University of Würzburg
Institute of Computer Science
Research Report Series

# CSPF Routed and Traffic-Driven Construction of LSP Hierarchies

Michael Menth and Andreas Reifert

Report No. 297 June 2002

Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
tel: (+49) 931-8886644, fax: (+49) 931-8886632
E-Mail:{menth|reifert}@informatik.uni-wuerzburg.de

# CSPF Routed and Traffic-Driven Construction of LSP Hierarchies

**Michael Menth and Andreas Reifert**
Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
tel: (+49) 931-8886644, fax: (+49) 931-8886632
E-Mail:{menth|reifert}@informatik.uni-wuerzburg.de

**Abstract**

The objective of this work is the analysis of reservation aggregation and the description of a network architecture for scalable Quality of Service (QoS) support. This architecture applies Differentiated Services (DiffServ) for packet forwarding, admission control (AC) is done on a per flow basis at the access, and resource allocation is based on reservations. The reservations of individual flows are aggregated recursively to achieve scalability in the core. Multiprotocol Label Switching (MPLS) is applied to reflect these aggregates in label switched paths (LSPs). The construction of the LSP hierarchy is traffic-driven and based on explicit routes that are determined by constraint shortest path first (CSPF) routing. We describe the architecture of that system and suggest several mechanisms to operate it also in networking scenarios with heavy signaling load.

**Keywords:** QoS, resource allocation, admission control, CSPF, MPLS, LSP hierarchy

## 1 Introduction

One of the challenges in future data communication networks is the provisioning of toll quality data transport for real-time applications, i.e. Quality of Service (QoS) for the traffic in terms of loss and delay bounds for transported packets must be met.

For this purpose, the Internet Engineering Task Force (IETF) proposed the Integrated Services (IntServ) approach [1, 2] for IP networks. This, however does not scale in large networks with respect to reservation handling, packet classification and scheduling in presence of a large number of flow reservations in a router. In contrast to that, the Differentiated Services (DiffServ) [?] allows only for a few relatively differentiated transport service classes which makes this paradigm is highly scalable. However, it does not provide reservations and absolute QoS guarantees. Multiprotocol Label Switching (MPLS) [3] is a promising technology that offers various means for traffic engineering (TE). When aggregated flows are tunneled in a label switched path (LSP), their reservations can be subsumed to a single one along that path and their packets are unified under a common MPLS label for classification and scheduling. This achieves scalability in the label switching routers (LSRs) [4]. Furthermore, MPLS allows for the integration of Constraint Shortest Path First (CSPF) routing leading to a better utilization of network resources [5].

In this work, we investigate a network architecture that combines principles from IntServ, DiffServ, and MPLS. It offers good real-time QoS support and it is scalable in large networks.

We suggest protocol actions to build LSP hierarchies in a traffic-driven and distributed manner. We propose mechanisms that protect the network from signaling overload not only in extreme networking scenarios [6].

This paper is structured as follows. Section 2 gives a short introduction to important aspects of IntServ, DiffServ, MPLS, and CSPF. In Section 3, we describe our network architecture that is based on these concepts and explain the required protocol actions. We introduce further enhancements to the basic structure to achieve signaling scalabiltiy even in extreme networking scenarios. Section 4 illustrates the performance of this approach with respect to resource utilization and signaling stability. Section 5 concludes this work and gives an outlook on further activities.

# 2 Concepts for QoS Support in IP Networks

The architecture under study comprises many known concepts of today's methods for QoS support in IP networks. Therefore, we briefly describe their most important aspects in this section. The IETF has suggested two main alternatives to enhance IP networks with real-time capabilities. These are the IntServ and the DiffServ approach. Recently, MPLS has been defined to facilitate the process of traffic engineering, e.g. data tunneling and route pinning. The latter feature may be used in combination with constraint based routing.

## 2.1 Integrated Services

IntServ is characterized by the separate handling of each individual end-to-end (e2e) micro flow. The resource reservation protocol (RSVP) [7] is used to establish for an e2e flow path and reservation states with knowledge about this flow in all routers along its path. These states contain among other information a flow specification that comprises the $Tspec$ and $Rspec$ parameters to indicate the expected data rate and the desired QoS for the reservation. They are used to manage the capacity on every outgoing interface and to enforce policing on a per flow basis. In particular, AC uses these data to decide whether an additional flow can be admitted. A separate queue and a scheduling state are maintained for each flow to meet the required QoS objectives. This, however, is clearly a difficult task for routers as soon as the number of flows is in the order of a few ten thousands which can be easily reached in backbone networks. Hence, IntServ does not scale in large networks and can not be applied. Therefore, reservation aggregation [8] has been suggested to overcome this drawback.

## 2.2 Differentiated Services

The DiffServ approach allows only for a few traffic classes. The Differentiated Services Code Point (DSCP) in the IP header is used to mark the different Per Hop Behaviors (PHBs) that tell the routers to treat the corresponding IP packet with low or high priority in the forwarding process.

No per flow information is stored and, as a consequence, this architecture scales well for large networks because the forwarding process operates on aggregated traffic and not on single micro flows. Policers and shapers at the network edges try to control the traffic volume

entering the network. But simple traffic conditioning impairs the transport QoS of all flows with the same DSCPs in the same way since the approach lacks AC. It can not support high QoS for some flows at the expense of the rejection of others.

A so-called bandwidth broker solves that problem by introducing AC on a per flow basis at the network edges [9]. The packet classification and scheduling inside the network is still done according to the DSCPs. The bandwidth broker needs to know all flows and their routes in the network to avoid congestion on the links. Hence, AC is done in an almost central manner and faces similar scalability issues like IntServ [10]. Distributed and hierarchically structured bandwidth brokers try to mitigate that effect [11, 12, 13]. The remaining key feature of DiffServ is that the packet classification and scheduling relies only on the DSCP in the packet headers and keeps the forwarding engine simple.

### 2.3 Multiprotocol Label Switching

MPLS is a mechanism to allow packet switching instead of routing over any network layer protocol [3]. The ingress label switching router (LSR) of a label switched path (LSP) equips an IP packet with a label of 4 bytes and sends it to the next LSR. The LSRs classify a packet according to its incoming interface and to its label. Based on this information and the incoming label map (ILM), label swapping is performed and the packet is forwarded to the particular outgoing interface. The egress LSR only removes the label from the IP packet header. In practice, modern routers are capable to process both IP and MPLS packets. Hence, the label swapping process requires entries for every LSP in the management information base (MIB) of the LSRs, so there is again a state per session like in IntServ.

There are two major protocol alternatives for establishing an LSP. RSVP Tunneling Extensions (RSVP-TE) is a modification to RSVP [14] and is able to distribute the labels and the Constraint Based Label Distribution Protocol (CR-LDP), [15] has been designed particularly for that goal, however, the IETF seems now to go along with RSVP-TE. An LSP may be established and associated with bandwidth reservations, e.g. using the primitives of RSVP. Thus, the LSP represents then a virtual link that borrows its resources from the links connecting its LSRs. The more general Label Distribution Protocol (LDP) is not able to make reservations [16].

The label distribution and the label switching paradigm allows for explicit route pinning which facilitates fast rerouting and load balancing. Furthermore, packets from different flows can be tunneled through an LSP. The label makes the aggregation visible in the LSRs and eases the mapping of a packet to a specific aggregate. It also bypasses control messages that are related to the individual flows at the LSRs.

MPLS implements the connection concept. Therefore, it is often viewed as modified version of the Asynchronous Transfer Mode (ATM) with variable cell size. But there is a profound difference: ATM enables a two-fold aggregation with its virtual connection and virtual path concept while MPLS allows for many-fold aggregation using multiple label stacking, i.e. an LSP may be transported over other LSPs. This feature helps to build scalable network structures, so-called LSP hierarchies [17, 18, 19, 20].

### 2.4 Constraint Shortest Path First (CSPF) Routing

Open Shortest Path First (OSPF) is the common routing algorithm in the Internet. The routing in the Internet has two major drawbacks. All packets with the same destination are routed along the same path. This can lead to overloaded and poorly utilized links at the same time. Packets that require real-time transportation need a path where enough resources are available to avoid extensive waiting times and packet loss in the router internal queues. But the IP routing mechanism is unaware of the free link capacities. In contrast, CSPF takes the free resources of the links into account and finds the shortest path through a network whose links offer a desired QoS - if there is any such path available. This route may differ from the shortest path that is taken by OSPF. LSPs may be used to provide tunnels along these routes to bypass conventional IP routing. The blocking probability for data flows that require stringent QoS can be reduced this way when default paths are highly loaded.

## 3 Reservation Aggregation

The hierarchical partition of networks into access and core implicates that the number of flows increases towards the core. This is a problem for per flow reservations because the large number of flows in the core is not manageable in the routers. Flow aggregation towards the core achieves scalability for flow classification and resource reservation. We briefly characterize what we understand in general by aggregation and deaggregation, describe the tunnel and funnel concept, and explain how these are signaled by existing protocols. Finally, we also present another kind of aggregation that is implemented in a distributed bandwidth broker.

### 3.1 Aggregation In General

**Aggregation.** Several flows with the same or similar requirements are summarized along a common subpath of their routes to a single flow from the viewpoint of a router. That reduces the information quantity of the flows which is stored in the routers along the common subpath. Aggregation can be applied recursively.

**Deaggregation.** When a flow aggregation terminates, the individual flows become visible again in the deaggregating router. The attributes of the original flows (e.g. final destination) are restored. The aggregation method influences how much of the original information can be recovered.

**Aggregation Aspects.** In DiffServ, packets with the same demand for QoS are marked with the same DSCP. This aggregates many streams with respect to queuing and scheduling. Routing in the Internet is also performed on traffic aggregates: only a subnet mask is decisive for the routing decision. All the information needed for packet scheduling and routing is recorded in the packet header.

The flowspecs (e.g. peak rate) of reservations are needed for the resource management of the links. Their information is not related to a single packet but to the whole flow and it is

stored in the routers. Hence, when reservations are aggregated, the flow related information in the routers must also be aggregated. If it is needed again after deaggregation, it must be preserved. Therefore, reservation aggregation is more difficult than aggregation for scheduling and routing.

There are basically two ways how reservations can be aggregated. We explain these fundamental concepts using MPLS terminology. Aggregation using MPLS means that two different flows are equipped with the same label and are forwarded in the same manner. In principle, there are two alternatives to accomplish this: tunnels and funnels.

### 3.2 Tunnel Aggregation

We talk about tunneling flows if the uppermost label in their packet headers remains in place and a new common label is put onto the label stack. This new label corresponds to a new connection with the aggregated context (cf. Figure 1), i.e. a new aggregating LSP is set up. The individual reservations of the contained flows are (naively spoken) summed up to compute the size of the aggregate reservation. The flows are transported over the LSP and when the egress router of that LSP removes the uppermost label, the original flows are restored. In particular, their connection context and reservation information is present in the deaggregating router.

The LSP acts as a logical link and the intermediate LSRs do not see the individual reservations because their control messages are bypassed as MPLS packets at the interior LSRs of the LSP. Hence, tunnels reduce the state information in the intermediate routers, they allow for reservation aggregation and deaggregation because the original flows are recovered. MPLS tunnels are applied in [21] whereas other tunnels are applied for reservation aggregation in [8] and [12].

We sketch out how tunnels in MPLS are set up using RSVP-TE. The ingress LSR issues a PATH message with the resource demand and it is forwarded hop by hop to the future egress LSR. This pass is used to install a path state in every participating router to indicate the previous hop, to store flow related information, and to make label requests to downstream next hops. In addition, the PATH message contains information about the available capacity on the already traversed route so that the demand can be adapted if there is a shortage of resources. The egress LSR triggers a RESV message back to the ingress LSR that distributes the MPLS labels upstream and establishes the reservation for the LSP by setting up a reservation state. As RSVP-TE is only an extension of RSVP, the path and reservation states are soft and are refreshed by periodically sent PATH and RESV messages.

The tunnel concept scales poorly in some network topologies. When full connectivity in a star network is to be realized, the center node has to handle all possible tunnels which amounts to exactly $N \cdot (N-1)$ LSPs. Thus, the number of aggregates scales quadraticly with the network size. Fortunately, mostly not all of them are in the MIB of a single router but this is still not a good scaling behavior.

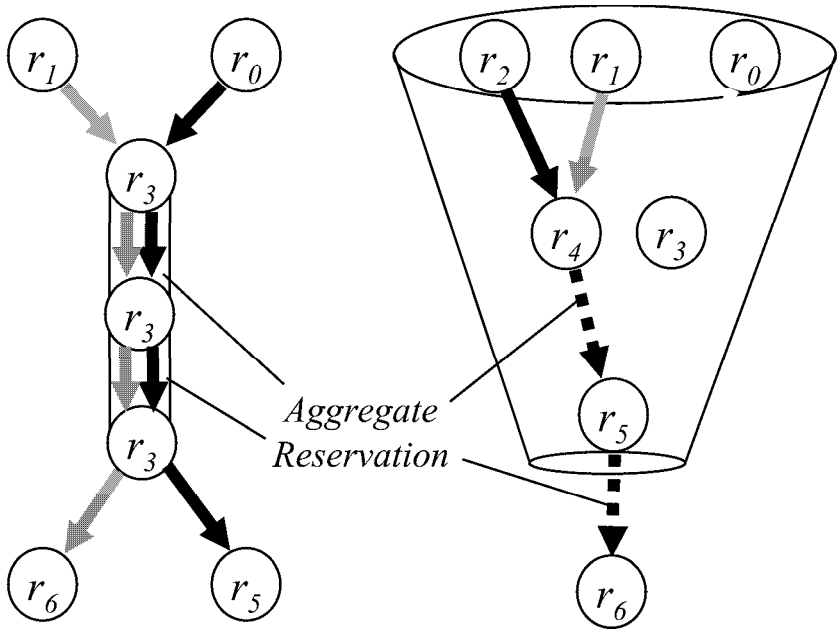


Figure 1: Tunnel aggregation in contrast to funnel aggregation.

### 3.3 Funnel Aggregation

We say that LSP flows are merged into a new aggregate if the uppermost label in their packet headers is substituted by a new common label. Figure 1 visualizes the resulting sink tree towards a common destination and motivates the name funnel for this kind of aggregation. In contrast to tunnel aggregation, no new connection is created to carry the aggregated context but the aggregate information of the new aggregate is associated with the merged flow in the downstream LSP context. The information about the individual flows is lost and can not be recovered at the end of the sink tree.

To achieve full connectivity in a network with $N$ nodes, every node needs to hold $N - 1$ LSPs since every other router can then be reached by equipping the packets with the corresponding label for the destination machine. This means that the number of paths scales linearly with the network size.

**LSP Multipoint-to-Point Trees.** In MPLS it is possible to construct multipoint-to-point forwarding trees that have a common sink as destination of the transported traffic. So far, the LDP has been developed for label distribution in MPLS but no reservations can be set up with LDP. CR-LDP is able to set up reservations but it explicitly declares the construction of multipoint-to-point LSP for further study. In RSVP-TE, there are different filter styles. The fixed filter (FF) style only allows for point-to-point LSPs. With the wildcard filter (WF) reservation style, a single shared reservation is used for all senders to a session. The total reservation on a link remains the same regardless of the number of senders. This reduces the amount of reservation information at the egress on the one side but on the other side, the size

of the aggregated reservation can not be adapted to the number of sending sources. Hence, this is not a scheme for reservation aggregation. The shared explicit (SE) style allows a receiver to explicitly specify the senders to be included in a reservation. There is a single reservation on a link for all the listed senders and this option may be used to support sink tree reservations provided that the explicit route objects (ERO) of the different sessions are the same. But the bandwidth computation of the merged reservations is not appropriate: "When SE-style reservations are merged, the resulting filter spec is the union of the original filter specs, and the resulting flowspec is the largest flowspec. [7]. For sink tree reservation aggregation we rather need the sum and not the maximum of the individual reservation sizes. Hence, there is no label distribution protocol that supports sink tree reservations for the purpose of reservation aggregation.

**Signaling of Sink Tree Reservation Aggregation with BGRP.** The Border Gateway Reservation Protocol (BGRP) [22] has been conceived for inter-domain use and to work in cooperation with the Border Gateway Protocol (BGP) for routing. It is used for reservations between border routers only. BGRP aggregates all inter-domain reservations with the same destination autonomous system (AS) gateway into a single funnel reservation, no matter of their origin.

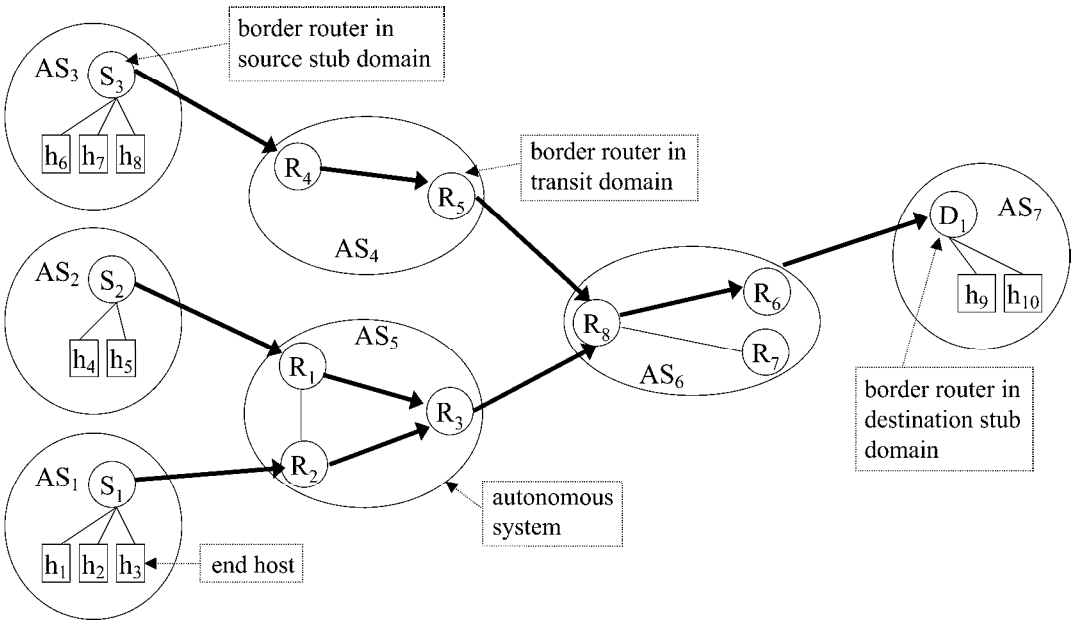


Figure 2: Signaling in BGRP.

Figure 2 shows the setup of sink tree reservations in BGRP. A PROBE message is sent from a source border router $S_1$ to the destination border router $D_1$. It is processed at the intermediate border routers, collects them as well as information about available transit capacities in their AS. Unlike in RSVP, no "PATH" state is established, i.e. stateless probing is done. A sink tree is uniquely identified by the IP address of the destination AS and an ID for the destination

border router $D_1$. The destination border router returns a GRAFT message together with the tuple (IP address of destination AS, ID) along the reversed path collected in the PROBE message. The required reservation states are established or updated if they already exist.

So far, this is very similar to RSVP: one pass is used to collect path information in order to reserve on the way back an appropriate amount of resources. Hence, both protocols consist of a path information pass (PIP) and a resource reservation pass (RRP). There are also simpler protocols like Boomerang [23, 24] that combine the PIP and the RRP. They require only a single signaling pass but they need a second pass back to the sender to notify the successful establishment of the reservation.

The difference between BGRP and RSVP is that the reservation request is expressed by a relative offset because the issuing source can not know the resulting size of the aggregate reservation on the downstream links. Such an offset must be signaled exactly once and must arrive at the sink of the reservation tree. Therefore, BGRP requires reliable communication for signaling whereas RSVP messages are sent in unreliable datagrams. BGRP is also a soft state protocol but in contrast to RSVP, only neighboring routers exchange explicit REFRESH messages. They keep the reservation alive and interchange absolute reservation values.

### 3.4 Source Tree Flow Aggregation in Aquila

The project Aquila [11, 25, 26] implements a distributed and scalable bandwidth broker architecture which gains its scalability also from aggregation. The capacity of all links is controlled by a central resource control agent (RCA) which distributes shares of link capacities to so-called admission control agents (ACA). An ACA is a bandwidth broker is associated with a single edge router and handles only local admission requests. The resource assignment from the RCA to the ACAs can be viewed as a reservation for aggregated flows starting from a common ingress border router to all egress border router. In contrast to BGRP, these flows form a source tree instead of a sink tree. The reservations for all flows are basically known for all links in the network but the scalability comes from the fact that only the edge router knows about them. Therefore, the reservations do not need to implement as RSVP states. Note that this kind of aggregation can not be applied to MPLS or RSVP-like concepts that rely on reservation states in the network. The Aquila architecture is flexible since the RCA and ACA are able to negotiate bandwidth and to make their interaction more scalable, the RCA may be implemented as a hierachically structured and distributed entity.

## 4 Distributed and Traffic-Driven Setup of an LSP Hierarchy with Integration of CSPF

In this section, we describe how CSPF routed e2e reservation may be established within an LSP hierarchy using a traffic-driven LSP setup.

We envision a hierarchically structured multi-service network that is capable for real-time transport. QoS is realized for flows in the network by e2e reservations, i.e. AC is performed for all links in the network (cf. Figure 3). For scalability reasons we apply reservation aggregation using MPLS technology. That introduces additional virtual links for which AC also

applies. Databases at the ingress routers store the available capacity of all links in the network. This information is used to compute for QoS flow requests the shortest path with a minimum capacity through the network. These constraint-based routes increase the success probability of AC when the network is highly loaded. The IETF discusses a similar concept with a centralized solution: A Path Computation Server (PCS) computes routes based on a link database tracking the network-wide resource utilization [27, 28].

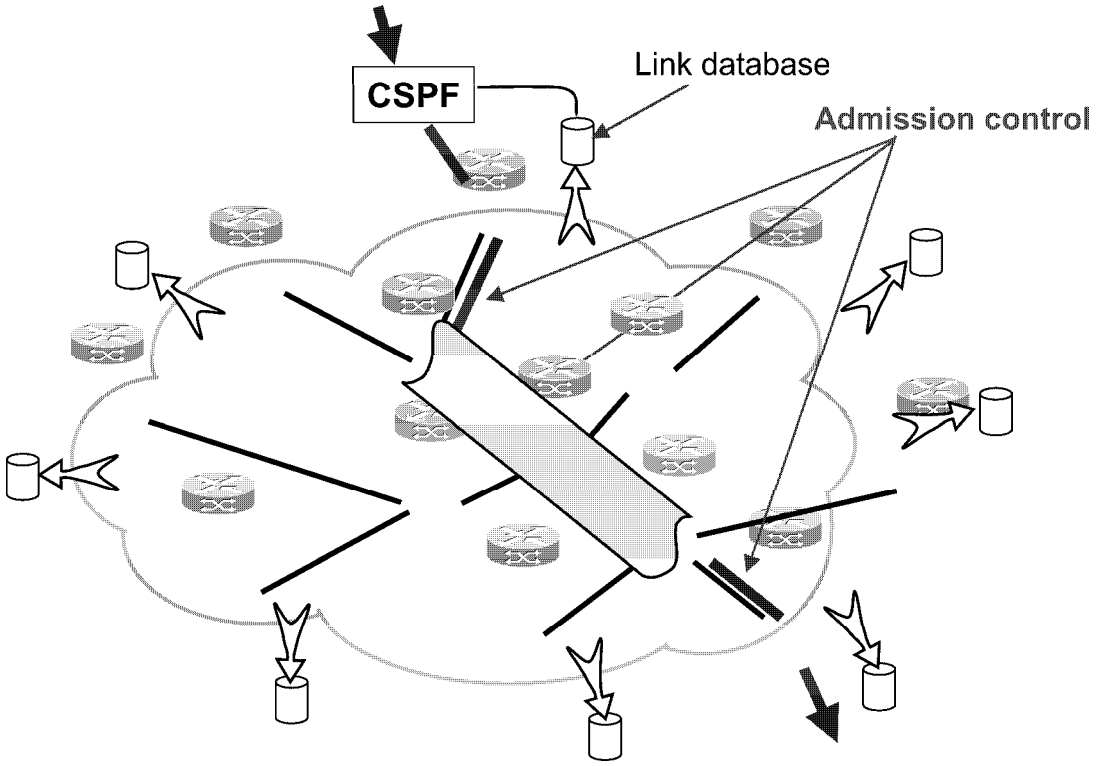


Figure 3: Link databases allow CSPF computation and AC is performed in the network.

In the following, we suggest mechanisms to realize a low degree of overreservation for reservation aggregates to reduce their signaling frequency. We propose that a protocol signals the available capacity of physical and virtual links to the link databases at the ingress routers. Based on this information, explicit routes (ER) are computed and enhanced by LSP hierarchy information. We explain modifications to RSVP(-TE) that are necessary for hierarchical LSP setup. Finally, we delimit this architecture against existing approaches.

### 4.1 Resource Management and Hierarchical Aggregation

As outlined before, reservations are a possible means for resource management. If parts of the resources of a link have been dedicated to some other – virtual – links, they can not be reused until they are released. Reservation aggregation reduces the number of reservation states in intermediate routers. Tunnels and funnels may be used and they may also be applied

recursively. In the following, we suggest the static structure of some basic aggregation options using MPLS technology because it also supports scalability for packet classification.

**No Reservation Aggregation.** If no reservation aggregation is performed, individual RSVP reservations initiate states along their paths through the network. This is the standard IntServ architecture.

**Simple Tunnel Aggregation between Access Routers Using LSPs.** LSPs may be used between access routers to tunnel e2e reservation through the core network. The number of LSPs in the core is limited by the square of the number of access routers and is independent of the traffic load. If alternative paths are possible, this number can be exceeded because new LSP tunnels are required for additional alternative routes. As outlined before, LSP tunnels can be established by already existing signaling protocols.

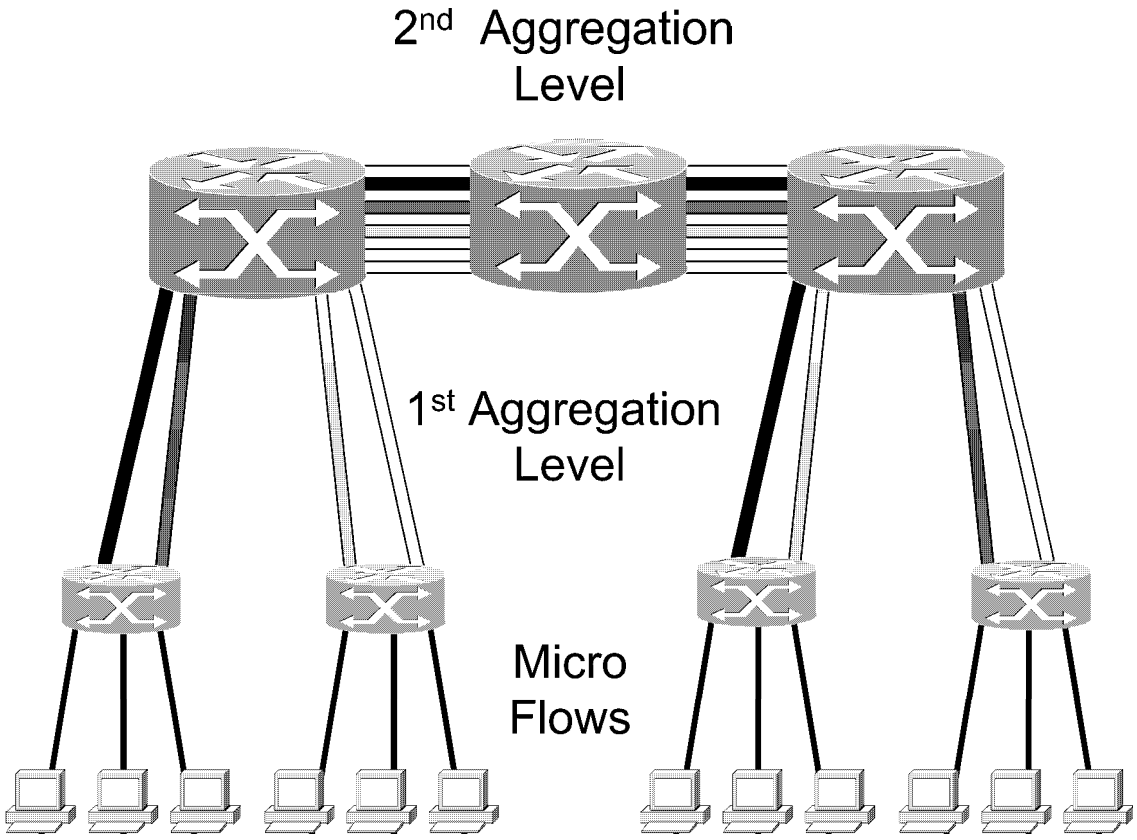


Figure 4: Hierarchical tunnel aggregation between access routers.

**Hierarchical Tunnel Aggregation between Access Routers** LSPs may be used between routers of the same depth whose path starts and ends with a link between two different depths

(cf. Figure 4). The depth of the LSP is the depth of its ingress and egress router. If possible, the LSP of depth $d$ is carried over another LSP of depth $d-1$.

**Simple Aggregation between Access Routers Using a Sink Tree LSP.** LSPs may be used between access routers to tunnel e2e reservations through the core network. The LSPs themselves are arranged as a sink trees. The number of LSPs in the core is limited by the number of access routers and is independent of the traffic load. If alternative paths are possible, this number can be exceeded because for an additional alternative path a new LSP funnel may be required. As outlined before, LSP sink trees with reservations can not be established by already existing label distribution protocols. But enhancements to MPLS signaling protocols similar to BGRP would solve that problem.

### 4.2 Description of the PIP and RRP

Explicitly routed E2E reservations as well as LSPs are set up using a PIP and an RRP. They are also used to increase or to decrease the reserved rate of reservations, therefore, we explain them briefly beforehand. The RSVP(-TE) sender triggers a PATH message that contains an explicit route object (ERO) and follow the ER towards the destination. The PATH message carries a desired minimum and maximum capacity value, collects the available bandwidth on the path, and adjusts the flowspec ($Tspec$). The destination router returns an RESV message with an appropriate $Tspec$ parameter and the required resources are reserved on the way back, i.e. the used links dedicate $Tspec$ to the new reservation. When the RRP is back at the sender, it is informed about the new $Tspec$ for future PATH messages that refresh the PATH and RESV states periodically. If the reservation failed because of lack of bandwidth, this is marked at the router to prevent another such reservation setup or increase of reserved bandwidth during the next *NoResourceInterval* for that reservation. This failure should be propagated to the lower hierarchy levels until the e2e reservation is notified by an error message. Since the error message returns to the ingress border router, this failure information should be added to the local link database for the next *NoResourceInterval* in order to prevent the same requests in the near future.

### 4.3 Reduction of Signaling Frequency by Overreservation

Reservation aggregation increases the scalability in real-time networks by reducing the number of states in the router MIBs. If an aggregate reservation corresponds exactly to the sum of the size of the aggregated reservations, it is updated whenever an RSVP flows starts, changes in rate, or ends. This change is also propagated to higher level LSPs if their aggregate reservations are also tight. Therefore, the amount of signaling is rather increased than reduced due to LSP capacity updates. Fortunately, it is possible to trade signaling frequency for bandwidth efficiency. When the bandwidth of an LSP is updated, its reservation is set to a larger value than the required sum of aggregated reservations in order to serve future requests from the residual bandwidth.

We define several attributes for physical links and LSPs. $Tspec$ is the amount of bandwidth that is assigned to an LSP by its reservation from the links along its path to itself. The

"$Tspec$" of a physical link corresponds to its physical bandwidth which does not change. Only the link itself can dispose of that capacity. When a flow passes the AC of a link, some of the link capacity is dedicated to the reservation of the flow. The *AllocatedBandwidth* is the sum of all capacities of a (virtual) link that are already assigned to reservations. The *FreeBandwidth* of a link is the difference between its $Tspec$ and its *AllocatedBandwidth*. This is the capacity that may be used for serving new requests. For conservative AC without overbooking $FreeBandwidth \geq 0$ is always true. The *AllocatedBandwidth* may be larger than the sum of individual e2e reservations (*UsedBandwidth*) that are indeed carried over the link. *UsedBandwidth* is a value that is in general not available in the router MIBs since it is intentionally concealed by aggregation and overreservation.

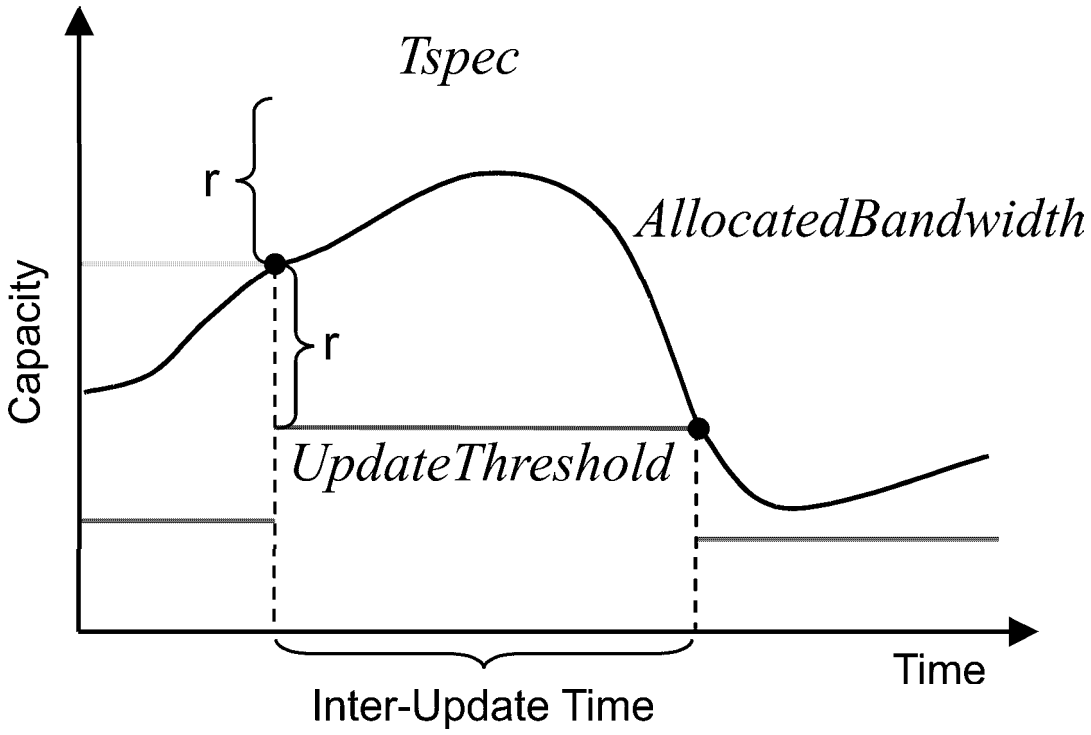


Figure 5: Overreservation decreases the mean inter-update time of an LSP reservation.

According to Figure 5, the LSP orders more bandwidth from the links supporting it (if possible) when its *AllocatedBandwidth* does not suffice to serve a new request. With RSVP-TE this may be done using the SE style [14] and a PIP with a new LSP_ID is triggered. When the *AllocatedBandwidth* falls below a predefined *UpdateThreshold*, some of the $Tspec$ has to be redistributed to the links supporting the LSP. This is done by sending a RESV message with reduced capacity requirements. During the signaling of a decrease of the LSP resources, the LSP is in an inconsistent state and should store arriving signaling packets in its sleep queue (SQ) which will be explained later. The overreservation dynamics have been studied in [6, 29].

### 4.4 Signaling Available Bandwidth for LSPs

The *AvailableBandwidth* of a physical link is its $FreeBandwidth$ and the *AvailableBandwidth* of an LSP at a certain LSR is the minimum of the *AvailableBandwidth* of all links supporting this LSP from this LSR to the egress LSR. Hello messages [14] provide a means for rapid node failure detection and are exchanged every 5 ms. We propose to use either these HELLO messages or extra messages to exchange additional information about *AvailableBandwidth* of LSPs. This signaling pass is illustrated in Figure 6. The *AvailableBandwidth* of an LSP at a certain LSR is computed as the minimum of the *AvailableBandwidth* at the next hop LSR and the available bandwidth on the link from the LSR to the next hop LSR. With this enhancement, the LSPs have an approximated knowledge about the bandwidth that they can potentially allocate. Hence, the value of the *HelloInterval* influences the system accuracy.

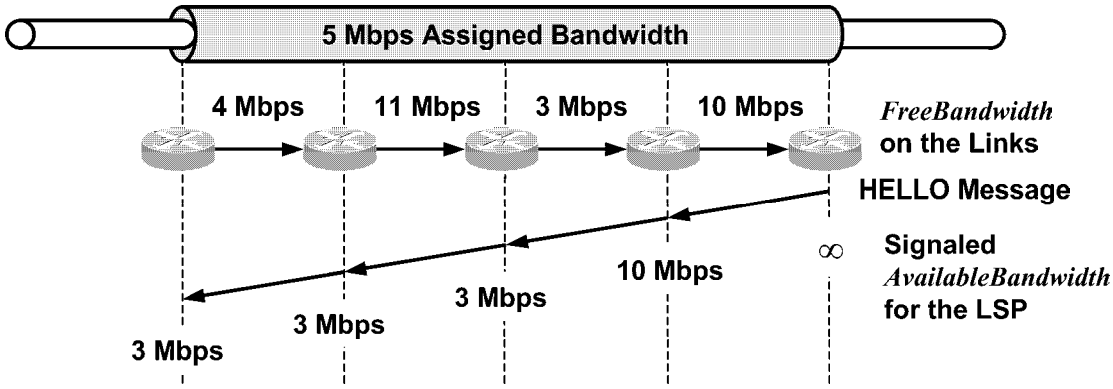


Figure 6: The signaling of available bandwidth for an LSP using Hello messages.

### 4.5 Monitoring the *AvailableBandwidth* in the Link Databases at the Border Routers for CSPF

To enable CSPF computation at the border routers, the link database needs the available resources of the links in the network. For the provision of the required information we suggest that every router in the network multicasts the *AvailableBandwidth* of all links that are under its control towards all border routers. The information quantity rises with the network size and, which is worse, the update frequency with the number of successfully admitted flows. Therefore, the simple approach where changed resource utilization is signaled to all local databases can not scale for large network structures from the signaling point of view. Therefore, it is important that the update frequency is limited, i.e. not all changes are signaled to the databases. Therefore, the router should send the *AvailableBandwidth* information only once in a *DatabaseUpdateInterval*. Hence, the CSPF algorithm operates on obsolete information. Therefore, the *DatabaseUpdateInterval* should be small but it must be sufficiently large to keep the amount of signaling traffic in the network low. A simple protocol like RTP should be used to perform that task.

### 4.6 Computation of an ER

When a new flow requests a reservation through a network, AC is performed for every link of the path. This process must succeed at all AC points, otherwise the desired reservation can not be set up. Based on its link database, the ingress border router computes using CSPF a constraint-based route that has enough capacity and that can serve as an ER for the reservation setup. The ER may deviate from normal IP routing, therefore, it is convenient that LSPs can realize route pinning. The output of the CSPF algorithm is a constraint-based route consisting of physical and virtual links. This constraint-based route is resolved into a hierarchical ER (HER) that contains the ER together with some information about the intended LSP hierarchy.

### 4.7 Hierarchical Explicit Routes (HER)

An ER is a predefined path and a hierarchical ER (HER) is an ER that contains information about the intended LSP hierarchy concerning the ER. From the properties of an LSP hierarchy [4] one can conclude that the flow-specific LSP hierarchy can be marked by parentheses. The HER $(l_0, (l_1, (l_2, l_3), l_4), l_5)$ (cf. Figure 7) denotes that the flow is first transported over a simple physical link $l_0$, over an LSP along the links $l_1$, $l_2$, $l_3$, and $l_4$, and, finally, over physical link $l_5$. The used LSP consists of the physical link $l_1$, another LSP consisting of physical links $l_2$ and $l_3$, and the physical link $l_4$. This notation does not determine whether LSPs are tunnels or funnels. This may be configured in the routers if there is a suitable signaling protocol.

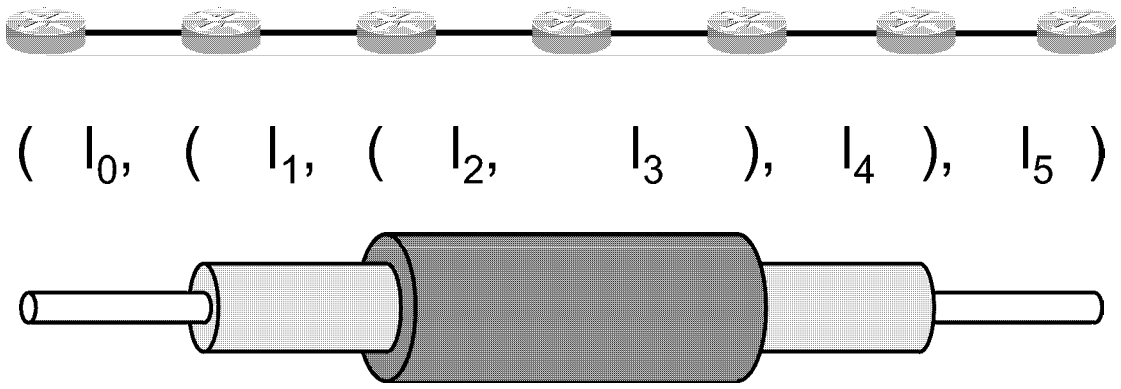


Figure 7: The LSP hierarchy for the hierarchical explicit route $(l_0, (l_1, (l_2, l_3), l_4), l_5)$.

Hence, the ingress border router that determines the ER must also add some hierarchy information to tell the e2e reservation where LSP should be used. The appropriate aggregation hierarchy depends on the used technology and is still a matter of research. The LSPs do not need to exist yet, they may also be constructed on demand which requires some changes to the signaling protocols RSVP and RSVP-TE. The advantage of that approach is that not all possibly required direct and detour LSPs need to be established in advance but only when needed. This reduces again the number of simultaneous states in the routers while keeping the system flexible. Based on the HER, the e2e reservation is set up.

### 4.8 Reservation and LSP Setup

E2E reservations as well as LSPs are set up by initiating a PIP with an ERO and a flowspec as described above. Upon receipt of the PATH message, the receiver triggers the RRP. When the used links dedicate $Tspec$ to the new LSP reservation, they increase in turn their $AllocatedBandwidth$. If all required logical links (i.e. LSPs) in the ERO do already exist, the e2e reservation setup does not differ from normal RSVP signaling.

The setup procedure becomes more complex when LSPs in the ERO do not exist yet. The PIP proceeds until a router recognizes that a desired LSP is not yet established. The router suspends the PIP of the ongoing e2e reservation or LSP setup because the path state can not be set up for the missing link. The construction of the missing LSP is triggered and the first action is the setup of a sleep queue (SQ) that stores all signaling packets as long as the LSP is in an inconsistent state, e.g. until it is established. The PIP of the missing LSP is triggered with an adapted ERO by the router. This may further cascade if other LSPs are missing. When a PIP arrives at its destination, the RRP starts and reserves the resources on every link. If a logical link has not enough free resources anymore, the RESV message may be suspended and put into the SQ of the LSP and the capacity of the link is increased if possible. Note that this may cascade recursively. When the result returns, the RRP is resumed. If the increase succeeded, the RRP continues up to the initiator of the PIP and the e2e reservation or LSP is set up (cf. Figure 8). Otherwise, the setup is rolled back and an error is signaled. If the PIP or RRP have failed, an error is returned to the initiator of the PIP. In both cases, all items in the SQ are processed and appropriate actions like forwarding of PATH messages or returning of error messages are performed.

### 4.9 Reservation Teardown

When a reservation is torn down, it gives its capacity ($Tspec$) back to its supporting links and they decrease their $AllocatedBandwidth$. If their $AllocatedBandwidth$ of the LSPs among the supporting links falls below their $UpdateThreshold$, these trigger a decrease of their bandwidth for resource efficiency. This action may cascade to higher levels in the hierarchy. In contrast to capacity increases, this is done asynchronously concerning the reservation teardown process. When a supporting link has finally no other reservations or LSPs to support and its SQ is also empty, it may wait for another $TeardownDelay$ time until it triggers its own teardown. This should be done to leave as few LSPs alive as possible. There are two possibilities how the LSP can realize that it is not needed anymore. If PATH states are used, there must be no PATH state for the LSP waiting for the establishment of a RESV state. If stateless PIPs are used (like PROBEs in BGRP), they should leave a timestamp at the ingress LSR. After sufficiently long time one can assume that no corresponding RRP message will return and the empty LSP can be torn down.

### 4.10 Transport Resilience

In this work we have left out the resilience aspect. RSVP(-TE) has the capability of rerouting when network links fail and the routing protocols converge again. This is not intended in the described system. The carried traffic has real-time requirements and, therefore, it is crucial that

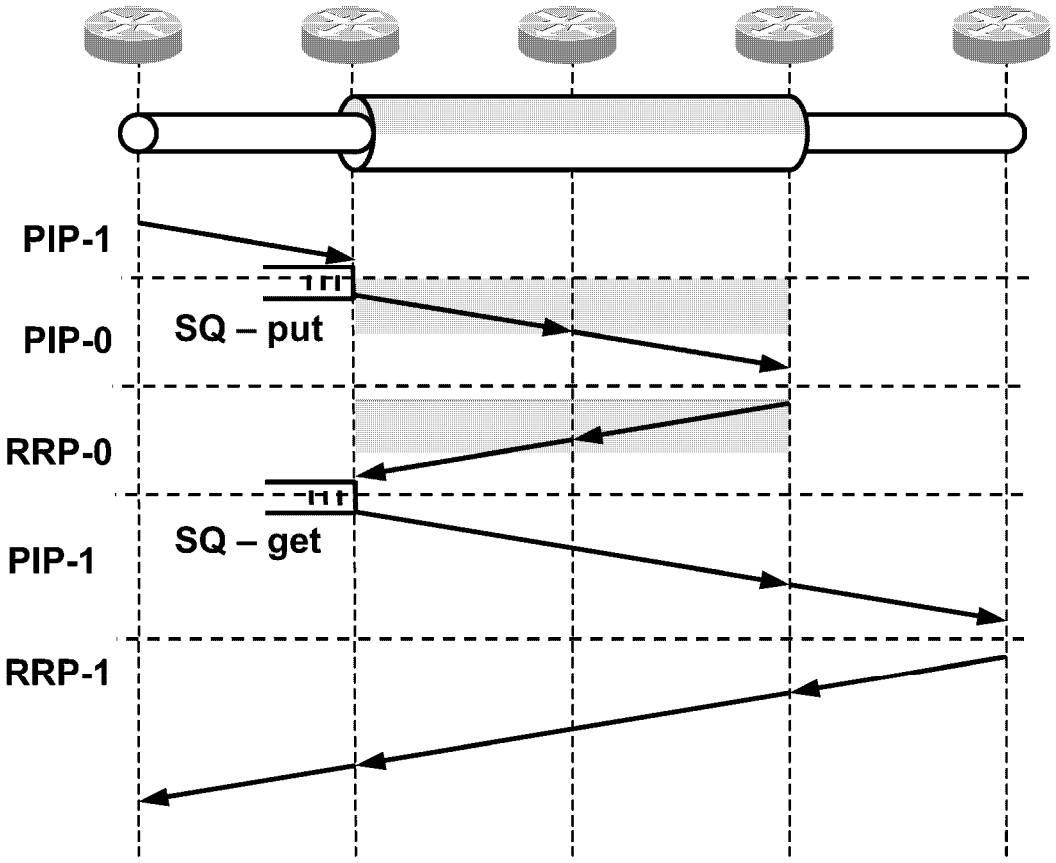


Figure 8: The PIP is suspended for the construction of a higher level LSP.

there are enough resources available on the rerouted path. Otherwise, AC fails and so does the rerouting approach. MPLS technology offers fast rerouting mechanisms for failure scenarios using precomputed backup LSPs with some shared backup capacity. Backup capacities may also rely on other resources whose traffic may be preempted if necessary. This is favorable especially in muli-service transport networks. This is also subject for further research.

### 4.11 Differences to IntServ

At first sight, the presented network architecture is pretty close to IntServ because we have only considered the reservation process so far. But our approach is more scalable due to reservation aggregation. The aggregated reservations are supported by LSPs whose packet can be also classified in a scalable manner using only a few MPLS labels. Tracking of individual flows for single reservations is not anymore required in the core. Packets are policed and shaped only on an aggregate basis.

Another important difference to IntServ is that traffic handling works on an aggregate basis in the routers. As in DiffServ, packets are scheduled for transmission according to the DSCP in their IP packet header which can also be inherited by an LSP. Only traffic with the

same DSCP value may be aggregated in an LSP, i.e. this attributed can be stored with the forwarding tables and the DSCP for labeled packets can be inferred. This allows for simple service differentiation while admission control is maintained.

The traffic with less stringent QoS requirements may consume the bandwidth left over by the real-time flows. This is realized by appropriate scheduling mechanisms in the routers [30, 31] that are not considered in this work.

### 4.12 Differences to Conventional DiffServ Bandwidth Brokers

The above proposed architecture has two different types of databases.

The routers store the flows and aggregates thereof that use their links in their MIBs. This information is required for local AC and resource management purposes. Every individual flow in the network is known at least in the ingress and egress border router. But inside the network they are concealed by aggregate tunnels. All these data are timely accurate because they must prevent overload on links such that real-time guarantees can be granted. Note that a hierarchical network structure is important for the scaling of this approach.

In contrast to DiffServ without bandwidth brokers, our architecture is able to support real-time guarantees. We explicitly state that our approach is different from a central DiffServ bandwidth broker because of the better scaling properties. A bandwidth broker in DiffServ stores all flows in the network in a more or less central database. The amount of that information is proportional to the overall number of flows in the network which does not work in large networks for scalability reasons. In addition, flow policing can be done in the network based on LSP classification and not only according to DSCP aggregates.

The link databases have a global view on the *AvailableBandwidth* of all links in the network and provide this information for CSPF computation at the ingress routers. Hence, the size of the link databases is proportional to the number of links in the network that are also advertised by routing protocols. The information quantity is independent of the number of transported flows. If these data are obsolete, the derived EROs might lead to increased blocking probability of reservations inside the network. However, obsolete data might decrease the resource efficiency but they can not corrupt the QoS of already admitted flows. This feature can reduce the flow blocking probability and increase the network utilization for real-time traffic. This does neither exist in DiffServ prototypes because route pinning is hard to be done with conventional IP routing.

## 5 Conclusion

In this paper, we gave a short overview of current mechanisms to support QoS in data communication networks. We have presented the aggregation technique as a means to achieve reservation state scalability in the routers. Aggregation for reservations can be implemented using tunnels or funnels. Tunnel reservations can be realized using LSPs while funnel reservations are only signaled by BGRP in an inter-networking context so far. We explained the basic signaling procedures for both approaches.

Our main contribution is the description of a DiffServ-based network architecture that

relies on e2e reservations. Individual reservations are aggregated by bandwidth-adaptive LSPs in the core which leads to reservation state scalability. We suggested to use a low degree of overreservation for the aggregate reservations coupled with a hysteresis to achieve signaling scalability. We suggested operations for an automated, dynamic, and incremental setup of such an LSP hierarchy. These operations reuse existing label distribution and resource reservation protocols as much as possible.

The available resources on the network links are signaled regularly to the resource utilization databases at the ingress border routers which use that information to compute constraint-based routes using the CSPF algorithm. MPLS helps to establish these explicit routes. The integration of CSPF reduces the blocking probability for new requests in a highly loaded network.

Our solution scales well if the network structure is hierarchical. The LSP hierarchy setup is automated, therefore, it is still a low-cost solution since no human interaction is required. Apart from AC, traffic engineering is performed by the integration of CSPF and leads to an optimized resource utilization. We believe, therefore, that this architecture may be one step towards scalable next generation QoS networks.

For future work, there are still some open issues concerning the presented network architecture. Those may be evaluated by simulations or analytical investigations. Parameters like the *DatabaseUpdateInterval*, *NoResourceInterval*, and others need to be set in an appropriate way. MPLS may also be used for fast link and node failure recovery. For resource efficiency, thorough planning of backup LSPs using shared resources or preemption of other reservations is a crucial issue.

## References


[1] B. Braden, D. Clark, and S. Shenker, "RFC1633: Integrated Services in the Internet Architecture: an Overview." http://www.ietf.org/rfc/rfc1633.txt, June 1994.

[2] J. Wroclawski, "RFC2210: The use of RSVP with IETF integrated services." ftp://ftp.isi.edu/in-notes/rfc2210.txt, Sep. 1997.

[3] E. C. Rosen, A. Viswanathan, and R. Callon, "RFC3031: Multiprotocol Label Switching Architecture." http://www.ietf.org/rfc/rfc3031.txt, Jan. 2001.

[4] M. Menth and N. Hauck, "A Graph-Theoretical Concept for LSP Hierarchies," Technical Report, No. 287, University of Würzburg, Institute of Computer Science, Nov. 2001.

[5] P. Ashwood-Smith, B. Jamoussi, D. Fedyk, and D. Skalecki, "Improving Topology Data Base Accuracy with Label Switched Path Feedback in Constraint Based Label." http://www.ietf.org/internet-drafts/draft-ietf-mpls-te-feed-05.txt, Nov. 2002.

[6] M. Menth, "A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks," in $17^{th}$ *International Teletraffic Congress*, (Salvador de Bahia, Brazil), pp. 211–222, Dec. 2001.

[7] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification." ftp://ftp.isi.edu/in-notes/rfc2205.txt, Sep. 1997.

[8] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie, "RFC3175: Aggregation of RSVP for IPv4 and IPv6 Reservations." http://www.ietf.org/rfc/rfc3175.txt, Sept. 2001.

[9] A. Terzis, J. Wang, J. Ogawa, and L. Zhang, "A Two-Tier Resource Management Model for the Internet," in *Global Internet Symposium'99*, Dec. 1999.

[10] M. Günther and T. Braun, "Evaluation of Bandwidth Broker Signaling," in *International Conference on Network Protocols ICNP'99*, pp. 145–152, Nov. 1999.

[11] G. A. Politis, P. Sampatakos, and I. Venieris, "Design of a Multi-Layer Bandwidth Broker Architecture," in *Interworking*, (Bergen, Norway), 2000.

[12] B. Teitelbaum, S. Hares, L. Dunn, V. Narayan, R. Neilson, and F. Reichmeyer, "Internet2 QBone: Building a Testbed for Differentiated Services," *IEEE Network Magazine*, Sep. 1999.

[13] Z.-L. Z. Zhang, Z. Duan, and Y. T. Hou, "On Scalable Design of Bandwidth Brokers," *IEICE Transaction on Communications*, vol. E84-B, pp. 2011–2025, Aug 2001.

[14] D. O. Awduche, L. Berger, D.-H. Gan, T. Li, V. Srinivasan, and G. Swallow, "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels." http://www.ietf.org/rfc/rfc3209.txt, Dec. 2001.

[15] B. Jamoussi *et al.*, "RFC3212: Constraint-Based LSP Setup using LDP." http://www.ietf.org/rfc/rfc3212.txt, Jan. 2002.

[16] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification." http://www.ietf.org/rfc/rfc3036.txt, Jan. 2001.

[17] M. Menth and N. Hauck, "A Graph-Theoretical Notation for the Construction of LSP Hierarchies," in $15^{th}$ *ITC Specialist Seminar*, (Würzburg, Germany), July 2002.

[18] K. Kompella and Y. Rekhter, "LSP Hierarchy with Generalized MPLS TE." http://www.ietf.org/internet-drafts/draft-ietf-mpls-lsp-hierarchy-08.txt, March 2002.

[19] H. Hummel and J. Grimminger, "Hierarchical LSP." http://www.ietf.org/internet-drafts/draft-hummel-mpls-hierarchical-lsp-01.txt, May 2002.

[20] H. Hummel and B. Hoffmann, "O(n**2) Investigations." http://www.ietf.org/internet-drafts/draft-hummel-mpls-n-square-investigations-00.txt, June 2002.

[21] T. Li and Y. Rekhter, "RFC2430: A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)." ftp://ftp.isi.edu/in-notes/rfc2430.txt, Oct. 1998.

[22] P. Pan and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations," *Journal of Communications and Networks*, vol. 2, pp. 157–167, June 2000.

[23] G. Fehér, K. Németh, M. Maliosz, I. Czslényi, J. Bergkvist, D. Ahlard, and T. Engborg, "Boomerang - A Simple Protocol for Resource Reservation in IP Networks," in *"IEEE Workshop on QoS Support for Real-Time Internet Applications "*, (Vancouver, Canada), June 1999.

[24] M. Menth and R. Martin, "Performance Evaluation of the Extensions for Control Message Retransmissions in RSVP," in $7^{th}$ *International Workshop on Protocols For High-Speed Networks (PfHSN 2002)*, (Berlin, Germany), 2002.

[25] T. Engel, E. Nikolouzou, F. Ricciato, and P. Sampatakos, "Analysis of Adaptive Resource Distribution Algorithsm in the Framework of a Dynamic DiffServ IP Network," in $8^{th}$ *International Conference on Advances in Communications and Control (ComCon8)*, (Crete, Greece), June 2001.

[26] B. F. Koch, "A QoS Architecture with Adaptive Resource Control: The AQUILA Approach," in $8^{th}$ *International Conference on Advances in Communications and Control (ComCon8)*, (Crete, Greece), June 2001.

[27] C.-Y. Lee, B. Ganti, S. Hass, and V. Naidu, "Path Request and Path Reply Message." http://www.ietf.org/internet-drafts/draft-lee-mpls-path-request-04.txt, Nov. 2002.

[28] J.-P. Vasseur, C. Iturralde, R. Zhang, X. Vinet, S. Matsushima, and A. Atlas, "RSVP Path Computation Request and Reply Messages." http://www.ietf.org/internet-drafts/draft-vasseur-mpls-computation-rsvp-03.txt, June 2002.

[29] H. Fu and E. Knightly, "Aggregation and Scalable QoS: A Performance Study," in *Proceedings of IWQoS 2001*, (Karlsruhe, Germany), June 2001.

[30] M. Menth, M. Schmid, H. Heiß, and T. Reim, "MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in the UTRAN," in *IEEE INFOCOM'03*, (San Francisco, USA), Mar. 2003.

[31] T. Bonald and J. W. Roberts, "Performance of Bandwidth Sharing Mechanisms for Service Differentiation in the Internet," in $13^{th}$ *International Teletraffic Congress Specialist Seminar*, (Monterey, USA), September 2000.