Service Level Agreement Trading for the Differentiated Services Architecture

George Fankhauser David Schweikert Bernhard Plattner
{gfa,dws,plattner}@tik.ee.ethz.ch
Computer Engineering and Networks Lab
Swiss Federal Institute of Technology, Zürich, Switzerland

Abstract— The differentiated services (DS) architecture provides a framework for the scalable provisioning of multiple service levels in the Internet. Its definition and initial work have concentrated mainly on per-hop behaviors (PHB) and mechanisms at each DS domain. Equally important is what happens *between DS domains*. Thanks to the flexibility of the architecture, it is the providers' choice how to interconnect with peers.

In principle, traditional, static peering agreements work together with DS, but they do not offer the flexibility and dynamics needed in an electronic market for network capacity. Therefore, we look at dynamic service level agreements at the interdomain level. Such agreements are established by software entities called *traders*. These traders follow market-based principles to decide which contracts will be beneficial. In particular, traders compare the offers made by neighbor providers and select the most interesting ones. This selection of peer services creates competition among providers and integrates route selection based on service level *and* destination.

We describe and implement a framework for service level agreement trading. We show the basic workings and first performance results of SLA trading using specific traders and an experimentally defined PHB in a simulation environment.

Keywords— Network architecture, differentiated services architecture, service level agreement, pricing, trading, interdomain QoS routing.

I. Introduction

The Internet was designed with connectivity and robustness in mind. It was primarily targeted towards data transport. Later, it was tried to transport media streams over this *best-effort service*. Although this lead to fruitful work in application adaptivity, enhancements to the Internet's service architecture were proposed. One approach, *integrated services (intserv)* [1], is an architecture to enhance the Internet service with *per-flow reservations*. Unlike circuit-switched telecommunication systems, intserv uses the same network and transport layer as the best-effort Internet. To implement resource reservation and the associated signaling, corresponding protocols have been developed and standardized, most notably RSVP [2].

While intserv and RSVP were adopted by application programmers and some access providers, the architecture was thought of being too complex and too resource hungry when used across transit networks serving millions of flows. To remedy this, i.e. to provide a globally scalable service architecture, a simple framework, called *differentiated services (DS)*, was proposed within the IETF [3].

In a nutshell, DS builds on the classification of packets by a short mark in every IP-packet. According to that mark a service is implemented at each network. Unlike per-flow scheduling of packets, DS provides different service levels for classes. This results in the aggregation of flows which is fundamental to scalability in the backbone.

A. Diffserv Basics

Important to the DS concept is the marking of packets at the edge of the network according to a customer profile. Such a profile defines which amount of packets belongs to what service class and level. Packets that exceed a specified contract are detected by meters and marked as *out of profile*. Out of profile packets will be treated by the Internet Service Providers (ISP) differently, for example as best-effort. The edge of the network is also a convenient point to apply charging and accounting mechanisms efficiently.

The DS architecture [3] currently describes local network behavior known as *per-hop behaviors (PHB)*. This includes mechanisms such as the use of packet schedulers, classifiers, DS code point (DSCP, i.e. the short marks) definitions, and traffic conditioners (meters, markers, droppers, and shapers). PHBs describe the externally observable forwarding behavior of packets which is provided for a class of traffic. The architecture defines several PHBs providing different levels of service for a variety of applications.

One of the defined PHBs is the *expedited forwarding (EF)* service (formerly known as premium service) [4]. It provides a high probability of available bandwidth for packets being classified as EF. This is achieved by exchanging precise *Service Level Agreements (SLA)* definitions between provider networks including information about the traffic's destination. The main application of EF

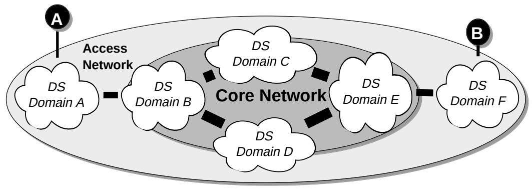


Fig. 1. ISP networks at access and core network level.

are virtual leased lines across the Internet (e.g. to build virtual private networks).

Another PHB, assured forwarding (AF), defines basically a priority forwarding service based on traffic classes and drop precedences [5]. Other defined PHBs cover best-effort traffic (default PHB) and network control traffic.

One of the key points in a working DS network is the interaction between DS domains. [1] Technically, this includes a means of communicating SLAs between ISP including the SLA definition itself, and an appropriate protocol. This aspect of DS, the signaling of SLAs, is still in its early stages (wrt. to the DS framework) and the focus of this paper.

B. *Network Architecture and Service Level Agreements*

Figure 1 shows the basic division of the network into a core part (backbones, transit networks) and access networks connecting the individual users. DS based solutions, as we discuss them here, focus on the core network and assume aggregated traffic flowing to and from the access networks. The access networks form the edge of the network.

While we could assume that SLAs are rather static and handled manually by operators at providers, the idea of an *automatic exchange* according to the network load is very appealing. Bandwidth brokers (BB) [6] have been proposed to a implement such an automatic exchange of SLA information. In our context, we call such entities *SLA Traders* (or simply *SLAT*) [2].

In contrast to BBs, SLATs provide a generic means for exchanging information needed for negotiating and trading services among DS, such as inter-domain path-selection, pricing and cost information. They implement a general approach to exchanging SLA information which includes:

- Different *service classes* described by an arbitrary QoS vector or predefined DS code points which refer to PHBs.
- *Source and destination* of traffic (scope of the traffic).
- *Cost* and contract *duration*.

[1] A DS domain is “a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions.” [3]

[2] Although similar to the bandwidth broker concept we thought the term was too limiting.

The motivation to follow and extend the trading approach is manifold:

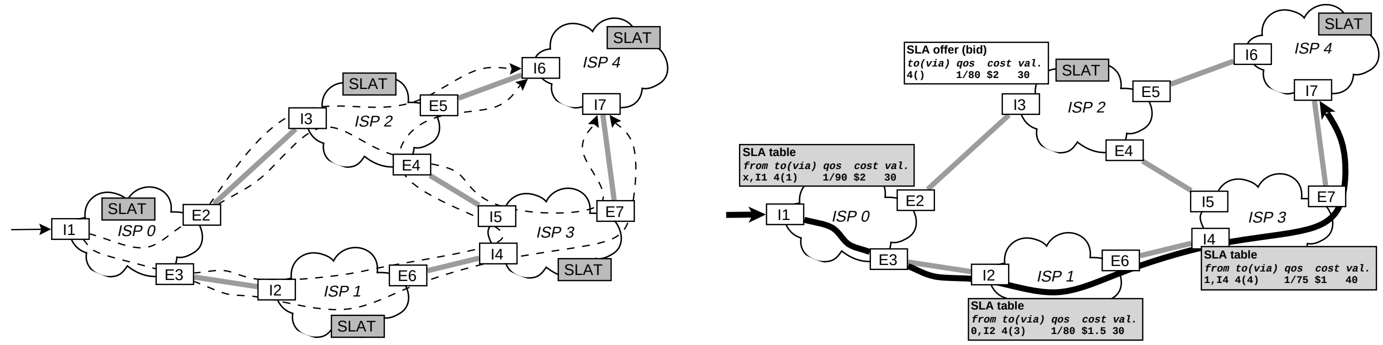
- Automation of SLA exchange means *faster service availability* and convergence compared to today’s SLA negotiation process.
- Bilateral trading promises to allow for *local optimization* and *incremental deployment*.
- *Cost and pricing* aspects can be integrated (e.g. cost-based optimization, profitability checking)
- An efficient electronic market for network resources is supported by the SLAT approach. In fact *Path selection* by the SLAT is the basis for *competition* among ISPs.

Although one could think of deploying SLAT on a global scale and even replace parts of the existing infrastructure (e.g. inter-domain routing) we need to keep in mind that incremental deployment is the wiser approach. Also, best-effort traffic will still make up a large portion of the Internet’s traffic. Furthermore, we will be able to relax some conditions on the SLA traders and protocols.

It is noteworthy that extending and automating the process of establishing SLAs with routing information coincides very well with the goals of QoS routing in the Internet. As RFC 2386 states, QoS routing should not only happen inside autonomous systems (AS) but also between ASs. The objective is to “Encourage simple, consistent and stable interactions between ASs...” [7]. Intra-domain resource allocation and routing is a well researched topic [8], [9], [10], [11], [12], [13], [14] and outside the scope of this paper.

Furthermore, SLA trading has direct economic implications. Each ISP in our system will implement a pricing function for its own resources. This, together with the offers of its surrounding peering partners, is used to construct and advertise new services. The objective is to maximize local (DS domain) profits. Among the basic ISP models, this corresponds to the hierarchical bilateral approach [15]. Although a hierarchy is formed, different technologies and terms of agreement may be deployed. Heterogeneity and incremental deployment are also important requirements for inter-domain pricing solutions [16].

In this paper we describe in detail how the SLAT framework is defined and how it fits into the DS architecture (Section II). In Section III and Section IV the trading entities, the protocol used and their integration into DS are described in greater detail. Using a comprehensive simulation framework we will provide quantitative results on the convergence of SLA trading and the resulting network utilization for specific scenarios in Section V. Section VI sums up related work and in Section VII we conclude and discuss future work.



(a) Alternative paths from *ISP* 0 to *ISP* 4 (dashed lines).

(b) Selected path (bold line) with SLAs.

Fig. 2. Example: Interconnected ISP networks using dynamic SLAs.

II. The SLA Trading Framework

As a first example of SLA trading we assume that an access ISP is connected to *ISP* 0 and has a quantity of aggregated traffic to send to *ISP* 4. Figure 2(a) shows this simple network of interconnected ISPs. Due to the meshed topology of the network multiple paths can be found to reach a destination network. As we can see, some ISPs find themselves in competition to others. In our example, four possible paths lead from *ISP* 0 to *ISP* 4. Consider the forwarding-service to *ISP* 4: *ISP* 3 will receive an offer for the service from *ISP* 2, but since it can go there directly, will probably refuse it (unless it's connection to *ISP* 4 doesn't have enough capacity). Say *ISP* 3 can go to *ISP* 4 with a bandwidth 1, delay 75 and it costs $1. *ISP* 1 could receive an offer from *ISP* 3 with a higher delay, same or lower bandwidth and higher price. *ISP* 1 and *ISP* 2 could then make proposals to *ISP* 0, which will, if it is interested in that service, decide which one to buy. In the example (cf. Figure 2(b)) *ISP* 0 finally decides to buy the cheaper service from *ISP* 1.

From this small example involving simple QoS metrics (delay and bandwidth), we see that bilateral agreements in form of SLAs build up in a nested manner providing finally an end-to-end service. Cost and delay increase at each ISP (additive metric) along the path while the bandwidth metric is concave and stays at its minimum. As [6] states, "...[the] observation [is] that multilateral agreements rarely work...". Of course, the advantage of bilateral agreements comes at the expense of a possible service setup delay. However, we can avoid such delays through clever and foresighted contracting (cf. Section III). Furthermore, we argue that SLA trading happens at a medium time scale (several minutes to hours) and operates on aggregated flows.

A. Service Level Agreements

The definition of an SLA provides a base for heterogeneous trading systems. [3] defines the SLA term as "A service contract between a customer and a service provider that specifies the forwarding service a customer should receive. A customer may be a user organization (source domain) or another DS domain (upstream domain). An SLA may include traffic conditioning rules...".

SLA trading protocols and the traders itself may change from location to location. In our scenario, SLAs include the destination of the traffic flow to ensure end-to-end service. However, in a relaxed form, SLAs may also describe services that do not have an end-to-end significance, i.e. provide lower assurance of QoS. In detail, we define SLAs at each ISP by the following parameters:

- A *traffic description*. This includes support for defined PHBs as well as a QoS-vector (e.g. bandwidth and delay) for a specific traffic description. Furthermore, information about traffic conditioning may be included. Using a specific parameters instead of a PHB has the advantage of being a universal metric understood by all ISPs. It is their obligation to map service requests and offers existing PHB in their respective domains. In Section IV we discuss how an end-to-end flow can be mapped to an experimental PHBs supporting bandwidth and delay.
- A *geographical scope* from the ISP's network to some other destination network.
- *Duration* of the agreements. All SLAs expire after an interval specified in the contract.
- *Cost* for the agreements. SLAs are always associated with a price. Local pricing methods and business strategies may be used to calculate prices for new offers [3].

[3]For the sake of simplicity we assume global currency. In a practical

B. SLA Trading

SLA trading is performed by SLA traders situated somewhere in the ISP's DS domain. For the time being we assume traders to be centralized. SLA traders make local decisions about what services are provided to which peers. Such decisions may be made spontaneously or they are the reaction to an external event.

Initially, SLA traders may offer services to peer ISPs only (i.e. one inter-domain hop). For such SLAs a price $p > 0$ has to be calculated. It is the ISP's business whether to employ a model that always covers its own cost or to decide to implement a long-term strategy where, e.g. heavy discounting may be used.

Once offers from other partners are received and accepted as an SLA, an ISP may build new services out of the existing ones. The price for such a service is the sum of the SLA price offered by the peer plus the cost of the ISP's own resource. Or, if all the nesting is uncoiled, the sum of all local prices set by all ISPs involved.

Each time an SLA trader wants to construct a new service it may compare offers made by all the peers. Usually the best offer, compared to the fitness of the service and the price will be taken. However, this is not a necessity: by adding policy mechanisms to peers' offers, our purely market-based approach can be distorted by regulation. In general, only policies about the peer ISPs are expressible [4].

As a preliminary conclusion, we see that SLA traders have a great deal of autonomy. It is possible to follow short- or long-term strategies, to take higher or lower risks, and to exploit the ISP's very own business strategy.

C. A Simple Protocol for Trading SLAs

Signaling *demand or supply* at ISPs needs an appropriate protocol. Fortunately, SLA trading is done only between peers which keeps the protocol very simple. Figure 3 shows a message sequence chart of the *SLA Trading protocol (SLATP)*.

By sending optional *ask messages*, an ISP may request the service of another one. The main reason for this part of the protocol is to speed up SLA setup and therefore convergence. In a perfect, coordinated world where each ISP would advertise all its available resource by using *bid messages* this might not be needed. Bids are mandatory since they initiate establishing agreements.

system a currency converter may be employed.

[4]One could also think about an extension to global policies by, e.g. excluding providers from the path of nested SLAs. Policy attributes would then become part of the SLA itself and could be propagated to the next provider.

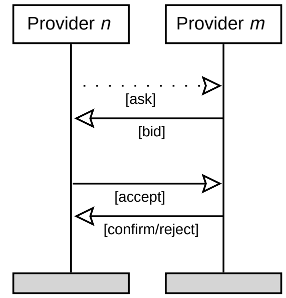


Fig. 3. The SLAT protocol enables ISPs to communicate offers and requests (bid/asks). Upon mutual agreement SLAs may be accepted or rejected.

After an ask/bid phase SLAs may be accepted or rejected by sending *accept* or *reject messages*. Upon an accept the bidding party will send a *confirm message* to seal the contract.

We should also mention here, that as long as ISPs adhere to the basic SLA structure they can deploy a trading protocol of their own choice (upon mutual agreement) [5] Even local definitions of experimental PHB are possible, as long as the end-to-end semantics are not disturbed (if this is a goal at all for the kind of traffic at hand).

III. SLA Traders

We will now examine the main functions of a SLA Trader.

A. Management of Bought and Sold SLAs

Every Trader maintains a database of bought and sold SLAs. It will set up DS classifiers/markers at ingress nodes and re-markers at egress nodes. Routing and switching traffic between ingress and egress node according to the SLAs is outside the scope of this paper and delegated to an appropriate switching architecture. SLA traders will also remove entries on expiration from that database and provision their internal networks accordingly.

B. Provisioning of New Resources

There are two types of resources: owned resources of the ISP and SLAs bought from other, external ISPs.

Owned resources are the raw material on which all the services are constructed. These are static or dynamic links from a ISP to some destination in the form of leased lines, ATM channels, optical fibers, etc., attached to routers of

[5]SLATP was a designed for simplicity and experimentation. In practice, one could also think about integrating SLAT messages into an existing inter-domain routing protocol, e.g. as BGP [17] attributes.

the ISP. In this paper we will make two assumptions about owned links:

- Every link is *unidirectional*
- Every link belongs to the ISP from which the link originates (i.e. the source of the traffic for that link).

SLA resources are the ones bought from other ISPs.

An SLA Trader has to decide how to provision it's external resources such as to make the most money possible by re-selling them to other ISPs. In other words, it will have to decide which SLAs to by buy trying to improve the value of its offer while keeping its cost as low as possible.

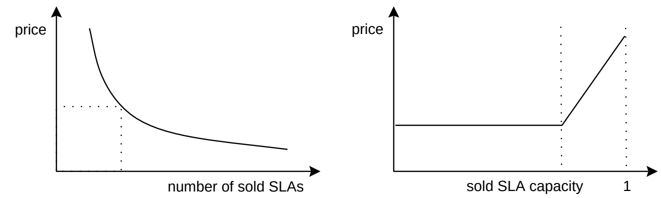
C. Bid Generation and Pricing

SLA Traders have to decide, based on available resources, which SLAs to propose to other ISPs. More fine-grained SLA bids will be more probable of being accepted because of the broader service palette, but will also involve more protocol overhead for the communication of them. Since a link is owned by the source ISP, frequent advertising of many bids will eat up part of its own link resources, leaving less space for services to sell. This is clearly an optimization problem, which will drive the development of new very efficient trading protocols and optimal customized offers for each peers matching the supply with their demand [6].

The goal of commercial ISPs is to maximize profit which is reflected in how SLATs will behave. The price of the sold services will have to be calculated to cover the costs of the ISP *and* to make profit.

Since SLA Trading involves *competition*, the lower the price for a service is, the more probable it is that someone will purchase it. An ISP will therefore try to *optimize the price* of the services trying to get the most possible profit. In a friction-free, single-good economic system where the demand curve is known, this would mean taking the price which does maximize the rectangular area shown in Figure 4(a). The economic system of networking services is however multiple-good and possibly not fair (two customers can be sold the same service for different prices), which does make this complex optimization problem better solvable with iterative algorithms.

Many pricing strategies are possible, for example *residual bandwidth pricing* as shown in Figure 4(b), where prices get higher the more the resources are used, based on the assumption that if the demand is high there will be customers willing to buy the services for a higher price. This aspect is a well studied economic problem, which goes beyond the scope of this paper. See [18] for further economic theory on this subject.



(a) Demand Curve

(b) Residual Bandwidth Pricing

Fig. 4. Example price functions.

D. Competition

Competition is of primary importance to the SLA Trading framework. If a provider has a much faster and less expensive link to a destination, it is better for global efficiency that its services are being preferred over the others.

If it is possible for everyone to setup the required resources and make SLA proposals, it is also probable, that if at a given time the price for a service is much higher than what could be made, another provider might take over the customers.

E. Trading Algorithms

To solve the problem of minimization of cost and maximization of profit, the SLATs may implement one or several of the following strategies:

What we call a *Lazy Trader* is one that does buy the services only on explicit demand of customers (trough *Ask Messages*). If every trader was lazy it would mean flooding the whole network with Ask Messages to even find the destination! It could, however, be a good strategy for a small ISP.

A *Greedy Trader* does simply buy all or too many of the bids it receives and is only a reference for a badly behaving trader.

A *Trendy Trader* does analyze current usage of the resources and buys bids by predicting future demands. The Trendy Trader is certainly a principal trader because it has the ability to book ahead for services with high demand.

Profitable Traders are an extension of Trendy Traders. They conduct in addition a profitability analysis based on past trades. In general, a window of past trades of the same type is maintained and the average price p_a is computed (this window may vary and even stretch beyond expiration of SLAs backwards in time but we currently use a window size of 1). p_a multiplied with an estimate of what amount will be sold (i.e. the bit volume) [7] is compared to each

[6] This is a kind of self-regulated signaling.

[7] This estimate is based on the history of sold SLAs, outstanding asks

available bid (bid price times volume that has to be purchased). If this balance is positive, the trade is considered being profitable and will be accepted.

Another aspect that has to be considered when implementing traders, is the aspect of temporal and spatial *fragmentation* of bids. Buying services that do not fit the requirements exactly impose a risk on the ISP (e.g. the needed service is indeed offered by a peer, but for too long or only bulk quantities are available). In our Profitable Traders bids are always offered as a selection ranging exponentially from a base bandwidth to a large bid. When bids are analyzed by traders both time and size are included in the profitability analysis.

F. Trading Anomalies

Since SLA Trading with competition involves selection of peers, and therefore routing decisions, two fundamental properties have to be ensured:

- *Loop freeness*
- Usage of *limited alternate routes*

We show that if intelligent enough traders are used, no routing loop will be made and the network usage will converge to a near-optimum. The reasons for the loop-freeness are:

- Strict QoS guarantees are made such as delay which would add indefinitely for each loop.
- “Stupid” traders which build loops will loose money doing that and will therefore be eliminated from the market.

Consider the forwarding service to *ISP* 0 in example Figure 2(b) of Section II. Suppose that *ISP* 2 did sell that service to *ISP* 4, which in turn sold it to *ISP* 3. Suppose further that *ISP* 2’s resources to *ISP* 0 become scarce and that *ISP* 2 wants to buy more because that service is in great demand. If *ISP* 2 behaves badly, it could buy further bandwidth for example from *ISP* 3. A loop is the result. This is different from a routing-loop, it’s rather a *service-loop*. What are the consequences of these service loops?

- Every ISP other than *ISP* 2 will gain money and the already existing services won’t be affected.
- *ISP* 2 will, for each service-loop as described before, rebuy it’s own service to *ISP* 3 paying also to every other ISP involved in the service loop. In other words, the QoS guarantees will remain satisfied, but will worsen (e.g. delay constraints will be higher and higher) and *ISP* 2 will go soon out of business because it will loose money.

For the limitation of alternative paths a similar argument holds: alternative paths are longer (in terms of DS hops) than the shortest path is. Longer paths might offer a competitive advantage in terms of price or service characteristics.

messages, and a target provisioning factor.

TABLE I
DS CODE POINT SPACE.

PHB	Code Space	# of DSCPs
Std. Pool 1	xxxxx0	32
Exp. Pool 2	xxxx11	16
Exp. Pool 3	xxxx01	16

IV. Integration of SLAT in DiffServ

We developed an experimental version of the SLA trading framework using a simulation environment [19]. We made several abstractions to keep complexity and runtime overhead low: (i) Packets belonging to the same flow (same source and destination network and traffic class) are modeled as a flow, and (ii) DS domains were abstracted into nodes. As a consequence, ingress and egress nodes of DS domains become links in the abstraction and interior nodes disappear.

Originally, 3 bits of the IPv4 header’s type-of-service byte were used as “IP precedence field”. Similarly, in IPv6 a “traffic class octet” was defined. In DS this octet was redefined and called *DS field*. It is split into 6 bits forming a code space of DSCPs and 2 bits which are currently unused [20].

Thus, the DS field has 64 entries for DSCPs. From Table I we can see that the code space is halved into a standard and experimental part. The standard pool is being filled with standardized PHBs as well as with a default PHB for best effort traffic and prioritized routing and other network control packets.

As mentioned earlier, SLA trading can be applied to any notion of PHB. However, in our simulation environment we chose to add an experimental PHB (called XF for experimental forwarding) that provides 16 code points with a specified delay, and the egress node. These combined entries are *dynamically allocated* by the SLA trader. The basic idea is to reserve a fixed number of DSCPs and end up with the ones most often asked for. Thus, the semantics of DSCPs become dynamic and the currently associated QoS features have to be communicated via SLATP to peers. In addition, this scheme is applied to each ingress node locally (cf. Figure 5 on the left side, local view of ingress node *i*), i.e. incoming packets from different peers having the same DSCP do not necessarily get the same service level.

Since the DSCP are a limited resource, two things can

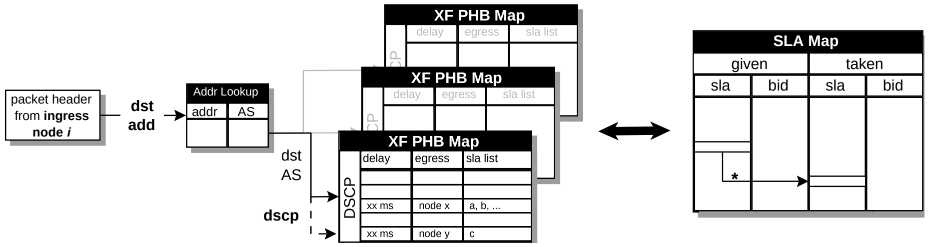


Fig. 5. Local per-destination and per-ingress node PHB tables; global SLA table.

be done to limit this problem: merging of flow aggregates with same service characteristics and service quantization. Merging can be done, if someone selects the same delay and egress node for multiple bandwidths, e.g. for 1 Mbps and 4 Mbps ($\rightarrow$ upgrading/extending SLAs). Note that the bandwidth is not defined in the PHB table.

Also, very fine grained delay specs are usually not needed and can be quantized to become “reasonably granular”. For the egress node information, a merger is not possible as long as we want to support multiple paths through the DS cloud.

A. Packet Handling at Ingress Nodes

Now that the usage of DSCPs is defined, we look into the basic operations of forwarding packets through DS domains. The two basic data structures used are local PHB maps for the dynamic allocation of code points (per ingress node and destination) and an SLA table (one for each DS domain). As shown in Figure 5, incoming packets select a PHB table depending on the entry node into the network, then the address is used to find the PHB table for that ingress/destination pair, and finally, the DS field itself is used to select the code point in this PHB table (for the destination AS address selection hashing is used while DSCP tables are short enough for direct lookup). Inside the DSCP’s table entry, a list of SLA identifiers is kept. The SLAs in this list share all the same ingress, egress, destination address and delay class.

Finding the list of SLAs from the packets destination AS and DSCP information as keys is a Multi-Field classification step (MF). Destination address to AS mapping can be done using traditional routing lookup methods. This conversion step is omitted in our implementation due to the collapsed DS domains.

The list of SLA identifiers contains also entries for the “new” DSCPs used for remarking (the ones that are valid in the next-hop network) and some domain-specific information how to get the packet to the egress node. Since we focus on inter-domain issues, we intentionally do not describe how the flows are switched from ingress to egress node. Many existing solutions exist for this problem, e.g. PASTE [21] could be employed, which in turn is based on MPLS and RSVP. Such solutions have to deal with resource allocation inside the DS domain, explicit routing through the cloud, and state setup in the interior routers/switches.

B. Packet Handling at Egress Nodes

As already mentioned, packets may need to be remarked at the egress node to conform to the SLA definition of the next hop on the path. If the DSCP is neither used nor changed while in transit through the ISP’s DS cloud, this step may be performed already at the ingress point. This has the advantage that the MF classification described above does not need to be repeated at egress nodes.

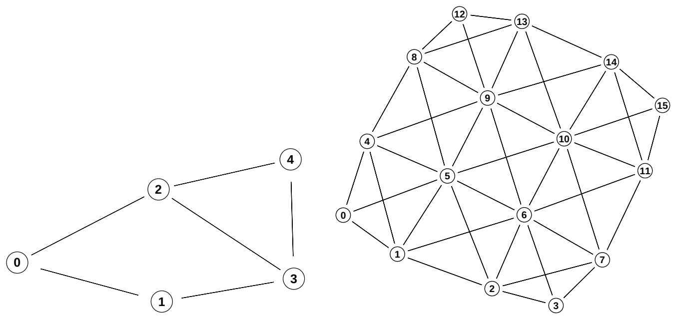
If it must be done at the egress node, the relationship between ingress node *i* and the flow used to transport these packets to the egress node must be forwarded via DS domain internal signaling.

Basically, we end up with one or possibly many DSCPs according to the SLAs for outgoing traffic. With a single DSCP, all the classified packets are remarked and sent out to the peer. If many contracts were used for same service characteristics and paths but differing in temporal scope and bandwidth, the outgoing packets have to be remarked according to their *share*. Packet scheduling algorithms, for example deficit round robin [22], are a good choice to implement this “fair-share remarking” process.

C. Interaction with the DS Domain SLA Table

As a central data structure in each DS domain, a table of “given” and “taken” SLAs and outstanding bids (offers) is maintained. As described in Section III, this table is the main instrument of the trader process. It contains all the dependencies of given SLAs on taken SLAs. Owned resources (the links to the peers) are just treated as a taken SLA that is always available.

When SLAs are set up, DSCP tables for the particular agreement have to be checked for code space first. It is the traders responsibility to do an economical allocation



(a) ISP example (b) Supergrid

Fig. 6. Inter-domain network topology examples.

of this scarce space. For each ingress node/destination pair a possible new combination of delay/egress node may be allocated when a new SLA is agreed upon.

To avoid fast consumption of code space, the number of egress nodes (multi-path options) has to be limited for a single destination and the number of delay classes has to be quantized. While multi-path routing has its clear benefits, an overly usage of the concept might not improve the networks utilization very much. For the quantization and resulting number of delay classes, many argue that two is enough (low delay and everything else). Therefore, we believe, that XF's dynamic allocation scheme of DSCPs can sustain even large and complex DS domains.

V. Evaluation

SLA trading is even in a simulated network a quite complex process. In this paper we conducted 4 experiments focusing on:

- Alternative path selection based on pricing.
- Network utilization with alternative paths.
- Resource distribution with different client behavior.
- Behavior of heterogenous network configurations.

For the evaluation of SLA trading we used an event driven simulator called *flowsim* [19]. In the following simulations, DS domains were collapsed to nodes at the inter-domain level (i.e. all nodes inside a DS domain are no longer visible, and ingress and egress routers of the DS domain become links of the collapsed node). Furthermore, traffic flows were mostly aggregated and directly fed into DS domains.

As topologies we used two simple, artificial examples. They are shown in Figure 6.

Our traffic model is an aggregate based on Poisson-distributed voice call arrivals with bandwidths between 16 kbps and 64 kbps which results in a smooth traffic aggregate.

A. Alternative Path Selection

In the simple example in Figure 6(a) a total demand of 1 Mbps for the path from *ISP* 0 to *ISP* 4 is applied. All links have 1 Mbps capacity. Using the residual bandwidth pricing function, starting to increase at 0.3, part of the load is shifted to the second path. Figure 7(a) shows the bandwidth of two links on the two paths. Since all bandwidths and initial prices are the same, the 2-hop route is selected first. With increasing demand, the price of this path increases and the 3-hop route becomes an alternative.

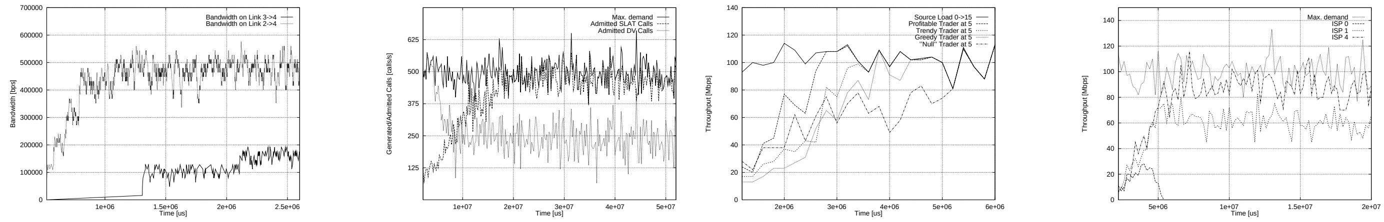
B. Network Utilization with Alternative Paths

The next experiment is run on the Supergrid topology (Figure 6(b)) with all links having 10 Mbps capacity. Aggregated voice calls (e.g. from a neighbor ISP) totaling 20 Mbps of offered load at a mean rate of 500 calls/s are generated at *ISP* 0 with destination *ISP* 15. In Figure 7(b) the admitted calls with SLA trading and DV routing (distance vector) are plotted (the experiment was run twice with the same setup for each routing method). While DV is clearly limiting the throughput to the bottleneck link speed of the shortest path (via *ISP* 5) SLA trading needs more time to setup the paths but achieves almost the maximum thoughput by using three paths via ISPs 1 and 4. The DV path is set up almost immediately resulting in a high call rate (until the shortest path is saturated). The Profitable Traders that were used for SLAT buy bandwidth incrementally and need almost 20 s in this configuration to converge to the maximum throughput. Of course, SLA traders can be configured using different strategy (e.g. shorter service setup time vs. higher risk).

C. Heterogeneous Traders: The "Black-Sheep" Setup

In this experiment we investigate the effect of heterogeneity in ISP networks. The "Supergrid" Topology with 200 Mbps links was used. We measure the throughput that an ISP connected to DS domain 0 receives from the network in function of the implementation of the trader at node 5 (the "Black Sheep"). The flow's destination is domain 15. All other traders were of type "Profitable". The experiment is repeated with the same setup for each trading type. The traffic load at *ISP* 0 has a mean bandwidth of 100 Mbps.

The result (Figure 7(c)) shows the time each trader needs to adapt to the new load. The *Profitable* trader (this is also the homogeneous case) is clearly the best implementation (convergence after 2.8 s). The *Trendy* trader follows, because it makes sensible decisions based on service usage (3.6 s). A *Greedy* trader makes quite sub-optimal



(a) Alternative route selection

(b) Load balancing

(c) Different trading strategies.

(d) Competition between 3 client ISP

Fig. 7. Simulation results.

decisions on the services to buy (4.4 s). Finally, the *Null* Trader which doesn't buy or sell anything at all requires the setup of alternate routes through ISPs 4 and 1 (convergence without using the path via 5 after 5.2 s).

D. Competition Among Client ISPs

Competition does not only involve the selection of the best bids from peers but also the selling to the highest bidders. Observing the link 5 $\rightarrow$ 10 (200 Mbps) in Figure 6(b) and the 3 ISPs at 0, 1, and 4, a Profitable trader at *ISP* 5 sells SLAs to these three clients. They are configured with an increasing willingness to pay (*ISP* 0 has the lowest, *ISP* 4 the highest). From Figure 7(d) we can see the maximum demand each ISP wants to generate (average 100 Mbps) and the actual throughput it gets according to its willingness to pay. The link 5 $\rightarrow$ 10 is beginning to fill and due to rising prices, *ISP* 0, with the lowest willingness to pay, stops buying the service at all after about 5 s. *ISP* 1 with a medium willingness to pay ends up buying about 60% of the ISP's demand and *ISP* 4 can buy almost all of its demand.

VI. Related Work

This section sums up several proposals for automatic provider information exchange related to SLAT.

Nichols et al. propose "bandwidth brokers" (BB) for DS in [6]. While their focus is on a global architecture including router mechanisms, packet marking, and integration of other architectures like intserv, one part of their work deals with BBs that are responsible for setting up bilateral SLAs. They observe that the information exchange between BBs can range from static agreements (i.e., classical peering) without any signaling to a dynamic setup even for a single changing flow.

Clark and Fang describe the "allocated-capacity" framework which is based on a single bit to differentiate services [23]. Similar to [4] and our SLAT approach, they base service allocation profiles on traffic specifications, geographic scope, and probability of assurance.

Li and Rekhter propose an architecture for DS and traffic engineering (PASTE) [21]. Using MPLS and explicit route objects in RSVP, they construct aggregated flows with forwarding and service level semantics. PASTE intends to provide an intra-domain solution.

Semret et al. review the DS framework in the context of a game theoretic approach [24]. Focus in this work is put stability and consistency of bandwidth allocation across several networks. First results indicate that stability can be achieved even for different service classes (AF and EF) and service levels affecting each other. However, they observed instabilities in situations of small networks (3 nodes were used in simulations) and tight provisioning.

In [25], Courcoubetis and Siris investigate how SLAs are priced when demand is measured as effective bandwidth. Furthermore they show how customers can optimally select traffic parameters of SLAs and how their model performs for real-time and non real-time service classes.

A good overview of general QoS routing problems, solutions, and architectural considerations is given in [14], [7]. As soon as different traffic classes, associated prices or other complex cost measures enter the picture, QoS routing (at least at the inter-domain level) of multiple, aggregated flows (in contrast to per-call path selection) defines quite similar problems as our SLA trading approach does.

VII. Conclusion and Future Work

SLA trading provides a framework for bilateral agreement negotiation. It supports local optimization, incremental deployment, and evolving definitions of services and PHBs. This is good news for providers since they can pick the mechanisms and policies they like best. Therefore, it is a good alternative to homogeneous QoS routing systems. But it is also good news for customers. The competition among providers will be perceptible even at the edge of the network where a single user is not able to choose which path her packets will take.

The downside of the scheme is that the gained local freedom complicates the evaluation of the employed algorithms. We can simulate or test the system, even with different strategies running at different DS domains, but the system as a global network will remain very dynamic.

The main contributions introduced by the SLA trading concept are summarized below:

- Automatic dealing of SLAs provides the basis for a market of network resources.
- The *combination of service provisioning and inter-domain path selection* improves network utilization and it can exploit alternative paths within shorter time.
- Having local pricing and provisioning methods is an approach favored by many ISPs. Bilateral SLAs serve as a common interface between peers but they do not restrict local behavior and implementation. This scheme also promotes rapid deployment.
- We experimented with *dynamic and adaptive PHB definitions* (XF PHB) that are valid for just the lifetime of an SLA. This new method introduces flexibility to DS and saves code point space.

We will address two major topics in future work. First, an interesting field of research will be the application of the SLAT approach to other service types than XF. Especially AF will be a challenge due to its more relaxed definition. Second, more simulations with larger AS topologies and more realistic traffic models or traces will be performed.

References

[1] R. Braden, D. D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC 1633, June 1994.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification," RFC 2205, September 1997.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.

[4] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.

[5] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[6] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the internet," `http://www-nrg.ee.lbl.gov/papers/bitarch.pdf`, November 1997.

[7] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet," RFC 2386, August 1998.

[8] Z. Wang and J. Crowcroft, "Routing algorithms for supporting resource reservation," *IEEE JSAC*, 1996.

[9] R. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions," November 1996.

[10] Q. Sun and H. Langendorfer, "A new distributed routing algorithm with end-to-end delay guarantee," 1997.

[11] G. Apostolopoulos, R. Guerin, and S. Kamat, "Quality of Service Based Routing: A Performance Perspective," in *ACM SIGCOMM 98*, September 1998.

[12] Q. Ma and P. Steenkiste, "On QoS Path Computation," in *IEEE INFOCOM 98*, 1998.

[13] S. Kweon and K. G. Shin, "Distributed QoS routing using bounded flooding," 1998.

[14] S. Chen and K. Nahrstedt, "An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions," *IEEE Network*, November/December 1998.

[15] L. W. McKnight and J. P. Bailey, "Internet Economics: When Constituencies Collide in Cyberspace," *IEEE Network*, vol. 1, no. 6, pp. 30–37, November/December 1997.

[16] S. Shenker, D. Clark, D. Estrin, and S. Herzog, "Pricing in Computer Networks: Reshaping the Research Agenda," *ACM Computer Communication Review*, vol. 26, no. 2, pp. 19–43, April 1996.

[17] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.

[18] Hal R. Varian, *Intermediate Microeconomics, a modern approach*, Norton, 1996.

[19] "*flowsim*, A Flow-based Network Simulator," `http://www.tik.ee.ethz.ch/~fs2`, 1999.

[20] K. Nichols, S. Blake, F. Baker, and D. Black, "Defintion of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, December 1998.

[21] T. Li and Y. Rekhter, "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)," RFC 2430, October 1998.

[22] M. Shreedar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin," in *SIGCOMM '95*, Aug 1995.

[23] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, August 1998.

[24] N. Semret, R. Liao, A.T. Campbell, and A.A. Lazar, "Peering and Provisioning of Differentiated Internet Services," Columbia University, CTR, Technical Report, 1999.

[25] C. Courcoubetis and V.A. Siris, "Managing and Pricing Service Level Agreements for Differentiated Services," in *IEEE/IFIP IWQoS '99*, May 1999.