



HAL
open science

Peering vs transit: A game theoretical model for autonomous systems connectivity

Giovanni Accongiagioco, Eitan Altman, Enrico Gregori, Luciano Lenzini

► To cite this version:

Giovanni Accongiagioco, Eitan Altman, Enrico Gregori, Luciano Lenzini. Peering vs transit: A game theoretical model for autonomous systems connectivity. Eitan Altman; Konstantin Avrachenkov; Francesco de Pellegrini; Rachid El-Azouzi; Huijuan Wang. Multilevel Strategic Interaction Game Models for Complex Networks, Springer International Publishing, pp.201-237, 2019, 10.1007/978-3-030-24455-2_11 . hal-02413046

HAL Id: hal-02413046

<https://inria.hal.science/hal-02413046v1>

Submitted on 16 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chapter 14

Peering vs transit: A game theoretical model for autonomous systems connectivity

Giovanni Accongiagioco, Eitan Altman, Enrico Gregori and Luciano Lenzini

We propose a model for network optimization in a non-cooperative game setting with specific reference to the Internet connectivity. The model describes the decisions taken by an Autonomous System (AS) when joining the Internet. We first define a realistic model for the interconnection costs incurred; then we use this cost model to perform a game theoretic analysis of the decisions related to link creation and traffic routing, keeping into account the peering/transit dichotomy. The proposed model doesn't fall into the standard category of routing games, hence we devise new tools to solve it by exploiting specific properties of our game. We prove analytically the existence of multiple equilibria

for several scenarios, and provide an algorithm to compute the stable ones. Thanks to the use of simulations we covered those cases for which analytic results could not be obtained, thus analyzing a broad variety of general scenarios, both ad-hoc and realistic. The analysis of the model's outcome highlights the existence of a Price of Anarchy (PoA) and a Price of Stability (PoS), originated by the non-cooperative behavior of the ASes, which optimize their cost function in a selfish and decentralized manner. We further observe the presence of competition between the facilities providing either transit or peering connectivity, caused by the cost differences between these two interconnection strategies.

14.1 Introduction

The Internet ecosystem is made of tens of thousands Autonomous Systems, interconnected together in a complex and dynamic manner. Roughly speaking, Autonomous Systems (AS) are independently administered networks that dynamically connect together to provide end-to-end reachability. ASes can be grouped in different tiers and categories, depending on the service they offer and the organization they belong to: content providers, access providers, transit providers and so on [204].

The late twentieth-century Internet ecosystem was largely dominated by transit links, where the relationship between the connecting ASes was of “customer to provider” type. This kind of relationship produces a hierarchical pricing scheme, where the customer AS pays its provider for the traffic flowing on the link, both incoming and outgoing; in return, the latter provides a default gateway to reach all Internet's routes. For example, an access provider AS wishing to grant Internet access to the eyeballs (i.e. its end users), needs to establish a link with a transit provider, and pay for the traffic flowing on this link. Transit providers are also known as Network Service Providers (NSP) [7] and their pricing strategy is typically volume-based, metered using the 95th percentile traffic sampling technique (this allows customer ASes to burst, for a limited time period, beyond their committed base rate) [202].

The beginning of twenty-first century brought a new paradigm into the environment, since more and more ASes found it beneficial to establish peering links between them [89]. This kind of relationship is “settlement-free”, meaning that the two ASes mutually agree to exchange traffic for free between them, and the only cost they incur is that of laying out the physical interconnection. Peers must agree to each other's policy, which is used to avoid abuse of the peering

relationship. Typical clauses include prohibition of using the peer as default gateway (therefore peers cannot be used to reach other Internet's routes) and traffic ratio balancing, meaning that the ratio between incoming and outgoing traffic over the link must not exceed some value (e.g. 2:1) [201].

The exponential growth of peering links was made possible mainly thanks to the deployment of Internet Exchange Points (IXPs) [120]. These interconnection points are facilities through which ASes can exchange traffic, typically by settlement-free (i.e. peering) relationships. The growth of IXPs, in number and in size, made it easy to establish more and more public peering relationships. In fact, by joining an IXP, an AS can potentially peer with all (usually a subset) of the other ASes connected to the same IXP. The pricing strategy of an IXP, with respect to its customers, is typically flat. Each one of them pays a monthly-based fee, depending on the size (speed) of the port they are using and the cost of maintaining the equipment. Thanks to this mechanism, the IXP can share maintenance costs among all its participants [203]. It is worth noting that this pricing strategy doesn't allow standard cost function modeling (like in [211]), since the addition of new participants potentially brings down the costs of an IXP customer.

When an Autonomous System joins the Internet, it needs to decide the connections to lay out with other ASes. While in the last century, as shown above, the decision space for ASes was substantially small, today they have many alternatives: transit or peering, joining one or more IXPs, dealing with distance and geographic issues. In principle, the best strategy for an AS is the one yielding the lowest cost. However, the outcome of its strategy also depends on that of other ASes dealing with the same problem, thus we find it straightforward to analyze the problem in a game-theoretical framework. We propose a model keeping into account the above factors, which can be used to compute the outcome of this problem and the strategy of the players. Realistic modeling of the whole decision space of an AS is an extremely difficult task, therefore in our work we restrict our analysis to the problem of peering versus transit. Nevertheless, a proper understanding of this problem is fundamental to get insight on the behavior of ASes in the Internet environment, as shown by the results of the analysis.

This work brings contributions both from a game theoretic perspective and an engineering perspective. First of all this is, to the best of our knowledge, a novel model to analyze the strategic choices of ASes living in an Internet environment with both technological and economic constraints. The modeling takes into account many realistic elements, which do not fall into standard game

frameworks, yet tries to keep the problem mathematically manageable. From a game-theoretic perspective, we prove that our game falls in a specific category for which we both demonstrate the existence of equilibria and provide an algorithm for computing stable solutions. From an engineering perspective, the outcome of the analysis is highly insightful as it shows both the suboptimality of the decentralized solution and the emerging competition, first observed in [18], between the two facilities enabling either transit or peering connectivity: Network Service Providers and Internet Exchange Points.

The remainder of this paper is organized as follows: in Section 14.2 we describe the related work. Section 14.3 defines the general model, while Section 14.4 gives analytical results for the general model, derives the existence of equilibria and the algorithm to compute them. Section 14.5 analyzes a subcase of the general model, so as to deepen the obtained results and show the inefficiency of the decentralized solution. In Section 14.6 we use simulations to study the behavior of the system for several configurations, and in particular to analyze a realistic case-study. Finally, we conclude in Section 14.7.

14.2 Related Work

This work relates to the characterization and modeling of the Internet AS-level topology. In this field the majority of graph theoretic models try to reproduce observed Internet topological properties, such as its power-law degree distribution [100], the small-world property [287] and other structural properties (communities, cliques, etc.) [122, 121]. The graph can be reconstructed [126] by either defining some attachment criteria, as in [287, 294, 283, 16] or solving constrained optimization problems for the different nodes, like in [99, 72, 71].

In our paper we rather try to understand network formation as the result of a game between ASes. In this context, it relates to game-theoretic network formation models, which populate both computer science and economics literature (see books [157, 119]). This research branch focuses on proving the existence of equilibria in networks with a fixed number of agents, where links are formed taking into account their preferences in the form of utility functions. The need for mathematical tractability requires simplifications that make these models unrealistic and unable to study real life networks such as the Internet. Other models simulating the dynamics of network formation are agent-based computational models such as GENESIS [183]. In this case authors can include more realistic considerations by skipping the analytical part and simulating the behavior of each agent, hoping to find one of the possibly many equilibria.

In order to keep the problem analytically tractable we do not aim at modeling the whole network formation process. We rather study the interaction between ASes which connect to an existing network in order to serve some demands. A possible modelization of a network where access providers need to select a subset of content providers and fetch traffic from them in a cost-efficient manner is given in [159]. However, the aim of this work is quite different from ours, as it concentrates on the economic analysis of neutral/non-neutral network features, without taking into account the difference between traffic and peering agreements. The models in [83, 194] focus on the competition between backbone providers and on the conditions under which they agree or refuse to establish bilateral peering connections with each other. In [241], authors perform an interesting analysis on network pricing and analyze the economics of private internet exchanges. This kind of peering, known as private peering, has different rules and costs compared with public peering. As explained in the introduction, nowadays Internet is largely dominated by public peering, occurring at IXPs, therefore in our work we concentrate on this last phenomenon, which allows us to give different insights on the present difference between transit and peering.

A partial presentation of the basic components of this work is given in [14, 13]. Specifically, in [14] we analyze a small-scale model, while in [13] some results are extended to the general model. Here we provide an exhaustive framework which: i) gives the analytic results for the general model and for the small scale model; ii) presents an organic set of simulations, integrated by a specific case-study used to infer the behavior of the model in the real Internet.

14.3 Scenario

In the following we describe the general scenario under investigation and derive the cost function.

14.3.1 Description

As said earlier, ASes can be grouped in different categories. Throughout the paper we will use the following categorization, closely resembling the one provided by PeeringDB [7]:

Internet Service Provider (ISP) this node gives to eyeballs (i.e. the end users) and lower tiers, access to the Internet and its contents. Each service

provider has a traffic demand, hereafter demand, which represents the amount of traffic (uplink+downlink) that it handles.

Content Provider (CP) this node has physical access to the contents users are looking after, therefore an ISP with a demand for his specific content, has to connect to it in order to serve this demand.

Internet Exchange Point (IXP) this is the facility that provides peering connection to all its participants. This means that all the nodes connected to a given IXP can potentially communicate with each other.

Network Service Provider (NSP)¹ this node is located at the highest hierarchical level of the network, meaning that each CP can be reached through it. ISPs can reach CPs by establishing a transit connection with an NSP. In particular, here NSPs can be either Tier-1 ASes, or high-level backbone service providers, whose main interest is to sell transit to lower tiers.

We consider a network with $i \in \{1, \dots, I\}$ ISPs, $n \in \{1, \dots, N\}$ CPs and $l \in \{1, \dots, L\}$ transmit facilities (TF), that can be either the NSPs or the IXPs. Without loss of generality, we impose that TFs $j_1 \in \{1, \dots, l_1\}$ are NSPs, while TFs $j_2 \in \{l_1 + 1, \dots, L\}$ are IXPs.

The main difference between transmit facilities is that while links to NSPs are established through transit connections, links to IXPs are established through peering connections.

In a *transit connection*, or customer-to-provider (C2P) connection, the cost to the customer is a function of the amount of traffic that crosses the link (typically expressed as \$/Mbps).

In a *peering connection*, the price is generally flat and depends on the size of the port the customer buys. Moreover, when peering connections are maintained by an IXP, the costs are shared among all the participants.

Each ISP i has a demand for a CP n , which we indicate as ϕ_i^n . The players of our game are the ISPs, which need to decide how to split their demand among all possible transmit facilities. Figure 14.1 depicts our network scenario.

We indicate with $x_{i,l}^n$ the flow from ISP i to CP n via TF l . The strategy of ISP i is given by the vector $\mathbf{x}_i = (x_{i,1}^1, \dots, x_{i,l}^n, \dots, x_{i,L}^N) \in \mathbb{R}^{L \times N}$, while the strategy of all the other players is expressed as $\mathbf{x}_{-i} \in \mathbb{R}^{(I-1) \times L \times N}$. The

¹ISPs, NSPs and CPs are typically ASes. IXPs are not ASes, even if their infrastructure is under a single administrative control.

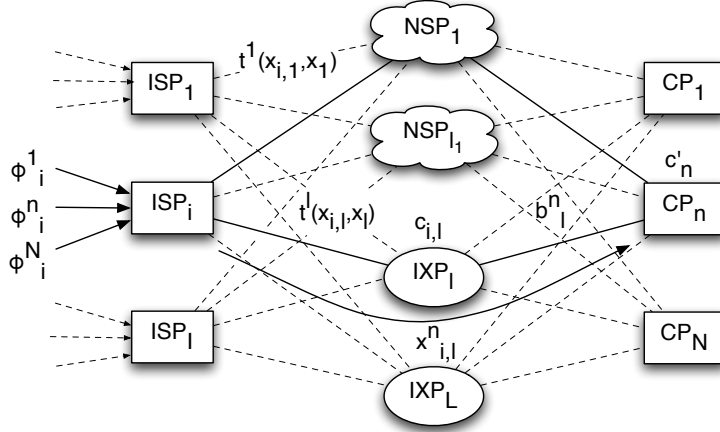


Figure 14.1: General model

goal of each player is to serve, at the minimum possible cost, his demand $\phi_i = (\phi_i^1, \dots, \phi_i^N) \in \mathbb{R}^N$ by splitting it into several flows $x_{i,l}^n$. Please note that we are not dealing with the issue of complete connectivity for the ISPs (in which case, it is sufficient to deploy a single transit connection): our aim is just to enable them to serve their specific demands.

We also indicate as:

- $x_{i,l} = \sum_{n=1}^N x_{i,l}^n$ the total flow from ISP i to TF l ;
- $x_l = \sum_{i=1}^I \sum_{n=1}^N x_{i,l}^n$ the total flow at TF l .

Each player, say i , for each transmit facility (NSP or IXP) it connects to, incurs some costs:

TF usage cost this cost depends on the transmit facility used. If it is an NSP, then it is a function of $x_{i,l}$, the flow from the player to the NSP. Otherwise the TF is an IXP and the cost is shared among all the participants, therefore it also depends on the other players, in the form of x_l , the total flow at the IXP. Consequently, this cost can be written as a function:

$$t^l(x_{i,l}, x_l) \quad (14.1)$$

TF capacity cost each link between an ISP and a TF has a fixed capacity $c_{i,l}$; this means that we have a constraint of the form $x_{i,l} \leq c_{i,l}$. While we may introduce it in the problem “as is”, this would make the model less manageable. Moreover, due to performance and congestion issues, network operators typically avoid reaching the capacity limit and keep a margin for traffic fluctuations. We can think of this performance degradation as a “virtual cost” for the ISP, and therefore model the constraint as a cost, that increases as the flow over the link approaches the capacity limit (as is typically done in the literature for M/G/1 Processor Sharing queues [211]):

$$\frac{1}{c_{i,l} - x_{i,l}} x_{i,l} \quad (14.2)$$

We are aware that, in reality, network operators adjust this capacity when there is more demand for it, and the interconnection cost grows accordingly. However, this situation can be avoided as long as our working region is sufficiently far away from the saturation point. We will always assume that capacities are symmetric w.r.t. the players, therefore $c_{i,l} = c_l \forall i$. Typically the capacity of the NSP can be assumed to be much larger than that of IXPs: $c_{NSP} \gg c_{IXP}$ (see [202] and [203]).

CP reachability cost let’s indicate with b_l^n the cost of transporting one unit of flow from TF l to CP n . This cost is not relevant from the player’s perspective (it is paid by the CP), however it can be used to express the reachability of a given CP. In fact, while all the CPs are connected to the NSPs, an IXP can be connected only to a subset of CPs. This phenomenon can be expressed by putting:

$$b_l^n = \begin{cases} 0 & \text{if } (l \leq l_1) \text{ or } (IXP_l \text{ connected to } CP_n) \\ \infty & \text{otherwise} \end{cases} \quad (14.3)$$

Thanks to all these considerations, the cost function for player i can be expressed as the sum of (14.1), (14.2) and (14.3):

$$C_i(\mathbf{x}_i, \mathbf{x}_{-i}) = \sum_{l=1}^L \left(t^l(x_{i,l}, x_l) + \frac{x_{i,l}}{c_l - x_{i,l}} \right) + \sum_{l=1}^L \sum_{n=1}^N x_{i,l}^n b_l^n \quad (14.4)$$

In order to serve all the demands, each player i has to satisfy the flow constraint: for every CP, the total flow has to be equal to the demand ϕ_i^n .

Therefore player i 's best response $BR_i(\mathbf{x}_{-i})$ is obtained by minimizing cost function (14.4), subject to the flow constraints (14.5):

$$\begin{cases} BR_i(\mathbf{x}_{-i}) &= \arg \min_{\mathbf{x}_i} C^i(\mathbf{x}_i, \mathbf{x}_{-i}) \\ s.t. & \sum_l x_{i,l}^n = \phi_i^n \quad \forall n \end{cases} \quad (14.5)$$

The vector $\mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_I^*) \in \mathbb{R}^{I \times L \times N}$ is an equilibrium of the game if and only if $\mathbf{x}_i^* \in BR_i(\mathbf{x}_{-i}^*) \forall i$, that is, if the strategy of each player is a best response to the strategies of other players.

Throughout the paper we will always refer to the description of Figure 14.1, however, mutatis mutandis, the results are still valid for scenarios where players are CPs or a mix of CPs and ISPs, as long as the demands are changed accordingly.

14.3.2 Transmit Facility Usage Cost

The TF usage cost is different between the NSPs and the IXPs. More specifically the NSP usage cost is linear in the amount of flow that each player sends to it [202]. Therefore we can write:

$$t^l(x_{i,l}, x_l) = a_l x_{i,l} \quad l \leq l_1 \quad (14.6)$$

where a_l , $l \leq l_1$ represents the transit price of NSP l per unit of flow. We are aware that, due to economies of scale in the traffic delivery, transit costs are subadditive in reality. However, introducing this aspect would overcomplicate the model, hiding the truly interesting differences between transit and peering. Nevertheless, we are able to show that some of our results still hold for more generic transit cost functions (see below, Theorem 5).

For the IXP usage [204], each player has to pay a share of the total cost of IXP maintenance. This share can be expressed as the ratio between the flow sent by player i on IXP l and the total flow crossing that IXP: $\frac{x_{i,l}}{x_l}$. Assume we can write the cost of maintenance of IXP l as a function h_l of the total flow through the IXP, therefore the usage cost is:

$$t^l(x_{i,l}, x_l) = \frac{x_{i,l}}{x_l} h_l(x_l) \quad l > l_1 \quad (14.7)$$

The cost of maintaining the equipment of an IXP is, in general, a non-linear function of several parameters. In order to keep the problem manageable, we will approximate this cost with that of a single port which handles x_l , the entire

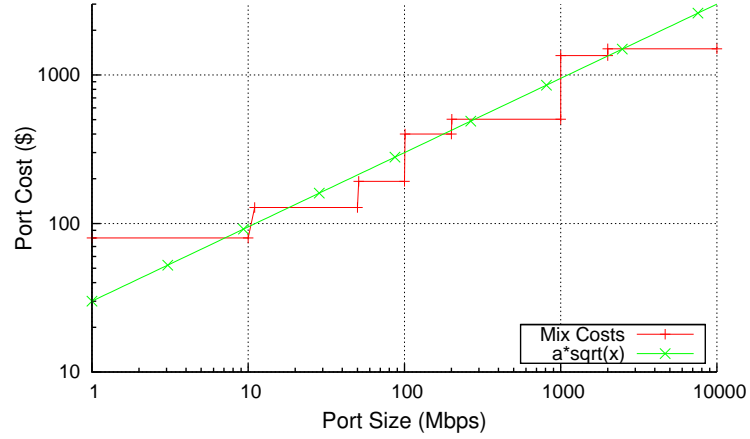


Figure 14.2: IXP port costs for MIX (log-log scale)

flow over the IXP. The cost of a port is a step-wise increasing function, as shown in Figure 14.2 for the MIX² [5], an Italian IXP. This type of cost functions can be modeled (see [192]) by using a function like x^α with $\alpha \in [0.4; 0.7]$. For simplicity, we take $\alpha = 0.5$ as this value provides a fairly accurate fit (shown in Figure 14.2). Therefore, we express the maintenance cost as:

$$h_l(x_l) = a_l \sqrt{x_l} \quad (14.8)$$

where $a_l, l > l_1$ is a constant relating the total flow through IXP l with its maintenance cost. By putting together definitions (14.6), (14.7) and (14.8), the cost function (14.4) can be rewritten as:

$$\begin{aligned} C^i(\mathbf{x}_i, \mathbf{x}_{-i}) &= \sum_{l=1}^{l_1} a_l x_{i,l} + \sum_{l=l_1+1}^L \left(\frac{a_l}{\sqrt{x_l}} x_{i,l} \right) + \\ &+ \sum_{l=1}^L \left(\frac{1}{c_l - x_{i,l}} x_{i,l} \right) + \sum_{l=1}^L \sum_{n=1}^N x_{i,l}^n b_l^n \end{aligned} \quad (14.9)$$

²Milan IXP - public peering costs available online: <http://www.mix-it.net>

Now, we define these new functions:

$$\begin{cases} f^l \left(\sum_i \sum_n x_{i,l}^n \right) = \begin{cases} a_1 & l \leq l_1 \\ \frac{a_l}{\sqrt{\sum_i \sum_n x_{i,l}^n}} & l > l_1 \end{cases} \\ g^l \left(\sum_n x_{i,l}^n \right) = \frac{1}{c_l - \sum_n x_{i,l}^n} \end{cases} \quad (14.10)$$

By using (14.10) and recalling that $x_{i,l} = \sum_n x_{i,l}^n$ and $x_l = \sum_i \sum_n x_{i,l}^n$, we can rewrite (14.9) as:

$$C^i(\mathbf{x}_i, \mathbf{x}_{-i}) = \sum_{l,n} x_{i,l}^n \left[f^l \left(\sum_{i,n} x_{i,l}^n \right) + g^l \left(\sum_n x_{i,l}^n \right) + b_l^n \right] \quad (14.11)$$

Equation (14.11) is the most general expression of the cost function for each player. Please note that (14.11) is in general a non-convex function of $x_{i,l}$, and therefore we cannot directly establish existence of pure equilibria. In particular it does not comply with the general assumptions used for link cost functions in the framework described in [211]. Nevertheless, we cannot avoid dealing with functions of this shape if we want to properly grasp the difference between transit and peering strategies offered, respectively, by NSPs and IXPs.

14.4 General Model

In this section we give analytical results for the general model presented above. Thanks to the peculiar properties of our game, we are able to prove the existence of equilibria under specific conditions, and provide an algorithm to compute them.

14.4.1 The Cost Function

The cost function of Section 14.3 takes into account the presence of many NSPs and IXPs. However, while having multiple IXPs is fundamental for understanding how players aggregate around exchange points, especially in presence of reachability constraints (IXPs may be connected to only a subset of CPs), this is not the case for NSPs, due to the fact that their cost is independent from other players' choice, and they are necessarily connected to all possible CPs. Therefore, without loss of generality for our problem, it is safe to consider

only a single NSP for $l = 1$, and $L - 1$, for $l = 2..L$, IXPs as we do here. In the following we will always refer to this system, which is a single NSP version of Figure 14.1.

If we consider a system with I ISPs, N CPs, and L TFs, with $l_1 = 1$, in a possibly disconnected topology, the cost function (14.11) can be rewritten separating the NSP component from the IXPs. Therefore we have:

$$C^i(\mathbf{x}_i, \mathbf{x}_{-i}) = \sum_n x_{i,1}^n \left(a_1 + \frac{1}{c_1 - \sum_n x_{i,1}^n} \right) + \quad (14.12)$$

$$+ \sum_{l \neq 1} \sum_n x_{i,l}^n \left(\frac{a_l}{\sqrt{\sum_i \sum_n x_{i,l}^n}} + \frac{1}{c_l - \sum_n x_{i,l}^n} + b_l^n \right)$$

where the CP reachability cost for the NSP has been removed since we know from (14.3) that $b_1^n = 0 \forall n$. Before making further considerations on cost function (14.12), in the following we review and extend the theory on supermodularity. The obtained results will be of great help when dealing with our game.

14.4.2 Supermodularity

Definition 1. Supermodular games [292]

Consider a generic game G , where user's payoffs are given by a utility function $u : \mathbb{R}^k \rightarrow \mathbb{R}$. The utility is supermodular if the following condition holds:

$$u(x \vee y) + u(x \wedge y) \geq u(x) + u(y) \quad \forall x, y \in \mathbb{R}^k$$

where $x \vee y$ denotes the componentwise maximum and $x \wedge y$ the componentwise minimum of x and y . The game is supermodular if the utility function of each player is supermodular. If u is twice continuously differentiable, this property is given by the following condition:

$$\frac{\partial^2 u}{\partial z_i \partial z_j} \geq 0 \quad \forall i \neq j$$

In our case we consider costs rather than utilities and minimization instead of maximization, therefore a game like ours is supermodular iff:

$$\frac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \leq 0 \quad \forall i \neq j \quad (14.13)$$

Theorem 1 of [21] proves the existence of equilibria for supermodular games, moreover it provides a way of computing them. The proof is based on showing that **best response sequences are monotone** and therefore converge to a limit which is then shown to be a Nash Equilibrium Point (NEP). The monotonicity is a consequence of the “strategic complementarity” of the players: if one of them chooses a strategy x_i that decreases its own cost, this decision is beneficial for the other players too.

Here we relax the results on the existence of equilibria and convergence of best response sequences in supermodular games.

Definition 2. Symmetric supermodularity

We define as symmetric supermodular games, those for which (14.13) holds for all strategies $\mathbf{x}_i = \mathbf{x}_j$, meaning that the property holds along the symmetric axis.

Definition 3. Symmetric best response sequence

We call symmetric best response sequence a path $(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)}, \dots)$, $(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}, \dots)$, $(\mathbf{x}_i^{(2)}, \mathbf{x}_j^{(2)}, \dots)$, ..., where $\mathbf{x}_i^{(0)} = \mathbf{x}_j^{(0)} = \dots$ and $\forall k$, $(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}, \dots)$ satisfies $\mathbf{x}_i^{(k)} = \mathbf{x}_j^{(k)} = \dots$

Theorem 4. *In symmetric supermodular games, pure equilibria exist and are given as the limit of symmetric best response sequences.*

Theorem 4, whose proof is reported in Appendix 14.8.1, not only proves the existence of equilibria for symmetric supermodular games, but also gives an algorithm for computing them. Please note that, for this theorem to hold, the game does not need to satisfy (14.13) for all possible strategies, but just along the symmetric path. This result can be applied to our game thanks to Theorem 5 and Corollary 6.

14.4.3 Analytic Results

Theorem 5 and Corollary 6 demonstrate existence of equilibria and convergence of the symmetric best response algorithm for the general model. The proofs are reported in Appendix 14.8.1.

Theorem 5. *The game defined in (14.12) is symmetric supermodular.*

Sketch of the Proof. The proof is based on showing that (14.13) holds along the symmetric axis for any possible combination of indexes:

$$\left. \frac{\partial^2 C^i(\mathbf{x}_i, \mathbf{x}_{-i})}{\partial x_{j,\bar{l}}^{\bar{n}} \partial x_{i,l}^n} \right|_{x_{i,l}^n = x_{j,\bar{l}}^{\bar{n}}} \leq 0 \quad \forall i \neq j, \forall l, \bar{l}, n, \bar{n} \quad (14.14)$$

□

It is interesting to note that, as long as the transit cost function t^l of one ISP does not depend on the other ISP, the mixed second derivative (14.14) does not change. Therefore, symmetric supermodularity can be applied to game (14.12) even for more general transit cost functions (as outlined in Section 14.3.2). Please note that without the symmetric assumption, the game is neither supermodular, nor submodular, because we cannot say anything about the sign of the mixed derivatives.

Corollary 6. *The game defined in (14.12) has at least one pure equilibrium for symmetric demands, given as the limit of a symmetric best response sequence.*

14.4.4 The Simulator

Thanks to the supermodularity property, we have not only proven the existence of equilibria for the symmetric version of our game, but also gave a mean for computing these equilibria, as the the limit of symmetric best response sequences. Therefore, it is natural to deploy a simulator, that exploits this property to compute the stable equilibria of our game.

We implemented in MATLAB [187] the general model (14.11) described in Section 14.3. Iteratively, each player performs its best response to the set of other players' strategies as shown in algorithm 1. If the simulation converges, the output **newx** is the NEP for the given input parameters, which are:

- The number of ISPs, TFs and CPs, respectively I, L, N .
- The cost function parameters a_l, c_l, b_l^n and demands ϕ_i^n .
- The *tolerance* and the **startingpoint**.

Please note that, while we have demonstrated convergence of the best repines dynamic (and therefore of our algorithm) in symmetric cases, we do not have convergence guarantees for asymmetric scenarios. However, we know that if the best response sequence algorithm converges, then it converges to an equilibrium [21]. Therefore we can use our simulator both to study our system in general cases, and to assess convergence for specific cases.

Algorithm 1 Best Response Sequence

```

1: startingpoint = ...                                ▷ Initial strategies
2: tolerance = ...                                    ▷ NEP stationarity tolerance
3: newx = startingpoint
4: repeat
5:   oldx = newx                                    ▷ current step game strategies
6:   for  $i = 1; i < I; i++$  do
7:      $\mathbf{x}_{-i} = \mathbf{oldx}_{-i}$                         ▷ other players strategy
8:      $\mathbf{x}_i = \arg \min_{\mathbf{x}_i} C^i(\mathbf{x}_i, \mathbf{x}_{-i})$         ▷  $i$  strategy
9:     s.t.  $\sum_l x_{i,l}^n = \phi_i^n \quad \forall n$ 
10:    newx $_i = \mathbf{x}_i$                                 ▷ next step game strategies
11:   end for
12: until  $\|\mathbf{newx} - \mathbf{oldx}\| < \textit{tolerance}$ 

```

14.5 Minimal Complexity Model (MCM)

After proving existence of equilibria for the general case, here we analyze some subcases in order to understand what kind of equilibria we should expect for specific scenarios. We especially concentrate on the Minimal Complexity Model (MCM), for which we perform an in-depth analysis of: i) the cost function; ii) the best response behavior; iii) the prices of anarchy, stability and fairness.

14.5.1 Subcases Analysis

Starting from the general model of the previous section, we have two main specializations:

Fully Connected Topologies Suppose that we have a fully connected topology, meaning that $b_l^n = 0 \forall l, n$. In such a case, we can take the summation over n and consider cumulative flows and demands:

$$\begin{cases} x_{i,l} = \sum_n x_{i,l}^n & \text{cum. flow } ISP_i \rightarrow TF_l \\ \phi_i = \sum_n \phi_i^n & \text{cum. demand } ISP_i \end{cases}$$

We can now substitute these two variables inside cost function (14.11), thus obtaining an equivalent problem where the strategy of each player is a vector $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,l}, \dots, x_{i,L}) \in \mathbb{R}^L$. This means that, in fully connected topologies, our system is equivalent to another one where we only have a single CP, and each

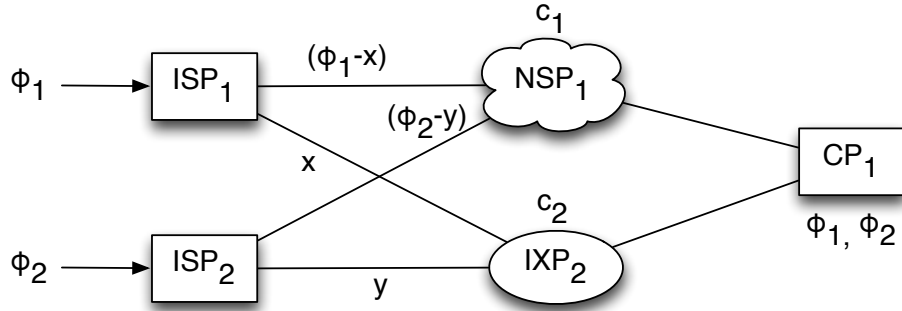


Figure 14.3: Minimal complexity model

player has to serve a cumulative demand ϕ_i for that CP. This happens because there are no reachability constraints, therefore from a player's perspective the specific CP from which he has to fetch data is not relevant.

Symmetric IXPs Suppose that all the IXPs have the same costs, capacities and reachability matrix: $a_l = a_{IXP}$, $c_l = c_{IXP}$, $b_l^n = b_{IXP}^n \forall l \neq 1$. Due to their symmetry, there is an equilibrium where traffic is split equally among them [102], therefore we might think of transforming this problem in an equivalent one having a single IXP with the same reachability matrix and transformed costs and capacities. Unfortunately, we were unable to perform this conversion due to the form of the cost function for the IXPs. In fact, as we see from (14.11) and (14.10), the non linear port cost h_l makes it quite different for players to share small traffic quantities rather than large ones.

The analysis of the two categories highlights that scenarios with multiple CPs can be highly simplified with fully connected topologies, while in the case of multiple IXPs, even if symmetric, the analysis can be quite difficult. Therefore, an easy setting to analyze is the one where we have fully connected topology and just one IXP, due to the fact that we can handle the multiple CPs as if there was just a single one.

Out of all the possible settings to analyze, the easiest is the one where we have just two players. Thus, it is straightforward to define as “**Minimal Complexity Model**” (MCM) the scenario with fully connected topology, $I = 2$ ISP players, $N = 1$ CP (possibly representing an aggregate of all CPs), and $L = 2$ transmit facilities, either the NSP ($l = 1$) or the IXP ($l = 2$). The MCM is depicted in Figure 14.3.

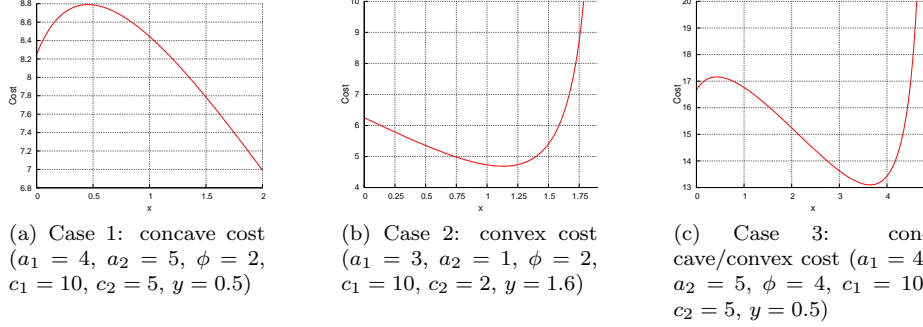


Figure 14.4: Cost function

Thanks to the simplifications of this scenario, the demand of player i can be simply expressed as ϕ_i . With some algebraic manipulations, explicitly shown in Appendix 14.8.2, we can rewrite cost function (14.11) for both players as:

$$\begin{cases} C^1(x, y) = (\phi_1 - x) \left(a_1 + \frac{1}{c_1 - (\phi_1 - x)} \right) + \\ \quad + x \left(\frac{a_2}{\sqrt{x+y}} + \frac{1}{c_2 - x} \right) \\ C^2(x, y) = (\phi_2 - y) \left(a_1 + \frac{1}{c_1 - (\phi_2 - y)} \right) + \\ \quad + y \left(\frac{a_2}{\sqrt{x+y}} + \frac{1}{c_2 - y} \right) \end{cases} \quad (14.15)$$

where x is the flow sent by player 1 through the IXP and $\phi_1 - x$ is, by constraint, the flow sent through the NSP. The same applies to y for player 2. The best response of player i is thus obtained by minimizing $C^i(x, y)$ defined in (14.15).

While simple, the MCM is interesting on its own as it provides a clear way to study the fundamental difference between transit and peering agreements, shedding light on the emerging competition between NSPs and large IXPs, first observed in [18].

14.5.2 Cost Function Analysis

In order to gain insights on the outcome of the behavior of the best response, here we analyze the cost function. Consider the cost function of player 1 and suppose that the strategy y of player 2 is fixed, so:

$$\begin{aligned}
C^1(x) &= (\phi_1 - x) \left(a_1 + \frac{1}{c_1 - (\phi_1 - x)} \right) + \\
&+ x \left(\frac{a_2}{\sqrt{x+y}} + \frac{1}{c_2 - x} \right)
\end{aligned} \tag{14.16}$$

Lemma 7 and Theorem 8, whose proofs can be found in Appendix 14.8.1, tell us the shape of the cost function.

Lemma 7. *The second derivative of the cost function (14.16) is a monotonically increasing function.*

Theorem 8. *The cost function (14.16) can be either: always concave, always convex, or first concave and then convex.*

Figure 14.4 shows the possible cases. Please note that the shape depends both on the parameters and the strategy y of the other player: while for specific values of y the function might be convex as in Figure 14.4b, it can also be concave (Figure 14.4a) and in general is neither convex nor concave, as shown in Figure 14.4c.

14.5.3 The best response behavior

The analysis of the cost function performed in the previous section, suggests that in our game (14.15), even if the best response procedure converges to a NEP, there might be multiple equilibria, because of the presence of multiple local minima. This assumption can be verified via simulation. First of all, let's show a specific case in which the NEP reached can change, depending on the starting point of the algorithm. We can use the implemented algorithm 1 on the symmetric MCM, which has fully connected topology and symmetric demands, by putting $I = 2$, $L = 2$, $N = 1$, $b_l^n = 0 \forall l, n$ and $\phi_i^n = \phi$. Given the selected scenario and the best response sequence algorithm, Theorem 6 ensures the convergence of the simulation for whatever cost function coefficients. Moreover, thanks to the symmetric property, we can just investigate the strategy of player 1 (x), because player 2 will show exactly the same behavior.

We simulate the following parameters: $a_1 = 4$, $a_2 = 5$, $\phi = 4$, $c_1 = 10$, $c_2 = 5$. In this case, as previously shown in Figure 14.4c, the cost function could present multiple local minima, depending on the players' strategies. The simulation has multiple outcomes: if we start from the mean point ($x = 2$,

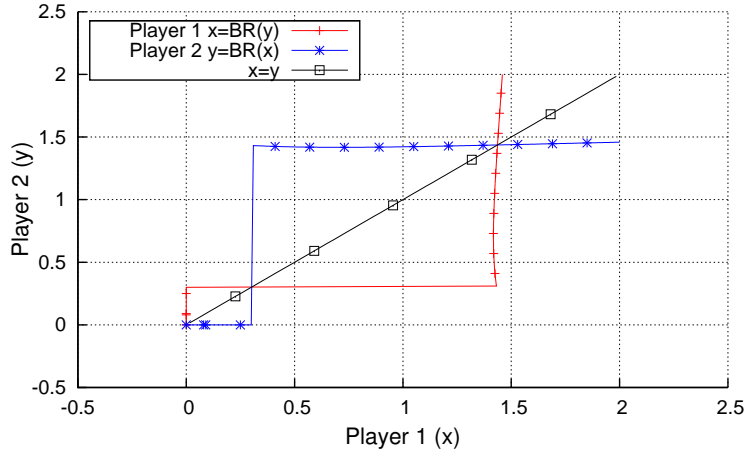


Figure 14.5: BRI - Case 1: $a_1 = 2, a_2 = 2, \phi = 2, c_1 = 10, c_2 = 3$

$\phi - x = 2$) we end up in an equilibrium where traffic is split between the IXP and the NSP: $x^* = 3.64; \phi - x^* = 0.36$. The IXP is preferred because the usage cost is shared among the two players, however it is not used exclusively due to its smaller capacity not being able to serve all the traffic. With a sufficiently larger capacity, all the traffic would have been routed through the IXP. Otherwise, if we start from a strategy where the majority of traffic is routed through the NSP ($x = 0.4, \phi - x = 3.6$), we end up in an equilibrium where all the traffic flows through the NSP: $x^* = 0; \phi - x^* = 4$. This happens because when the IXP is routing a small amount of traffic, the flat port cost is too high to justify its use, therefore the players prefer the NSP. Once the NSP is serving all the traffic no player has an incentive to deviate, because he would pay the whole IXP cost by himself.

As we see, the outcome of the game is highly dependent on the starting point: the IXP is preferred only if it already has, at the beginning, a good amount of flow passing through it, otherwise all players will stick to the NSP. This result is consistent with reality, in fact, the necessary condition for an IXP to emerge is that it has a critical mass (represented by a fraction of the traffic/users in the Internet) which makes the value perceived by a potential participant greater than the cost he would incur in by joining the facility [203].

With the purpose of understanding the number and position of NEPs, we draw the Best Response Intersection (BRI) picture. In this graph, shown in Figure 14.5, the line with tick marks represents the best response x of player 1

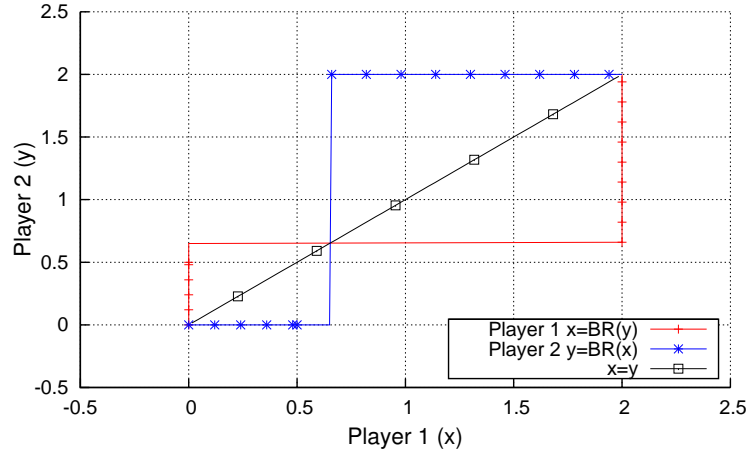


Figure 14.6: BRI - Case 2: $a_1 = 2$, $a_2 = 3$, $\phi = 2$, $c_1 = 10$, $c_2 = 4$

as a function of player 2's strategy y , while the line with cross marks does the exact opposite. The intersection points on the graph mean that both players are playing their best responses, therefore they are Nash Equilibrium Points. As we can see, there are three NEPs and, as expected due to the symmetric property, they are all on the symmetric axis [102]:

Left Equilibrium is in $x = x_L^* = 0$ and corresponds to the scenario where all the traffic is routed through the NSP.

Right Equilibrium is for $x = x_M^* = 1.43$ and is the one where traffic is split between the IXP and the NSP.

Middle Equilibrium happens for $x = x_M^* = 0.31$. This is however a repulsive equilibrium, in fact, as soon as one of the two players deviate, they will never return to this point and reach instead one of the two others equilibria.

These three equilibria can be understood by observing Figure 14.4c: x_L^* and x_R^* are attractive, and correspond to the minima of the cost function, while x_M^* corresponds to the maximum of the cost function, and is thus repulsive. Of course, the last picture corresponds to the cost function for a specific strategy, therefore it cannot assert the position or the existence of equilibria, however it gives an insight on their meaning.

As we change the game parameters we observe that the shape of the best response is always the same, while the position of x_M^* and x_R^* changes. In particular, as shown in Figure 14.6, If the ratio $\frac{a_2}{a_1}$ increases (meaning that IXP cost w.r.t. NSP cost increases) then x_M^* gets nearer to x_R^* , making the left equilibrium is easier to reach. Moreover, we observe that if the capacity c_2 of the IXP is large enough, than in the right equilibrium all the traffic will flow through him.

To conclude this analysis, we verify the behavior of the best response in a slightly more complex case, where the number of players is $I = 3$ (it would be difficult to represent more dimensions). The Best Response Intersection (BRI) graph is shown in Figure 14.7. Just like in the MCM, the picture shows three equilibrium points, obtained by the intersection of the three surfaces representing the players' best responses. As we see from the straight line crossing all such points, the three equilibria are symmetric, with the leftmost (traffic split between NSP and IXP) and the rightmost (all flows through the NSP) being the stable ones.

14.5.4 Price of Anarchy, Stability and Fairness

Social Optimum

We now exploit the MCM to compare the performance of the distributed system, where each Service Provider acts on its own, with that of an ideal centralized system where decisions are took by some external entity. In this case the objective is to minimize the total cost of the two players, given by the summation of the two costs in (14.15):

$$\begin{aligned} C(x, y) &= \sum_i C^i(x, y) = & (14.17) \\ &= (\phi_1 + \phi_2 - x - y) a_1 + \frac{\phi_1 - x}{c_1 - (\phi_1 - x)} + \\ &+ \frac{\phi_2 - y}{c_1 - (\phi_2 - y)} + \frac{x + y}{\sqrt{x + y}} a_2 + \frac{x}{c_2 - x} + \frac{y}{c_2 - y} \end{aligned}$$

Theorem 9 and Corollary 10, whose proofs can be found in Appendix 14.8.1, explain how to optimize this cost function.

Theorem 9. *The cost function (14.17) has a global minimum point. For symmetric demands this minimum point is attained at symmetric strategies, and it is either the left endpoint of the strategy space or the unique local minimum point of its convex part.*

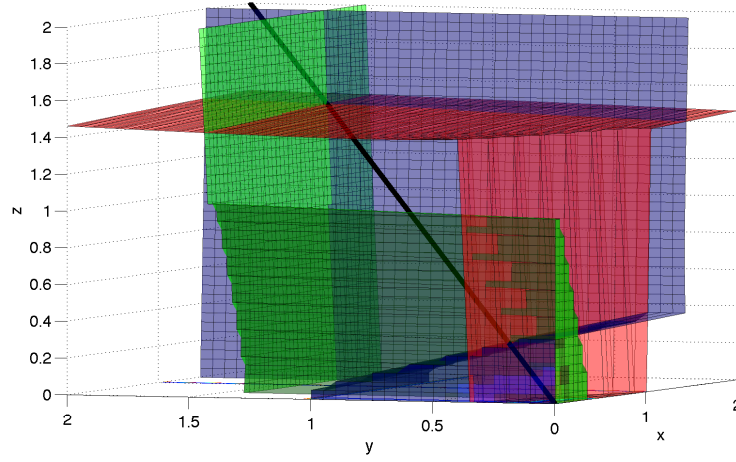
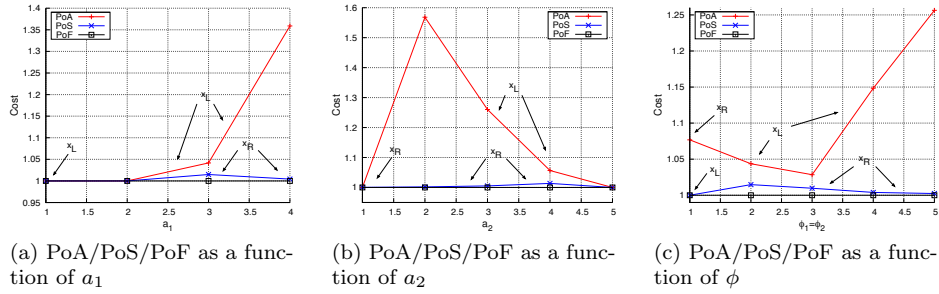


Figure 14.7: BRI for 3 players: $a_1 = 2$, $a_2 = 2$, $\phi = 2$, $c_1 = 10$, $c_2 = 3$



(a) PoA/PoS/PoF as a function of a_1

(b) PoA/PoS/PoF as a function of a_2

(c) PoA/PoS/PoF as a function of ϕ

Figure 14.8: Price of anarchy, stability and fairness

Corollary 10. *The global minimum point of (14.17) is, for symmetric demands, either the left endpoint of the strategy space or the output of a standard algorithm for convex function optimization that starts from the right endpoint.*

The globally optimal solution to problem (14.17) can thus be computed by comparing the two candidate points highlighted in Corollary 10.

Alpha-Fair solution

Another metric for comparison comes from the theory of fairness. A unifying mathematical formulation, known as α -fairness [171], says that given a set of users and utility functions $U_i(x)$, the α -fair solution to the problem of maximizing their utilities is given by:

$$\max_x \left(\sum_i \frac{U^i(x)^{(1-\alpha)} - 1}{1-\alpha} \right)$$

For $\alpha = 0$, this is the same as maximizing the sum of the utilities, thus it gives the social optimum for the problem. The case $\alpha \rightarrow 1$ yields the proportional fair share assignment, however this solution is not feasible when we have to deal with cost function rather than utilities, and for $\alpha \rightarrow \infty$ it is equivalent to the max-min fairness. For $\alpha = 2$, the formula gives us the “harmonic mean fair” solution, which we investigate here:

$$\max_x \left(\sum_i \frac{U^i(x)^{(1-\alpha)} - 1}{1-\alpha} \right) = \min_x \left(\sum_i -1 - \frac{1}{C^i(x)} \right)$$

The solution is computed numerically in the next subsection.

PoA/PoS/PoF Comparison

As is usually done in the literature [159, 171], we define the Price of Anarchy (PoA) as the ratio between the worst decentralized solution (equilibrium) and the social optimum. Similarly, the Price of Stability (PoS) is defined as the ratio between the best equilibrium and the social optimum:

$$PoA = \frac{C(x_{worst}^*)}{C(x_{opt})} \geq 1 \quad PoS = \frac{C(x_{best}^*)}{C(x_{opt})} \geq 1$$

In our case we have just two attractive equilibria, therefore the best and worst equilibria are either x_L^* or x_R^* . Following the same path, we define the price of fairness as the ratio between the fair and the optimal solution:

$$PoF = \frac{C(x_{fair})}{C(x_{opt})} \geq 1$$

Algorithm 1 has been extended to include numerical computation of the above-defined Prices of Anarchy, Stability and Fairness. We use as general

configuration: $a_1 = 3$, $a_2 = 4$, $\phi = 2$, $c_1 = 10$, $c_2 = 3$ (except for Figure 14.8c, where c_2 increases as ϕ increases), and show PoA, PoS and PoF as parameters a_1 , a_2 and ϕ change. Results are reported in Figure 14.8. As we see, it is always the case that $PoF = 1$, meaning that the harmonic mean fair solution is equal to the social optimum.

The PoA almost always corresponds to the left equilibrium. An exception to this is the case where there is a small amount of total traffic, shown in Figure 14.8c for $\phi = 1$: in this case the left equilibrium outperforms the right one, meaning that for small amounts of flow it is not convenient to share costs at the IXP. As ϕ increases, the advantages of sharing become obvious. Figures 14.8a and 14.8b show that the PoA increases as a_1 increases and decreases as a_2 increases. An exception to this is the case $a_2 = 1$ of Figure 14.8b: with these parameters the cost function resembles that of Figure 14.4b, therefore we have only one equilibrium. The PoS is almost always very low, and it is always caused by the fact that the competition between ISPs reduces the amount of traffic through the IXP, thus reducing their opportunities to share costs.

14.6 Simulations

While the analytic results obtained are interesting on their own, as they shed light both on the inefficiency of the decentralized solution and on the competition between an Internet Exchange Point and a Network Service Provider, driven by the clear differences between transit and peering, one might argue that this topology is a bit simplified to represent the Internet, especially due to the symmetric assumption. Here we explicitly tackle this problem by means of simulation, using our MATLAB implementation to test the behavior of the system. Simulations have been performed using the Best Response Algorithm 1: iteratively, each player performs its best response to the set of other players' strategies. If the simulation converges, the output is the equilibrium for the given input parameters, which are:

- the number of ISPs, TFs and CPs, respectively I , L , N ;
- the cost function parameters a_l , c_l , b_l^n and demands ϕ_i^n .

Moreover, it is important to set a **startingpoint**, because, as we saw, on startup IXPs need a critical mass, represented by a share of the total traffic in the system, in order to be able to attract players.

14.6.1 Simulating Ad-Hoc Configurations

In the first part of this section, we focus on the study of the system using ad-hoc configurations, in the sense that we use parameters not necessarily connected with the actual Internet. This kind of analysis can be used to study the behavior of the system for generic parameter combinations. This will prove to be especially useful to infer a set of conditions under which the system behaves “better”, in the sense that the players obtain a higher utility.

The second part of the section will focus on a regional case-study, where parameters are selected according to the real Internet environment.

Growing Number of ISPs/IXPs

We start with showing the behavior of the system for symmetric cases, for which convergence has been proven, and checking what happens as the number of agents in the system grows. The base configuration used is $I = 2$, $L = 2$, $N = 2$, $b_l^n = 0 \forall l, n$ and $\phi_i^n = \phi^n = 2$, and all tests have been performed with fully connected topology and symmetric demands. The cost coefficients used are: $a_1 = 1$, $a_2 = 1.5$, $c_1 = 10$, $c_2 = 6$. As long as flows and capacities are properly balanced, the existence of multiple CPs does not seem to affect the results of the simulations, therefore here we check what happens when we have either more ISPs or IXPs.

Generic Number of ISPs When the number I of players increase, the benefits of joining an IXP increases as well, due to the fact that costs are shared among multiple participants: in fact, as shown in Figure 14.9 on the y1 axis, the fraction of traffic flowing through the IXP at the equilibrium increases with I . We recall from Section 14.4.4 that IXPs need a critical mass to be used, which in our case corresponds to a fraction of the total traffic in the system. Very interestingly, the y2 axis of Figure 14.9 shows that this fraction decreases as the number of player grows.

Generic Number of IXPs In order to have an interesting case study as L grows, we test a scenario where the global IXP capacity does not change, therefore $c_l = c_2/(L - 1) \forall l \neq 1$. This means that instead of having one “large” IXP with a high capacity, we have multiple IXPs with less capacity. In order to have meaningful capacities for the small IXPs, we increased global flows and capacities to: $c_1 = 50$, $c_2 = 25$, $\phi_i^n = \phi^n = 10$. As shown in Figure 14.10 the

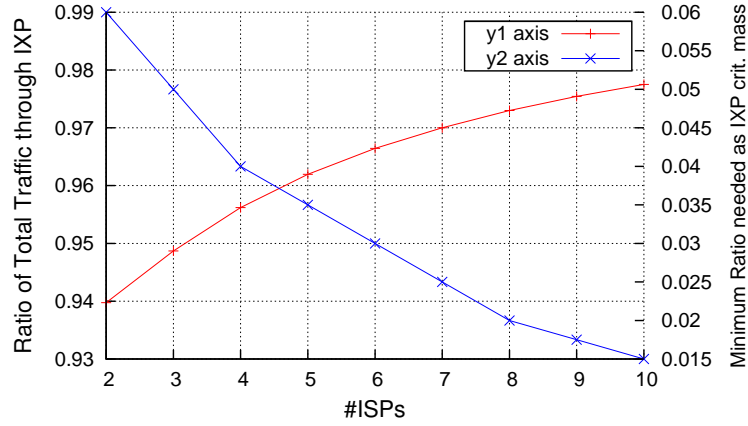


Figure 14.9: Traffic ratios and equilibrium breakpoint as I grows

fact that IXPs only offer small ports is detrimental for the players, and after a certain point they will all stick to the NSP.

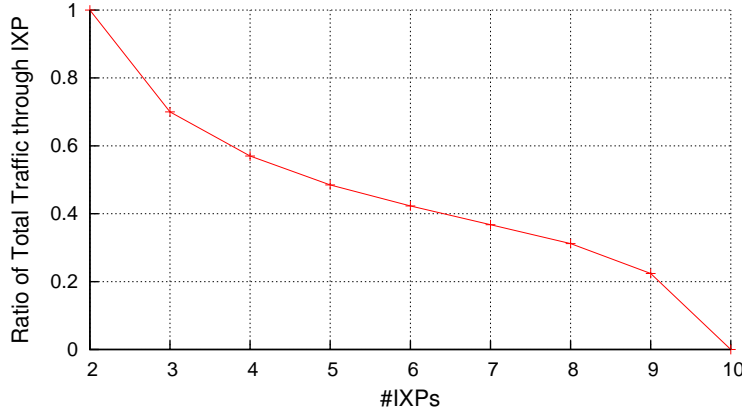
Simulations show that while the IXP critical mass decreases with a larger player base, this effect is counterbalanced by the fact that the critical mass increases with the number of IXPs. Therefore, the results found in the two player case still hold in more realistic scenarios: IXPs need a critical mass to emerge even in scenarios with more ISPs and IXPs, otherwise we still end up in an equilibrium with dominant NSP connectivity.

Flow Path Analysis

Here we show results of simulations regarding the path followed by traffic flows. The simulations were performed both for symmetric scenarios, for which convergence has been proven in the general case, and asymmetric scenarios, for which we have no proofs. In fact, as we'll see later on, in this last case it is possible for players to never reach an equilibrium.

Symmetric Case We simulate a scenario with $I = 10$ ISPs, $L = 4$ TFs ($L - 1$ symmetric IXPs) and $N = 4$ CPs. The connectivity matrix is:

$$b_i^n = \begin{cases} \infty & (l, n) = (2, 2) \vee (l, n) = (4, 1) \vee (l, n) = (4, 4) \\ 0 & \text{otherwise} \end{cases}$$

Figure 14.10: Traffic ratio as L grows

The cost parameter a_l has been chosen in order to be similar to present reality [202], therefore we choose $a_1 = a_{NSP} = 3$ and $\forall l \neq 1, a_l = a_{IXP} = 30$ as seen in the fit performed for the MIX (Figure 14.2). All users have symmetric flows $\phi_i^n = 12.5 \forall i, n$ and their capacities to the facilities are $c_1 = c_{NSP} = 100$ and $\forall l \neq 1, c_l = c_{IXP} = 20$, so that $c_{NSP} \gg c_{IXP}$. As already happened in the MCM, depending on the **startingpoint** we notice the existence of multiple equilibria. In fact, if the initial condition is such that one or more IXPs are underutilized, than at equilibrium those IXPs will not be used. This phenomenon corroborates the outcomes of the MCM, showing that indeed even in general scenarios the competition between NSPs and IXPs, and even between IXP themselves, strongly emerges. Differently from the MCM, in this case we observe more than two stable equilibria, since any combination with one or many unused IXPs can be an equilibrium. Suppose now that the **startingpoint** is such that flows are split equally among the facilities, so that all IXPs have the critical mass to attract players. Figure 14.11 shows the scatterplot at equilibrium. In this plot, each dot represents the flow quantity that each user sends on a given path (that is, to a fixed CP through a given IXP). Due to symmetry, we observe that all users will behave symmetrically on the same path, and this is exactly what happens in Figure 14.11. There is generally a low utilization of the NSP, which rises a little bit for those CPs with a worse reachability matrix (CP_1, CP_4).

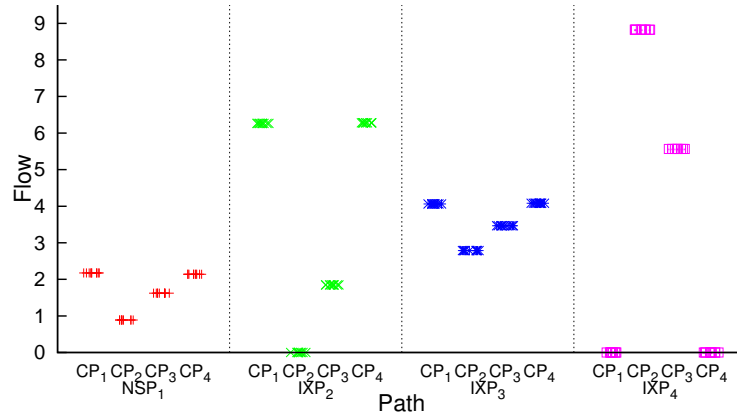


Figure 14.11: Symmetric case flows scatterplot: $\phi_i^n = 12.5$

Asymmetric Case We now show the impact of asymmetric players' demands. In this case, convergence of the best response sequence is not guaranteed by Corollary 6, however, we know that if the simulation converges we certainly reach an equilibrium. We simulate a scenario with exactly the same parameters as in the symmetric case, except that now the demands grow linearly from $\phi_1^n = 10 \forall n$ to $\phi_I^n = 15 \forall n$. The average demand is still 12.5, but now the demand of the last player is 1.5 times that of the first one. The scatterplot at equilibrium is shown in Figure 14.12. Very interestingly, even if demands are asymmetric, paths of flow tend to be almost symmetric for the IXPs, while they spread apart for the NSP. This happens because the benefits of sharing costs at the exchange points is bigger when the traffic ratio is approximately the same between participants, therefore players tend to "symmetrize" around the IXPs. Due to the fact that flows around the IXPs are more or less symmetric, players will send the traffic residual through the NSP, which will see highly asymmetric patterns. In previous case, the asymmetry in players' demands was not very pronounced. Let's now see what happens when the demands go from $\phi_1^n = 6.5 \forall n$ to $\phi_I^n = 18.5 \forall n$, meaning that last player demand is nearly three times that of player one. Again, Figure 14.13 shows the scatterplot at equilibrium. Due to the heavily unbalanced demands, the symmetric patterns around the IXPs are still present, but much less pronounced. While in the previous case equilibrium was driven by the simple rule of "symmetric behavior", in this case the outcome is more difficult to predict. In general, due to the asymmetry, cost benefits of players' for using exchange points decrease, therefore

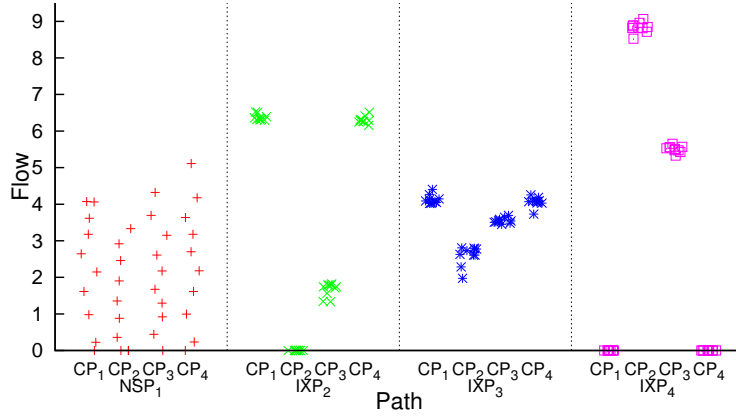


Figure 14.12: Asymmetric case flows scatterplot: $\phi_i^n = 10 \rightarrow 15$

we observe, on average, a higher quantity of flow going through the NSP.

The phenomena emerged through this analysis can provide some preliminary insight on how to devise optimal policies to handle peering traffic at IXPs. More specifically, the “symmetric behavior” rule highlights that it is beneficial to balance traffic as much as possible, therefore IXP owners should create few classes of traffic (namely, few different port sizes), and participants should try to aggregate traffic on these ports, since unbalanced flows must be handled at NSPs and bring to suboptimality. While simulations have been carried out with a limited network size due to computational constraints, the conclusions are fairly general, therefore we expect similar results to hold for larger-scale scenarios.

Non Convergence

Even quite simple scenarios for which we cannot apply Theorem 5, might lead to a situation where players’ behavior oscillates, never reaching an equilibrium. Consider a system with two symmetric IXPs and an asymmetric starting point s.t. a group of players send more traffic to one of the IXPs and less to the other, while the other group of players do the opposite. Due to the asymmetric assumption we cannot apply Corollary 6, and simulation shows that this scenario might never reach an equilibrium. This happens when players enter a never-ending oscillation between the first and the second group, as detailedly shown in Appendix 14.8.3.

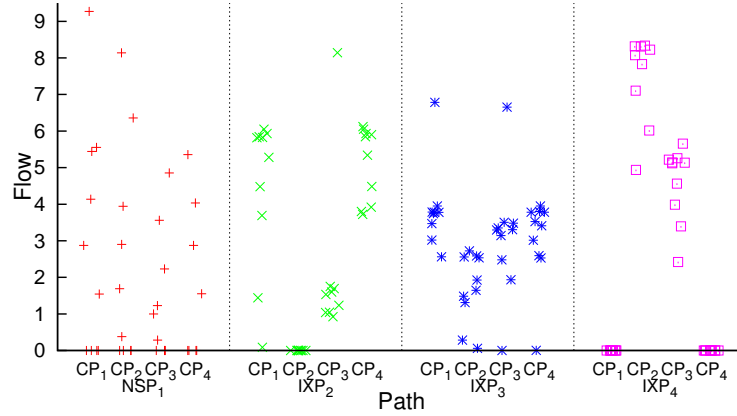


Figure 14.13: Asymmetric case flows scatterplot: $\phi_i^n = 6.5 \rightarrow 18.5$

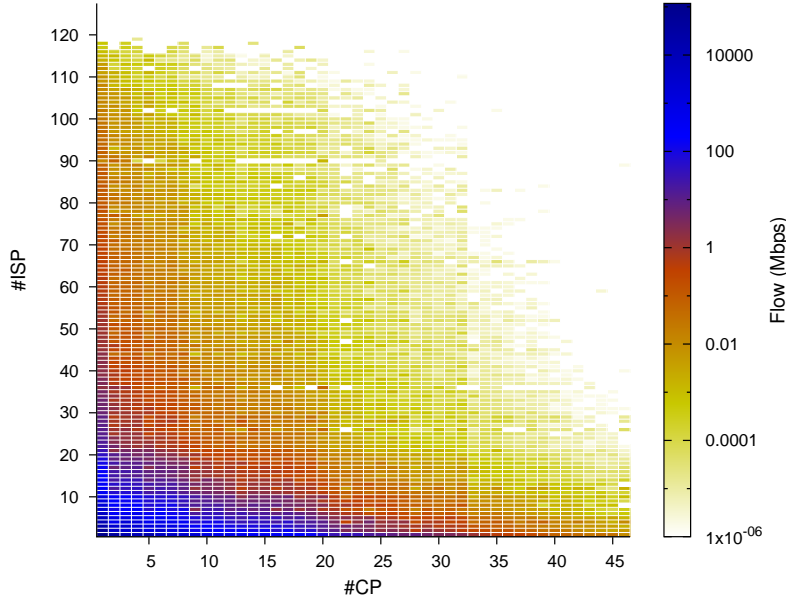
14.6.2 Simulation of a Realistic Scenario

In this part, we instantiate the model on a regional case-study, in order to inspect its behavior for a realistic scenario, and understand which properties of the model apply. We first describe how to set up the scenario, and then the results obtained through simulation.

Simulation Scenario

We focus on a relevant subset of the Internet by focusing on a restricted geographical region: the Italian country. The first thing to do is defining the network topology as in Figure 14.1, by selecting the ISPs, CPs, IXPs and NSPs. We take as IXPs the Top 3 Italian Internet eXchange Points: Milan Internet eXchange (MIX) [5], Torino Piemonte Internet Exchange (ToP-IX) [8], and Nautilus Mediterranean eXchange (NaMeX) [6]. In order to select the ISPs/CPs, we extracted the list of ASes participating to these IXPs (available on the IXPs websites), and labeled each AS as either ISP or CP, according to the information provided by PeeringDB [7]. Furthermore, we add to the topology a Network Service Provider (NSP), which provides transit services to the ISPs and CPs, so that they can exchange traffic even if they are not on the same IXP.

The connectivity matrix $\mathbf{B} = [b_l^n]_{(L \times N)}$ can be directly obtained by considering the IXP participants list: IXP l is connected to CP n only if the Content Provider appears in the IXP participants list. The cost parameter

Figure 14.14: Italian Interdomain Traffic Matrix Φ

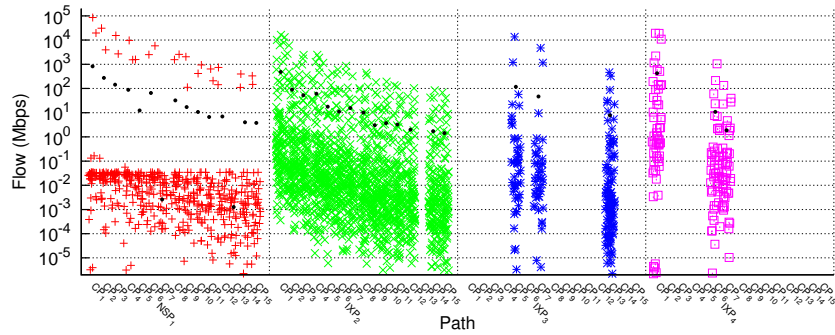
$a_1 = a_{NSP} = 3$ has been chosen in order to be similar to present reality [202], while for IXPs we perform the same fitting procedure as for the MIX (Figure 14.2) and obtain $\forall l \neq 1, a_l = a_{IXP} \approx 30$. As for the capacities, we assume that the capacity of the NSP is sufficient for routing even the heaviest flow, while each IXP has a capacity equal to 1/6 of the heaviest flow.

The Interdomain Traffic (flow) Matrix $\Phi = [\phi_i^n]_{(I \times N)}$ is obtained through the methodology thoroughly described in [15]. Basically, the methodology infers the Interdomain Traffic Matrix (ITM) by exploiting DNS data. To this end, the methodology was applied to the .it DNS traffic recorded at the Institute of Informatics and Telematics of the Italian National Research Council (IIT-CNR). A flavor of the methodology is given in Appendix 14.8.4, while for more information, refer to [15]. The ITM is shown in Figure 14.14. Please note that the matrix is heavily asymmetric, with a low number of large ISPs and popular CPs carrying almost all the traffic.

Table 14.15a sums up the parameters of our Italian case-study.

Parameter	Value
I (#ISPs)	127
N (#CPs)	46
L (#TFs)	1 NSP & 3 IXPs
a_{NSP}	3
a_{IXP}	≈ 30
$\mathbf{B} = [b_i^n]_{(L \times N)}$	IXP Participants
$\Phi = [\phi_i^n]_{(I \times N)}$	ITM in Fig. 14.14

(a) Scenario Parameters Table



(b) Flows Scatterplot

Figure 14.15: Realistic Italian Scenario

Results

Here we illustrate the results obtained by simulating the Italian case study presented above. The scatterplot at equilibrium is shown in Figure 14.15b. The plot shows results for all the ISPs and the 15 CPs carrying most traffic, due to space limits. Nevertheless, results shown here are still valid when considering the entire CP population. The figure shows a number of interesting results.

First of all, we observe a phenomenon which was missing in the simulations of the previous section, even if it was shown when analyzing the Price of Anarchy as a function of ϕ (Figure 14.8c: the NSP is preferred over the IXP when the amount of flow to be sent is very small. Very interestingly we notice that IXPs are the to-go facility for flows of average weight (100 kbps to 100 Mbps), while NSPs are only used in the other cases. The high volumes (over 1Gbps) are served by the NSP due to capacity limits of the IXP, while for the low volumes

the NSPs are used because they provide an economic advantage over the IXP. However, we still notice low traffic volumes flowing through IXPs. This happens due to an effect of traffic trunking: basically, if an ISP has already bought an IXP port for sending traffic and has not reached the capacity limit, it can take advantage of the IXP also for sending small flows. Instead, when the ISP is only involved in small traffic volumes, there is no need of purchasing IXP ports, therefore the NSP is preferred.

As for the “symmetric behavior” outlined in the previous section, we observe that the effect is mild, due to the fact that the ITM is heavily asymmetric, but still present, as shown by the fact that some zones of the clouds in Figure 14.15b are more dense than others.

Last, but not least, we observe that the average amount of flow (represented by the black circles in the figure) handled by the largest IXP, is approximately equal to that handled by the NSP, which is confirmed in [18].

14.7 Conclusions

The proposed model gives insight into the economy of different types of Autonomous Systems and the driving forces behind the decisions they make when joining the Internet. The peculiar pricing strategies of players doesn’t allow standard modelization, however, by exploiting peculiar properties of the game, we are able to prove analytically the existence of multiple equilibria, and provide an algorithm to compute the stable ones. From a game theoretic perspective, while the theory on supermodularity is well-developed, we relaxed this concept and introduced the new category of Symmetric Supermodular games. Thanks to this we were able to prove existence of equilibria and convergence of best response sequences in our game. This is the first case, to the best of our knowledge, where results on supermodularity are applied even if the property does not hold for the game in general, by showing that it holds along the symmetric path. From an engineering perspective, the outcome of the analysis is highly insightful as it shows different interesting aspects. First of all, we observe the suboptimality of the decentralized solution, originated by the non-cooperative behavior of the ASes, by showing the existence of a Price of Anarchy and Stability. Second, we have shown through a realistic setting that while IXPs should be preferred for medium-size flows, NSPs prevail in the other cases. Moreover, we have shown that also for asymmetric cases the system often reaches an equilibrium. Such equilibrium suggests that players have to “symmetrize” their traffic as much as possible with respect to the

peering exchange points, and send their asymmetric traffic quota via the transit service providers. This observation can provide insights on how to devise optimal policies to handle peering traffic at IXPs. Last, but probably most important, we highlight the growing competition between IXPs, providing customers the ability to lay out peering connections, and NSPs, high-level providers selling transit connections, even for large-scale realistic scenarios.

The proposed model was specifically tailored for the Internet environment. This complex system is the result of a human engineering process, which tries to balance social behavior, technological constraints and optimization in the interaction of its building blocks. While all these aspects make game theory a powerful investigation tool, they also require specific design of the models and analysis involved, making it difficult to derive general conclusions for networks which lack these characteristics. Nevertheless, obtained results shed light on the emerging competition, between the two facilities enabling either transit or peering connectivity: Network Service Providers and Internet Exchange Points. This phenomenon, first observed in [18], is a key point driving the evolution of the Internet, thus it is directly related to CONGAS.

14.8 Appendix

14.8.1 Proofs

Proof of Theorem 4. Consider a sequence of best responses $(\mathbf{x}_i^{(0)}, \mathbf{x}_j^{(0)}, \dots)$, $(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}, \dots)$, $(\mathbf{x}_i^{(2)}, \mathbf{x}_j^{(2)}, \dots)$, \dots . Due to symmetry we can choose this path to be a symmetric best response sequence. From definition 2 and by applying the same reasoning as in the original proof [21], we shall get monotone sequences whose limits are equilibria. \square

Proof of Theorem 5. First of all, we can use constraint (14.5) to reduce the number of variables of our system. In fact if we perform the summation over n on left and right member, and separate the NSP component from the IXPs, we obtain:

$$\sum_n x_{i,1}^n + \sum_{l \neq 1} \sum_n x_{i,l}^n = \sum_n \phi_i^n \quad (14.18)$$

Now we substitute $\sum_n x_{i,1}^n$ taken from (14.18) inside (14.12) and rearrange

terms, so as to obtain:

$$\begin{aligned}
C^i(\mathbf{x}_i, \mathbf{x}_{-i}) &= \sum_{l \neq 1} \sum_n x_{i,l}^n \left(\frac{a_l}{\sqrt{\sum_i \sum_n x_{i,l}^n}} + \frac{1}{c_l - \sum_n x_{i,l}^n} \right) + \\
&+ \left(\sum_n \phi_i^n - \sum_{l \neq 1} \sum_n x_{i,l}^n \right) \left(a_1 + \frac{1}{c_1 - (\sum_n \phi_i^n - \sum_{l \neq 1} \sum_n x_{i,l}^n)} \right) + \\
&+ \sum_{l \neq 1} \sum_n x_{i,l}^n b_l^n
\end{aligned} \tag{14.19}$$

where all the flow variables have $l \neq 1$. Now we need to compute the mixed second derivatives of equation (14.19). We observe that the second term, referring to the NSP, has no mixed components, due to the fact that the cost does not depend on other players' choice, therefore this term becomes zero in the computation. The same happens with the last term, therefore we have:

$$\begin{aligned}
\frac{\partial^2 C^i(\mathbf{x}_i, \mathbf{x}_{-i})}{\partial x_{j,\bar{l}}^{\bar{n}} \partial x_{i,l}^n} &= \frac{\partial}{\partial x_{j,\bar{l}}^{\bar{n}}} \left[\left(\frac{a_l}{\sqrt{\sum_i \sum_n x_{i,l}^n}} + \frac{1}{c_l - \sum_n x_{i,l}^n} \right) + \right. \\
&\left. + \sum_n x_{i,l}^n \left(-\frac{a_l}{2(\sum_i \sum_n x_{i,l}^n)^{3/2}} + \frac{1}{(c_l - \sum_n x_{i,l}^n)^2} \right) \right] \\
&\quad \forall i \neq j, \forall \bar{l}, \bar{n}, n
\end{aligned} \tag{14.20}$$

Following the same reasoning previously done, we observe that the second and forth term in (14.20) do not depend on $x_{j,\bar{l}}^{\bar{n}}$, therefore their contribution in the final derivative is zero. Moreover, we observe that first and third term only have flow variables with index l , therefore for any $\bar{l} \neq l$ the whole derivative becomes zero:

$$\frac{\partial^2 C^i(\mathbf{x}_i, \mathbf{x}_{-i})}{\partial x_{j,\bar{l}}^{\bar{n}} \partial x_{i,l}^n} = 0 \quad \forall i \neq j, \forall \bar{l} \neq l, \forall n, \bar{n} \tag{14.21}$$

while in the other case we have:

$$\begin{aligned}
\frac{\partial^2 C^i(\mathbf{x}_i, \mathbf{x}_{-i})}{\partial x_{j,l}^{\bar{n}} \partial x_{i,l}^n} &= -\frac{a_l}{2 \left(\sum_i \sum_n x_{i,l}^n \right)^{3/2}} + \frac{3a_l \sum_n x_{i,l}^n}{\left(\sum_i \sum_n x_{i,l}^n \right)^{5/2}} \\
&\quad \forall i \neq j, \forall l, n, \bar{n}
\end{aligned} \tag{14.22}$$

Please note that, regardless of the chosen n, \bar{n} , the derivatives are all the same. In order to prove symmetric supermodularity, we have to show that property (14.13) holds for both (14.21) and (14.22). While in the first case this is trivial, for the second one we multiply (14.22) by the positive quantity $\left(\sum_i \sum_n x_{i,l}^n\right)^{3/2}$, thus obtaining that:

$$\operatorname{sgn} \left(\frac{\partial^2 C^i(\mathbf{x}_i, \mathbf{x}_{-i})}{\partial x_{j,l}^n \partial x_{i,l}^n} \right) = \operatorname{sgn} \left(\frac{a_l}{4} \cdot \frac{\sum_n x_{i,l}^n - 2 \sum_{j \neq i} \sum_n x_{j,l}^n}{\sum_i \sum_n x_{i,l}^n} \right) \quad (14.23)$$

Along the symmetric axis we have that $x_{i,l}^n = x_{j,l}^n \forall i \neq j, \forall l, n$, meaning that each couple (i, j) of players send, to a fixed CP n through a given IXP l , the same quantity of flow. With this condition, equation (14.23) is always negative, therefore (14.12) is symmetric supermodular. \square

Proof of Corollary 6. By hypothesis the demands satisfy $\phi_i^n = \phi_j^n \forall i \neq j, \forall n$. Therefore ISPs keep playing along the symmetric axis [102], and we obtain this result by combining Theorems 4 and 5. \square

Proof of Lemma 7. The second derivative of function (14.16) w.r.t. x is:

$$\begin{aligned} \frac{\partial^2 C^1(x)}{\partial x^2} &= \frac{2c_1}{(c_1 - (\phi_1 - x))^3} + \frac{2c_2}{(c_2 - x)^3} + \\ &+ \frac{3a_2 x}{4(x+y)^{\frac{5}{2}}} - \frac{a_2}{(x+y)^{\frac{3}{2}}} \end{aligned} \quad (14.24)$$

We have that $c_1 \gg c_2, \phi_1$, therefore the first term is negligible when trying to check the sign of this derivative. Given that $x, y > 0, c_2 > x$ and $x \leq \phi_1$, the second term is an always positive increasing function. For a fixed value of y , the summation of the third and fourth term is an always negative increasing function. Therefore (14.24) is a monotonically increasing function, as it is the summation of two increasing functions. \square

Proof of Theorem 8. According to Lemma 7 the second derivative of (14.16) is monotonically increasing. Therefore we can only have three cases:

$$\left\{ \begin{array}{ll} \frac{\partial^2 C^1(x)}{\partial x^2} < 0 \forall x & \Rightarrow C^1 \text{ always concave} \\ \frac{\partial^2 C^1(x)}{\partial x^2} > 0 \forall x & \Rightarrow C^1 \text{ always convex} \\ \exists! \bar{x} \text{ s.t. } \frac{\partial^2 C^1(x)}{\partial x^2} \Big|_{x=\bar{x}} = 0 & \Rightarrow C^1 \text{ concave in } [0; \bar{x}] \\ & \text{and convex in } [\bar{x}; c_2] \end{array} \right.$$

□

Proof of Theorem 9. The strategy space is $x \in [0, \min(c_2, \phi_1)[$ and $y \in [0, \min(c_2, \phi_1)[$. Within this set, C is continuous, therefore, by Weierstrass' theorem, it has a global minimum, which might be either a stationary point or one of the interval endpoints. In order to find all the stationary points, we study the function gradient:

$$\begin{cases} \frac{\partial C(x,y)}{\partial x} = -a_1 - \frac{c_1}{(c_1 - (\phi_1 - x))^2} + \frac{a_2}{2\sqrt{x+y}} + \frac{c_2}{(c_2 - x)^2} \\ \frac{\partial C(x,y)}{\partial y} = -a_1 - \frac{c_1}{(c_1 - (\phi_2 - y))^2} + \frac{a_2}{2\sqrt{x+y}} + \frac{c_2}{(c_2 - y)^2} \end{cases} \quad (14.25)$$

By hypothesis we have symmetric demands: $\phi_1 = \phi_2 = \phi$. If we sum the two equations in (14.25) we have that:

$$-\frac{c_1}{(c_1 - (\phi - x))^2} + \frac{c_2}{(c_2 - x)^2} = -\frac{c_1}{(c_1 - (\phi - y))^2} + \frac{c_2}{(c_2 - y)^2}$$

which clearly is true only for symmetric strategies, therefore we must have $x = y$.

The capacity of the NSP is typically $c_1 \gg \phi, x$ therefore we can simplify $\frac{c_1}{(c_1 - (\phi - x))^2} \approx \frac{1}{c_1}$. In order to find the stationary points we need to find the roots of equation:

$$-a_1 - \frac{1}{c_1} + \frac{a_2}{2\sqrt{2x}} + \frac{c_2}{(c_2 - x)^2} \quad (14.26)$$

Unfortunately, this is a fifth-degree polynomial, therefore we don't have an explicit solution. Consider now the derivative of equation (14.26):

$$-\frac{a_2\sqrt{2}}{8x\sqrt{x}} + \frac{2c_2}{(c_2 - x)^3} \quad (14.27)$$

This is a monotonically increasing function (as it is the sum of two increasing functions), that goes to $-\infty$ for $x \rightarrow 0$ and to $+\infty$ for $x \rightarrow c_2$, therefore it has one root. As a consequence, we know that eq. (14.26), representing the first derivative of C , is convex and has limits $+\infty$ for $x \rightarrow 0, c_2$. Therefore, it has a unique minimum point corresponding to the root of equation (14.27).

Given the form of its first derivative (14.26), the cost function (14.17) is concave in its first part (where the derivative decreases) and convex in the second part (where the derivative increases). The points of minimum of the concave part are its two endpoints, while the convex part has a unique minimum point. The right endpoint of the concave part is inside the convex

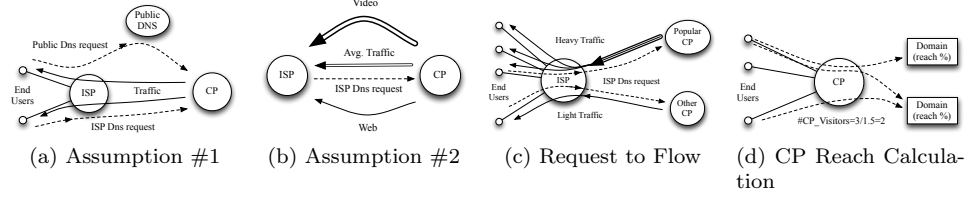


Figure 14.16: DNS to Traffic: Assumptions and Calculations

part (C is continuous), therefore the convex minimum is an improvement over it. Henceforth, the global minimum point is either the minimum point of the convex part, or the left endpoint of the concave part, which is also the left endpoint for $C(x, y)$. \square

Proof of Corollary 10. Given such an algorithm, we execute it on the function giving as initial point the right endpoint of the strategy set, where we know that the function is convex. The output of the algorithm is the local minimum of the convex part, therefore according to theorem 10 the global minimum point is either this point or the left endpoint. \square

14.8.2 MCM Cost Function Derivation

Consider the model in Figure 14.3. First of all, given that the topology is fully connected, meaning that $b_l^n = 0 \forall l, n$, we can erase all those terms. Moreover, we have $I = 2$, $N = 1$, $L = 2$, where $l = 1$ is the NSP and $l = 2$ is the NSP, therefore we can explicitly rewrite the general cost function (14.11) for the two players as:

$$\begin{cases} C^1(\mathbf{x}_i, \mathbf{x}_{-i}) &= x_{1,1}^1 \left(a_1 + \frac{1}{c_1 - x_{1,1}^1} \right) + \\ &+ x_{1,2}^1 \left(\frac{a_2}{\sqrt{x_{1,2}^1 + x_{2,2}^1}} + \frac{1}{c_2 - x_{1,2}^1} \right) \\ C^2(\mathbf{x}_i, \mathbf{x}_{-i}) &= x_{2,1}^1 \left(a_1 + \frac{1}{c_1 - x_{2,1}^1} \right) + \\ &+ x_{2,2}^1 \left(\frac{a_2}{\sqrt{x_{1,2}^1 + x_{2,2}^1}} + \frac{1}{c_2 - x_{2,2}^1} \right) \end{cases} \quad (14.28)$$

Constraints shown in (14.5) can also be rewritten as:

$$\begin{cases} x_{1,1}^1 + x_{1,2}^1 = \phi_1 \\ x_{2,1}^1 + x_{2,2}^1 = \phi_2 \end{cases} \quad (14.29)$$

where ϕ_1 and ϕ_2 are the demands from player 1 and 2 to the CP, respectively. For the sake of readability, we apply the following variable renaming to our problem:

$$\begin{cases} x = x_{1,2}^1 \\ y = x_{2,2}^1 \end{cases} \quad (14.30)$$

Meaning that x is the flow sent by player 1 to CP 1 (the only one present) through the IXP, while y is the analogous for player 2. Due to constraints (14.29) we have that:

$$\begin{cases} x_{1,1}^1 = \phi_1 - x \\ x_{2,1}^1 = \phi_2 - y \end{cases} \quad (14.31)$$

By substituting in (14.28) the variables defined in (14.30) and the ones obtained in (14.31), we obtain the final form of the cost functions for the two players, shown in (14.15).

14.8.3 Non convergence case-study

Consider a system with fully connected topology, $I = 16$ ISPs, $L = 3$ TFs (one NSP and two symmetric IXPs) and $N = 1$ CP. Once again we use cost parameters $a_{NSP} = 3$ and $a_{IXP} = 30$. Capacities are $c_{NSP} = 200$ and $c_{IXP} = 70$ and we even take symmetric demands $\phi_i^n = 50 \forall i, n$. Furthermore, we separate players in two groups: players i with $i \leq 8$ belong to the first group, while players for $i > 8$ take part into the second group.

According to Corollary 6, equilibrium can be reached by following a sequence of symmetric best responses. Instead, we set an asymmetric starting point, depending on the group:

$$\begin{cases} x_{i,l} = 24.9 & \text{if } (l = 1 \wedge i \leq 8) \vee (l = 2 \wedge i > 8) \\ x_{i,l} = 25.1 & \text{if } (l = 2 \wedge i \leq 8) \vee (l = 1 \wedge i > 8) \\ x_{i,l} = 0 & \text{otherwise} \end{cases}$$

Simulation shows that players never reach an equilibrium, and go through a never-ending oscillation between two points:

$$\begin{cases} x_{i,l}^- = 0 \ x_{i,l}^+ = 50 & \text{if } (l = 1 \wedge i \leq 8) \vee (l = 2 \wedge i > 8) \\ x_{i,l}^- = 50 \ x_{i,l}^+ = 0 & \text{if } (l = 2 \wedge i \leq 8) \vee (l = 1 \wedge i > 8) \\ x_{i,l}^- = x_{i,l}^+ = 0 & \text{otherwise} \end{cases}$$

This happens because, on each iteration, each player of the first group sees the second group of players on a different IXP, and finds it beneficial to deviate on that TF. The same happens for the players of the second group, which in turn deviate altogether to the IXP of the first group. After the deviation, situation is reversed, therefore the two groups keep deviating all the time, never reaching an equilibrium. Please note that using a symmetric starting point would immediately lead to an equilibrium where $x_{i,l} = 25$ if l is an IXP, and zero otherwise.

14.8.4 Inferring ITM from DNS data

Here we briefly summarize our methodology for inferring the Interdomain Traffic Matrix (ITM) by exploiting DNS data [15]. The DNS data we use, was obtained by exploiting the Passive DNS monitoring system developed at IIT-CNR [88] and used to collect data for all the .it domains. The monitoring system produces logs containing all the information regarding each DNS request performed by a client to the DNS server, including the client AS and the query performed. Therefore, the first step consists of building a DNS Requests Matrix. This matrix contains the number of requests performed by a Source AS to a Destination AS. Please note that since we use DNS traffic for inferring this matrix, we are only capturing traffic flowing in client-server environments, while we are discarding peer-to-peer traffic.

The second step consists of transforming the DNS request matrix into an Interdomain Traffic Matrix. The transformation is carried out through 4 steps, which implement a set of assumptions so as to map each DNS request to an amount of traffic generated by a single client. The 4 steps are depicted in Figure 14.16b, and are reported below:

Step 1 The DNS request of a client can be performed by either its ISP or a Public DNS Server (e.g. Google DNS). While, in the first case, the endpoints of the request will be the same as that of the traffic flow, in the second case this is not true (see Figure 14.16a). Therefore, when converting DNS requests to Traffic flows, we have to discard all those Source ASes which are not ISPs.

Step 2 The amount of traffic flowing between two ASes due to a DNS request, depends on the application type (e.g. Video, Web, etc..). Since we are not interested in differentiating between flows, we can easily solve this problem by using a constant quantity obtained by averaging the traffic generated by different sources (see Figure 14.16b). We indicate as $\#avg_request_traffic$ the average traffic generated by a request from a single user. This can be estimated by averaging the amount of data downloaded from several webpages, with either text, video, etc.

Step 3 In order to transform a DNS request into a traffic flow, we have to consider that each DNS request performed by a client to the DNS server is cached at the ISP, and potentially used for all the other clients. Moreover, the number of clients interested in the same content, depends on the popularity of the content itself (see Figure 14.16c). More precisely, we can represent the relationship between a request between source and destination AS and the number of users that can benefit from it as:

$$\#ISP_Users * \frac{\#Content_Visitors}{\#Total_Visitors}$$

The $\#ISP_Users$ is calculated, for each ISP, by computing the ratio between the IP address space of each ISP and the total IP space; afterwards we estimate the user population by multiplying this quantity by the total population of the country.

Step 4 The quantity $\frac{\#Content_Visitors}{\#Total_Visitors}$ is also known as *Reach* (%), and has been measured for several websites by Alexa [4]. It basically expresses the popularity of a given content, with respect to all other contents. While the metric has been measured for websites, we need to convert it to ASes, since we are interested in interdomain traffic. In order to do so, we: i) discard all Destination ASes which are not Content Providers hosting websites ranked by Alexa, since these CPs are the only ones for which we are able to estimate the *Reach* metric; ii) calculate the *Reach* of a CP as the the sum of all *Reaches* of its websites, normalized by the average number of requests of the CP clients. To understand this formula, we shall look at Figure 14.16d. In the Figure, there is a CP hosting two websites and 2 clients making requests. As we see, there is one visitor for the first domain, and two visitors for the second domain, therefore, if we simply sum the $\#Content_Visitors$ for the CP, we obtain 3, while the users effectively accessing the CP are only 2. However, if we normalize this quantity

Algorithm 2 Traffic Generator

```

1: // Input Data
2: DNS_Req = < Source_AS, Dest_AS, #Requests >
3: AS_population = < ISP, #Users >
4: AS_reach = < CP, Reach % >
5: // Output Data
6: Traffic = < Source_AS, Dest_AS, Flow >
7:
8: avg_request_traffic = ...                                ▷ Average Traffic Generated
9:                                                         by a Request (Single User)
10: for all Row ∈ DNS_Req do
11:   src = Row[Source_AS]
12:   dst = Row[Dest_AS]
13:   req = Row[#Requests]
14:   if src ∈ ISP && dst ∈ CP then
15:     Traffic[src, dst] = req * avg_request_traffic *
16:       * AS_population[src] * AS_reach[dst]
17:   end if
18: end for

```

by the average number of requests of the CP clients ($((2 + 1)/2 = 1.5$ in the example), we can correct this overestimation.

Once we have computed all the input data, the DNS Request Matrix can be transformed into an Interdomain Traffic Matrix. Basically each request between a CP and an ISP is transformed to a flow quantity equal to:

$$\#avg_request_traffic * ISP_Users[ISP] * Reach[CP]$$

The final conversion procedure is described in Algorithm 2. For more information on the methodology used for inferring the ITM by exploiting DNS data, and on the results obtained through it, refer to [15].