CS 856
Internet Transport Performance

Network Control:
Routing & Signalling

Martin Karsten

School of Computer Science, University of Waterloo
mkarsten@uwaterloo.ca

Contents

Routing

Multi-Protocol Label Switching

Network Signalling

Discussion

Naming & Addressing

Name

- human readable identification of host, service, etc.
- location-dependency of name: centralized or distributed lookup?
- complexity/overhead of name lookup?

Address

- topological relevance: encoding of network access point
- entity which is used for routing

Example: Mobile Phone Number

- strictly speaking: neither name nor address?

Datagram Networks

- simple/limited addressing required
  - routing of each packet
- vs. virtual circuit: complex addressing more acceptable
  - routing of path setup only

Flat vs. Hierarchical Routing

Flat Routing

- **large global routing tables (distributed storage possible)**
- **global scope of routing updates → overhead & error-prone system**

Hierarchical Routing

- **goal: reduce size of routing tables**
- **address: encoding of network access point & host identifier**
- **old Internet addressing: subnetwork classes**
  - class A: 2^7 networks with up to 2^{24}-2 hosts
  - class B: 2^{14} networks with up to 2^{16}-2 hosts
  - class C: 2^{21} networks with up to 2^8-2 hosts
  - plus some special classes
- **observation: most networks are between class B and C**
  - exhaustion of class B address space
  - potential administrative solution: enforcement of network structure
    - multiple smaller networks need to team up as a class B network
    - and internally structure themselves as set of class C networks
  - not a very good solution!
    - still need modification is routing system → classes are hard-coded
    - lack of flexibility

Dynamic Routing Hierarchy

Classless Inter-domain Routing (CIDR)

- **explicit representation of subnet length in routing information**
  - e.g. 10.4.12.0/22 represents all IP addresses in 10.4.12.0 - 10.4.15.255
- **more flexible allocation of IP addresses to networks**
- **route aggregation on contiguous addressing ranges**
  - e.g. 10.4.12.0/22 and 10.4.8.0/22 → 10.4.8.0/21
  - e.g. 10.4.8.0/21 and 10.4.0.0/22 → no aggregation without 10.4.4.0/22
  - when forwarding route advertisements
- **arbitrary aggregation possible**

Route Lookup for Packet Forwarding

- **critical for datagram networks → performance**
- **multiple routing entries may exist**
  - e.g. entry for 10.4.0.0/20 and 10.4.8.0/22
  - 10.4.0.0/20 is a possible route, but a better route is known to 10.4.8.0/22
    - e.g. learned from a different peer router
    - 10.4.8.0/22 may be multi-homed through different ISPs
  - prefer 2 overlapping entries over 16 disjoint entries
- **address matches multiple entries in routing table → find longest match**

Address Space Limitations

IPv6

- extends IP address space to 128 bit
- deployment slower than earlier predictions

Network Address Translation (NAT)

- local address within local network
- dynamic address translation at access gateway
  - side effect: no disclosure of internal structures
- additional level of hierarchy
  - hosts in 2nd level have restricted capabilities

Generic Overlay Networks

- IP over ATM
- IPv6 over IPv4
- IPv4 over IPv4
- etc.

Fault Recovery in IP Routing

Failure Detection

- **explicit link monitoring**
- **HELLO messages (periodic)**

System Healing

- **link-state routing**
  - propagation of global updates
  - local route computation at each node
- **path/distance-vector routing**
  - local updates and count-to-infinity problem
  - even longer propagation delays for changes
- **frequency of faults vs. speed of convergence?**

Multi-Protocol Label Switching (MPLS)

IP Forwarding

- **stateless**
- **packet-switched**
- **packet forwarding: longest-prefix lookup**

Label Switching

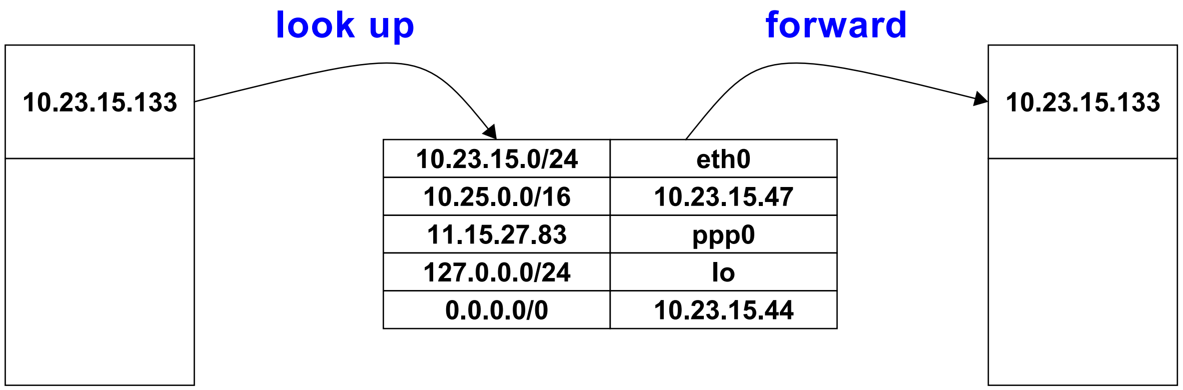
- **virtual-circuit approach**
  - "connection" here: **FORWARDING EQUIVALENCE CLASS** (FEC)
  - arbitrary topological scope of FEC (flows, trunks, etc.)
- **assign local label to FEC**
- **forward packets according to label**
- **multiple links form LABEL SWITCHED PATH (LSP)**
- **control protocol needed for label distribution**

MPLS

- **technical functionality: network layer**
- **conceived as intermediate layer between various data link layers and IP**
  - particularly: exploit ATM switching technology without ATM signalling
- **inter-operates with any link and any network protocol**

Packet Switching vs. Label-Switching

Packet Switching



look up

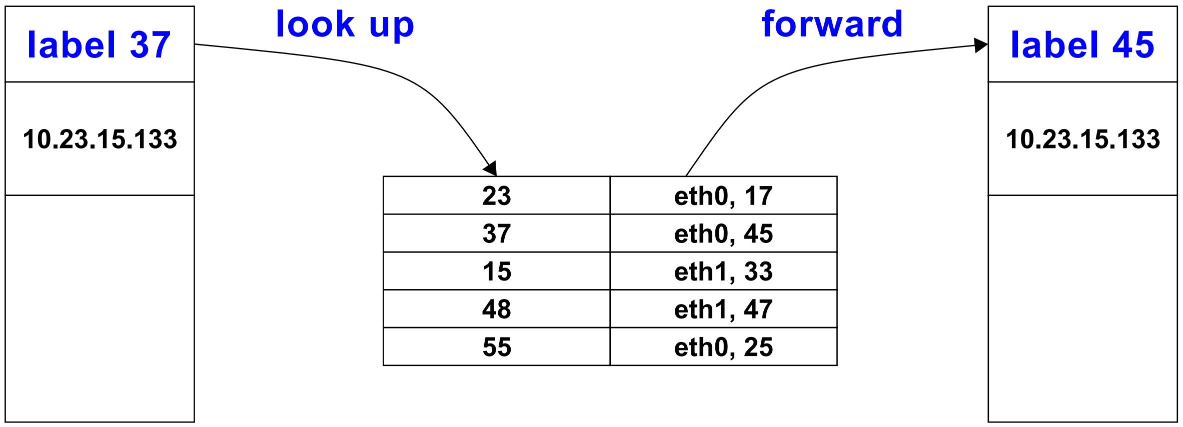
forward

10.23.15.133

10.23.15.0/24	eth0
10.25.0.0/16	10.23.15.47
11.15.27.83	ppp0
127.0.0.0/24	lo
0.0.0.0/0	10.23.15.44

10.23.15.133

Label Switching



label 37

10.23.15.133

look up

forward

23	eth0, 17
37	eth0, 45
15	eth1, 33
48	eth1, 47
55	eth0, 25

label 45

10.23.15.133

Label Encoding

32 Bit Shim Header between L2- and L3-Header

0 10 20 23 31

Label	Exp	S	TTL

- Label: Label Value, 20 bits
- Exp: Experimental Use, 3 bits
- S: Bottom of Stack, 1 bit
- TTL: Time to Live, 8 bits

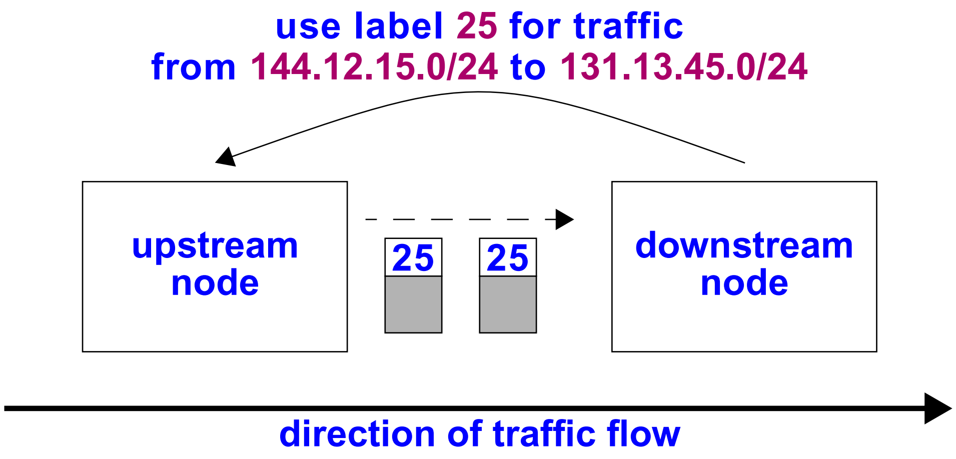
Label Stacking

- push label in front of stack
- create aggregate trunks while preserving flow identification
- extended version of ATM’s VCI/VPI

Label Assignment

Unique Label Binding Needed

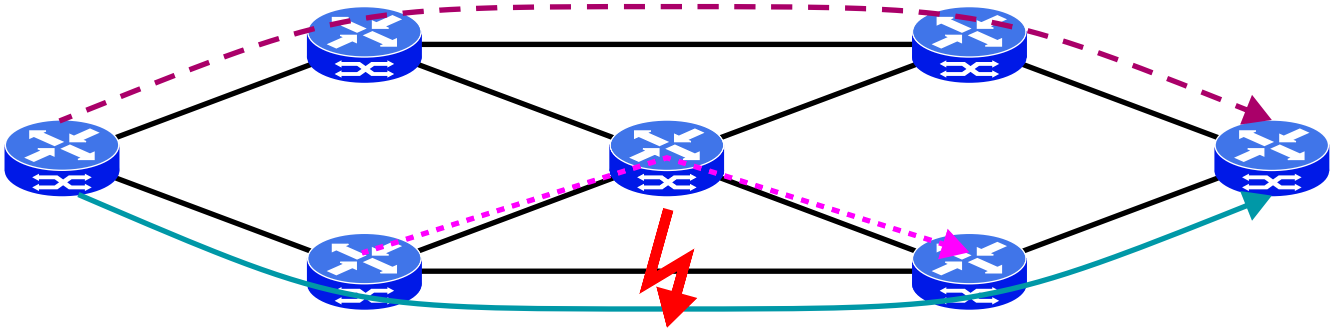
- **downstream node announces label to upstream node**
- **downstream node chooses label scope**
  - global scope: (label) → lookup
  - interface scope: (label, interface) → lookup
  - no previous hop information available


use label 25 for traffic
from 144.12.15.0/24 to 131.13.45.0/24
upstream node
25
25
downstream node
direction of traffic flow

Label Assignment Modes

- **downstream on demand → upstream node requests label**
- **unsolicited downstream**

Recovery Options



working LSP

protection LSP

local rerouting

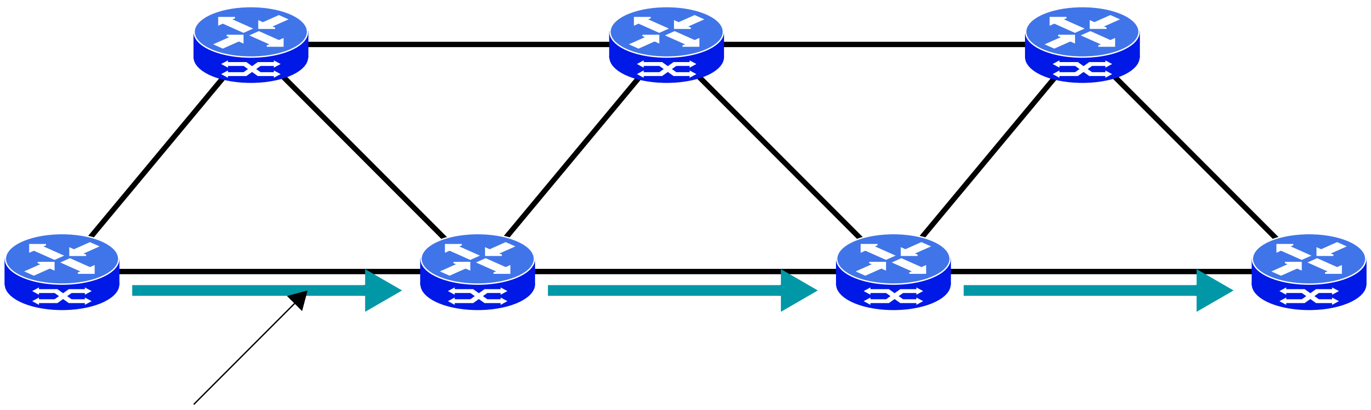
LSP Setup

- **explicit routing of LSP**
- **resource allocation for LSP**
  - protection LSP: pre-reserved or on-demand

Protection LSP

- **disjoint path (if possible)**
- **fast failure detection needed**

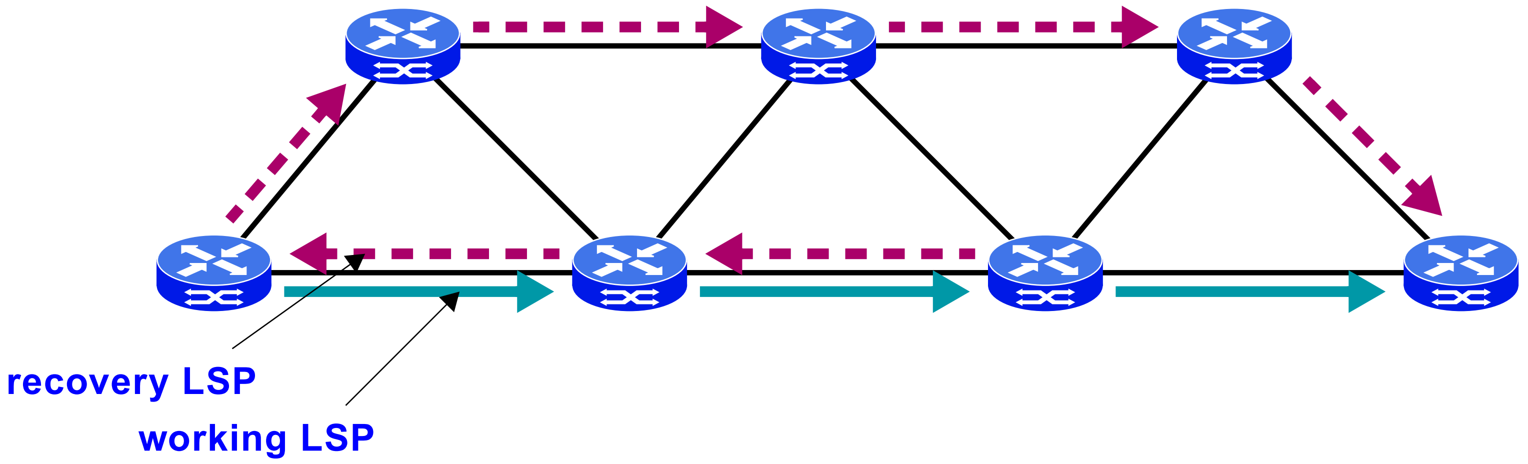
Fast Reroute



working LSP

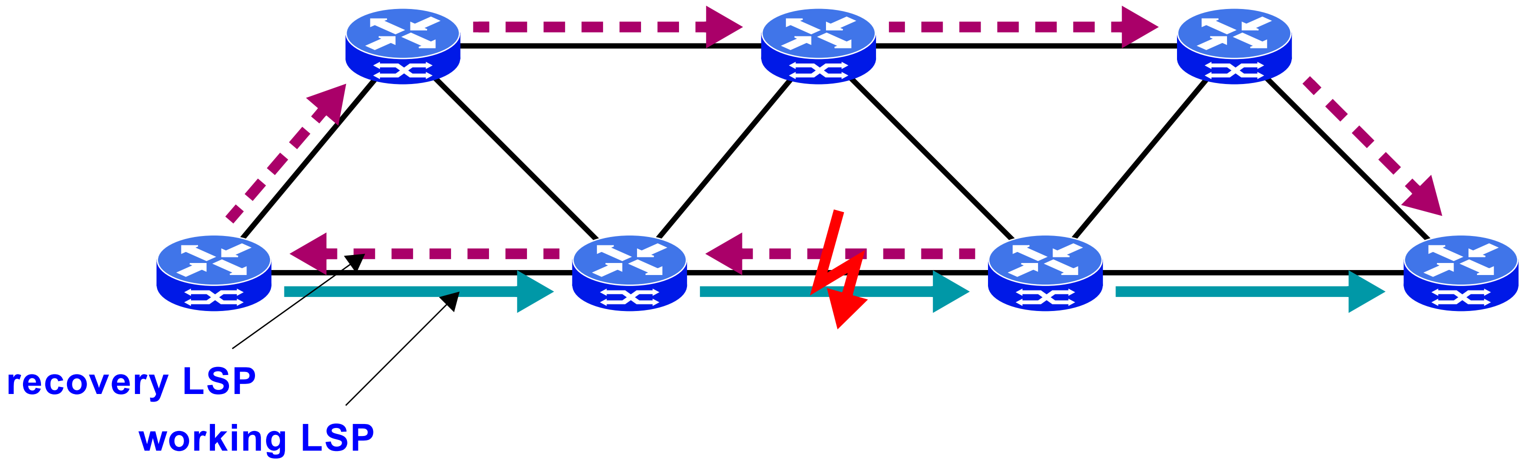
Establish Primary LSP

Fast Reroute



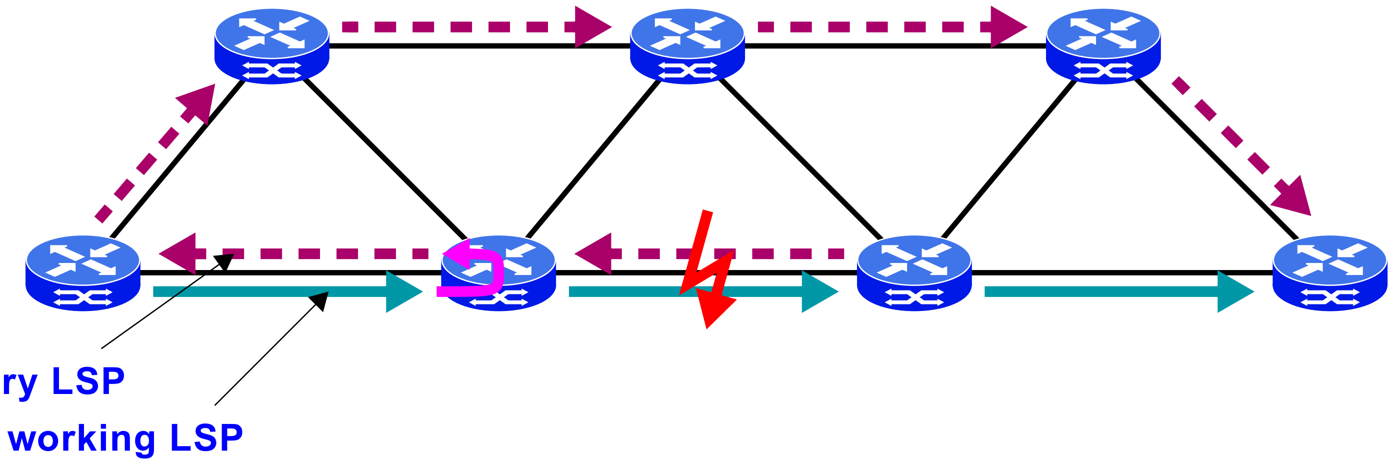
Establish Backup LSP

Fast Reroute



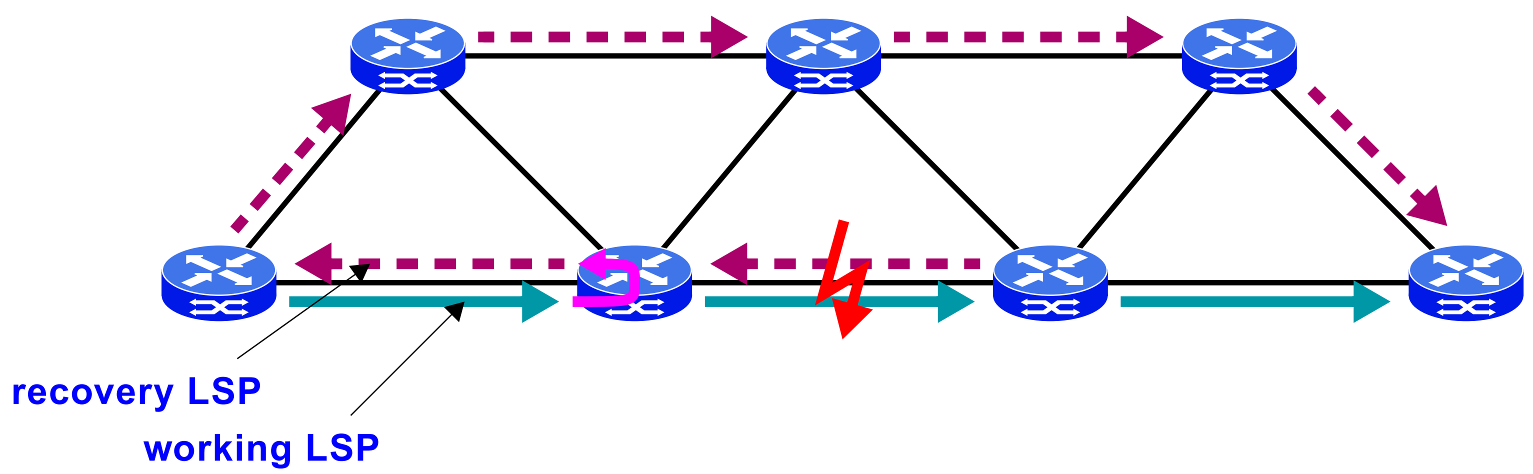
Intermediate Node Discovers Link Failure

Fast Reroute



Node Can Immediately Reroute Traffic

Fast Reroute



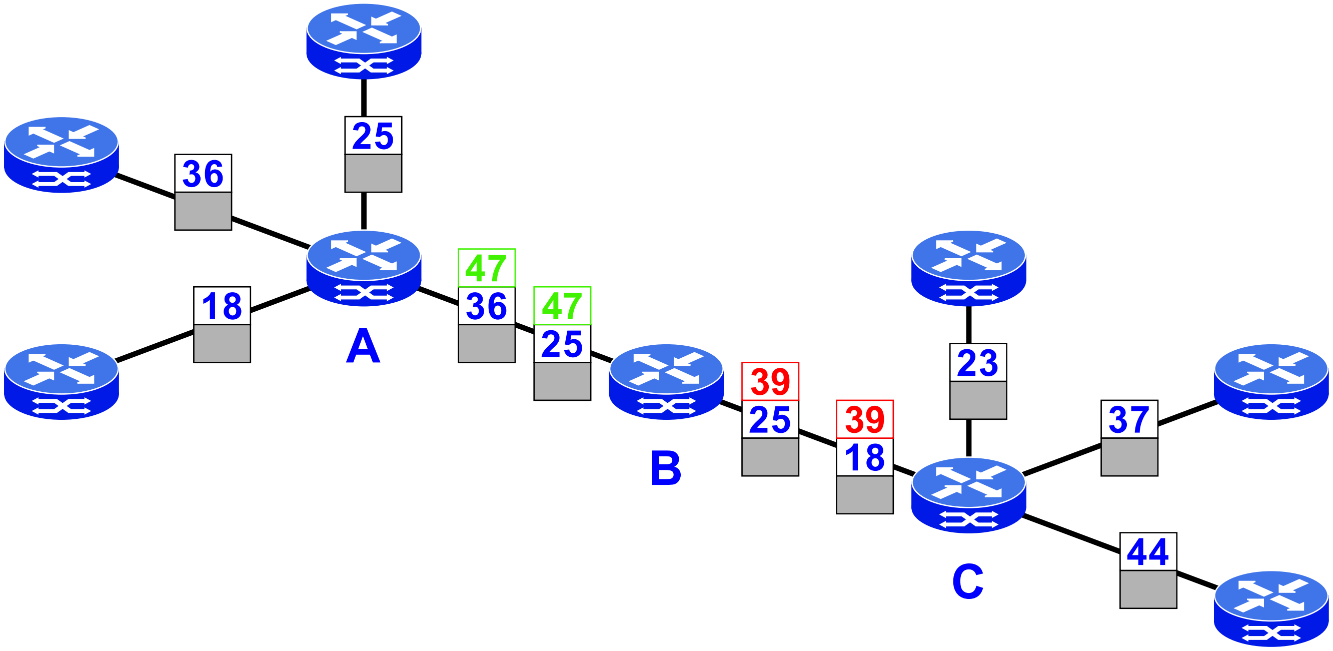
Recovery LSP

- **automatic establishment from last-hop switch in reverse direction**
- **along disjoint path from source to destination**

Upon Failure

- **adjacent upstream node redirects traffic**
  - similar to e.g. FDDI ring protection
- **later: source node redirects traffic**
- **lossless recovery possible**
  - depending on speed of link failure detection

Tunnels and Hierarchies



Tunnel A → C

- tunnel ingress A
- LSP via B (label 47), label push
- from B to C (label 39), label switching
- tunnel egress C, label pop, label switching & forwarding

Routing Mix in the Internet

Inter-domain Routing

- BGP
- long-term traffic contracts
- packet forwarding: IP

Intra-domain Routing

- OSPF
- IS-IS
- static configuration
- packet forwarding: IP or MPLS

Other NBMA Technologies (Intra-Domain)

- NBMA = Non-Broadcast Multiple Access
  - subnet technology with own addressing/routing function
- ATM
- Sonet/SDH

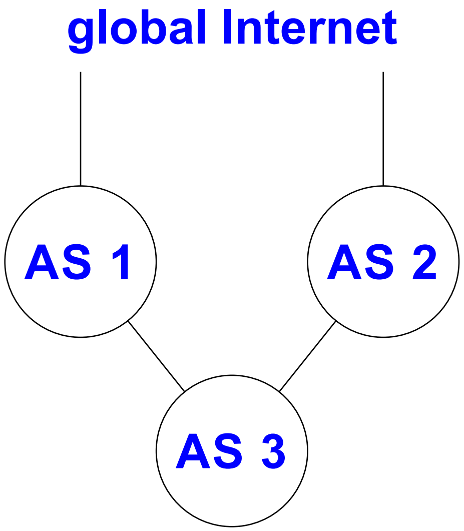
Routing Problems in the Internet

Changes in Network Structure ⇒ Routing State "Explosion"

- **global Internet evolves from tree towards denser mesh**
  - end-user multi-homing
  - regional peering between ISPs

Example

- **AS3 receives address range from AS1**
- **AS3 also advertises through AS2**
- **AS2 cannot aggregate AS3 info**
- **later in the network:**
  - AS3 via AS2 info is more specific than AS1 aggregate ⇒ longest-prefix matching directs all traffic via AS2
  - AS1 needs to announce AS3 specific rather than aggregated ⇒ more state information



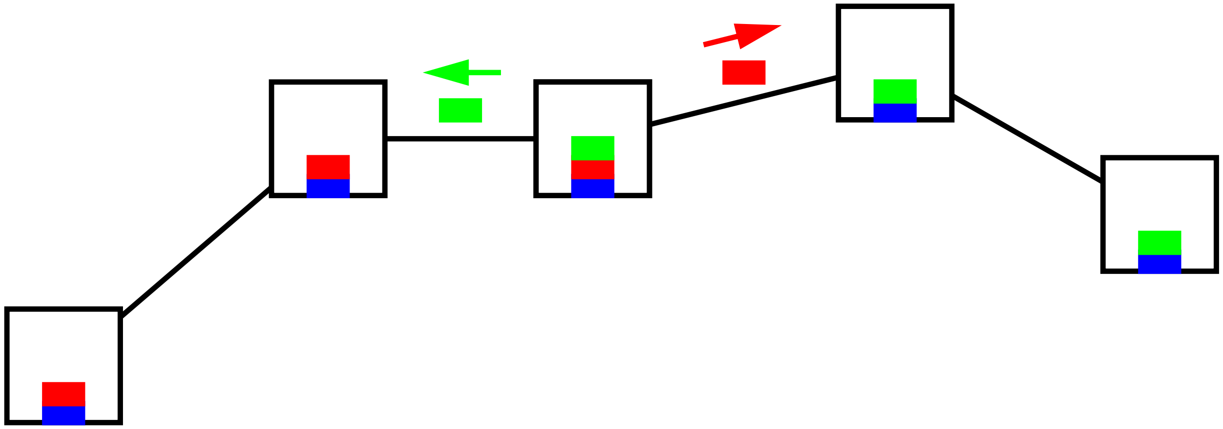
Routing Convergence

- **fast reaction to routing changes → route flapping**
- **⇒ reaction to changes on the order of seconds and minutes**
- **⇒ slow global convergence**
- **local configuration & policy vs. global goals**

Network Signalling

Transmission of State Information between "Neighbours"

- relationship to network path (explicit or not)



- path setup
- QoS signalling
- firewall traversal

Issues

- state complexity
- message transmission overhead
- protocol complexity
- consistency → system convergence

Hard State vs. Soft State

Goals

- system convergence → stability after state change
  - avoid manual maintenance
- fast recovery → immediate problem resolution after failure

Hard State

- transmit state once, receive acknowledgement
- detect all errors
- correct errors
- combination of convergence and recovery

Soft State

- transmit state periodically, no acknowledgement
  - idempotent messages
- ignore errors
- automatic error correction
- optimisation (fast recovery): detect and correct errors
- convergence + optional fast recovery

Resource Reservation Protocol (RSVP)

RFC 2205

Conceived as Signalling Protocol for Integrated Services Architecture

- not limited to this scenario

Design Goals

- multi-sender and multi-receiver
- heterogeneous multicast
- dynamic multicast group membership
- aggregation within multicast group and for multiple senders
- selection of senders
- independent of routing
- adaptive to routing changes
- robustness
- controlled protocol overhead

Design Principles

Receiver-Initiated Reservation

- receiver knows best which QoS to ask for
- adopt IP multicast model
- allow for heterogeneous receivers

Separating Reservation from Packet Filtering

- allow for dynamic filter changes

Different Reservation Styles

- multi-sender applications
- shared vs. fixed reservations
- explicit vs. wildcard reservations

	shared	fixed
explicit	SE	FF
wildcard	WF	--

Design Principles (2)

Soft State

- periodic refresh of state information (otherwise state times out)
- compromise between stateful and stateless
- stateful, but robust
- "hard state" vs. "soft state"

Protocol Overhead Control

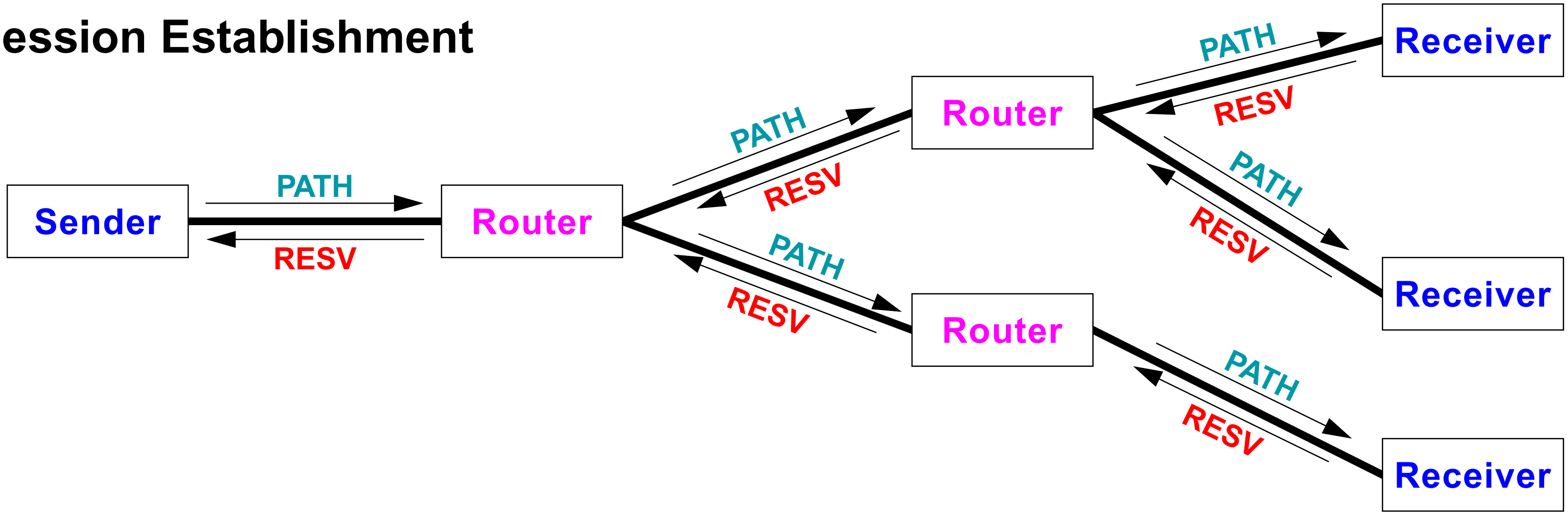
- merging of reservation messages along the multicast tree
- configurable refresh timeout for soft state

Modular Architecture

- decoupling of services from signalling protocol
    - decoupling of service enforcement (admission control and traffic control)
- decoupling of signalling and routing
    - RSVP does not influence routing
    - eventually RSVP & routing should cooperate
    - see discussion later

RSVP Operation

Session Establishment



Two-Way Session Setup

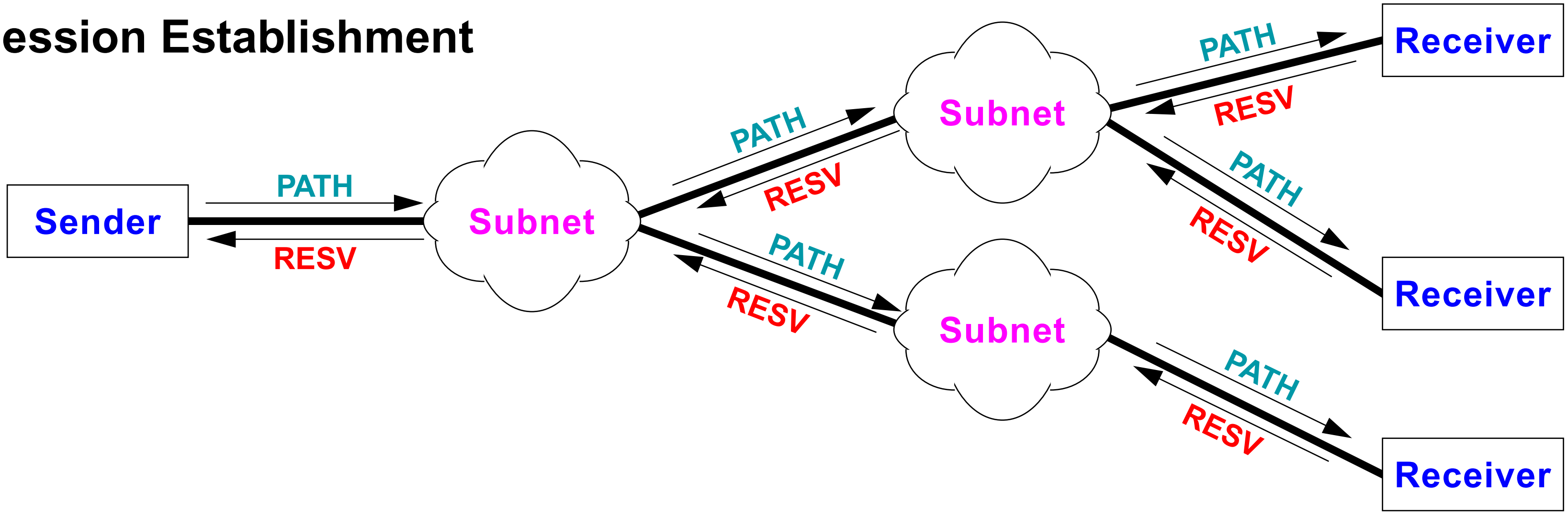
- one-pass with advertising
- PATH message follows data path
- reverse path is stored hop-by-hop at intermediate nodes
- RESV message is transmitted along reverse path

Soft State

- asynchronous refresh between nodes
- independent refresh frequency

Alternative Operation

Session Establishment



Traversal of RSVP-unaware clouds

- PATH message is regularly routed through subnet
- RESV message is addressed to previous RSVP-capable hop
- service guarantees have to be ensured by other means

Protocol Messages

Messages Composed of Objects

SESSION
- destination address, destination port, protocol number

SENDER_TEMPLATE/FILTER_SPEC
- sender address, port number

SENDER_TSPEC
- traffic description: token bucket

FLOWSPEC
- QoS description: rate allocation

ADSPEC
- characteristics of transmission path

RSVP_HOP
- sending node of protocol message

Others
- INTEGRITY, TIME_VALUES, ERROR_SPEC, SCOPE, STYLE, POLICY_DATA, RESV_CONFIRM

Message Types

PATH

- sender → receiver
- traffic announcement
- establishment of path
- path characteristics: intermediate nodes → receiver

RESV

- receiver → sender
- QoS request
- reverse transmission along established path

PTEAR

- sender → receiver
- path teardown

RTEAR

- receiver → sender
- reservation teardown

Message Types (2)

PERR

- **intermediate node → sender**
- **error when establishing the path**

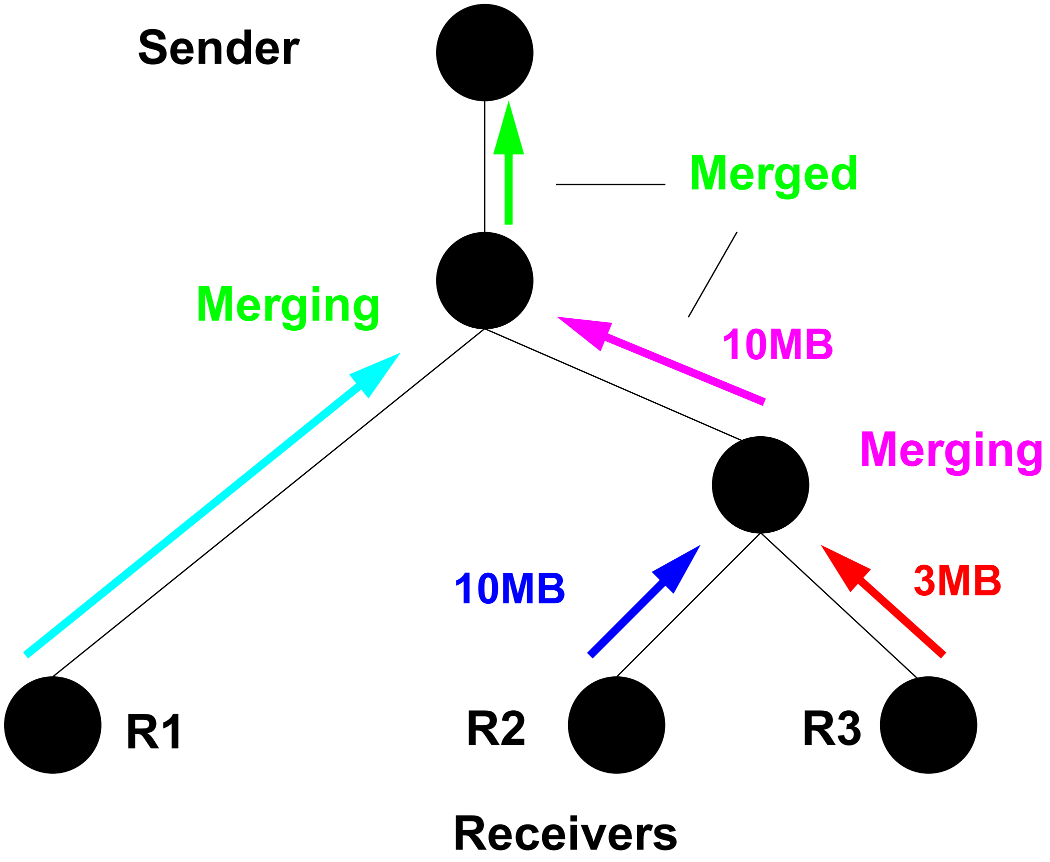
RERR

- **intermediate node → receiver**
- **error when establishing the end-to-end reservation**
  - e.g. admission control failure

RCONF

- **intermediate node/sender → receiver**
  - depending on previously established reservation
  - branching node in multicast tree
- **confirmation of reservation**
  - not reliable

RSVP – Merging of Reservations



Merging – Fixed-Filter Style

s*: senders

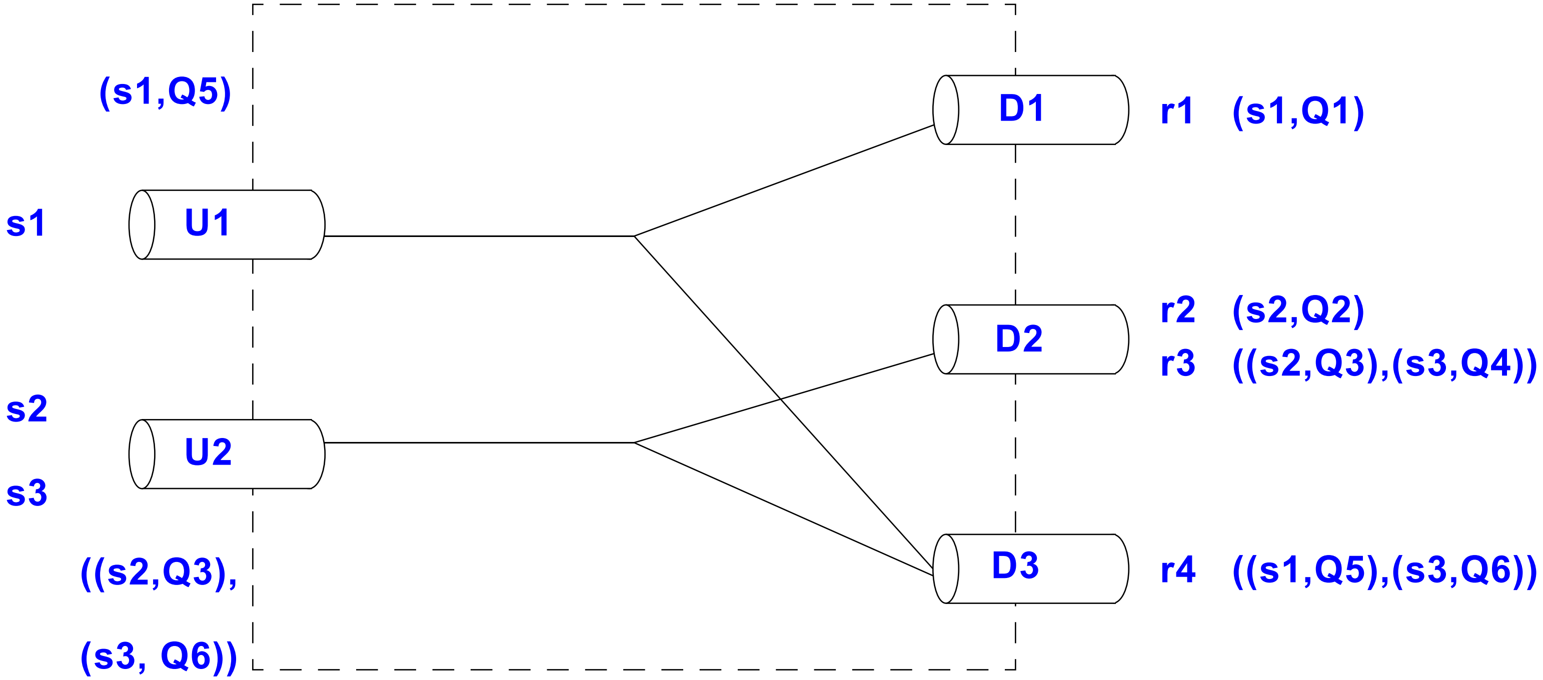
r*: receivers

Q*: FlowSpec

U*: upstream interfaces

D*: downstream interfaces

assume: Q1 < Q2 < Q3 < Q4 < Q5 < Q6



(s1,Q5)

s1

U1

s2

U2

s3

((s2,Q3),

(s3, Q6))

D1

r1 (s1,Q1)

D2

r2 (s2,Q2)

r3 ((s2,Q3),(s3,Q4))

D3

r4 ((s1,Q5),(s3,Q6))

Each interface reserves maximum of received reservations for each source

Separate reservation sent to each requested source

Merging – Shared-Explicit-Filter Style

s*: senders

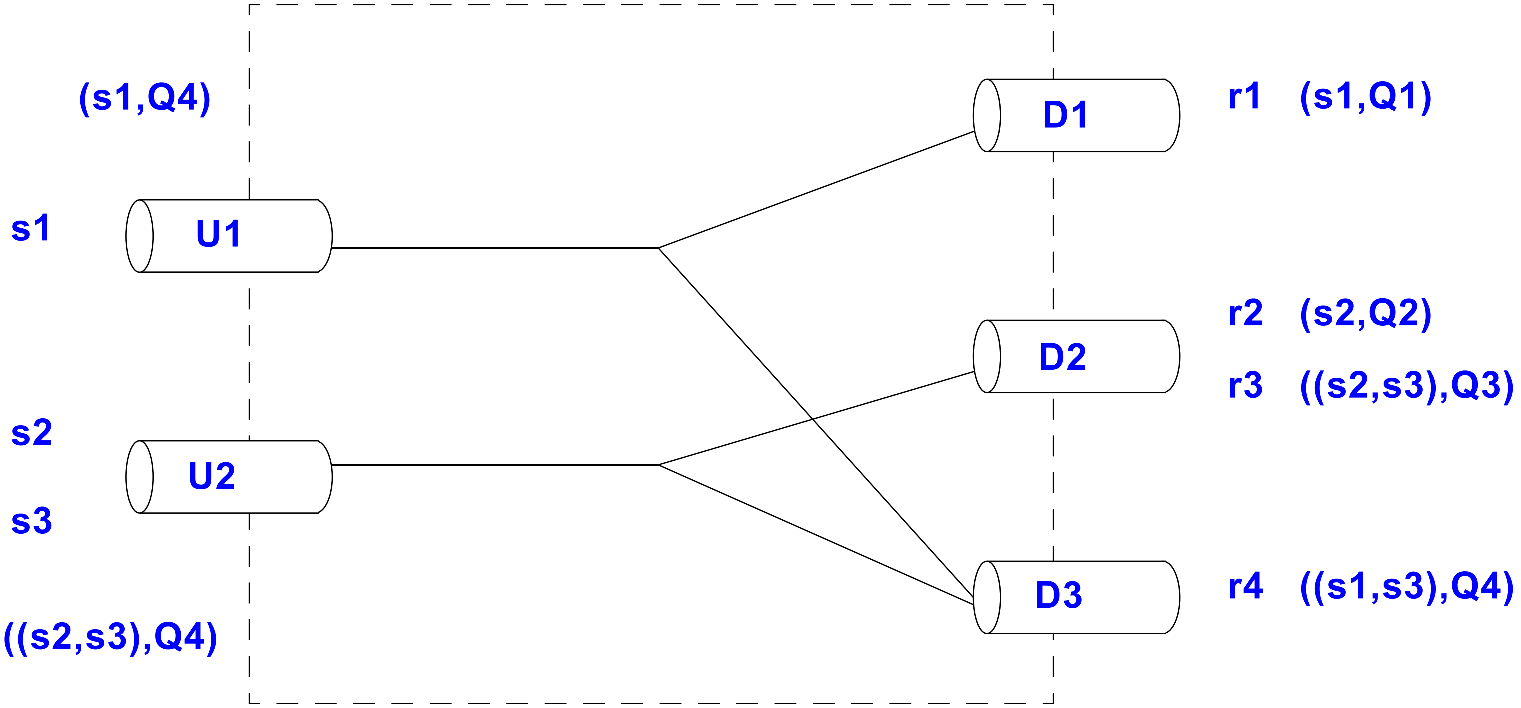
r*: receivers

Q*: FlowSpec

U*: upstream interfaces

D*: downstream interfaces

assume: Q1 < Q2 < Q3 < Q4 < Q5 < Q6



FilterSpec of merged reservations is union of FilterSpecs

FlowSpec of merged reservations is maximum FlowSpec

Merging – Wildcard-Filter Style

s*: senders

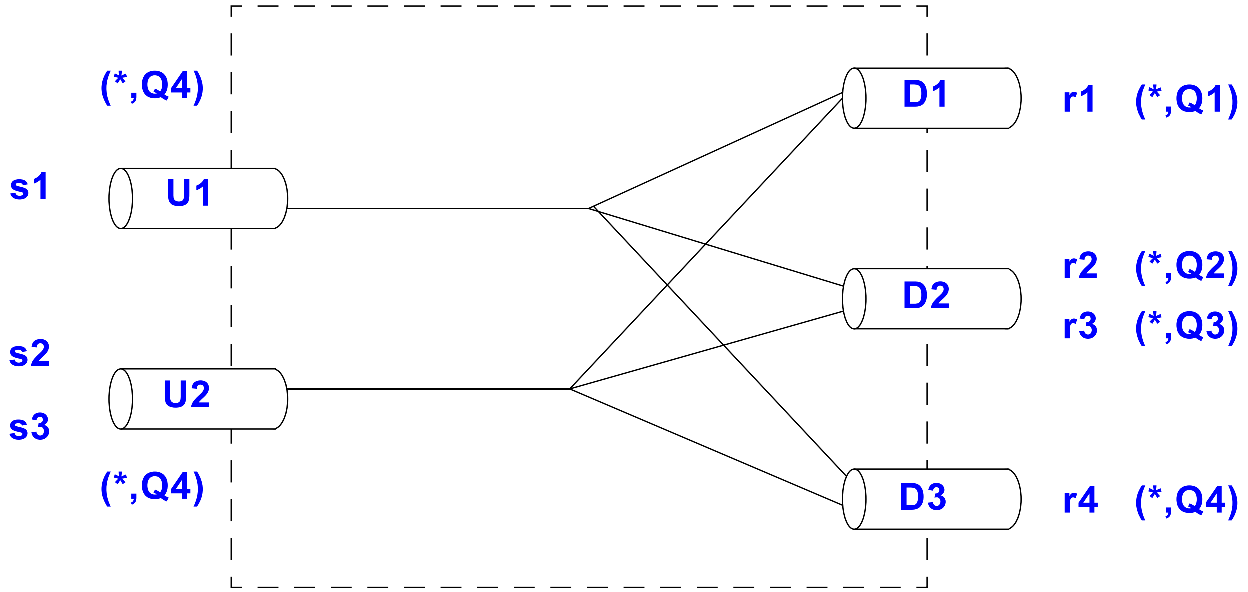
r*: receivers

Q*: FlowSpec

U*: upstream interfaces

D*: downstream interfaces

assume: Q1 < Q2 < Q3 < Q4 < Q5 < Q6



Each interface reserves maximum of received reservations

Maximum of all reservations is sent to all sources

RSVP & Routing

Data forwarding tree is set up by routing protocol (esp. IP Multicast).

RSVP Messages

- independent from reservations
- before knowledge about reservations is available
- data transmission possible without reservation
- various routing protocols could be used

Decoupling of RSVP and Routing

- simple handling of link failures
- adaptation to route changes → delay of adaptation?
- route flapping possible
- ⇒ no hard QoS guarantees

Other Routing Issues

- path selection: find path that can handle new flow
- load balancing → traffic engineering

Evaluation of IntServ/RSVP

IntServ

- **extensible**
- **vague definition of Controlled Load service**
- **often wrongly assessed as enforcing fine-grained flows**

RSVP

- **linear scaling per number of flows**
- **tuned for multicast**
- **handles multi-sender conferencing**
- **relatively complex for unicast**
- **only host and group addressing → enforcing fine-grained flows**
  - no support for topological aggregation (e.g. subnet to subnet)
  - limitation easy to eliminate (replace IP addresses by subnet prefixes)
- **heterogeneous reservations not always sufficient**
  - traffic filtering needed, as well
- **service reliability?**

RSVP as Label Distribution Protocol

Extensions to RSVP: RSVP-TE

PATH messages

- label request
- tunnel request
- explicit routing
- route recording
  - nodes
  - labels

RESV messages

- label distribution
- resource allocation

Decoupling of Functionality

- other setup protocols exist for MPLS
- RSVP could even be used for other signalling purposes

Message Objects

LABEL_REQUEST

- network layer protocol identification

LABEL

- 20 bit label

EXPLICIT_ROUTE

- strict route: node addresses
- loose route: network addresses / AS numbers

RECORD_ROUTE

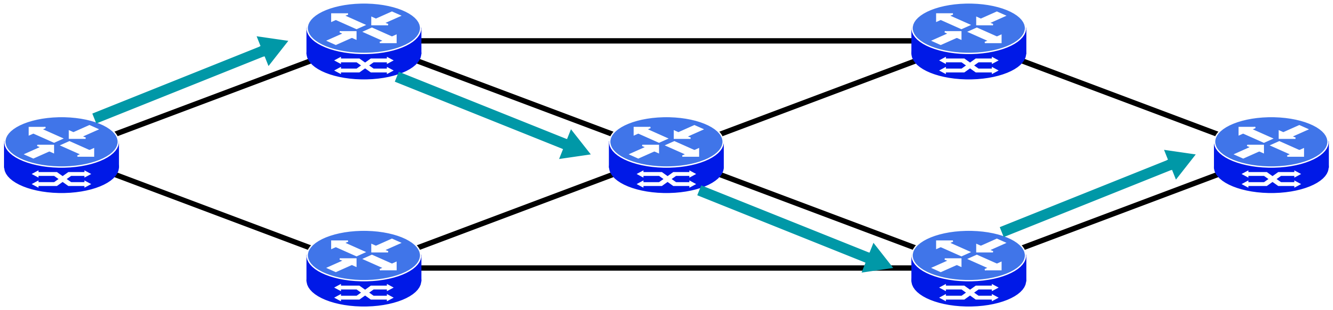
- node addresses
- labels at each link

LSP_TUNNEL

- refinement of SESSION object
- ingress/egress addresses
- identifier (to distinguish primary from backup/reroute path)

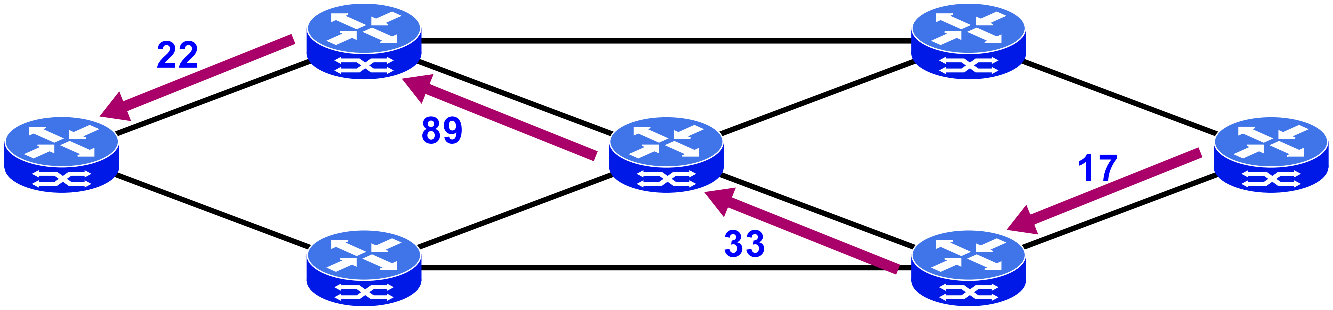
Protocol Operation

PATH Message including LABEL_REQUEST object



- implicit or explicit routed

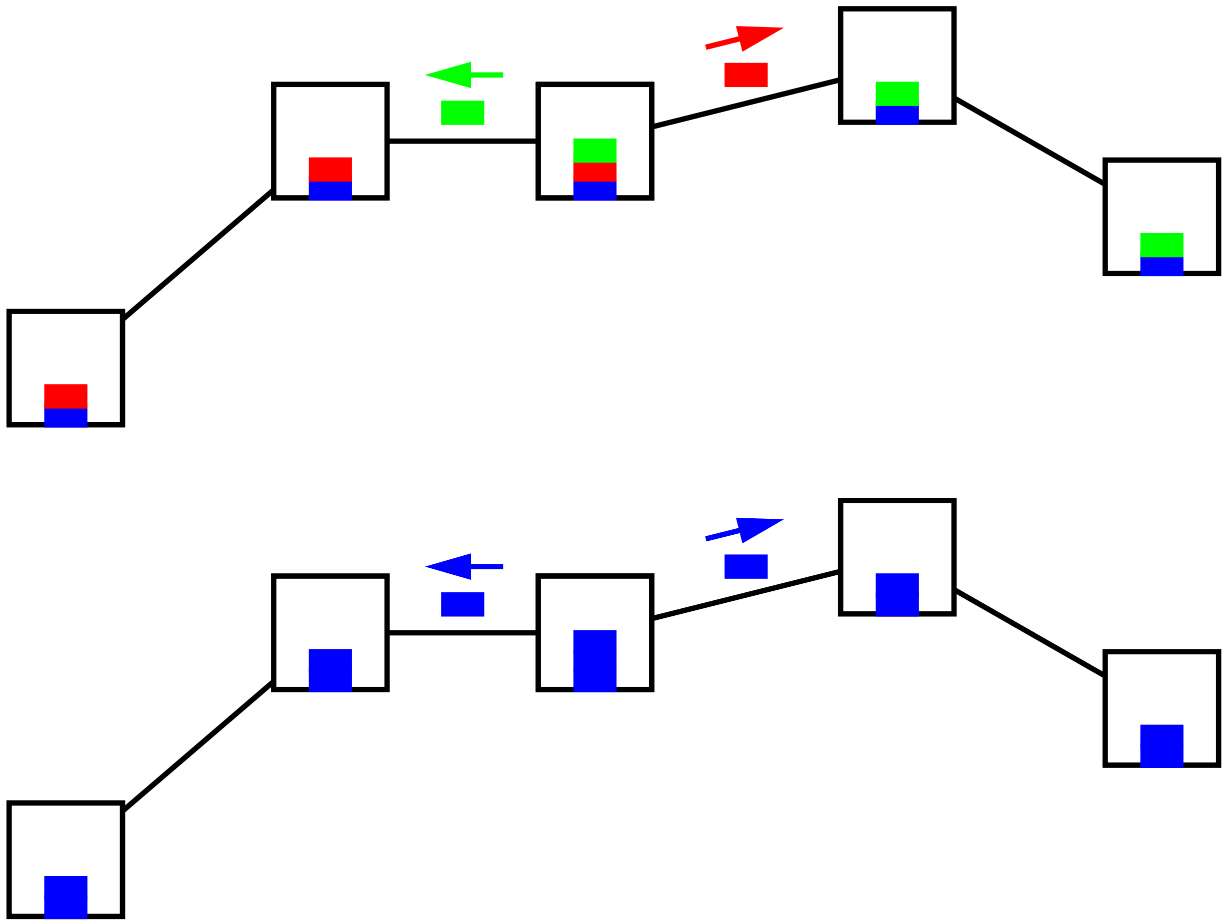
RESV Message including LABEL object



- message follows reverse path (established through PATH message)
- distribution of locally unique label
  - unique for downstream node

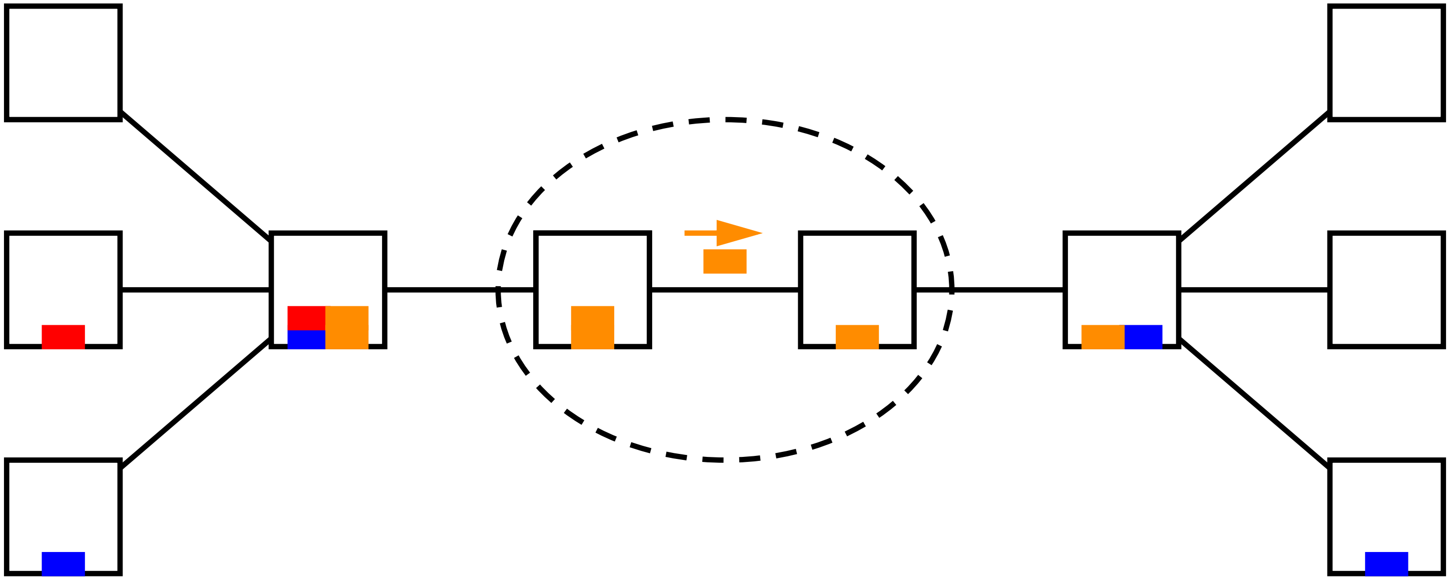
Aggregation

Assumption: Addition of Individual Requests Yields Same Service



Multicast Merging (RSVP) → Special Case (not considered here)

Gateway-based Aggregation

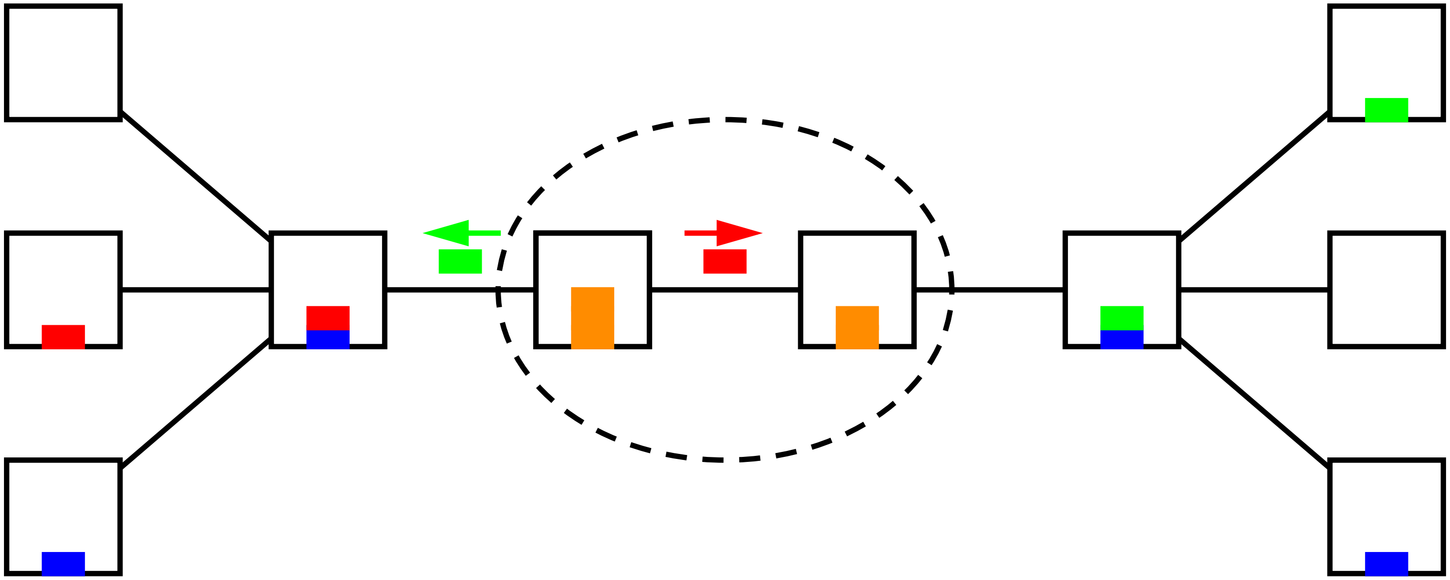


Gateway-based Aggregation

- **gateways control internal network**
- **encapsulation/decapsulation necessary**
  - control path
  - data path

State Complexity: O(n^2) per class (for n gateway nodes)

Conceptual Excursion: Automatic Aggregation



Individual Requests / Internal Aggregation

- **messages are interpreted as increase or decrease request**
- **data path: aggregation mechanism needed (could be DiffServ-like)**
- **soft state: message loss → distinction between setup and refresh?**
  - sequence numbers, timeouts, etc. → complex management
- **hard state**
  - temporary node failure: complex detection and recovery
  - transient node failure: all state for all nodes along the path affected

YESSIR

YEt another Sender Session Internet Reservation

Piggybacked on RTP/RTCP

- **RTCP periodically transports *Sender Reports* and *Receiver Reports***
- **IP option: router alert → routers intercept packets**
- **soft state**

Differences to RSVP

- **sender-initiated reservations**
  - end-to-end transport of path information
- **partial reservations**
  - but: if segment is overloaded → why end-to-end reservation at all?
- **synchronous state refresh → no refresh timers**
  - triggered by end systems
- **simpler filter styles**
- **smaller messages → less overhead**
- **learn classification from RTCP packets**
- **possibly: learn resource requirements from RTCP packets**

⇒ No Complete Protocol, but Extension to RTCP

Border Gateway Reservation Protocol (BGRP)

Proposal

- claim/idea: simple protocol in access networks (e.g. YESSIR)
- trunk reservation needed in the core

State Complexity: $O(N^2) \rightarrow O(N)$

- state for pair of edge nodes

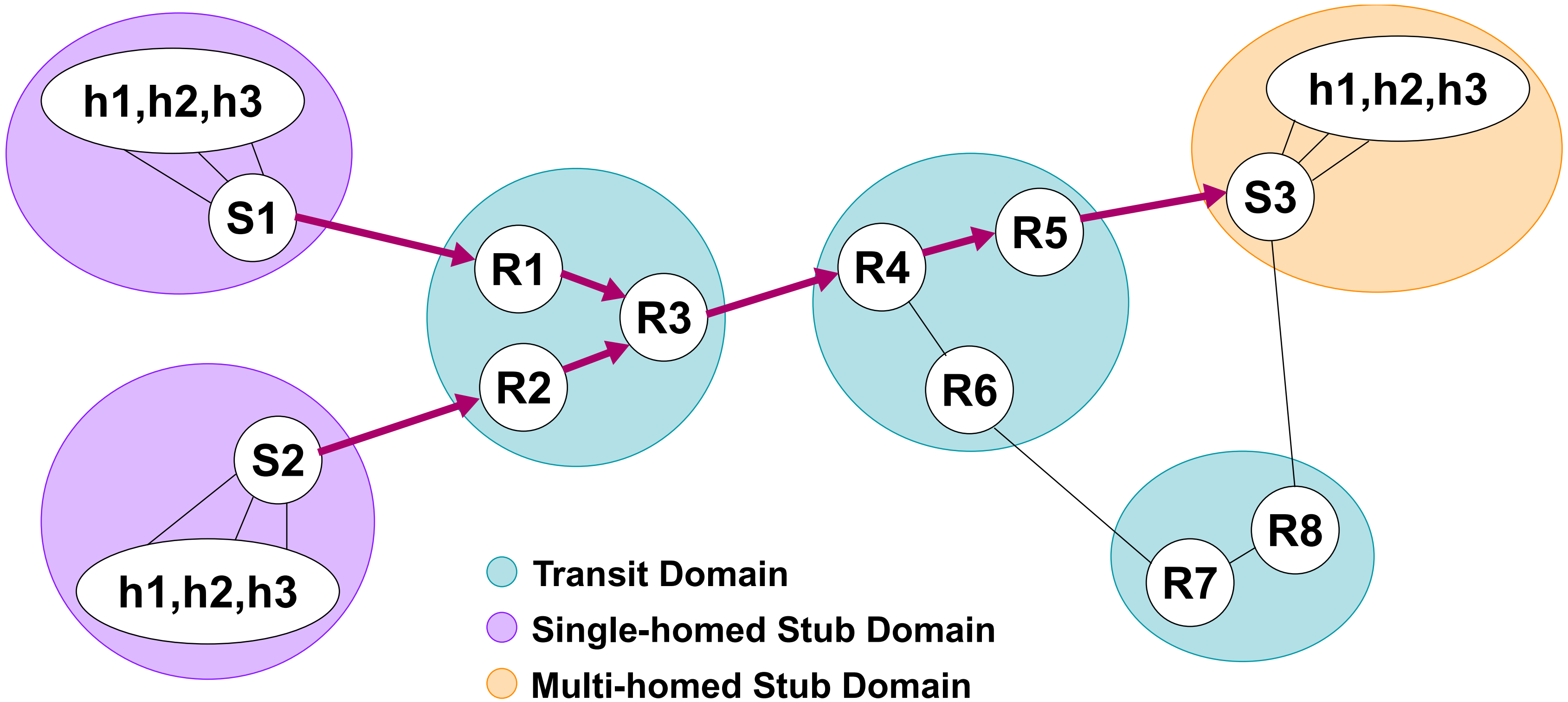
⇒ **Sinktree-based Aggregation (aligned with route aggregation in BGP)**

Message Types

- PROBE downstream message to probe network
- GRAFT upstream message to reserve resources
- REFRESH upstream/downstream message to refresh state
- ERROR upstream/downstream message to report erros
- TEAR upstream message to release resources

Similarities to RSVP

Border Gateway Reservation Protocol (BGRP) (2)



Differences to Traditional RSVP

- **no PATH state → record route in packet**
- **sink-tree reservations: sum of individual reservations on leg**
  - delta reservations ⇒ reliable message transmission required!
    - egress keeps track of aggregated reservation
  - ⇒node failure & other error management becomes highly complex
- **bundled refresh - refresh multiple reservations with one message**

Discussion

Gedankenexperiment: End-to-End MPLS

- pros and cons?

Comparison of RSVP and BGRP

- can BGRP be done with RSVP mechanisms?
- what are the fundamental differences?

QoS System = Admission Control & Scheduling

- pros and cons of different combinations?

MPLS & QoS

- can MPLS help? if yes, how?