Computer Networks 55 (2011) 3959–3974

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Trade & Cap: A customer-managed, market-based system for trading bandwidth allowances at a shared link

Jorge Londoño [a,1], Azer Bestavros [a,*,2], Nikolaos Laoutaris [b]

[a] Computer Science Department, Boston University, Boston, MA 02215, USA
[b] Telefónica I+D, Via Augusta, 177, 08021 Barcelona, Spain

ARTICLE INFO

Article history:
Received 11 June 2010
Received in revised form 17 March 2011
Accepted 30 May 2011
Available online 14 June 2011
Responsible Editor: A. Popescu

Keywords:
Bandwidth management
Incentive mechanisms
Selfish users

ABSTRACT

We propose Trade & Cap (T&C), an economics-inspired mechanism that incentivizes users to voluntarily coordinate their consumption of the bandwidth of a shared resource (*e.g.*, a DSLAM link) so as to converge on what *they* perceive to be an equitable allocation, while ensuring efficient resource utilization. Under T&C, rather than acting as an arbiter, an Internet Service Provider (ISP) acts as an enforcer of what the community of rational users sharing the resource decides is a fair allocation of that resource. Our T&C mechanism proceeds in two phases. In the first, software agents acting on behalf of users engage in a strategic *trading* game in which each user agent selfishly chooses bandwidth slots to reserve in support of primary, *interactive* network usage activities. In the second phase, each user is allowed to acquire additional bandwidth slots in support of a presumed open-ended need for *fluid* bandwidth, catering to secondary applications. The acquisition of this fluid bandwidth is subject to the remaining "buying power" of each user and by prevalent "market prices" – both of which are determined by the results of the trading phase and a desirable aggregate *cap* on link utilization. We present analytical results that establish the underpinnings of our T&C mechanism, including game-theoretic results pertaining to the trading phase, and pricing of fluid bandwidth allocation pertaining to the capping phase. Using real network traces, we present extensive experimental results that demonstrate the benefits of our scheme, which we also show the salient features of an efficient implementation architecture, settling the basis for a practical implementation of the system.

1. Introduction

Motivation: The ever increasing appetite for Peer-to-Peer (P2P), media streaming, and Video on Demand (VoD) content is forcing service providers to constantly upgrade their infrastructures to keep-up with customers bandwidth demands. This state-of-affairs is significantly exacerbated by the prevalence of flat-pricing schemes and hence the lack of an incentive for users to moderate their hunger for network bandwidth, especially around periods of peak network utilization, which are the primary determinants of an Internet Service Provider (ISP) costs (both in terms of infrastructure upgrade cycle and inter-AS traffic volume costs du e to the 95/5 rule). Attempts by ISPs to deviate from flat pricing (including field-tested per-byte pricing [1]) have been widely rejected by customers [2]. This is also reinforced by the prevalence of flat pricing in the telephony market [3].

In addition to the significant capital investments that ISPs must shoulder to ensure that their networks are well provisioned during the few hours of peak demand, the new "Internet world order" of seemingly unbounded

* Corresponding author.
E-mail addresses: jmlon@cs.bu.edu (J. Londoño), best@cs.bu.edu (A. Bestavros), nikos@tid.es (N. Laoutaris).

[1] Supported in part by the Universidad Pontificia Bolivariana and COLCIENCIAS–Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología "Francisco José de Caldas".
[2] Supported in part by NSF awards #0720604, #0735974, #0820138, #0952145, and #1012798.

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.05.019

hunger for bandwidth further complicates fundamental issues that have confounded the networking community for decades, including the adoption of an acceptable notion of fairness as it relates to congestion management. Congestion increases delay and losses, reducing the perceived Quality of Service (QoS) of *interactive* applications such as web browsing, VoIP, and video streaming. Dealing with congestion requires that users (flows) "pay" for their share of the congestion they cause [4], resulting in a degradation in QoS (the congestion price). But, when interactive applications are forced to compete with non-interactive applications – such as P2P filesharing, background backup services, or VoD downloads – the degradation in QoS becomes unacceptable.

Under flat pricing, during periods of peak demand, current congestion control practices could be seen as particularly "unfair" to users of low-volume, mostly-interactive applications who would be effectively subsidizing "bandwidth hogs." This has prompted some ISPs to act as arbiters, proactively shaping user traffic by setting quotas,[3] or by preferentially treating different traffic payloads (*e.g.*, web browsing vs. bittorrent downloads) during periods of peak demand.[4] These efforts have backfired, eliciting a public relations quagmire regarding violation of "Net Neutrality," [17,18] which is perceived as the prime reason for the Internet being the cradle of innovation it is [19]. Proactive ISP intervention based on traffic payload also raises concerns regarding monopolistic practices, *e.g.*, blocking or taxing Video/VoIP services not provided by the same ISP [19].

Although QoS mechanisms have been around for a while, their deployment on the Internet is minimal. The wide-spread adoption of flat-rate pricing models prevents the utilization of currency-based mechanisms for assigning priority classes [20–22]. Other mechanisms that do not rely on currency, such as Re-feedback [23] handle congestion by policing flows on the network, but do not give the end-user the means to expresses preferences between different classes of traffic, and thus make it impossible to guarantee the QoS level for certain applications. In contrast, the T&C mechanism provides the means to trade bandwidth allocations, so that the minimum expected demands are ensured through the provider's network (or the last mile), where bottlenecks typically materialize. Giving the end-users a guaranteed minimum bandwidth allocation makes it possible to prioritize the traffic right at the edge, where policies are under the control of the end-user and existing mechanisms can be used to provide guarantees. While the general strategy used in T&C could be adopted for handling bandwidth allocations at the core of the network, in this paper we restrict our attention to its use at the edge.

We note that the term "Cap and Trade" is widely known in (and typically associated with) the market mechanism used to control the emission of pollutants into the environment [24]. The mechanism we propose in this paper also uses marketplace supply (or allowances) and demand as a mechanism to control congestion in the network. As our mechanism operates first by trading and then by capping, we call it "Trade & Cap".

Scope and contributions: Rather than having ISPs act as arbiters who set the rules regarding what constitutes fair usage of a shared resource, in this paper, we propose a market-based T&C system in which user software agents converge on what *they* perceive as an equitable allocation of resources, irrespective of what these resources are used to support (HTTP vs P2P traffic) and irrespective of the absolute resource allocation (traffic volume) per user.[5] In our setting, the role of the ISP is that of providing a *mechanism* that supports any privately-defined user *policy* [26].

Effectively, our proposed T&C mechanism sets up a marketplace. Given the fixed (flat-rate) payment to the provider, customers enter this marketplace with equal buying power, but their use of this fairly-allocated buying power depends on their flexibility. This allows customers to trade "volume" during low-utilization periods for "quality" during peak-utilization periods (or vice versa). The direction of the trade (not to mention the user's willingness to even engage in trading) depends entirely on user preferences and flexibility (*e.g.*, tolerance for delaying a scheduled network backup job).[6] In addition to empowering customers to trade bandwidth allocations, T&C has the desirable side effect of smoothing traffic utilization over time, thus reducing the ISP's cost which is determined primarily by the peak rate.

Outline and summary of results: We start this paper in Section 2 by overviewing the T&C mechanism as it applies to a Digital Subscriber Line Access Multiplexer (DSLAM) setting, and in Sections 3 and 4 by presenting analytical results pertaining to convergence and efficiency of the marketplace underlying T&C. Formulating the problem as a game is not only useful for purposes of modeling and understanding the marketplace dynamics, but also it serves as the basis of a real mechanism that can be implemented and applied in practice. Thus, in Section 6 we discuss the salient features of an implementation architecture for T&C in a DSLAM setting. Our implementation allows the marketplace interactions to be carried out by software agents that run on behalf of the users and the ISP, and thus (with the exception of minimal configuration and parametrization) is quite transparent to the end user. Next, in Section 7, we demonstrate the significant advantages of T&C by presenting results from extensive trace-driven simulations. For instance, we show that introducing a relatively small level of flexibility in the scheduling of user activities results in significant gains for both the users and the ISP. For example, allowing user agents to reposition bandwidth

[3] Incidentally, when demand is well below the provider's nominal capacity, supporting bandwidth hogs is basically free, bringing to question the use of traffic volume "quotas" [5].

[4] Along these lines, there is a growing body of academic [6–11] and industry [12–16] work on delineating interactive from non-interactive traffic in order to police/balance consumption. Many of these systems depend on Deep Packet Inspection (DPI) techniques, raising concerns about consumer privacy. Moreover, the scalability and resilience of these techniques is also questionable as applications adapt quickly to avoid detection, *e.g.*, by using encryption and randomization of port numbers.

[5] We note that recent polls [25] indicate that consumers would accept traffic allocation mechanisms that ensure fairness as long as these mechanisms do not trample on net neutrality, privacy, *etc.*

[6] We use the term "user" liberally since in practice, customer-side software agents would make most decisions on behalf of the user.

allocations within relatively small windows of time enables them to increase their share of fluid bandwidth (supporting non-interactive applications) by 20–40% depending on their flexibility. This benefits the ISP as well, resulting in as much as 16–31% reduction in the 95th percentile of the ISP's 5-min traffic volume, and (even more impressively) resulting in smoothing traffic volume, reducing the 95th-percentile/50th-percentile ratio from 1.58 to an almost perfect ratio of 1.004. We conclude the paper in Section 8 with a review of the related literature.

2. T&C in a DSLAM setting

While our T&C mechanism is applicable to any setting in which it is desirable to coordinate the fractional acquisition by a set of rational parties of the *shared* capacity of a single resource, in this paper, and without loss of generality, we restrict ourselves to a specific setting – that of coordinating the utilization of a shared DSLAM link.

Fig. 1 illustrates the basic architecture of Digital Subscriber Line (DSL) access technology. In this setting, DSL modems on the customer side connect hundreds to thousands of users to a single DSLAM server on the provider network. DSLAMs connect to a Broadband Remote Access Server (BRAS) which relays traffic to/from the Internet. In this setting, the DSLAM-BRAS link poses the most significant traffic management problems for ISPs and is thus the shared resource managed using our T&C mechanism.[7]

As we alluded before, we envision a marketplace where DSL customers are empowered to trade capacity over time, so as to facilitate the exchange of traffic volume for QoS. This exchange is desirable given the different utility that various applications attribute to traffic volume vs. QoS (*e.g.*, Fluid-Traffic (FT) applications value traffic volume whereas Reserved Traffic (RT) applications value QoS).[8] In the envisioned marketplace, the DSLAM server's role is to enforce the capacity allocations agreed upon by the DSL customers. By doing so the ISP will benefit as well. The T&C marketplace dynamics result in a more balanced load over time, improving user satisfaction. The more balanced load also reduces de pressure for infrastructure upgrades to accommodate peak demand.

For our purposes, we assume that the marketplace will operate over fixed, non-overlapping periods of time, which we call *epochs* (*e.g.*, days), and that the trading and allocation of capacity will occur within T subdivisions of an epoch, which we call *time-slots*, *e.g.*, 288 5-min slots per day to match a *de facto* industry standard of 5 min for traffic accounting and pricing.

At the beginning of each epoch, the operator assigns each agent $i = 1, 2, \ldots, n$ an allowance or *budget* B_i in accordance with the user's Service Level Agreement (SLA) (*e.g.*, "Business" versus "Residential" plans). Under flat pricing, which we assume in this paper, all customers receive an equal budget. User agents also collect usage statistics to profile user demand for the RT applications. This information provides an estimate of the user demand and will be used as the basis for reserving bandwidth for the user's RT.

Our T&C mechanism proceeds in two phases:

(1) *The Bandwidth Trading Phase:* This phase proceeds as a pure-strategies, non-cooperative game among agents, who are allowed to rationally and selfishly decide *when* to schedule bandwidth allocations in support of their RT. Reserved Traffic is the traffic belonging to applications requiring a specific (minimum) bandwidth during a continguous period of time. RT may be flexible in terms of start and end times, but not in terms of reserved bandwidth over time. If RT is flexible, the agent's goal is to minimize the cost incurred in adquiring the fraction of the link capacity necessary to support the RT demand. The scheduling of RT sessions is subject to preset user preferences and constraints. The outcome of this game is a Nash-Equilibrium (NE) of RT bandwidth allocations to all participating agents, along with the corresponding cost incurred by each agent.
(2) *The Bandwidth Capping Phase:* This phase proceeds as a market-clearing phase, in which the operator distributes any remaining capacity among agents. The amount of "remaining" capacity distributed in this phase is set based on a desirable nominal utilization of the link (*e.g.*, determined by the 95/5 rule threshold). The allocation of bandwidth in the capping phase rewards agents who were able to preserve more of their budgets in the trading phase (due to a low RT volume or due to flexibility in scheduling such traffic), ensuring a market equilibrium of the resulting allocations. The capping phase is executed after the trading phase to make it possible for the agents to know what time-slots have lower demands, and therefore allocations will result in a larger allocation for the same cost.

Both phases are designed in such a way so as to provide users with the means via which they are able to maximize the benefit (utility) they obtain from the network. In the case of RT, this is done by ensuring that users have a minimum guranteed capacity to satisfy their expected interactive demands. In the case of FT, this is done by ensuring that users are able to extract as much fluid bandwidth as possible from the network. Observe that the classification between RT and FT is done at the user-side, and the user has control over what belongs to each class. This way, T&C provides the mechanism, but leaves the policies up to the users. The mechanism provides added value to the user in two ways: The improved performance experience for interactive applications during periods of high demand, and the ability to allocate larger capacities (and not being blocked or rate-limited) during periods of low demand.

It is important to highlight that the bandwidth allocations that the agents obtain after the conclusion of the two phases of T&C do not constitute a hard reservation, and do not interfere with lower-level QoS and congestion management mechanisms. In fact, the resulting reservations

[7] T&C is equally valuable and practical if the resource to be managed is not "physical" but rather "virtual" – *e.g.*, the aggregate inter-ISP (transit) traffic of a subnetwork. Our distributed implementation architecture discussed in Section 6 is particularly suited for managing such resources.

[8] Our T&C mechanism ensures that resource allocations are based on true valuations by the users themselves (rather than assumed by the ISP).

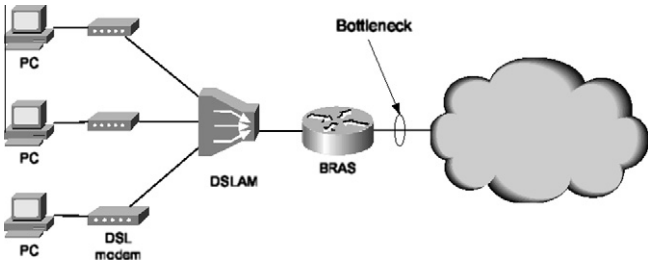

Fig. 1. Illustration of the DSL "last-mile" architecture.

constitute minimum guarantees during corresponding time-slots, but as we detail in Section 6 an implementation using a work-conserving scheduler would ensure that traffic from active users can exceed their reservation when other users are inactive. It should also be mentioned that the implementation of the mechanism may enforce a minimum reservation on all time-slots (independent of the profile). This ensures that user activities in atypical (or unexpected) time periods are not starved during periods of high demand.

As it is well established that traffic at the consumer edge of the network is highly asymmetric – with bottlenecks almost exclusively in the downstream direction – in this paper we restrict our use if the T&C mechanism to the management of the shared downstream capacity to end-users. Section 6 describes how this allocations are enforced from the implementation perspective.

3. The bandwidth trading phase

Each agent i represents its RT demand as a vector of requested bandwidth allocations: $T_i = (t_{i1}, \ldots, t_{il_i})$. An assignment of an agent's demand is a mapping that pins each one of the components of the vector to a different time slot. A set of such assignments (one per agent) comprises a potential configuration, or *schedule* of RT utilization at the DSLAM.

Let $k = m_i(j)$ be the time slot assigned to the jth component of agent i's request vector. We denote by x_{ik} the actual allocation for agent i in time-slot k, where $x_{ik} = t_{i,m_i(j)}$. The x_{ik} notation implicitly represents the mapping $m_i()$, noting that for time-slots that are not used in the mapping, we assume that $x_{ik} = 0$. Thus, x_{ik} is defined for all time-slots. The vector of requested allocations, $X_i = (\ldots, x_{ip}, \ldots)$ will be called *the bid* in what follows.

The benefit that the user accrues by completing its tasks is usually expressed in the form of a utility function. In the context of network applications, quantifying the utility is impractical due to the large number of applications and the granularity of the tasks involved (accessing a web page, reading an e-mail, *etc.*). Instead, we make the following observation: Interactive applications are sensitive to response times and they are valuable to a user if completed within their timing constraints. Failure to complete an interactive task in time yields a negligible or zero utility. On the other hand, if some tasks are flexible, their utility is equally realized independent of the time at which they are completed within their *validity window*. So, if there are two or more feasible mappings $m_i()$, their utility is the same, but it may be the case that some of the mappings are more convenient for the rest of the users and the network itself. More concretely, a mapping is more desirable if it results in a reduction in the congestion during a given time-slot. To make this difference explicit to the user we define the following cost function:

Definition 1 (*Cost Function*). The cost of the RT vector T_i is

$$c_i = \begin{cases} \frac{1}{C}\sum_{p=1}^{T} x_{ip}U_p & m_i() \text{ is feasible,} \\ \infty & \text{otherwise,} \end{cases} \quad (1)$$

where $U_p = \sum_{i=1}^{n} x_{ip}$ is the aggregate reservation on slot p, and C is a constant.

The above cost function (which is proportional to the product of the current utilization and the demand of the agent over all time slots) can be interpreted as a *cost-sharing* scheme where each agent pays its fair share of the price of each time slot, which depends on the *square* of the time-slot's utilization

$$c_i = \frac{1}{C}\sum_{p=1}^{T} U_p^2\left(\frac{x_{ip}}{U_p}\right).$$

The motivation for the cost function in Eq. (1) is twofold. First, in schemes where cost is constant or proportional to the user's demand, there is no incentive for an agent to avoid congested time-slots – a given level of resource (bandwidth) usage costs the same in either case. Our cost function creates the desired incentive of steering agents away from congested time slots (if they possess the flexibility to do so). Second, our cost function is fair in the sense that users sharing the same time slot pay the same unit-price. Non-linear cost functions (of which ours is an instance) have been used before [27] to control congestion and achieve "proportional fairness".

The strategy space S_i^* for agent i is the set of permutations of its request vector. As such, the strategy space is finite with cardinality $|S_i^*| = P_{l_i}^T$. The game's strategy space is the Cartesian product of the strategy spaces of all agents: $\mathcal{S} = \times S_i$. Initially, we will assume that all the points in the strategy space are feasible, and later we will incorporate various feasibility constraints. The trade proceeds by each user-agent obtaining the current demand (U_p) on the time-slots of the next epoch and submitting a bid X_i. All the bids are processed asynchronously by the server-agent

(running for example in the DSLAM), which facilitates the implementation (since no distributed synchronization mechanism would be required). The following theorem guarantees that the bidding process comes to an end:

Theorem 1. *The pure strategies game in which agents adopt better/best responses to allocate atomic units of traffic in per-user, mutually-exclusive time-slots converges to a NE.*

Proof. We define the following potential function:

$$\Phi = \sum_{i=1}^{n} c_i = \frac{1}{C}\sum_{p=1}^{T} U_p^2,$$

when an agent makes a cost-reducing move, $\Delta c_i < 0$,

$$\frac{1}{C}\sum_{p}\left(x'_{ip}U'_p - x_{ip}U_p\right) < 0. \tag{2}$$

Notice that for any other agent $k \neq i$, its utilization of interval p does not change, but the change in the total utilization affects its cost as follows

$$\Delta c_k = \frac{1}{C}\sum_{p} x_{kp}(U'_p - U_p).$$

Adding the changes of the agents other than i we get

$$\begin{aligned}\sum_{k\neq i}\Delta c_k &= \sum_{k\neq i}\left(\frac{1}{C}\sum_{p} x_{kp}(U'_p - U_p)\right)\\ &= \frac{1}{C}\sum_{p}\left((U'_p - U_p)\sum_{k\neq i} x_{kp}\right)\\ &= \frac{1}{C}\sum_{p}\left((x'_{ip} - x_{ip})\sum_{k\neq i} x_{kp}\right),\end{aligned} \tag{3}$$

where in the last step we used the fact that $U'_p - U_p = x'_{ip} - x_{ip}$ because agents other than p did not change their allocations. Since the components of x'_{ip} are the same as those of x_{ip} (but in different positions), we observe that $\sum_p x'^2_{ip} = \sum_p x^2_{ip}$. With this, we can reorganize expression (2) as follows

$$\frac{1}{C}\sum_{p}\left(x'_{ip}U'_p - x_{ip}U_p\right) = \frac{1}{C}\sum_{p}\left((x'_{ip} - x_{ip})\sum_{k\neq i} x_{kp}\right) < 0,$$

which is exactly the same as (3), *i.e.* $\sum_{k\neq i}\Delta c_k = \Delta c_i < 0$. As the sum of negative quantities is negative, we get

$$\sum_i \Delta c_i = \Delta\Phi < 0,$$

i.e. the potential is monotonically decreasing. Being a sum of squares it is lower-bounded by zero, therefore the game converges to a Nash Equilibrium. □

As we alluded before, it may be the case that an agent is subject to additional constraints that limit its strategy space – *e.g.*, a 2 h-long RT fixed bandwidth allocation must be assigned in consecutive time-slots, and be scheduled to start between 6 pm and 8 pm. Such constraints are easily captured by defining the agent's strategy space as a subset of $S_i \subseteq S_i^*$. Three practical examples of such constraints are: (1) *RT sessions* to enforce the atomicity of reservations for application sessions that span several consecutive time-slots, (2) *Capacity constraints* to ensure that the shared link capacity is never exceeded by the aggregate allocation – $\forall p : \sum_{i=1}^{n} x_{ip} \leqslant C$, and (3) *Budget constraints* to ensure that no agent is able to reserve resources beyond its "fair" share, which is upper-bounded by the agent's allowance – $\forall i : \frac{1}{C}\sum_{p=1}^{T} x_{ip}U_p \leqslant B_i$. Notice that these sets of constraints correspond to the elimination of infeasible points in the strategy space $\mathcal{S}$. This removal can be easily accomplished by setting to ∞ the cost for the agent at unfeasible points.

Theorem 2 (*Convergence to NE under constraints*). *Given a pure strategies game, such that each agent's action space is finite, and the game converges under better/best response dynamics to a NE, then after adding constraints to the action space of one or more agents, the game still converges, given that there exists feasible configurations after the addition of the constraints.*

Proof. Consider the following directed graph $G = \langle V, E\rangle$: There is a vertex $v_j \in V$ for every possible point in the strategy space $v_j = (a_{1j_1}, \ldots, a_{nj_n})$, where a_{ij} denotes the jth action of agent i. There is and edge $e_{pq} \in E$ for any valid transition[9] on the strategy space, *i.e.* the cost associated with agent i at vertex p is larger than the cost at vertex q: $c_p(i) > c_q(i)$ and $a_{-i,p} = a_{-i,q}$, meaning that the actions of all agents other than i are the same in p and q. Let us call G the transition graph of the game. Then, if the game always converges to a NE in the unconstrained case, G is a Directed Acyclic Graph (DAG). Any path (sequence of actions) the agents traverse when following their rational-selfish goal will always reach a vertex with no outgoing edges corresponding to a NE (of possibly many) of the game. The addition of constraints to the agents actions, corresponds to removing unfeasible vertices from V as well as the edges coming into or out of these vertices. Let G' be the residual transition graph after removing unfeasible vertices and edges. Suppose the new game with constraints does not always converge to a NE. Then, there exists at least one cycle in the residual transition graph G'. Being G' a subgraph of G this implies the same cycle must exist in the original graph G, contradicting the fact that G is a DAG. □

Fig. 2 illustrates the construction used in the proof. Fig. 2(a) shows the DAG corresponding to the transitions of some hypothetical game, where states v_6 and v_8 are NE. Fig. 2(b) v_4 and v_6 have been removed with their respective edges because they are unfeasible. The NE in the residual graph are v_3 and v_8. Notice that the set of NE vertices after the addition of constraints need not to be the same as those of the unconstrained game. In particular, feasible vertices that were not a NE will become a NE if all their outgoing edges are removed.

An important consideration about equilibria of non-cooperative games is the Price of Anarchy (PoA) – the ratio of the social cost at the worst-case equilibrium compared to the best possible. In the case of the Bandwidth Trading game, the social cost (understood in our case to be the

[9] Observe that the set of edges in not limited to best-responses, but includes any feasible move.

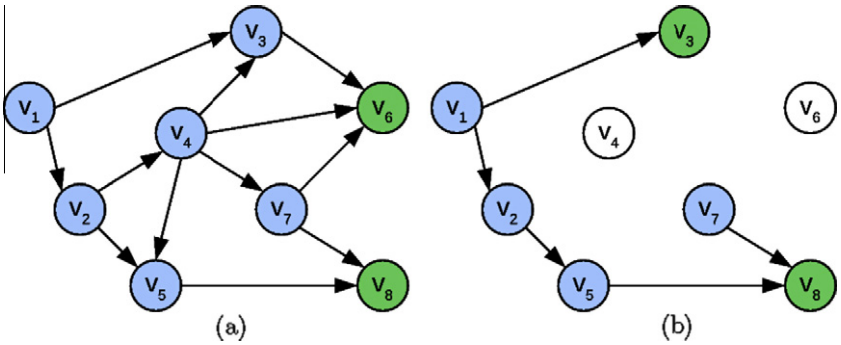

Fig. 2. Transition graphs for a pure strategies game. (a) Without constraints, (b) With constraints.

system metric we want to optimize) is the maximum slot utilization.

Theorem 3 (*Price of Anarchy for Bandwidth Trading*). *When user sessions are described as finite sequences of fixed size allocations, the PoA on the per-slot load is n.*

Proof. A loose bound on the PoA for the trading game is trivial: Given a maximum allocation per agent, X_{max}, it may be the case that all the n agents have an equally-large demand, and there exists a NE where these demands coincide in the same time slot. On the other hand, there is always going to be a slot with utilization of at least X_{max}, therefore this is a lower-bound on the slot utilization. Therefore we have the bounds:

$$X_{max} \leqslant \max\{U_p\} \leqslant nX_{max}.$$

These loose bounds immediately imply that

$$PoA \leqslant \frac{\text{worst-case}\max\{U_p\}}{\text{optimal}\ \max\{U_p\}} = n.$$

To show that this bound is tight, we present in Fig. 3 an instance that realizes it. In this example there are n agents, each one having a session of length $n + k$ n, and the total number of time-slots is $n + k \quad n + n - 1 = n(k + 2) - 1$. Fig. 3(a) shows the optimal allocation which yields a $\max\{U_p\} = X_{max}$, and Fig. 3(b) shows a NE whose $\max\{U_p\} = n\,X_{max}$. The second case is a NE because any unilateral deviation by any agent, gives an higher cost. In fact the agent cost at NE is

$$c_i = \frac{1}{C}\sum_p x_{ip}U_p = \frac{nX_{max}^2 + k}{C}.$$

And the cost for a agent if he moves any integral number of positions (within the allowed time-slots) is

$$c_i' = \frac{1}{C}\sum_p x_{ip}U_p = \frac{X_{max}^2 + 2k}{C}$$

and $c_i' > c_i$ whenever $k \geqslant (n-1)X_{max}^2$. □

It is important to note that realizing the above PoA bound requires a carefully crafted problem instance. In practice it is very unlikely to find instances with these characteristics. In fact, to evaluate the practical behavior of the PoA we conducted a series of simulations following the procedure below:

1. Create a problem instance whose optimal allocation is known. The load-balancing problem itself is NP-Complete.[10] On the other hand, constructing an instance with a known optimal solution is simple: Take the slots, assume they are all equally filled say with 1 unit. Split the content of each slot in several fractions and then take sequences of elements from different slots to be the tasks of the agents. Finally, shuffle around the tasks of the agents to get a problem instance.
2. For different number of agents (this defines the game size) and of time-slots we create multiple problem instances. In our case we created 100 instances for each game size.
3. Run the game by letting the agents take turns and play their best response until the game reaches a NE. Take the maximum among all the instances of the same size, and then compute the ratio with respect to the known optimum. This gives the empirical ratio of the worst-case to the optimal.

Fig. 4(a) shows the results of these simulations with 5 slots, and Fig. 4(b) show the results with 10 slots. In practice, the PoA for the trading phase (game) is almost always below 2, and tends to be insignificant as the number of agents (size of the game) increases, which bodes well for our setting.

4. The bandwidth capping phase

The Capping Phase computes a market-clearing solution that allocates the left-over budget of the agents in such a way that maximizes the aggregate FT allocation for each user. Let $W_i = (w_{i1}, \ldots, w_{iT})$ be the vector of FT allocations, where $w_{ip} \in \mathbb{R}^+$ is the allocation of FT for agent i in time-

[10] It is easy to see this by reduction to the 2-PARTITION [28] problem. If we had a polynomial algorithm that solves the load-balancing problem, we could run this algorithm on an instance of the partition problem with two slots. If the sum of the elements in the two slots is equal, the answer to the PARTITION problem is "yes", otherwise is "no".

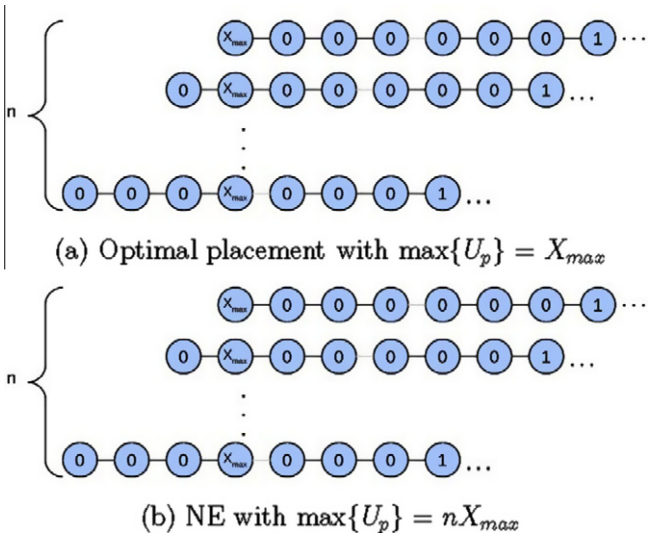

Fig. 3. Tight example for the PoA of the bandwidth trading game.

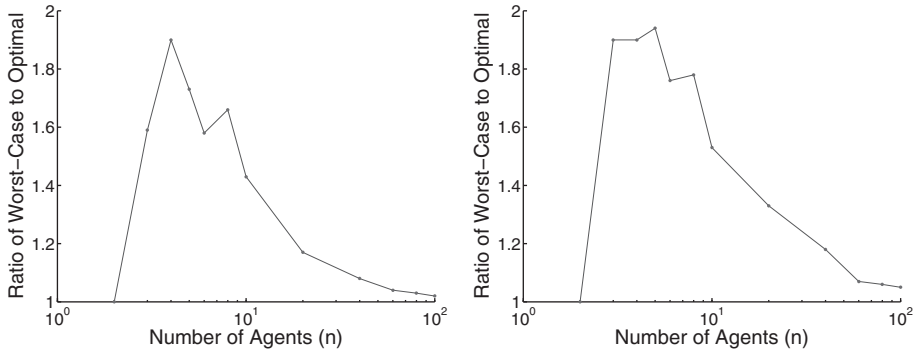

Fig. 4. Empirical price-of-anarchy based on synthetic worksets.

slot p. We adjust the definition of the cost function to take into account the allocation of FT as follows:

Definition 2. The cost to agent i for the combined allocation of RT (x_{ip}) and FT (w_{ip}) is

$$c_i(W_i) = \frac{1}{C}\sum_{p=1}^{T}(x_{ip} + w_{ip})U_p, \tag{4}$$

where $U_p = \sum_{i=1}^{n}(x_{ip} + w_{ip})$ is the aggregate reservation on slot p, and C is a constant.[11]

The implicit assumption of the Capping Phase is that RT allocations have priority, and are fixed once determined by the Trading Phase. FT allocations have no scheduling constraints: the value accrued by FT applications is strictly increasing with the aggregate allocation of FT bandwidth. Thus, self-interested agents select allocations so as to:

$$\text{Maximize} \quad b_i(W_i) = \sum_{p=1}^{T} w_{ip} \tag{5}$$

$$\text{subject to} \quad c_i(W_i) \leqslant B_i, \tag{6}$$

$$w_{ip} \geqslant 0 \quad \text{for } p = 1, \ldots, T. \tag{7}$$

A fundamental question that arises is the existence of an equilibrium solution for the FT marketplace. The following theorem shows that an equilibrium always exists.

Theorem 4. Existence of Nash-Equilibrium for FT Bandwidth Allocation *There exists a set of per-user allocation vectors that, when feasible for each user, maximizes the total per-user allocation and is a NE.*

In order to prove this theorem, we need first the following lemmas:

Lemma 1 (*Existence of the per-user solution*). *When the per-user FT maximization problem is feasible, there is a globally optimal solution (for a given set of allocations by the other agents).*

[11] The results in this section can be generalized for cost functions of the form $c_i(W_i) = \frac{1}{C}\sum_{p=1}^{T}(x_{ip} + w_{ip})f(U_p)$, where $f()$ is a continuous and twice differentiable convex function. See [29].

Proof. (Sketch) If the cost $c_i(W_i) < B_i$ when $w_{ij} = 0$, then there are feasible allocations of the fluid components w_{ij}. Notice also, that the feasible space defined as

$$\mathcal{D} = \{W_i \in \mathbb{R}^T | w_{ij} \geqslant 0 \quad \text{for } j = 1, \ldots, T \quad \text{and} \quad c_i(W_i) \leqslant B_i\}$$

is convex. This follows from the fact that the constraints of Eqs. (7) and (6) are concave functions. Then, by the Khun and Tucker (KT) theorem under convexity[12] there is vector W_i^* that maximizes the objective function $b_i()$ with associated Lagrange multipliers $\lambda_{iq}^*, \gamma_i^*$, such that the Kuhn-Tucker first order conditions

$$Db_i(W_i) + \sum_{q=1}^{T} \lambda_{iq}^* DW_i + \gamma_i^* Dc_i(W_i) = 0, \tag{8}$$

$$\lambda_{iq}^* \geqslant 0, \quad \gamma_i^* \geqslant 0, \quad \sum_{q=1}^{T} \lambda_{iq}^* w_{iq} + \gamma_i^* c_i(W_i) = 0 \tag{9}$$

are satisfied at $W_i = W_i^*$. □

The Lagrangean of the per-agent optimization problem is

$$L(W_i, \lambda_i, \gamma_i) = b_i(W_i) + \sum_{q=1}^{T} \lambda_{iq} w_{iq} + \gamma_i c_i(W_i)$$

and Eq. (8) can be succinctly written as

$$DL(W_i, \lambda_i, \gamma_i) = 0.$$

Lemma 2 (*Uniqueness of the per-user solution*). *The user's optimal solution is unique*

Proof. Suppose it is not. Let X and Y be two distinct global maximizers of $b_i()$. Let $Z = \alpha X + (1-\alpha)Y$ for $\alpha \in (0,1)$. By the convexity of $\mathcal{D}$, it is always the case that $Z \in \mathcal{D}$. By the linearity of $b_i()$

$$b_i(Z) = \alpha b_i(X) + (1-\alpha) b_i(Y) = b_i(X),$$

the second step because being X, Y global maximizers, it is the case that $b_i(X) = b_i(Y)$. This means that all the points Z in the hyperline segment defined by X, Y are also global maximizers.

Define the left-over budget at point Z as

$$\ell(Z) = B_i - \sum_p (x_{ip} + z_{ip}) U_p.$$

Then, $\ell(X) = \ell(Y) = 0$, otherwise if there is a positive left-over budget and the agent could increase its benefit and X, Y would not be maximizers. It is also the case that $\ell(Z)$ is strictly concave (this follows from $D^2\ell(Z)$ being a negative definite matrix), therefore

$$\ell(Z) > \alpha\ell(X) + (1-\alpha)\ell(Y).$$

This contradicts the previous observation that Z is a global maximizer, because whenever there is a positive left-over budget, the agent can increase the allocation in at least some time-slot thus increasing its total benefit. □

[12] See theorem 7.16 [30].

Proof of Theorem 4. Define the following global fluid maximization problem:

$$\text{Maximize} \quad \sum_{i=1}^{n} \sum_{p=1}^{T} w_{ip} \tag{10}$$

$$\text{subject to} \quad c_i(W_i) \leqslant B_i \quad \text{for } i = 1, \ldots, n, \tag{11}$$

$$w_{ip} \geqslant 0 \quad \text{for } i = 1, \ldots, n \quad \text{and} \quad p = 1, \ldots, T. \tag{12}$$

The Lagrangean of this problem is

$$L(W, \lambda, \gamma) = \sum_{i=1}^{n} \left(\sum_{p=1}^{T} w_{ip} + \sum_{p=1}^{T} \lambda_{ip} w_{ip} + \gamma_i c_i(W_i) \right), \tag{13}$$

where $W = (W_1, \ldots, W_n)$ is the concatenation of the per-user allocation vectors, and λ, γ are the concatenations of the per-user Lagrange multipliers. Observe that Eq. (13) is the sum of the corresponding Lagrangeans for the user problems, therefore a feasible W^* that maximizes (10), is also a global maximum for the per-user problems (all the terms in $DL(W, \lambda, \gamma) = \sum_{i=1}^{n} DL(W_i, \lambda_i, \gamma_i) = 0$ have to be zero, as none can be negative). Being the per-user allocations a global maximum, no agent can improve by unilaterally deviating from this allocation vector, hence W^* is a NE. □

5. Other application scenarios

Load balancing problems arise in a multitude of situations, of which the DSLAM scenario we have considered so far is but one example. The model we have presented is general and can be applied in other scenarios where the customer tasks can be modeled as a combination of atomic and fluid processes and all the customers compete to complete their tasks with the lowest cost.

An example setting in which T&C is applicable is given by Greenberg et al. [31] – namely provisioning datacenter resources. In this setting, resources such as energy and network capacity are typically priced according to the 95/5 rule. For a datacenter, this is a direct cost, making the incentive for the reduction of peak utilization more direct, but without changing the fundamental characteristics of the resource marketplace we have presented.[13] In particular, energy requirements of different tasks can be described as vectors of power consumption per time-slot, $T_i = (t_{i1}, \ldots, t_{il_i})$. Tasks may also be constrained to be executed within some time-interval and the charge associated with the execution of the tasks is determined by the total energy consumption according to Eq. (4). Accordingly, customers can schedule the execution of their tasks using the trading mechanism already described in Section 3. Similarly, there are fluid tasks that are able to use all the capacity made available to them, and which run forever. Examples of such fluid tasks are the crawling, indexing and ranking processes of web search engines. Such fluid tasks can be assigned a variable amount of resources per time-slot in such a way so as to maximize the total amount of work they are able

[13] In the DSLAM case with flat-rate payments, the incentive comes from exchanging flexibility for interactive applications with volume for fluid applications.

to achieve at the lowest cost. In addition, the possibility of assigning budgets to different tasks permits adjusting the fraction of the resources they get. In fact Greenberg et al. suggest using pricing and "urgency of execution" as parameters to reduce the peak-to-valley ratio on the utilization of these resources, which are precisely the notions captured by our mechanism.

6. Implementation of a T&C DSLAM marketplace

Architecture: We describe a distributed implementation of the T&C marketplace, where there is one provider agent (running at the DSLAM for example), and a client-side agent running on the customer's local router. The general architecture of the system is illustrated in Fig. 5. In this architecture, the client-side agent is responsible for: (1) profiling the customer's RT demand, (2) bidding for allocations during the bandwidth trading phase, and (3) shaping applications' traffic according to the reserved allocations. The provider-side agent provides two functionalities: (1) it runs the marketplace phases – bandwidth trading and bandwidth capping – just before the start of each epoch; and (2) once the epoch starts, enforces the allocations settled by the marketplace agents by using a traffic shaper for each customer line. The traffic shaper on the provider side enforces the total allocation determined by the T&C marketplace, but does not need to classify traffic, thus overhead is minimal.

The traffic shapers – both on the client-side and the provider-side – need not to be based on strict reservations. The drawback of a strict reservation system is that it does not take advantage of the statistical multiplexing between the flows sharing the link. To avoid this limitation, we use a work-conserving scheduler, namely a derivative of the hierarchical link-sharing scheduler [32] – the Hierarchical Token Bucket (HTB)–which is currently available in the Linux kernel [33]. When using a work conserving scheduler, if some of the sources are idle, the unused capacity is distributed between the other sources. As a consequence, the reservations established in the T&C marketplace are minimum guarantees, but the aggregate utilization can always reach the total reserved capacity.

Handling traffic on the customer side requires the implementation of a two-level priority queuing system as illustrated in Fig. 6, with the high priority assigned to RT demand and the low priority assigned to FT demand. This way, packets belonging to RT applications preempt any pending packets in the FT queue. The root traffic shaper ensures that the customer does not exceed its total allocated bandwidth.[14]

The routing of packets to each one of these queues could be implemented in a number of ways: (a) manual configuration on a per application basis, (b) using an automatic traffic classifier, or [6,7,9,11], (c) using special APIs that allow applications to bind to specific virtual interfaces. It is also worth mentioning that this system can be implemented on top a conventional QoS mechanism, *e.g.*, Diffserv, so that RT flows can be prioritized according to the application they belong to. This makes it possible, for example, for packets belonging to a VoIP call to receive priority over packets say of a HTTP request.

For accounting and policing purposes, the system would need to uniquely identify each customer. Authentication – in many cases already in place at the physical or link layers, depending on the underlying technology (*e.g.*, xDSL) – is needed to protect against "identity theft" whereby a customer would spoof the MAC address of another in the same DSLAM to avoid having its traffic counted against its own budget. Notice that to account for traffic during each epoch, the provider agent only needs the total allocation per customer. This information is enough to ensure that the customer is adhering to the outcomes of the T&C mechanism for each time slot. From the providers perspective it is irrelevant if the customer is using a bandwidth allotment for RT or FT bandwidth. In fact, this assures that the provider's policing mechanism is indeed *neutral* with regard to the customer's traffic.

Priority/weighted queueing systems have long been used in the QoS literature. An implicit assumption in that literature is that priorities/weights are assigned consistently by the end systems. However, when self-interested agents compete for the same resource, their choice would be to assign themselves the highest priority, unless there is a cost associated with this choice. Our T&C mechanism incorporates such a cost, thus providing the needed incentive for agents to act truthfully.

Algorithmic Complexity and Efficient Distributed Implementation: A scheme like ours would not be practical if associated processes are not efficient to compute.

To compute the best-response in the trading phase, we developed a dynamic programming solution which is pseudo-polynomial (complexity depends on the product of the number of sessions per user and the number of time slots) and which runs in a few seconds on current hardware for instances of practical sizes of hundreds of users and hundreds of time slots (108 and 288, respectively in our simulations). The dynamic programming solution for finding the best response for user i proceeds as follows:

1. Let k be number of sessions of agent i, and T the number of time-slots
2. Initialize the matrix A of dimension $k \times T$. Each element a_{jp} of A will represent the cumulative cost of sessions $1\ldots,j \leqslant k$, when the jth session is allocated in slot p. All the matrix elements are initialized to infinity.
3. The first row is computed by assuming session 1 is placed in slot p and computing the resulting cost.
4. Subsequent rows ($j = 2,\ldots,k$) are computed according to Eq. (14). Here, $c(j,p)$ represents the cost of session j at slot p (from Eq. (1)).

$$a_{jp} = \begin{cases} \infty & \text{if session } j \text{ is unfeasible at slot } p, \\ \min\{a_{j-1,1\ldots p-1}\} + c(j,p) & \text{otherwise.} \end{cases} \tag{14}$$

[14] A future extension would allow for a distributed implementation of the hierarchical scheduler, such that the capacity of the shared resource can be statistically multiplexed among the customers sharing the link. This way, the system would not have a per-customer cap due to the hard reservations.

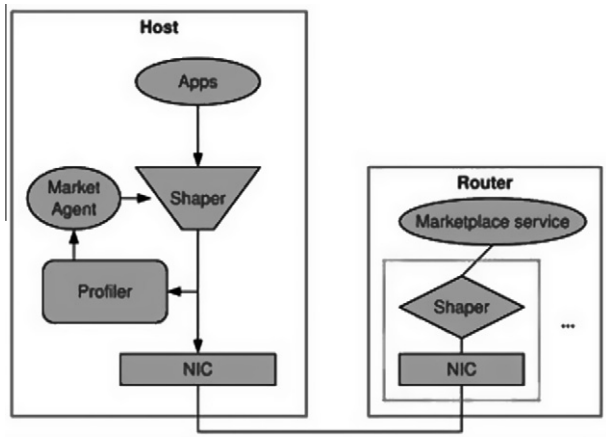

Fig. 5. Overall T&C architecture.

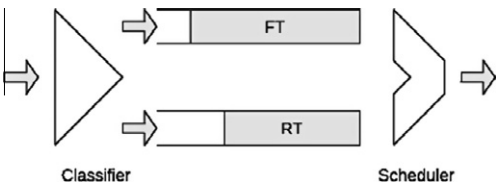

Fig. 6. Implementation using priority queues.

Observe also that a_{jp} is the minimum cost at which all the sessions up to j can be allocated in the time-slots up to p. Therefore, the minimum of the last row min $\{a_{k,1\ldots T}\}$ will give the optimal cost for the entire set of sessions of the user.

The feasibility condition in Eq. (14) refers to the constraints of the problem. Basically, different sessions do not overlap; all the components of the session fall within the allowed time-slots $(1,\ldots,T)$; and the cumulative cost is less or equal to the budget. In particular, in the case of a user exceeding its budget, we adopt the policy of dropping arbitrary sessions until the budget constraint is satisfied. In our experiments we did not implement the capacity constraint, although it could be easily incorporated into the procedure. In doing so, we allow for the utilization to grow as much as demanded, which gives even more conservative estimates of the worst-case performance metrics.

The entire system operates on the local domain defined by the DSLAM and the finite (customer) population attached to it. This is important as it ensures both, that the number of agents iterating is relatively small – the number of customers connected to a DSLAM typically in the order of a few hundred –, and that the RTT is small, which is important as the total time to clear the marketplace depends on the number of bidding rounds and the RTT. According to the results presented in the experimental evaluation, the bidding process for a population of about 100 users take around 600 bidding rounds, and assuming a RTT of 10*ms* (which is high for the local loop) this means that the equilibrium for the trading phase is obtained in just a few seconds. It is also worth mentioning that the messages to exchange are the bids (X_i vectors) and the current total allocations (U_p), which are both of dimensionality equal to the number of time slots. In our case, with 288 time-slots, and asuming 4-byte int values, the messages are just about 1*KB* in size.

As for the fluid allocation computation in the capping phase, the solution using Lagrange multipliers presented in Section 4 constitutes a straightforward efficient distributed implementation, whereby at the Customer Premises Equipment (CPE) each agent computes its best response iteratively until it gets close enough to the global optimum.

Running both the trading and capping processes at the CPE is consistent with a network-neutral implementation. The only support needed from the DSLAM would be to offer a blackboard service where all the participants are able to register their (RT and FT) allocations and query the totals (U_p) per time-slot. Once the market reaches an equilibrium, the posted schedule is committed for the next epoch.

7. Experimental evaluation

In this section we use trace-driven simulations to (1) highlight the benefits that a user in our system begets by exhibiting some flexibility in scheduling its RT sessions under T&C, (2) demonstrate the gains that an ISP stands to realize as a result of the overall smoother traffic profile of T&C, and (3) illustrate how various parameters affect the performance of T&C.

Traces and Trace Pre-Processing: As an alternative to direct DSLAM traces (which unfortunately are not available), we used publicly available WAN traces [34] to extract a slice of traffic associated with a customer access network. Table 1 shows the main characteristics of these WAN traces. Capturing a slice (portion) of the customer network's traffic results in less pronounced diurnal peak-to-valley ratios, which limits the performance gains realized by T&C. It is also like likely that the traces already include the effects of some traffic shaping which also reduces the peak-to-valley ratio. For both reasons, the performance gains reported in this section should be viewed as "conservative". Fig. 7 shows the traffic aggregated over 5 min

Table 1
Characteristics of the WAN trace used in our evaluation.

Period	2009-03-31 00:00 – 2009-03-31 23:59
Total packets	1,551,089,845
TCP packets	1,194,409,653
UDP packets	4,321,852
Total TCP bytes (payload)	924,540,189,060

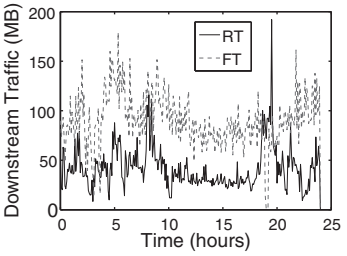

Fig. 7. Downstream trace for a subnet of broadband users.

time-slots for the subnetwork we selected for our evaluation.

To extract traffic associated with a customer access network, we applied the following pre-processing steps. First, we identified subnets most likely associated with broadband users, based on the upstream/downstream ratios, the activity per port number, and diurnal activity patterns. Next, assuming that each IP address is a single user/household, we classified the traffic per user as either RT or FT. This was done based on association of traffic activity with privileged port numbers. Finally, we identified the various RT sessions per user, with their corresponding demands per time-slot. Session identification was done by setting a threshold on the length of periods of high activity. We call this threshold S_{max} and it is given as a number of time-slots. For most of our experiments we considered the values $S_{max} = 6$ and $S_{max} = 12$ corresponding to half an hour and one hour respectively. If any sequence of time-slots has length greater than S_{max}, then we subtracted the minimum from this interval under the assumption that it was due FT. By repeating this process on any subinterval of length greater than S_{max} we obtained a set of disjoint RT sessions for the user.

T&C operates by letting user agents express their flexibility or willingness to move IT components (forward or backward in time) some number of time slots. We define a session's *slack* to be the number of time slots that an agent is willing to shift its session (back or forth in time). A slack of 0 implies no flexibility. A slack of 1 implies a willingness to shift sessions by 5 min (our time slot) back or forth, if such a shift is advantageous. Notice that *moving a session* means a shift of the traffic attributed to that session for *all* time slots spanned by that session (*i.e.*, traffic in all time slots of a single session is shifted equally to preserve session atomicity). In our simulation we also enforced the condition that no shifting sessions could overlap. This is consistent with users not doing more activities on the same time-slot. Similarly, we also enforced the condition of preserving the session ordering. Although not required by our model, it implies less effort on the part of the agent, and any results thus obtained are even more conservative.

How does T&C impact the ISP's bottom line?: Our first experiment aims to evaluate how the 95th percentile of the ISP's 5 min traffic volume (the 95% traffic envelop) changes as a result of letting users schedule their RT sessions according to the trading phase of T&C. For brevity, we assume that all agents adopt the same *slack* value for all their sessions. Fig. 8 shows two examples of the outcome after the market reaches an equilibrium. On the left is the traffic per time-slot, and on the right is the CDF of traffic per time-slot. Top row is for session length threshold of $S_{max} = 6$, and the bottom row is for $S_{max} = 12$ time-slots. Clearly, the session thresholding process has little effect on the trace, being the most noticeable effect the larger peak (from 130 MB to 150 MB). Table 2 shows the values of the 95% traffic envelop. These results underscore that selfishly scheduling RT sessions yields an equilibrium with *significant* reduction in the 95% traffic envelop – up to 31% reduction when slack is 1 h. Even for a small slack of 15 min, the savings amount to 16%.

We emphasize that the benefit from bandwidth trading quantified in the results in Table 2 (and elsewhere in this paper) is rather conservative given the nature of the WAN traces used in our evaluation, in which the peak-to-valley ratio is much lower than those observed in most characterization studies, *e.g.*, [35]. With workloads exhibiting typical variability, the benefits are likely to be even more significant.

We now consider experiments in which both phases of T&C are carried out. In particular, after completing the trading phase – thus scheduling all RT sessions in the trace – agents allocate as much fluid traffic as possible in accordance with their remaining budgets. Thus, an important consideration in setting-up these experiments is the budget assignment. In particular, we used the following policy: Let V denote the nominal traffic per time-slot that results in a total volume equal to the total traffic originally in the trace. We introduce a control parameter R (for resistance) which allows the provider to adjust the resulting traffic on the shared link. By setting $C = V/R$ (this is the C of the cost function in Eq. (4)), and the budget per customer to $B_i = CT/n$, the expected utilization (without RT) is precisely C. In our traces (as observed generally on the Internet) the FT component is much larger than the RT component, therefore the RT stage is rarely affected by the budget constraint.[15]

Fig. 9 shows the outcome of the two phases of T&C for a value of $R = 1.0$ and various slack values. The y-axis is normalized with respect to V (the nominal volume under perfectly balanced conditions, with no RT components). Due to the presence of RT components, this quantity is always (slightly) larger than 1.0. The session identification process also capture a much larger peak in the case of $S_{max} = 12$. Table 3 shows the 95% and 50% (median) of the time-slot utilizations, as well as the ratio between them. These results

[15] For large values of R, the budget constraint may impact RT allocations. In the rare event when this happens, the policy we adopted was to randomly drop user sessions in case the user runs out of budget in the trading phase.

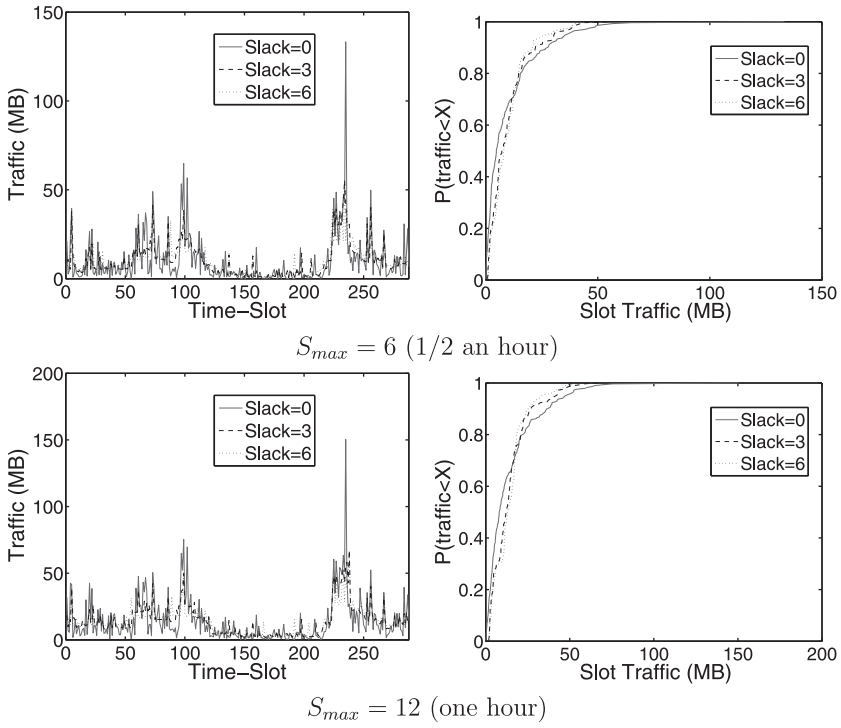

Fig. 8. Utilization over time for RT sessions with various slack values.

Table 2
95% utilization resulting from bandwidth trading.

Slack	S_{max} = 6		S_{max} = 12	
	95% (MB)	Savings (%)	95% (MB)	Savings (%)
0	36.3	0.0	47.7	0.0
3	30.6	15.6	42.1	11.7
6	27.4	24.4	33.6	29.6
12	24.9	31.4	30.9	35.2

suggest that with T&C in place, the ratio is nearly 1.0, resulting in a perfect flattening of traffic over time slots, thus eliminating cost problems derived from spikes when using the 95/5 rule.

How does T&C enable an ISP to cap its aggregate traffic volume? The ISP is able to specify a target total traffic volume on the managed link through its choice of the resistance parameter R (which directly affects the constant C and hence the budget B_i allocated to each agent). Fig. 10(a) shows the total allocation per time-slot as a function of R, when *slack* = 0 (which is the worst-case in the sense that under this scenario, the budgets are constrained the most). As expected, R effectively controls the aggregate traffic volume resulting from T&C. This volume is almost flat due to the "fluid" nature of FT bandwidth allocation. The exception is due to spikes underscoring the presence of large RT sessions that could not be smoothed out under the chosen slack value. Naturally, these spikes dissipate when larger slack values are used (see Fig. 9).

How does ISP resistance impact the allocation of FT traffic relative to RT traffic? Fig. 10(b) compares the per-user

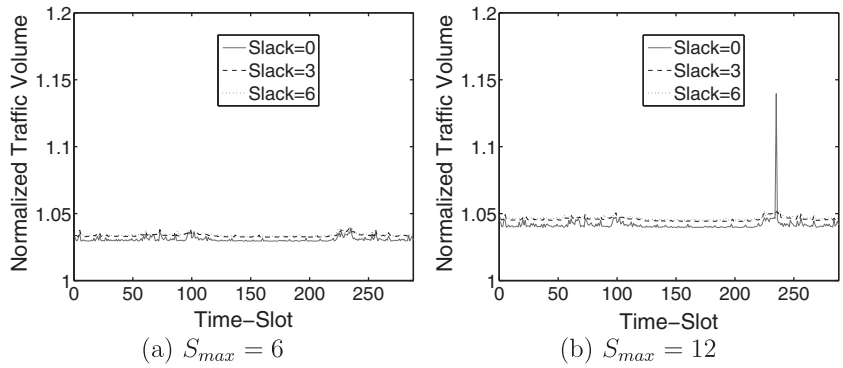

Fig. 9. Total Traffic (RT + FT) for various slack values.

Table 3
Traffic volume statistics (in MB) with and without T&C.

		95%	Median	Ratio
Original		197.15	124.56	1.583
T&C	$S_{max} = 6$	136.52	135.93	1.004
T&C	$S_{max} = 12$	138.05	137.33	1.005

Table 4
FT bandwidth gain for various values of R and RT demand.

Slack	$R = 4.0$ 100 MB	$R = 2.0$ 200 MB	$R = 1.0$ 400 MB
3	1.3190	1.2836	1.1931
6	1.3497	1.3338	1.2329
12	1.4079	1.3769	1.2520

bandwidth allocations for different values of the resistance, R. As before, the general trend is that the more RT bandwidth requested by an agent during the trading phase, the less FT allocation the agent is able to secure during the capping phase. Increasing the values of R results in a corresponding reduction in the aggregate allocation of FT bandwidth, with large RT bandwidth consumers impacted the most.

How does T&C impact the user's bottom line? To evaluate T&C on a per-user basis, we compare how RT and FT allocations vary across users. Fig. 11(a) shows a clear negative corelation between the allotment of FT and RT bandwidth. The relationship is not monotonic or deterministic because it depends on the outcomes from the trading phase, which affect the left-over budget for each agent. It is always the case though that the larger the slack, the larger the FT allocation for any given user (points along the same vertical line in the plot). An agent with fixed RT demand increases its allocation of FT bandwidth when it adds more flexibility to its RT sessions. The results in Table 4 expose this trade-off for selected levels of RT demand and resistances. For example, when $R = 4$, an agent with a nominal 100 MB of RT bandwidth is able to capture 32% more FT traffic by accepting a minimal slack of 3 for its RT sessions. A rather surprising (and also desirable) finding – evident from Fig. 11 and Table 4 – is that the user begets *most* of the benefit by introducing a minimal amount of slack. Increasing the slack much beyond that results in only marginal increases in FT allocation. In the above example, by doubling its slack from 3 to 6, the user is able to capture only 3% more FT traffic. The message is clear: it "pays" to be flexible, even if minimally so.

Fig. 11(b) shows the same results on a semi-log scale to expose the outcome for users with negligible demand for RT bandwidth. In this case, the capping phase assigns to all such users almost an equal share of the capacity (as expected). It is only the heavy RT bandwidth hogs who are unable to claim much FT bandwidth, which is precisely the premise of T&C.

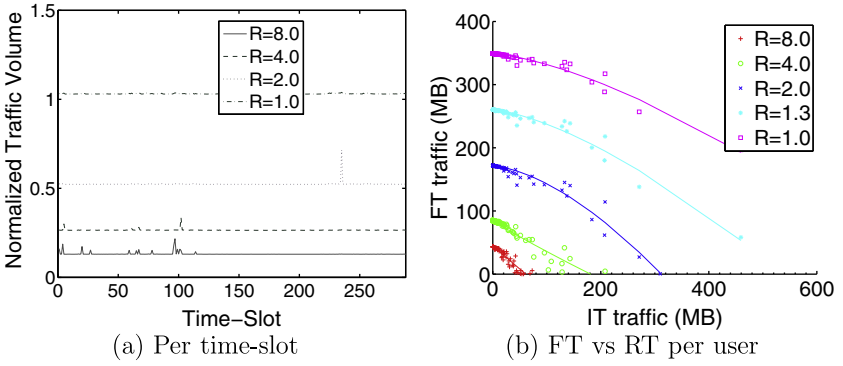

Fig. 10. Traffic allocations for variable R.

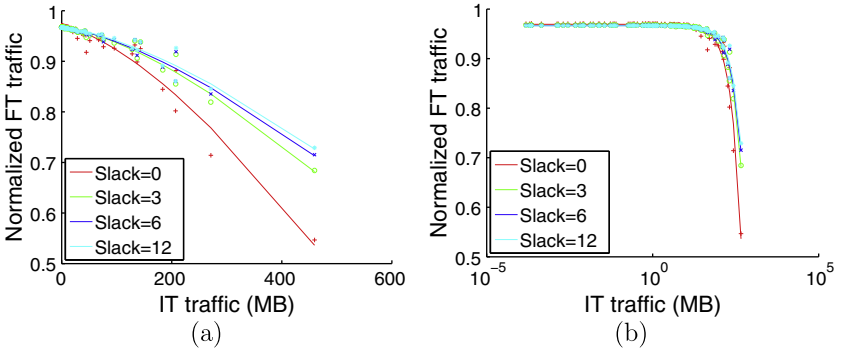

Fig. 11. RT and FT allocations per user for different slack values.

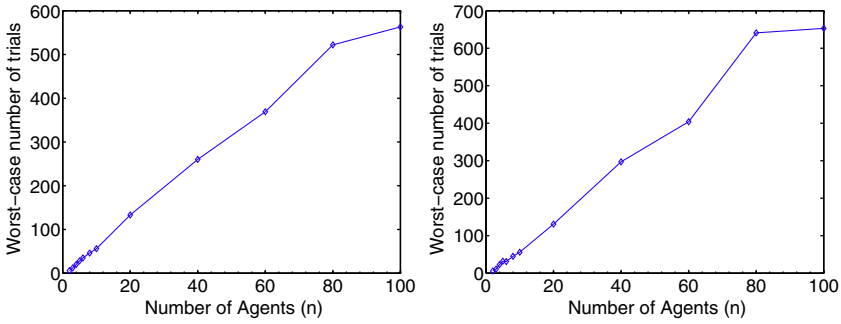

Fig. 12. Number of trials until convergence. Left – 5 time-slots, Right – 10 time-slots.

Convergence to Equilibrium and Scalability: Earlier, in Section 6, we discussed the algorithmic complexity of computing best-responses during the trading phase, and gave a pseudo-polynomial algorithm for its solution. It was also shown that the computation of market bids for the capping phase is polynomial. To evaluate the convergence speed, and how well the marketplace scaled for large numbers of participating agents, we conducted a series of simulations where we vary the number of agents and register the number of trials until the market reaches equilibrium. A trial is a single agent iteration and includes both, computing and submitting the bid to the marketplace. For a given number of agents n, we conducted 100 different experiments and counted the aggregate number of trials of all the agents. The worst-case among those 100 experiments is registered in Fig. 12. The experiments show that the total number of trials until convergence follows a linear trend, and timing measurements show that the market-clearing allocations can be computed in less than a few seconds (less than 10 s in a 2.4 Ghz P4 system).

8. Related work

While the application of game-theoretic and micro-economic approaches to networking problems is not novel [36,27,4,37,21], our approach of strategically trading-off allocation slots based on desirable properties for different traffic classes is new and quite promising.

Laoutaris and Rodriguez [5] recognized that the problems associated with rampant FT traffic are due to the lack of incentives for end-users to properly schedule their FT traffic and the lack of network mechanisms to identify and handle such traffic. As a solution to the first problem, they suggest giving users "higher-than-purchased" access rate during off-peak hours as a reward for time-shifting their FT traffic. As a solution to the second problem, they propose the introduction of a store-and-forward service to handle the network transfer of bulk FT data during off-peak hours.

Fairness is a controversial issue with no universally-accepted definition. The most commonly used definition is that of *max–min fairness*, whereby no user can increase its rate at the expense of other users with lower rates. Max–min fairness deals with instantaneous rates, and thus is useless over long time scales under time-varying demands. In many contexts, fairness is a property established across flows (*e.g.*, TCP's max–min fairness). Clearly, this definition breaks when a single entity (user) is able to open multiple concurrent flows, as it is indeed the case in many applications. Briscoe [38] gives a very thorough discussion of the issues involved. He advocates a notion of *cost fairness* between economic entities, thus avoiding both the per-flow and the instantaneous connotations. This is consistent with T&C's assignment of budgets to user agents as the primary mean for ensuring fairness.

Recently, Briscoe et al. [23] proposed an architecture that operates at the network edges and realizes the *cost fairness* model without directly charging users (hence, compatible with flat pricing). This work introduces *re-feedback*, a mechanism that allows measurement of downstream path metrics, such as delay and congestion. This information can then be used to police the compliance of end-users with a predetermined policy (*e.g.* backoff the sending rate in case of congestion). The network itself can perform the policing function requiring only a shaper at the ingress point and a dropper at the egress point. When doing so, it is the dominant strategy for end-points to report the correct metrics. Re-feedback operates as a congestion control mechanism. It provides the feedback needed by both, the flows and the network. Flows use this feedback to adjust their rates. The network uses the feedback to police the end-points response to congestion. Re-feedback is a best-effort scheme, and unlike T&C it does not provide the means for applications with specific QoS goals to make trade-offs that help them satisfy their requirements.

Approaches for *congestion-pricing* with explicit payments have been considered in a number of studies. Henderson et al. [22] present a review of the benefits and limitations of these proposals. Examples include *Smart Markets* [37,20] and *Split-Edge Pricing* [39]. Of particular interest is the scheme proposed by Ganesh et al. [27], which assigns costs to packets depending on congestion. Under a family of non-linear cost functions that depend on the utilization of the congested link and the flow's demand, they showed convergence to steady-state equilibrium. While our mechanism and system model are entirely different, our cost function has similar characteristics.

Several works [21,40,41] have considered priority queueing systems (a la Diffserv) under game-theoretic frameworks. Marback [21] analyzes a priority queueing

scheme where packets get charged based on their priority, and selfish users compete for bandwidth. Among other things, he shows that such a scheme leads to a Wardrop equilibrium and that allocation does not depend on the prices of each traffic class. A fundamental distinction in this case is that T&C enables different valuations for different classes of traffic, and uses these valuations to leverage the trading system. Park et al. [40] consider a QoS class assignment game where users share a single Generalized Processor Sharing (GPS) queue and they can assign the class for the traffic. Users do so, to meet the QoS requirements of their application at the minimum possible costs (as higher priority also means higher cost). In this work, they consider both, the case where traffic may be arbitrarily split between the many service classes and the *unsplittable* case where all the traffic is assigned to the same class. In the splittable case, NE need not exists, but it is proven that in the unsplittable case NE always exists. In [41], the authors consider the assignment of service classes to each user's traffic at each one of the routers in a path. In this analysis, each user provides a QoS vector and a utility function, and the user actions are the choices of service classes at each router, such that his traffic will meet the QoS goals with minimum cost. This model is limited to the *unsplittable case*, meaning that all the traffic from a user is assigned the same service class. The incentive for the users is implicit in the price-by-class scheme, where users requesting higher priority classes pay more. In addition, payment has to be made to all intermediary nodes on a route. Chen et al.[42] also provide an efficient distributed implementation and evaluation of their multi-switch QoS assignment game, where agents running at the routers and end-points compute the game outcome on behalf of the users. The performance evaluation shows a significant improvement on the per-application QoS metrics with respect to a static reservation mechanism.

A fundamental distinction between T&C and the various congestion pricing schemes considered in the literature ([23,43,22,27]) is that none of these schemes takes into account the dual nature (RT vs. FT) of applications. Therefore, all these schemes impose penalties (e.g. larger cost, increased drop rates) to *all* the traffic from a user during congestion periods. Because they operate over short-timescales (targeting an instantaneous response to congestion), none of these approaches exploits the extra degree of freedom offered by the possibility of time-shifting the execution of RT tasks, or adjusting the rate of FT tasks.

9. Conclusion

Trade & Cap is an effective bandwidth management mechanism that enables self-interested user agents to collectively converge on what *they* perceive to be an equitable allocation, based on their individual, private valuation of network utility (*e.g.*, raw volume vs. QoS over time). T&C not only benefits users by allowing them to extract better utility from the network, but also benefits the ISP by yielding smoother aggregate traffic volumes, which lowers traffic transit costs and reduces the currently unsustainable pressure on ISPs to upgrade their networks in order to keep up with peak demand. Under T&C, rather than acting as an arbiter, an ISP acts as an enforcer of what the community of rational users sharing the resource decides is a fair allocation of that resource. This is a welcome departure from current practices that force ISPs to use artificial notions of fairness to police shared bandwidth use, with negative implications to privacy and network neutrality.

References

[1] M.H. Bosworth, Time warner: metered broadband will prevent "internet brownouts", 2009.
[2] W. Gruener, Time warner shelves metered internet plans – for now, 2009.
[3] A. Odlyzko, Internet pricing and the history of communications, Comput. Networks 36 (2001) 493–517.
[4] F. Kelly, Charging and rate control for elastic traffic, Eur. Trans. Telecommun. 8 (1997) 33–37.
[5] N. Laoutaris, P. Rodriguez, Good things come to those who (can) wait or how to handle delay tolerant traffic and make peace on the Internet, in: HotNets'08.
[6] L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, CoNEXT '06: Proceedings of the 2006 ACM CoNEXT Conference, ACM, New York, NY, USA, 2006, pp. 1–12.
[7] T. Karagiannis, D. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, SIGCOMM '05: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, New York, NY, USA, 2005, pp. 229–240.
[8] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, Transport layer identification of P2P traffic, Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC), ACM, New York, NY, USA, 2004, pp. 121–134.
[9] A.W. Moore, D. Zuev, Internet traffic classification using bayesian analysis techniques, SIGMETRICS, ACM, New York, NY, USA, 2005, pp. 50–60.
[10] A.W. Moore, K. Papagiannaki, Toward the accurate identification of network applications, in: Passive and Active Network Measurement (PAM'05), pp. 41–54.
[11] S. Sen, O. Spatscheck, D. Wang, Accurate, scalable in-network identification of P2P traffic using application signatures, WWW '04: Proceedings of the 13th International Conference on World Wide Web, ACM, New York, NY, USA, 2004, pp. 512–521.
[12] L.G. Roberts, A radical new router, IEEE Spectr. 46 (2009) 34–39.
[13] F5 Networks, Inc., Bandwidth management for P2P applications, 2009.
[14] iPoque, Bandwidth management with deep packet inspection, 2009.
[15] E. Orion, Comcast internet throttling is up and running, 2009.
[16] N. Anderson, DPI vendor says 90% of ISP customers engage in traffic discrimination, 2009.
[17] J. Crowcroft, Net neutrality: the technical side of the debate: a white paper, SIGCOMM Comput. Commun. Rev. 37 (2007) 49–56.
[18] N. Anderson, Network neutrality in congress, round 3: fight, 2009.
[19] A. Odlyzko, Pricing and architecture of the internet: historical perspectives from telecommunications and transportation, in: Proceedings of TPRC.
[20] J. MacKie-Mason, H. Varian, Pricing congestible network resources, IEEE J. Sel. Areas Commun. 13 (1995) 1141–1149.
[21] P. Marbach, Analysis of a static pricing scheme for priority services, IEEE/ACM Trans. Networks 12 (2004) 312–325.
[22] T. Henderson, J. Crowcroft, S. Bhatti, Congestion pricing. Paying your way in communication networks, IEEE Internet Comput. 5 (2001) 85–89.
[23] B. Briscoe, A. Jacquet, C.D. Cairano-Gilfedder, A. Salvatori, A. Soppera, M. Koyabe, Policing congestion response in an internetwork using re-feedback, SIGCOMM Computer Communication Review, vol. 35, ACM Press, 2005, pp. 277–288.
[24] Wikipedia, Emissions trading, 2010.
[25] The Canadian Press, Most canadians support reasonable internet traffic management, poll suggests, 2009.
[26] D.D. Clark, J. Wroclawski, K.R. Sollins, R. Braden, Tussle in cyberspace: defining tomorrow's internet, SIGCOMM, ACM, New York, NY, USA, 2002, pp. 347–356.
[27] A. Ganesh, K. Laevens, R. Steinberg, Congestion pricing and user adaptation, in: Proceedings IEEE INFOCOM, pp. 959–965.

[28] M. Garey, D. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W.H. Freeman and Co., San Francisco, CA, 1979.
[29] J. Londoño, Embedding Games: Distributed Resource Management with Selfish Users, Ph.D thesis, Boston University, Boston, MA, 2010.
[30] R.K. Sundaram, A First Course in Optimization Theory, Cambridge University Press, 1996.
[31] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The cost of a cloud: research problems in data center networks, CCR Online (2009).
[32] S. Floyd, V. Jacobson, Link-sharing and resource management models for packet networks, IEEE/ACM Trans. Networks 3 (1995) 365–386.
[33] M. Devera, HTB Home, Web page, 2003.
[34] MAWI Working Group, Traffic archive, 2009.
[35] N. Laoutaris, G. Smaragdakis, P. Rodriguez, R. Sundaram, Delay tolerant bulk data transfers on the internet, SIGMETRICS, ACM, New York, NY, USA, 2009, pp. 229–238.
[36] J. Feigenbaum, C. Papadimitriou, R. Sami, S. Shenker, A BGP-based mechanism for lowest-cost routing, Distrib. Comput. 18 (2005) 61–72.
[37] J. MacKie-Mason, H. Varian, Public Access to the Internet, MIT Press.
[38] B. Briscoe, Flow rate fairness: dismantling a religion, SIGCOMM Comput. Commun. Rev. 37 (2007) 63–74.
[39] B. Briscoe, The direction of value flow in connectionless networks, in: Networked Group Communication, pp. 244–269.
[40] K. Park, M. Sitharam, S. Chen, Quality of service provision in noncooperative networks: heterogenous preferences, multi-dimensional QoS vectors, and burstiness, ICE '98: Proceedings of the First International Conference on Information and Computation Economies, ACM, New York, NY, USA, 1998, pp. 111–127.
[41] S. Chen, K. Park, An architecture for noncooperative QoS provision in many-switch systems, in: Proceedings IEEE INFOCOM, vol. 2, pp. 864–872.
[42] S. Chen, K. Park, A distributed protocol for multi-class QoS provision in noncooperative many-switch systems, in: Proceedings of the Sixth International Conference on Network Protocols, pp. 98–107.
[43] F.P. Kelly, A. Maulloo, D. Tan, Rate control in communication networks: shadow prices, proportional fairness and stability, J. Oper. Res. Soc. 49 (1998) 237–252.

Jorge Londoño recently obtained his Ph.D. degree from the Computer Science Department at Boston University, Boston, MA. He obtained a MA from Boston University in 1999 and a BS at the Universidad Pontificia Bolivariana in 1992. His research interests include distributed systems, and applications of game-theory and micro-economics to resource management problems in these systems.

Azer Bestavros is professor and former chairman of Computer Science at Boston University, which he joined in 1991 after completing his Ph.D. at Harvard University. He is the Chair of the IEEE Computer Society Technical Committee on the Internet and a distinguished speaker of the IEEE. He has received distinguished service awards from both the ACM and the IEEE, and was the recipient of the United Methodist Scholar/Teacher Award in 2010. Funded by more than $15 M of government and industry grants, his research is mostly applied to networking and real-time systems, culminating so far in 14 Ph.D. theses, two startup companies, and over 3,000 citations. Some of his early groundbreaking contributions include stochastic extensions of classical rate-monotonic analysis for real-time systems in the early 1990s, pioneering the CDN push content distribution model in the mid 1990s, and seminal Internet traffic characterization and reference locality modeling in the late 1990s. His more recent research has focused on network transport, caching, and streaming media delivery, adversarial exploits of system dynamics, economics-inspired approaches to resource management in overlay, P2P, and cloud settings, and formal approaches to the design and implementation of safety–critical cyber-physical systems.

Nikolaos Laoutaris is a researcher at the Internet research group of Telefonica Research in Barcelona. Prior to joining the Barcelona lab he was a postdoc fellow at Harvard University and a Marie Curie postdoc fellow at Boston University. He got his Ph.D. in computer science from the University of Athens in 2004.