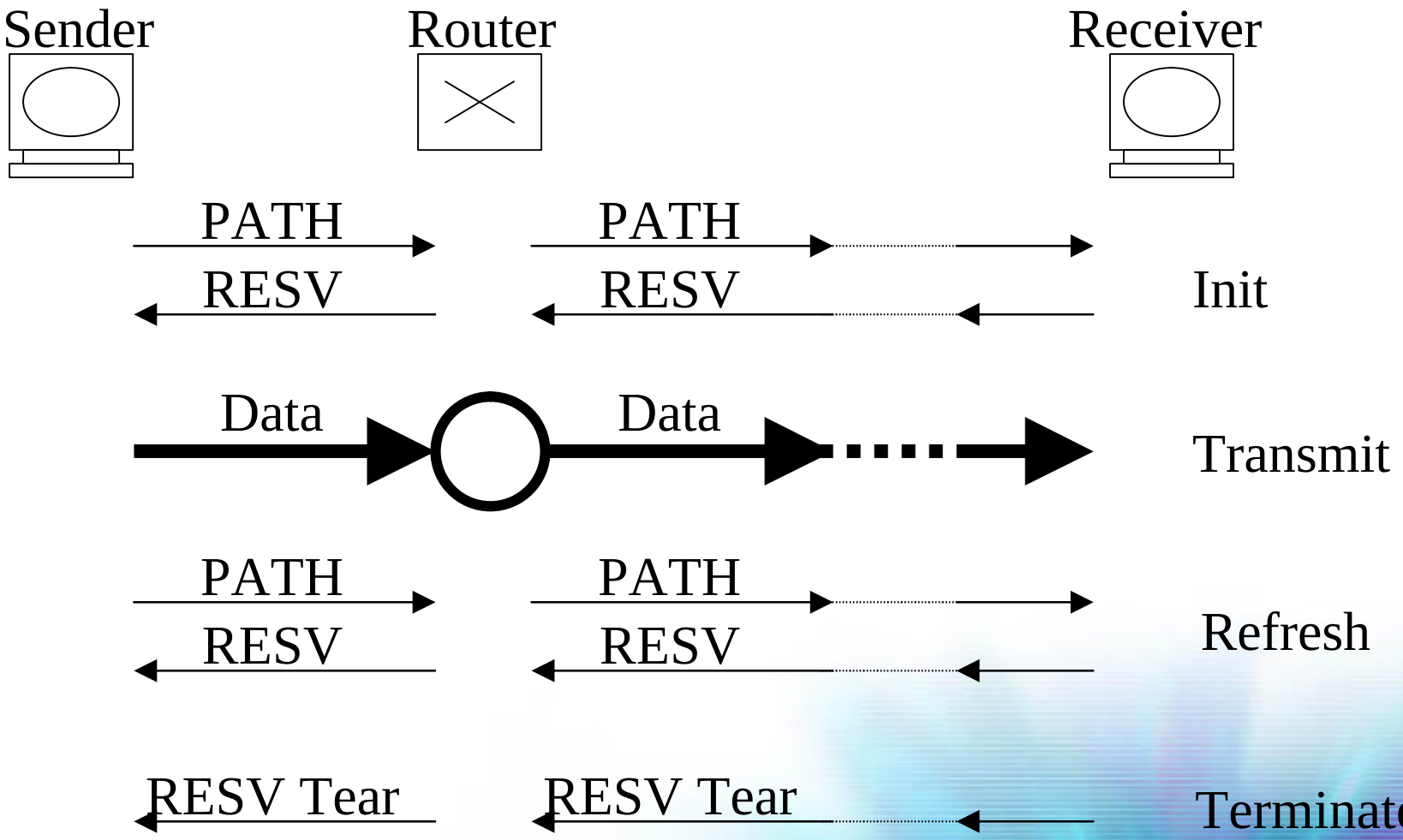
Next Steps in QoS Signalling: Do we need RSVP version 2?

Marcus Brunner, Rosella Greco, Juergen Quittek
Network Laboratories, NEC Europe Ltd.
Heidelberg, Germany
{brunner,greco,quittek}@ccrle.nec.de

Resource reSerVation Protocol (RSVP)

- Hop-by-hop protocol
- Routing protocol independent
- Path-coupled (“in-band” signaling)
- Sender advertisements
- Receiver-issued reservations
- Soft state design
- Support for multicast

RSVP Signaling



RSVP Issues under Discussion

- No direct support for mobile terminals
- Multicast support
- End-to-end
- Coupled reservation identifier and flow identifier
- Uni-directional reservation
- Receiver-orientation
- Soft state
- Path-coupled (“in-band”) reservation
- Scalability
- Complexity of implementation

Scalability and Complexity

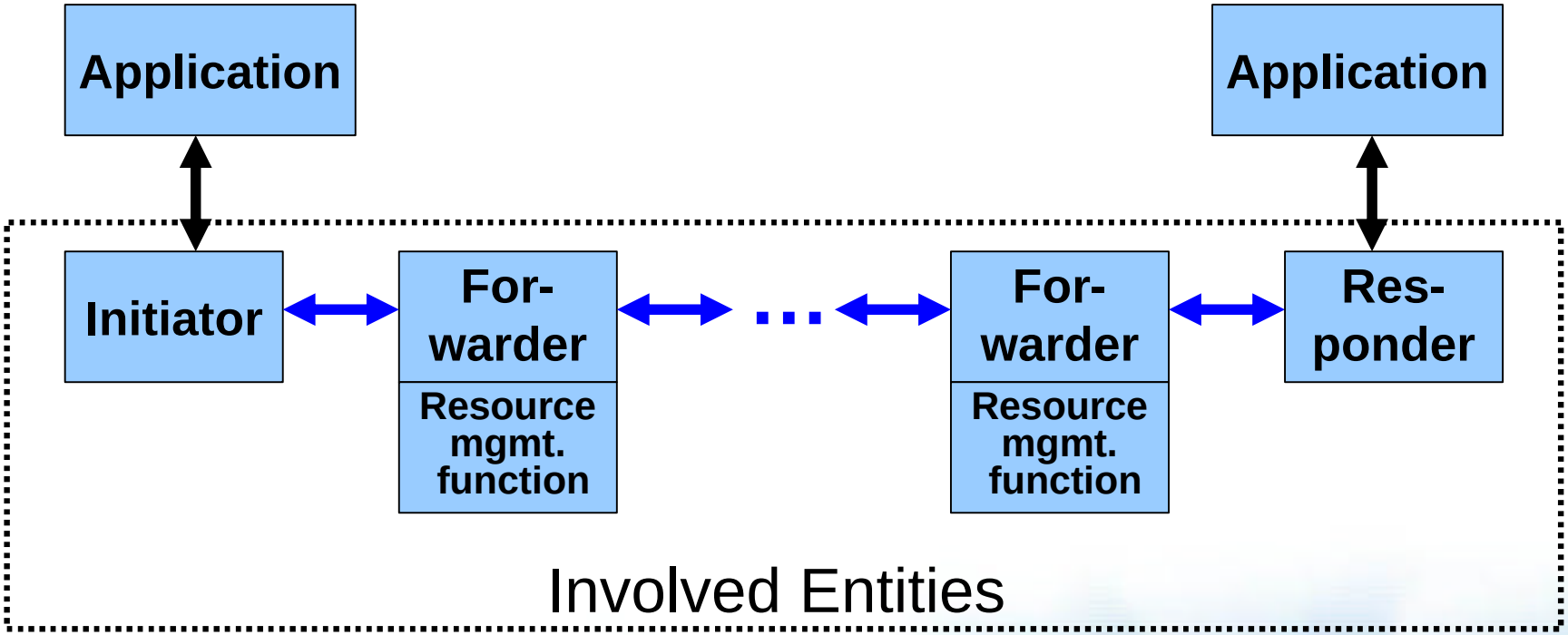
- Complexity of implementation
  - large state machine
    - support of multicast
  - several message types
- Scalability
  - per-flow reservation
    - potential overload of RSVP signaling daemon (soft state)
    - timer per flow
  - per-flow traffic handling not applicable to backbone core routers: too many flows
    - potential overload of classifier
    - potential overload of scheduler (number of queues)

Next Steps In Signaling (NSIS): Do we need RSVP version 2?

- Investigated by the IETF NSIS WG
  - started in December 2001
- Major contributors include Nokia, Siemens, Ericsson, Alcatel, NEC
- Requirements analysis
  - selection of scenarios
- Framework development

==> Decision

Entity Roles



Considered Scenarios

- Terminal mobility
  - initiator and/or responder
  - sender and/or receiver
- Cellular network
- UMTS access network
- Session mobility
- Reservation from access to core network
- QoS negotiation and reservation across administrative boundaries
- QoS signaling between PSTN gateways and IP backbone
- PSTN trunking gateway
- Application requested end-to-end QoS

Requirements for RSVPv2

- Not just pure protocol requirements, also framework and architecture requirements
- Some might be technically contradictory
- Categories
  - Architecture and design goals
  - Signaling Flow
  - Additional service information
  - Control information
  - Performance
  - Flexibility
  - Security
  - Mobility
  - Inter-working with other protocols and techniques
  - Operational requirements

Architecture and Design Goals

- Applicability to different QoS technologies
- Resource availability information on request
- Modular design
- Extensibility, even for non-QoS purposes
- Clear separation of signaling protocol and carried control information: extensibility, safety
- Independence of signaling and QoS provisioning paradigm
- Application independence

Signaling Flow

- Free placement of signaling end points
  - End-to-end, end-to-edge, edge-to-edge, network-to-network
- No constraint of the signaling and the forwarders to be in data path
- Concealment of topology and technology information
- Support of hierarchical reservation scenarios

Additional Service Information

- Explicit release of resources
- Automatic release of resources
- Upstream notifications
- Feedback on success of service request
- Local information exchange

Control Information

- Mutability information on parameters
- Possibility to add and remove local domain information
- Independence of reservation identifier and flow identifier
- Seamless modification of resource reservation
- Grouping of signaling for several micro-flows

Performance

- Scalability
- Low setup latency
- Low bandwidth consumption of signaling
- Ability to constrain load on devices
- Highest possible network utilization

Flexibility

- Flow aggregation
- Placement of initiator
- Initiation of re-negotiation
- uni-directional and bi-directional reservation

Security

- Authentication of resource requests
- Resource authorization
- Integrity protection
- Replay protection
- Hop-by-hop security
- Identity confidentiality
- Location privacy
- Prevention of denial-of-service attacks
- Confidentiality of signaling messages
- Ownership of reservations
- Hooks for authentication and key management protocols

Mobility

- Allow efficient QoS re-establishment after hand-over

Inter-working with other protocols and techniques

- Inter-work with IP tunneling
- IPv4 and IPv6
- independent of charging model
- Hooks for AAA protocols
- Inter-work with seamless hand-over protocols
- Inter-work with non-traditional IP routing

Operational Requirements

- Ability to assign transport quality to signaling messages
- Graceful fail-over

Conclusion on QoS Requirements

- Meeting all of them might be not desirable
  - too complex
  - one size fits all not necessarily a good idea
  - some are classified as “SHOULD” or “MAY”
- Note, that there is no requirement for multicast
- Being aware of all requirements is very useful when designing your own QoS signaling protocol
- A careful selection is necessary
  - Which application scenario is in your focus?
  - Which requirements harmonize and integrate well?

Non-QoS Issues

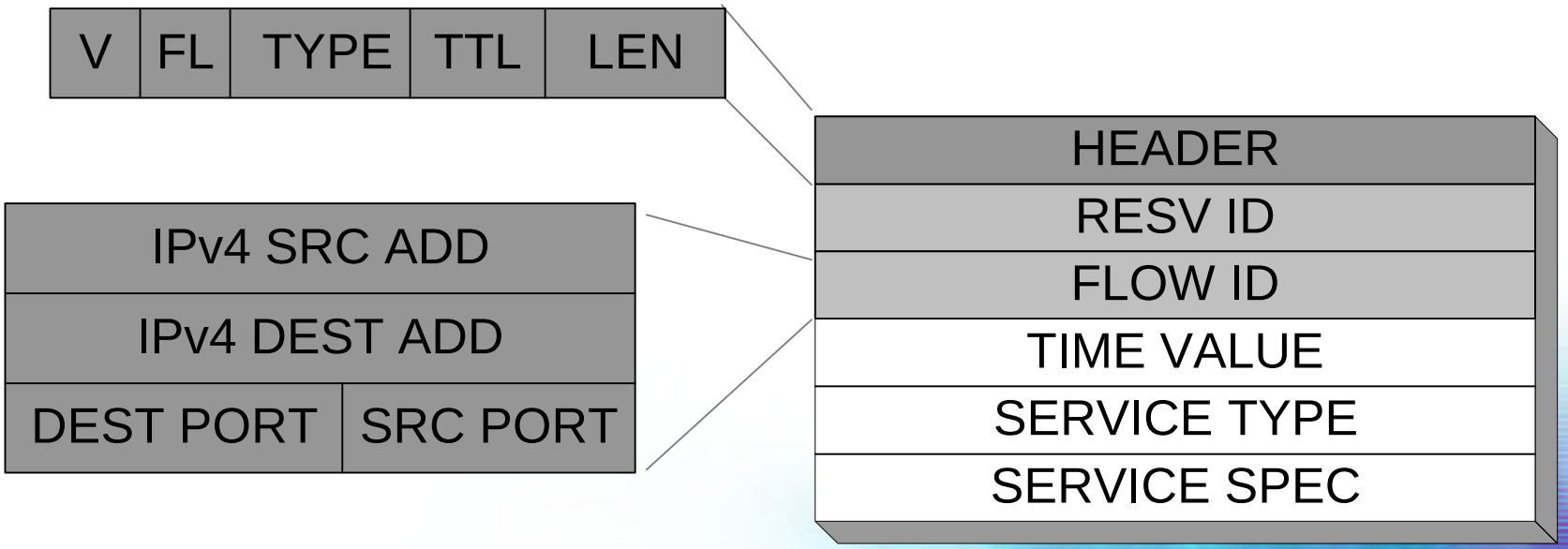
- Extending the extensibility requirement
- Middlebox configuration
  - Implicitly or explicitly request configuration of
    - Firewalls on the data path
    - network address translators on the data path
  - They are big obstacles for UDP-based services
- Gateway configuration

A Simple Design Example

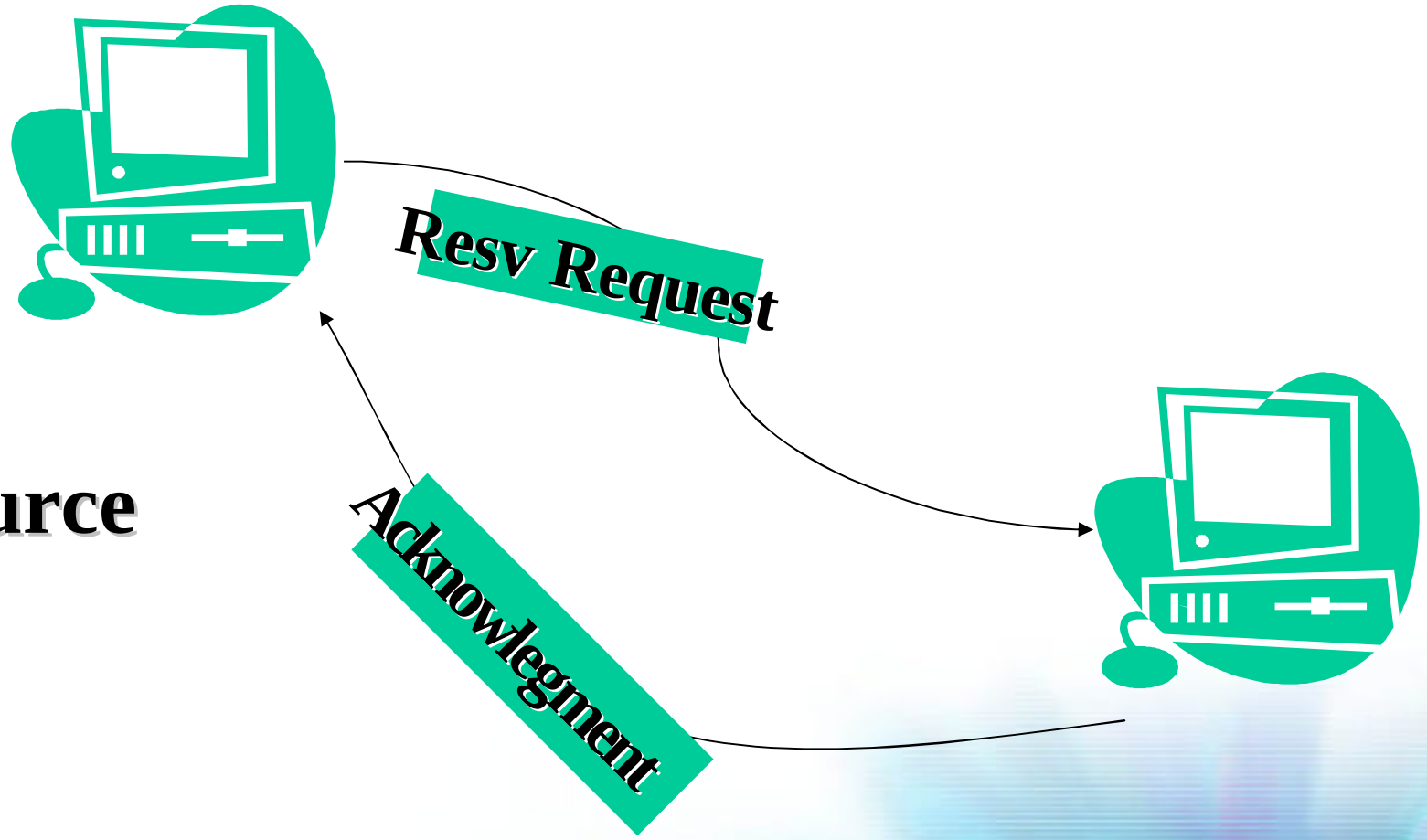
- Scenarios:
  - Wireless mobile terminals
  - IP telephony
- Design Goals and Choices
  - Coexistence with RSVPv1
  - Uni-cast reservations only
  - Sender-oriented approach
  - Small and efficient core protocol + service specific part
  - path-coupled and path-decoupled signaling
  - Flow ID separated from reservation ID
  - Soft-state

Message Types

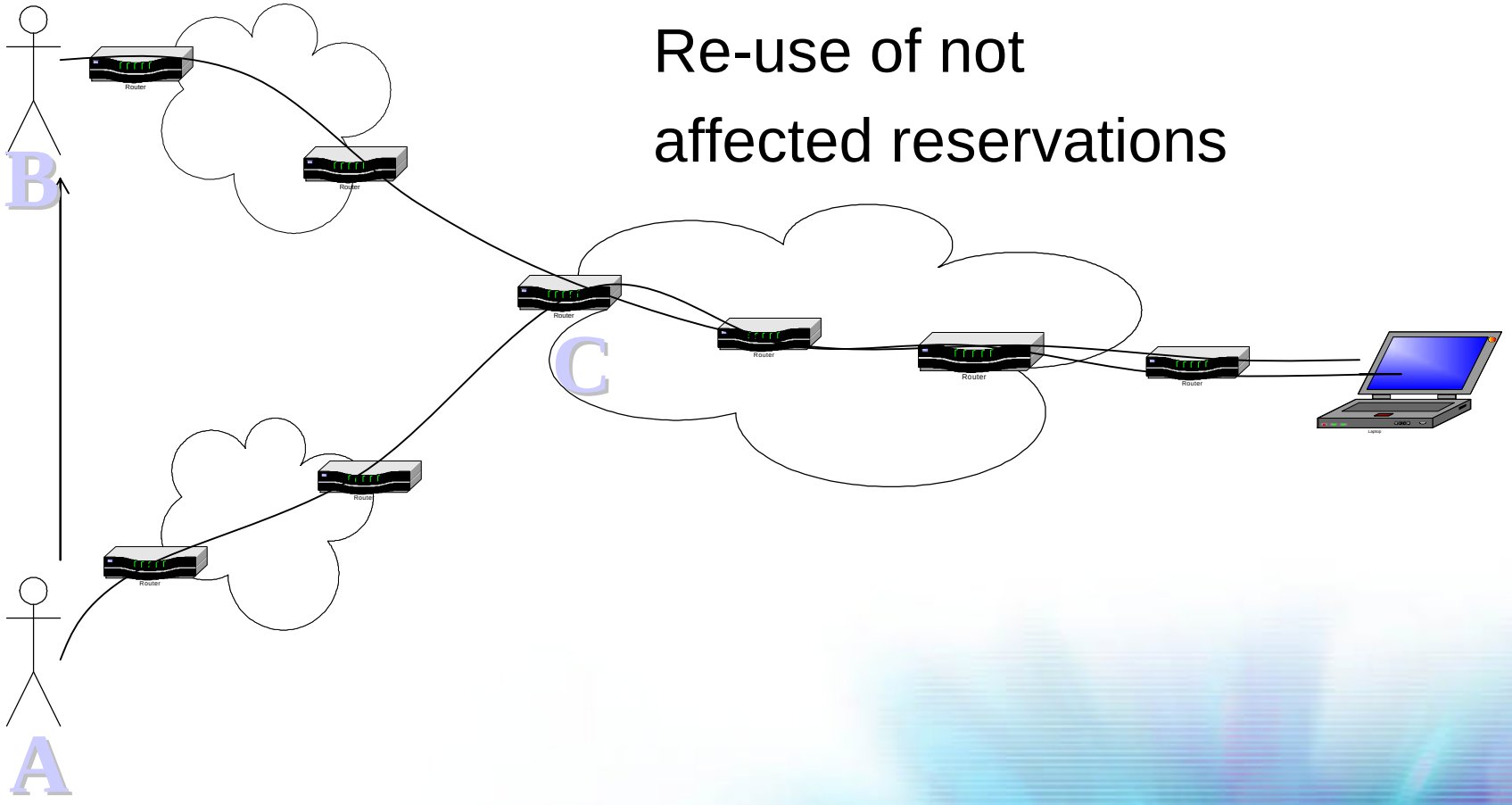
- RESV
- ACK/NACK
- REFRESH
- TEAR DOWN



Basic protocol operations

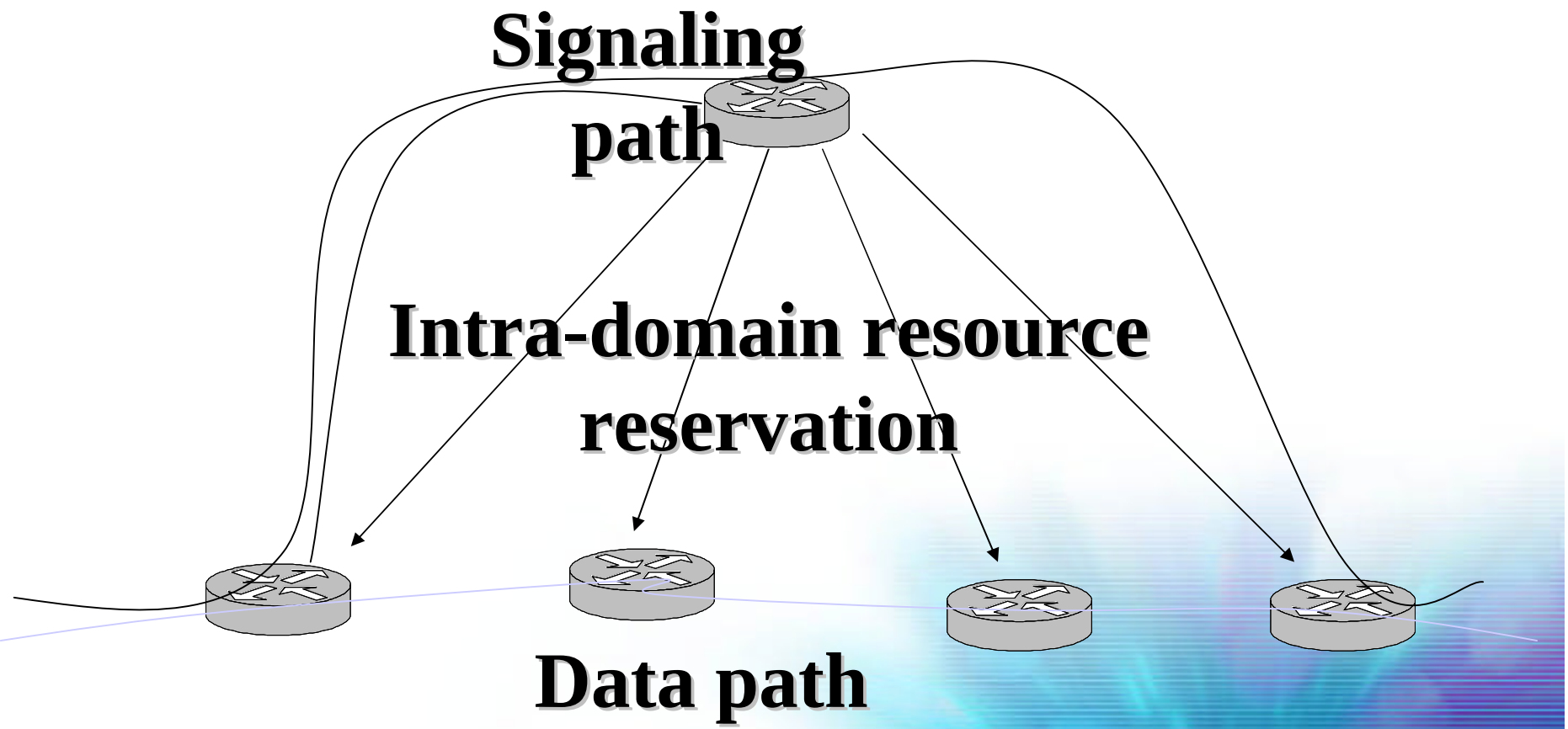


Mobile Scenario



Re-use of not affected reservations

Network with centralised management of resources



Implementation issues

The protocol has been implemented in C++

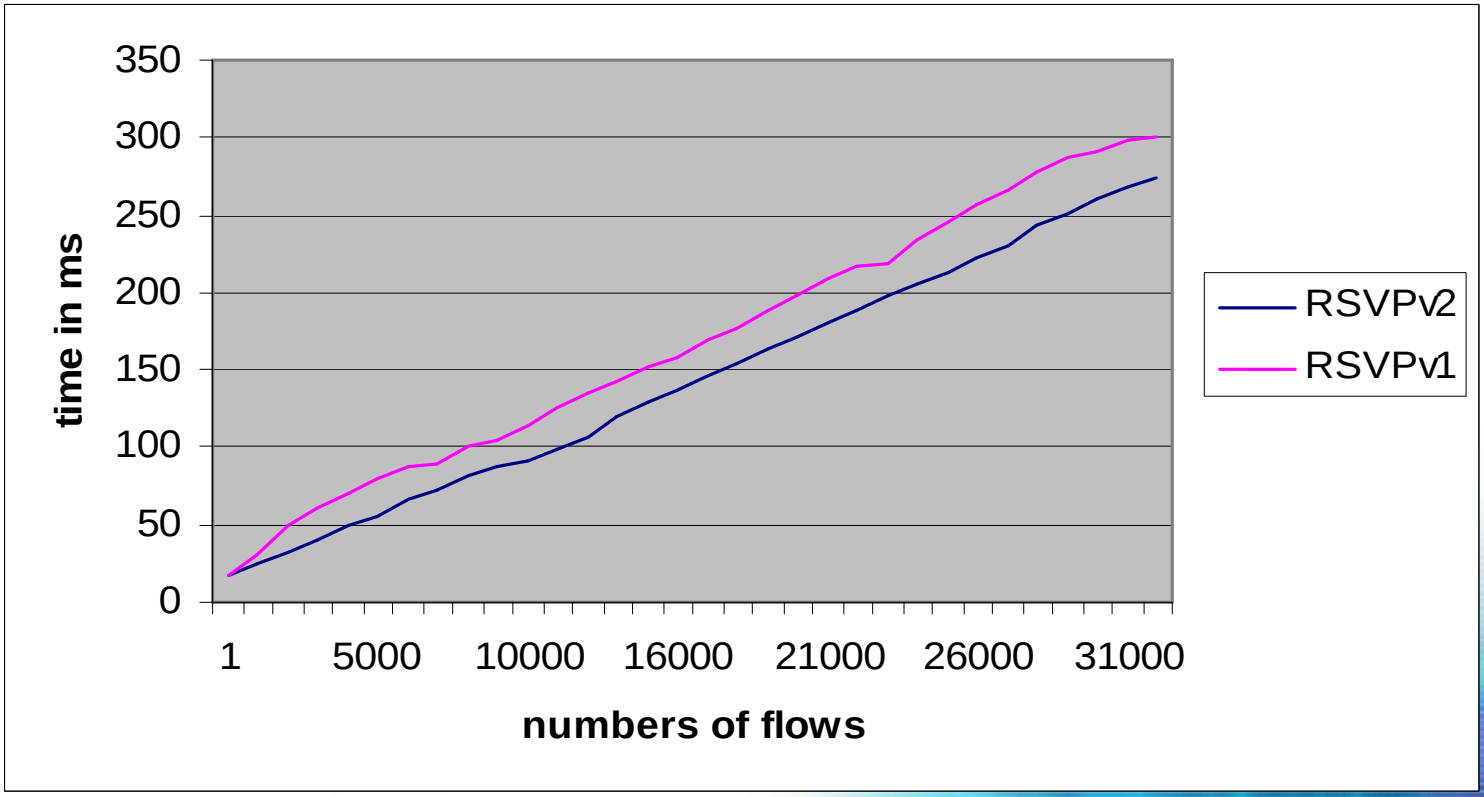
- One timer for each RSVPv2 daemon
- Few state information saved
- QoS signaling (type of services)
  - Assured Bandwidth
  - Integrated Service: Guaranteed Load and Assured Service (Same as RSVP)
  - DiffServ Interoperability
- Implicit Firewall and NAT configuration
- IPv4 only

Performance Studies

- Small testbed with 4 PC:
  - 2 terminals, 2 DiffServ routers
- Protocol performance compared with the RSVP KOM Engine developed by the Darmstadt University

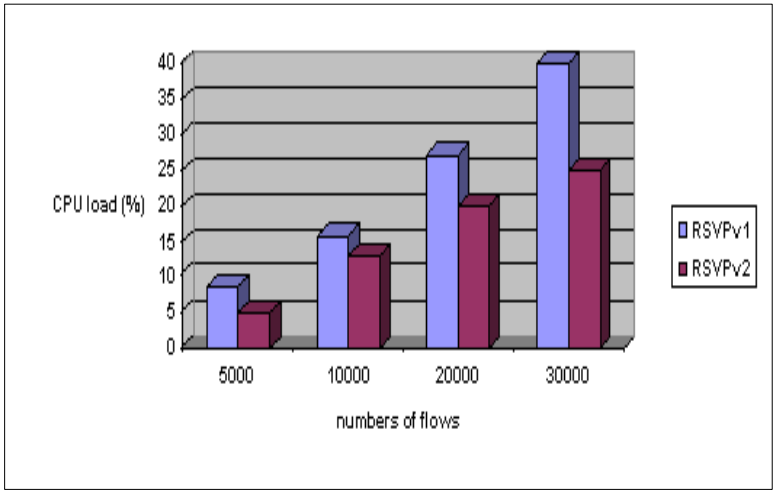
Setup time

Time needed to set up a reservation in the real testbed.

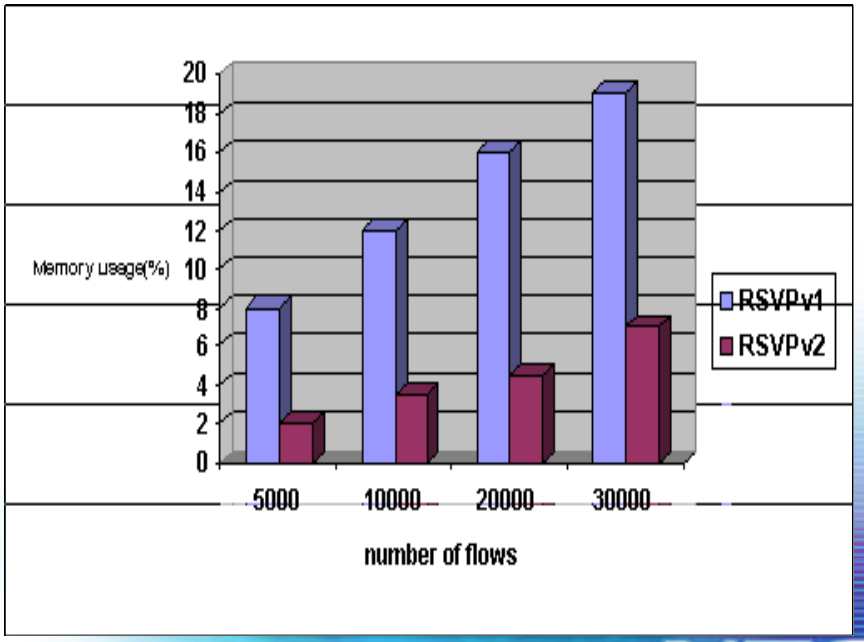


CPU usage and Memory

- CPU usage

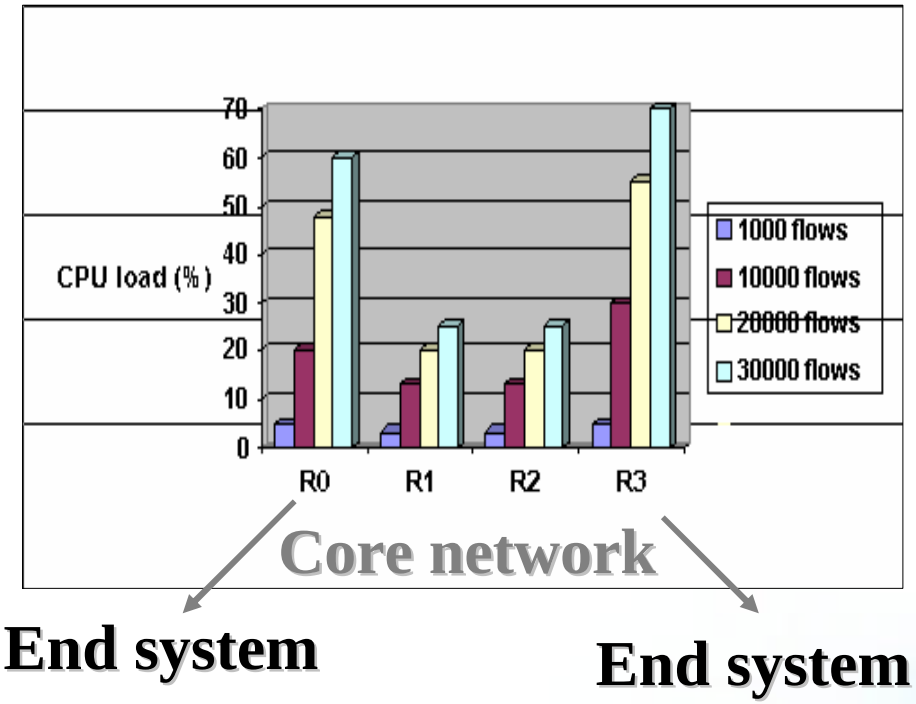


- Memory usage



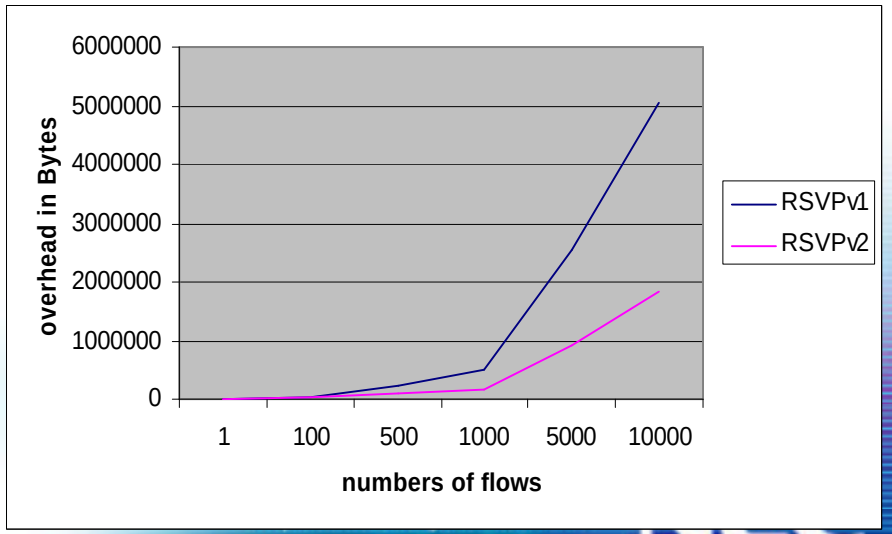
Other measurements

- CPU load in the topology



- Communication overhead for signaling:

Number of Bytes used to set up a reservation, refresh it 10 times and tear it down.



General Conclusion

- We do not need a completely different protocol.
- We learned a lot from using and studying RSVP, now we should make use of this when evolving it.
- The list of requirements we can identify is long.
- We should carefully select scenarios and a requirements set when designing a new protocol.
- Support for mobility becomes essential.
- Openness to supporting non-QoS signaling is highly desirable.
- Reservation should not be restricted to end-to-end.

Technical Conclusion

- Separation of reservation ID and flow ID is essential for mobile QoS.
- Separation of signaling and control information allows to support non-QoS signaling.
- Path-decoupled signaling can be integrated with reasonable effort.
- Omitting multicast reduces complexity and resource consumption.
- However, without omitting soft state, no significant improvement of performance can be achieved.