regret further.

4.5 Related work

Path Brokering, Multilateral Contract Negotiation, novel Source Routing:
Currently, there are bilateral economic contracts between ASes for end-to-end Internet routing and connectivity. To provide better performance and economic benefits, multilateral contract formation where access ASes contract multiple transit providers for end-to-end routing has been researched. Path brokering provides interconnection facilities for such a multilateral contract. MINT [95] shows how path brokers can stitch paths by means of end-to-end bandwidth transit markets. Lane et al. [17] propose the use of MPLS to support end-host path selection via path brokering as a novel path-query and billing method, where paths are the unit of commerce instead of connectivity. Castro et al. [15] leverage path brokering with pathlet routing [96] to develop a multilateral contract negotiation. They utilize a public cryptographic legder, a trustful entity to serve as an authenticated path broker between access networks and transit providers.

BGP Path-vector Contract Routing:
In this work, the end-to-end contract paths are computed in an on-demand manner. Each ISP advertises its contract links with fields embedding prices and performance levels to its neighbors. They expect contracts on the timescale of minutes. Yuksel et al. [107] propose a Contract-switched Inter-network with dynamic contracting over multiple providers for enabling flexible and economically efficient management of value flows and risks. Their architecture provides end-to-end capacity contracts for ASes over long time-scales in a fully decentralized manner and switches to BGP path-vector contract routing for shorter time-scales.

Our research differs from the above two research lines by proposing cost optimization of access and transit ASes participating in a multilateral contract arrangement. It is an interesting problem given the possibility of access networks contracting multiple providers, and thereby complimenting BGP routing. Such a setting with selfish entities offers many opportunities for exploring various trade-offs, interaction and optimization objectives. Therefore, our work focuses on developing algorithmic models in an environment that is dynamic since cost optimization is continuously re-evaluated. In previous works, there was not much focus on dynamic settings. We leverage bandit learning algorithms as a solution strategy for pricing transit ASes. [16] also uses a bandit algorithm for transit pricing, but with learning theory as their motivation and obtain the cumulative regret by considering edges of random and Erdos-Renyi graphs as levers for a capacity or price distribution. On the other hand, we study and use the most common bandit algorithm after finding its various practical applications, from online ad placement optimization to online shortest path problem [24].

4.6 Conclusions and future work

In this chapter, we put a focus on the interaction between transit and access ASes, and optimization under the multilateral interdomain contract formation setting. First, we considered the problem of access ASes buying path segments from transit ASes for an end-to-end path they are interested in. We modeled it as a linear assignment problem where there are access ASes as buyers, transit ASes as sellers, and our objective was to maximize the net benefit of access ASes. We employed Bertsekas auction [98] as the solution strategy for solving a sequence of linear assignment problems to obtain a preferred end-to-end path offered by transit ASes. We implemented the auction algorithm and simulated the bidding and assignment phases for a path segment with a bipartite graph composed of two sets of ASes, namely, transit and access ASes. We considered graphs of different sizes and performed a sensitivity analysis on the auction algorithm results. We observed the convergence time (in seconds) for the final assignment to grow at a nearly exponential rate, with 0.46 seconds and 9.8 seconds for the graphs with 200 x 200 and 1000 x 1000 ASes respectively. We computed the number of bids submitted by each access AS until their final assignment. We found that having a larger increment value (ζ) results in a smaller number of bids and reassignments.

Next, we modeled the pricing of transit ASes under uncertainty as a multi-armed bandit problem after finding motivating examples of bandit formulation for various practical applications: online ad placement optimization and online webstore pricing. We utilized the most commonly used epsilon-greedy bandit algorithm as the solution strategy and implemented it for evaluation with numerical pricing data. In particular, we showed the algorithm's ability to learn via exploration-exploitation trade-offs by plotting the variability of the mean regret over time for different values of ϵ.

Looking forward, we are interested in evaluating existing multi-armed bandit algorithms for end-to-end path selection, such as for on-line shortest paths [24] in the context of multilateral contract arrangements. Additionally, we plan to simulate the bandit algorithm at each transit AS of an end-to-end path for price optimization, since in this work we presented the working of the epsilon-greedy algorithm for only a single transit AS. However, such a simulation with multiple transit ASes as agents is challenging and requires developing an appropriate reward function. Adaptive learning with multiple agents has been successfully studied for efficient multi-robot motion towards a destination [108], and we find it as a motivating example. Finally, application of multilateral transport games [109] and multilateral negotiations [110] for studying economic aspects of the multilateral contractual framework constitutes an interesting direction for future work.

Chapter 5

Summary and Future Work

The AS-level Internet ecosystem consists of thousands of ASes interconnected on the interdomain level and belonging to different types such as transit, access, content-provider ASes. BGP routing implements the business relationships between ASes for global end-to-end reachability, and thus the interconnected network of ASes forms the underlay. The Internet ecosystem also involves CDNs that provide faster content delivery to end users in access networks by constructing an overlay network. The routing paths in the Internet underlay and overlay are very different, and CDNs identify and use shorter routes for improved performance. CDNs utilize hundreds of ASes for deploying caches in order to meet their content delivery objectives. Furthermore, CDN operators learn from topology, routing, and performance data to continuously optimize their design and deployments.

In this thesis, we first looked into the CDN cache deployment optimization (CaDeOp) problem that determines the optimal set of cache ASes and how much energy, bandwidth, and server resources to provision in each cache AS while satisfying the performance constraints. By leveraging realistic Internet topologies, traffic demands distribution, AS path lengths, we modeled and evaluated the deployment for various settings. For example, we assessed the trade-offs for performance vs. cost, deployment footprint vs. cost, cache ASes to served ASes mapping, and also observed the geographical footprint of cache ASes.

While, we utilized knowledge of the Internet core topology for CDN cache deployment, Chapter 3 shifted the focus on topology inference, and in particular on detection of provider-free ASes (PFS). We used public inter-AS relationship data and proposed the customer-cone topological metric. We developed a temporal cone (TC) algorithm and used Wikipedia as a validation source. Moreover, the very existence of provider-free ASes is because of current bilateral contractual model that is recursively applied in the

Internet hierarchy for end-to-end reachability. The novel customer cone topological metric which we have proposed for our TC algorithm has found application in a later work recently [111]. Due to performance limitations with BGP routing from bilateral contracts, there is a need to think on high-performance and economically viable contractual models.

Finally, we emphasized on multilateral contractual modeling that compliments existing bilateral contracts for BGP routing, by looking into the key aspects of economics. In particular, we proposed algorithms to optimize cost objectives for transit and access ASes under multilateral arrangement. First, we implemented and evaluated Bertsekas auction algorithm for optimal cost assignment of access ASes to transit ASes for path segments of an end-to-end path. Next, we implemented the epsilon-greedy bandit algorithm for optimizing the price of transit ASes and evaluated it with numerical pricing to show its learning potential through exploration-exploitation trade-offs.

The work in this thesis has explored research questions which are of interest to researchers, network operators and architects working in the area of Internet economics, Internet AS-level topology inference and evolution. The cache deployment optimization (CaDeOp) work has shown interesting results that are useful to CDN operators and architects to develop new models and templates for their evolving CDN deployments. With our work on PFS and the results from its evaluation highlighted the importance of characterizing Internet ASes with respect to economics, topological centrality, and their critical role for the future. The results from our multilateral contract work exhibited the potential of bandit and auction algorithms for price optimization of transit and access ASes respectively. We present the summary of the main contributions from each part of this thesis below:

- **Trade-offs in Optimizing the Cache Deployments of CDNs:** We studied the trade-offs in optimizing the cache deployment of CDN caches in the Internet core ASes [26]. We found that when the end-user performance requirements become more stringent, the CDN footprint expands rapidly, requiring cache deployments in additional ASes and geographical regions. With higher performance requirements, the CDN cost also rises by several times. While the server costs remain about the same, the costs of energy and bandwidth grow because the CDN loses some of the economies of scale in procuring these resources. We also found that the traffic distribution among the cache ASes stays relatively even, with the top 20% of the cache ASes serving around 30% of the overall CDN traffic. It is notable that the Pareto principle, which applies in many related domains, does not apply to CDN deployments, in part due to the highly distributed nature of the Internet traffic. We also explored the incremental cache deployment optimization (InCaDeOp) problem since it is related to upgrading cache deployments.

The results of our study can be applied in various contexts. First, it is of significant practical relevance since it formalizes the planning process that all real-life CDN operators must follow to reduce the operational cost of their overlay networks, while meeting the performance requirements of their end users. Second, our modeling efforts can help studies that look into CDN economics and CDN deployments evolution.

- **Obscure Giants: Detecting the Provider-Free ASes:** We explored PFS (provider-free ASes) and developed a algorithm to detect them from public AS-relationship datasets [27]. The developed TC (temporal cone) algorithm employs the customer-cone topological metric and exploits the temporal diversity of the datasets to infer PFS with a significantly higher accuracy. While the knowledge of PFS is highly valuable, validation of PFS inference results constitutes a major challenge because the ground truth lies outside the public domain. To tackle the validation challenge, we utilized trustworthy but non-verifiable sources such as Wikipedia. Whereas it seems practically impossible to obtain the complete ground truth from network operators, the non-verifiable source insights form the best available baseline for result validation in this important domain.

- **Optimizing Cost of Multilateral Interdomain Contracts:** We proposed algorithms for solving the multilateral-contract problem for transit and access ASes. The solution approach is two-fold. First, we adopted an auction algorithm with the objective to find optimal cost assignment of access ASes to transit ASes for path segments of an end-to-end Internet path. Our evaluation results with the auction algorithm showed the convergence time for assignments (in seconds) and how it grows exponentially with increase in the AS size. We also observed that the larger value of increment ζ results in a smaller number of bids and reassignments by access ASes. Second, we proposed an epsilon-greedy bandit algorithm as the solution strategy to optimize the pricing of transit ASes. Our evaluation results with the bandit algorithm and numerical pricing showed how the algorithm adapts in choosing high-rewarding prices in comparison to the optimal one by means of the regret metric. In particular, we presented the mean regret variability with time to show the potential of bandit learning for transit AS price optimization. The results from this study is useful to economists and optimization experts working with network operators. Moreover, it is also helpful to studies that attempt to propose novel interconnections and contractual frameworks among ASes in the evolving Internet.

5.0.1 Future work

This thesis presented few steps towards understanding the important role of topology and contracts in Internet content delivery. We identified a couple of directions for future work as natural extensions of the work in this thesis.

The natural extension is to study the incremental CaDeOp (InCaDeOp) that examines upgrading cache deployments given an initial deployment. We have already developed the InCaDeOp model and strive to evaluate it in future with realistic inputs. Exploring energy cost optimization for Internet-scale CDNs is of much significance for the future. Evaluation of CaDeOp with Internet topologies from the last 10 years and making appropriate changes to other inputs show promise in unraveling interesting patterns of cache deployments across time. These studies will shed more light on the significance of topology in CDN economics and deployment evolution.

With vast amounts of topology and routing data available, future work will strive to develop novel topological metrics and algorithms to study Internet evolution by identifying tangible economic implications of PFS. Furthermore, the issue of how the changing landscape of the flattening Internet topology affects overall Internet resilience is also interesting. We will utilize the bandit algorithm for online learning of shortest paths [24] to our problem of optimal cost assignment of access ASes to transit ASes for preferred end-to-end Internet paths under dynamic conditions of price and performance. Applications of multilateral transport games [109] and multilateral negotiations [110] for studying economic aspects of the multilateral contractual framework will be an interesting work for the future.

References

[1] Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” RFC 1771, 1995.

[2] L. Gao, “On Inferring Autonomous System Relationships in the Internet,” *IEEE/ACM Transactions on Networking*, 2001.

[3] J. Postel, “Internet Protocol DARPA Internet Program Protocol Specification,” RFC 791, 1981.

[4] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, “Towards an AS-to-Organization Map,” in *Proceedings of ACM IMC*, 2010.

[5] A. Dhamdhere and C. Dovrolis, “The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh,” in *Proceedings of ACM CoNEXT*, 2010.

[6] I. Castro, J. C. Cardona, S. Gorinsky, and P. Francois, “Remote Peering: More Peering without Internet Flattening,” in *Proceedings of ACM CoNEXT*, 2014.

[7] R. T. B. Ma, D. M. Chiu, J. C. S. Lui, V. Misra, and D. Rubenstein, “On Cooperative Settlement Between Content, Transit and Eyeball Internet Service Providers,” in *Proceedings of ACM CoNEXT*, 2008.

[8] E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani, “BGP and Inter-AS Economic Relationships,” in *Proceedings of IFIP Networking 2011*.

[9] J. Xia and L. Gao, “On the Evaluation of AS Relationship Inferences,” in *Proceedings of IEEE Globecom, 2004*.

[10] P. Bangera and S. Gorinsky, “Economics of Traffic Attraction by Transit providers,” in *Proceedings of IFIP Networking*, 2014.

[11] P. Bangera and S. Gorinsky, “Impact of Prefix Hijacking on Payments of Providers,” in *Proceedings of COMSNETS*, 2011.

[12] A. Dhamdhere, C. Dovrolis, and P. Francois, "A Value-based Framework for Internet Peering Agreements," in *Proceedings of ITC*, 2010.

[13] D. Clark, P. Faratin, P. Gilmore, S. Bauer, A. Berger, and W. Lehr, "The Growing Complexity of Internet Interconnection," *Communications Strategies*, 2008.

[14] A. Lutu, M. Bagnulo, and O. Maennel, "The BGP Visibility Scanner," in *Proceedings of IEEE INFOCOM Workshops*, 2013.

[15] I. Castro, A. Panda, B. Raghavan, S. Shenker, and S. Gorinsky, "Route Bazaar: Automatic Interdomain Contract Negotiation," in *Proceedings of USENIX HOTOS*, 2015.

[16] H. Esquivel, C. Muthukrishnan, F. Niu, S. Chawla, and A. Akella, "RouteBazaar: An Economic Framework for Flexible Routing," 2009, Technical Report 1654, Computer Sciences Department, University of Wisconsin, Madison.

[17] J. R. Lane and A. Nakao, "Path Brokering for End-host Path Selection: Toward a Path-centric Billing Method for a Multipath Internet," in *Proceedings of ACM CoNEXT*, 2008.

[18] E. Nordström, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Y. Ko, J. Rexford, and M. J. Freedman, "Serval: An End-host Stack for Service-centric Networking," in *Proceedings of USENIX NSDI*, 2012.

[19] S. Han, K. Jang, K. Park, and S. Moon, "PacketShader: A GPU-accelerated Software Router," *SIGCOMM Computer Communications Review*, 2010.

[20] S. E. A. Steven L. Scott, PhD, "Multi-armed Bandit Content experiments for Google Analytics," https://support.google.com/analytics/answer/2844870?hl=en.

[21] P. Krishnan, D. Raz, and Y. Shavitt, "The Cache Location Problem," *IEEE/ACM Transactions on Networking*, 2000.

[22] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks," in *Proceedings of IEEE INFOCOM*, 2010.

[23] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal Content Placement for a Large-scale VoD system," in *Proceedings of ACM CoNEXT*, 2010.

[24] A. Gyorgy, T. Linder, G. Lugosi, and G. Ottucsak, "The On-Line Shortest Path Problem Under Partial Monitoring," *Journal of Machine Learning Research*, 2007.

[25] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. K. Sitaraman, and B. Weihl, “Globally Distributed Content Delivery,” *IEEE Internet Computing*, 2002.

[26] S. Hasan, S. Gorinsky, C. Dovrolis, and R. Sitaraman, “Trade-offs in Optimizing the Cache Deployments of CDNs,” in *Proceedings of IEEE INFOCOM*, 2014.

[27] S. Hasan and S. Gorinsky, “Obscure Giants: Detecting the Provider-free ASes,” in *Proceedings of IFIP Networking*, 2012.

[28] E. Nygren, R. K. Sitaraman, and J. Sun, “The Akamai Network: a Platform for High-Performance Internet Applications,” *SIGOPS Operating Systems Review*, 2010.

[29] M. Podlesny and S. Gorinsky, “Leveraging the Rate-Delay Trade-Off for Service Differentiation in Multi-Provider Networks,” *IEEE Journal on Selected Areas in Communications*, 2011.

[30] J. Lee, “CDN Provider Akamai Expands Services to Costa Rica, Names Asia-Pacific Executives,” goo.gl/BRvvk.

[31] V. Mathew, R. K. Sitaraman, and P. Shenoy, “Energy-aware Load Balancing in Content Delivery Networks,” in *Proceedings of IEEE INFOCOM*, 2012.

[32] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, “Cutting the Electric Bill for Internet-Scale Systems,” in *Proceedings of ACM SIGCOMM*, 2009.

[33] B. Quoitin and S. Uhlig, “Modeling the Routing of an Autonomous System with C-BGP,” *IEEE Network*, 2005.

[34] “TeleGeography: Data-driven Telecommunications Market Research,” http://www.telegeography.com/.

[35] M. Adler, R. K. Sitaraman, and H. Venkataramani, “Algorithms for Optimizing the Bandwidth Cost of Content Delivery,” *Computer Networks*, 2011.

[36] W. B. Norton, “The Internet Peering Playbook: Connecting to the Core of the Internet,” *DrPeering Press*, 2012.

[37] I. Castro and S. Gorinsky, “T4P: Hybrid Interconnection for Cost Reduction,” in *Proceedings of IEEE NetEcon*, 2012.

[38] I. Castro, R. Stanojevic, and S. Gorinsky, “Using Tuangou to Reduce IP Transit Costs,” *IEEE/ACM Transactions on Networking*, 2014.

[39] A. B. Keha, I. R. De Farias, Jr., and G. L. Nemhauser, “Models for Representing Piecewise Linear Cost Functions,” *Operations Research Letters*, 2004.

[40] AMPL Optimization LLC, “AMPL Modeling Language,” http://www.ampl.com/.

[41] Gurobi Optimization, “Gurobi Optimizer 5.0,” http://www.gurobi.com/download/gurobi-optimizer/.

[42] IRL Topology, Cyclops, “UCLA Internet AS-level Topology Archive,” http://irl.cs.ucla.edu/topology/.

[43] CAIDA Data Server, “CAIDA AS Relationship Topology Data,” http://as-rank.caida.org/data/.

[44] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, “AS Relationships: Inference and Validation,” *ACM SIGCOMM Computer Communications Review*, 2007.

[45] Wikipedia, “Tier 1 Network,” http://en.wikipedia.org/w/index.php?title=Tier_1_network&oldid=555211028.

[46] H. Chang, S. Jamin, Z. M. Mao, and W. Willinger, “An Empirical Approach to Modeling Inter-AS Traffic Matrices,” in *Proceedings of ACM IMC*, 2005.

[47] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. De Prisco, and R. Sundaram, “A Methodology for Estimating Interdomain Web Traffic Demand,” in *Proceedings of ACM IMC*, 2004.

[48] Indiana GigaPoP Real-time Atlas, “Akamai Cache Traffic Statistics at Indiana GigaPoP, USA,” http://atlas.grnoc.iu.edu/atlas.cgi?map_name=Indiana%20Gigapop.

[49] Swedish Research Network (SUNET), “Akamai Cache Traffic Statistics at SUNET, Europe,” http://stats.sunet.se/stat-q/r-all?q=all&name=sunet-akamai.

[50] IT Department, CERN, “Akamai Cache Traffic Statistics at CERN, Europe ,” https://netstat.cern.ch/monitoring/network-statistics/ext/?q=CERN&p=EXT&mn=Akamai&t=Monthly.

[51] Akamai MRTG graphs, “Akamai Cache Traffic Statistics at Santa Clara NOC, USA,” http://noc2.sccoe.net/mrtg/akamai.html.

[52] T. L. Magnanti and D. Stratila, “Separable Concave Optimization Approximately Equals Piecewise-Linear Optimization,” *arXiv*, 2012.

[53] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, “On the Placement of Web Server Replicas,” in *Proceedings of IEEE INFOCOM*, 2001.

[54] J. Kangasharju, J. W. Roberts, and K. W. Ross, “Object Replication Strategies in Content Distribution Networks,” *Computer Communications*, 2002.

[55] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, “Distributed Cache Management in Information-Centric Networks,” *IEEE Transactions on Network and Service Management*, 2013.

[56] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, “Intelligent Placement of Datacenters for Internet Services,” in *Proceedings of IEEE ICDCS*, 2011.

[57] A. R. Curtis, S. Keshav, and A. Lopez-Ortiz, “LEGUP: Using Heterogeneity to Reduce the Cost of Data Center Network Upgrades,” in *Proceedings of ACM CoNEXT*, 2010.

[58] A. Dhamdhere and C. Dovrolis, “ISP and Egress Path Selection for Multihomed Networks,” in *Proceedings of IEEE INFOCOM*, 2006.

[59] A. Sharma, A. Venkataramani, and R. K. Sitaraman, “Distributing Content Simplifies ISP Traffic Engineering,” in *Proceedings of ACM SIGMETRICS*, 2013.

[60] M. Yu, W. Jiang, H. Li, and I. Stoica, “Tradeoffs in CDN Designs for Throughput Oriented Traffic,” in *Proceedings of ACM CoNEXT*, 2012.

[61] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar, “Using Batteries to Reduce the Power Costs of Internet-Scale Distributed Networks,” in *Proceedings of SoCC*, 2012.

[62] M. Stevens and E. D’Hondt, “Crowdsourcing of Pollution Data using Smartphones,” in *Proceedings of Ubiquitous Crowdsourcing*, 2010.

[63] N. Maisonneuve, M. Stevens, M. Niessen, P. Hanappe, and L. Steels, “Citizen Noise Pollution Monitoring,” in *Proceedings of DG.O 2009*.

[64] Wikipedia, “Tier 1 Network, 1/1/2009 revision,” en.wikipedia.org/w/index.php?&oldid=261328396.

[65] Wikipedia, “Tier 1 Network, 28/1/2009 revision,” en.wikipedia.org/w/index.php?&oldid=267026426.

[66] Wikipedia, “Tier 1 Network, 25/3/2009 revision,” en.wikipedia.org/w/index.php?&oldid=279646779.

[67] Wikipedia, “Tier 1 Network, 10/2/2011 revision,” en.wikipedia.org/w/index.php?&oldid=413097463.

[68] Wikipedia, “Tier 1 Network, 5/6/2009 revision,” en.wikipedia.org/w/index.php?&oldid=294566542.

[69] M. Brown, C. Hepner, and A. Popescu, “Internet Captivity and the De-peering Menace,” www.renesys.com/tech/presentations/pdf/nanog-45-Internet-Peering.pdf.

[70] Hurricane Electric, “Hurricane Electric BGP Toolkit,” http://bgp.he.net.

[71] University of Oregon Route Views Project, http://www.routeviews.org/.

[72] The RIPE Routing Information Service, http://www.ripe.net/data-tools/stats/ris/routing-information-service.

[73] L. Gao and F. Wang, “The Extent of AS Path Inflation by Routing Policies,” in *Proceedings of IEEE Globecom 2002*.

[74] Internet Routing Registry, http://www.irr.net/.

[75] B. Augustin, B. Krishnamurthy, and W. Willinger, “IXPs: Mapped?” in *Proceedings of ACM SIGCOMM 2009*.

[76] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, “A Measurement-based Analysis of Multihoming,” in *Proceedings of ACM SIGCOMM 2003*.

[77] Z. Ge, D. Figueiredo, S. Jaiswal, and L. Gao, “Hierarchical Structure of the Logical Internet Graph,” in *Proceedings of SPIE ITCOM 2001*.

[78] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, “Characterizing the Internet Hierarchy from Multiple Vantage Points,” in *Proceedings of IEEE INFOCOM*, 2002.

[79] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, “On AS-level Path Inference,” *Proceedings of ACM SIGMETRICS 2005*.

[80] G. Di Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, and T. Schank, "Computing the Types of the Relationships between Autonomous Systems," *IEEE/ACM Transactions on Networking*, 2007.

[81] H. Asai and H. Esaki, "Estimating AS relationships for Application-layer Traffic Optimization," in *Proceedings of ETM 2010*.

[82] H. Asai, H. Esaki, and T. Momose, "A Solution Approach for AS Relationships-aware Overlay Routing," http://tools.ietf.org/html/draft-asai-cross-domain-overlay-01, IETF Internet draft (Informational).

[83] R. Mahajan, M. Zhang, L. Poole, and V. Pai, "Uncovering Performance Differences among Backbone ISPs with Netdiff," in *Proceedings of USENIX NSDI 2008*.

[84] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet Routing Resilience to Failures: Analysis and Implications," in *Proceedings of ACM CoNEXT*, 2007.

[85] W. Deng, P. Zhu, N. Xiong, Y. Xiao, and X. Hu, "How Resilient are Individual ASes against AS-level Link Failures?" in *Proceedings of SCNC 2011*.

[86] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in)Completeness of the Observed Internet AS-level Structure," *IEEE/ACM Transactions on Networking*, 2010.

[87] M. Roughan, S. J. Tuke, and O. Maennel, "Bigfoot, Sasquatch, the Yeti and other Missing Links: What We Don't Know about the AS Graph," in *Proceedings of ACM IMC*, 2008.

[88] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao, "Where the Sidewalk Ends: Extending the Internet AS Graph using Traceroutes from P2P users," in *Proceedings of ACM CoNEXT*, 2009.

[89] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the Internet AS-level Topology," *ACM SIGCOMM Computer Communication Review*, 2005.

[90] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy, "A Systematic Framework for Unearthing the Missing Links: Measurements and Impact," in *Proceedings of USENIX NSDI*, 2007.

[91] A. Dhamdhere and C. Dovrolis, "Ten Years in the Evolution of the Internet Ecosystem," in *Proceedings of ACM IMC*, 2008.

[92] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations," in *Proceedings of ACM SIGKDD 2005*.

[93] E. Gregori, L. Lenzini, and C. Orsini, "K-dense Communities in the Internet AS-level Topology," in *Proceedings of COMSNETS 2011*.

[94] E. Gregori, L. Lenzini, and C. Orsini, "K-clique Communities in the Internet AS-level Topology Graph," in *Proceedings of SIMPLEX 2011*.

[95] V. Valancius, N. Feamster, R. Johari, and V. Vazirani, "MINT: A Market for INternet Transit," in *Proceedings of ACM CoNEXT*, 2008.

[96] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet Routing," in *Proceedings of ACM SIGCOMM*, 2009.

[97] D. P. Bertsekas, *Linear Network Optimization: Algorithms and Codes.* MIT Press.

[98] D. P. Bertsekas, *New Trends in Systems Theory*, 1991, ch. The Auction Algorithm for Assignment and Other Network Flow Problems.

[99] S. Bubeck and N. Cesa-Bianchi, "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems," *CoRR*, 2012.

[100] J. Vermorel and M. Mohri, "Multi-armed Bandit Algorithms and Empirical Evaluation," in *Proceedings of ECML*, 2005.

[101] R. Kleinberg, "Multi-Armed Bandit Problems, Lecture notes, Cornell University, 2007," http://www.cs.cornell.edu/courses/cs683/2007sp/lecnotes/week8.pdf.

[102] S. Pandey and C. Olston, "Handling Advertisements of Unknown Quality in Search Advertising," in *Advances in Neural Information Processing Systems 19.* MIT Press, 2007.

[103] D. P. Bertsekas, D. A. Castanon, and H. Tsaknakis, "Reverse Auction and the Solution of Inequality Constrained Assignment Problems," *SIAM Journal on Optimization*, 1993.

[104] B. Leloup and L. Deveaux, "Dynamic Pricing on the Internet: Theory and Simulations," *Electronic Commerce Research*, 2001.

[105] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Applications*, 2012.

[106] V. Kuleshov and D. Precup, "Algorithms for Multi-armed Bandit Problems," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1402.6028

[107] M. Yuksel, A. Gupta, and S. Kalyanaraman, "Contract-switching Paradigm for Internet Value Flows and Risk Management," in *Proceedings of IEEE INFOCOM*, 2008.

[108] J. E. Godoy, I. Karamouzas, S. J. Guy, and M. Gini, "Adaptive Learning for Multi-Agent Navigation," in *Proceedings of AAMAS*, 2015.

[109] F. B. Shepherd and G. T. Wilfong, "Multilateral Transport Games," in *Proceedings of INOC*, 2005.

[110] A. Gomes, "Multilateral Negotiations and Formation of Coalitions," *Journal of Mathematical Economics*, 2015.

[111] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy, "AS Relationships, Customer Cones, and Validation," in *Proceedings of ACM IMC*, 2013.