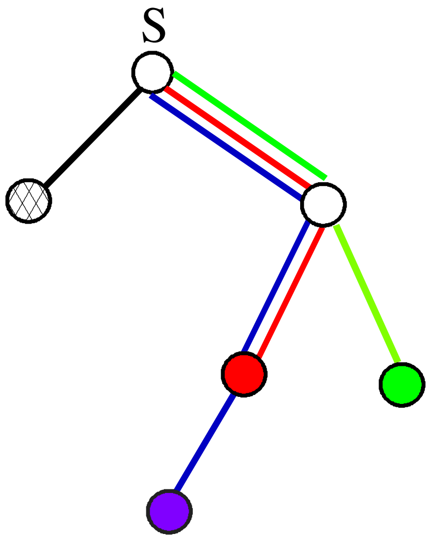
Multicast for Multimedia: Why is it so difficult?

Wolfgang Effelsberg
University of Mannheim

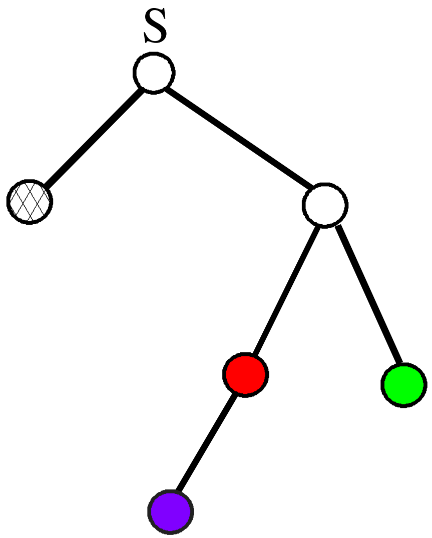
February 1997

1. Why is multicast important for Multimedia?

- Multicast applications very often require 1:n communication. Examples:
  - Videoconferencing
  - TeleCooperation (CSCW) with a shared workspace
  - near-Video-on-Demand
  - TeleTeaching, TeleSeminars
- Digital video and audio streams have high data rates (≥ 1,5 MBit/s)
- Realization with n point-to-point connections would overload the network:



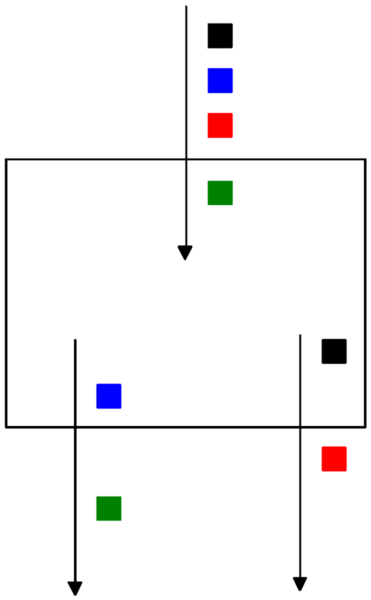
n point-to-point connections



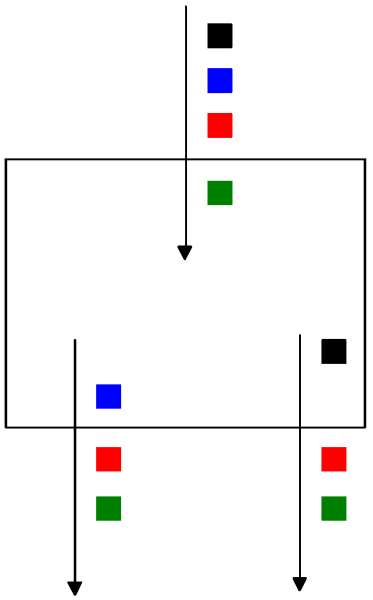
multicast connection

2. Packet or cell duplication is easy

- Copy incoming packets/cells and enqueue them into several output buffers. Can be done efficiently.
- Where and how to duplicate is controlled by entries in the routing table
- Should multicast be connection-oriented or connectionless?
- Should multicast be sender-oriented or receiver-oriented?



classical router



multicast router

3. Group addressing is difficult

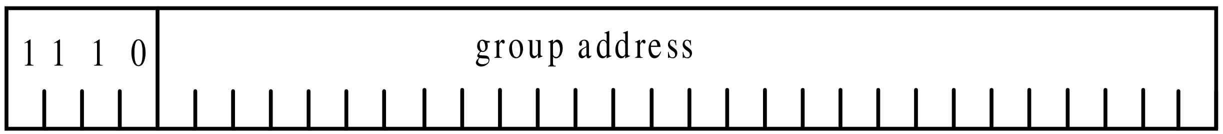
Internet Group adressing

- Group members are IP hosts
- Can be anywhere in the Internet
- The group address controls routing and cell duplication in the routers

But: Group addresses must be assigned dynamically! We need an efficient, scaleable group management protocol.

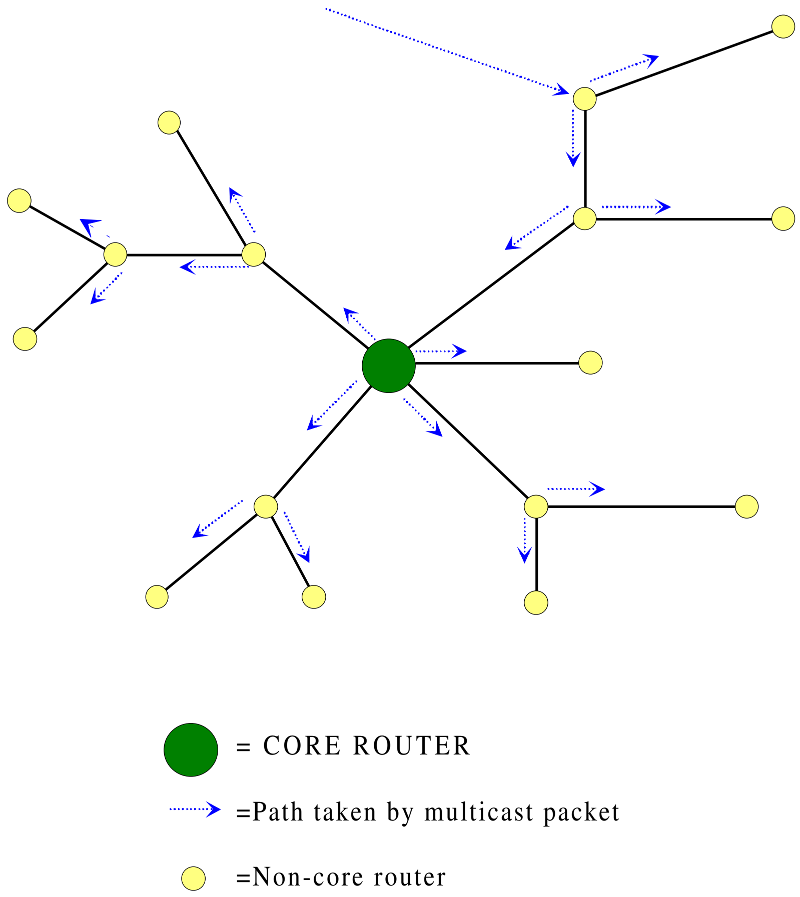
Group addresses should have a local scope so that address assignment does not require worldwide coordination.

Internet IP address of type D



4. Multicast tree routing is difficult

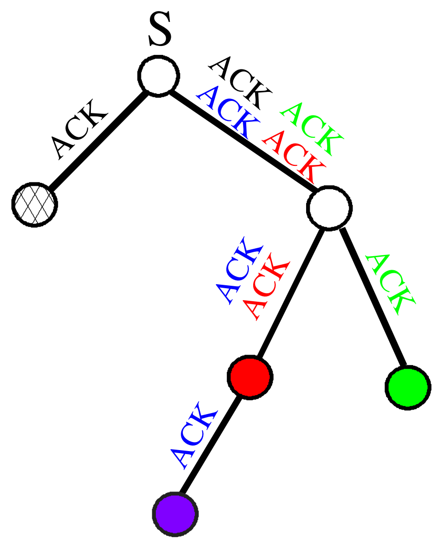
„Core Based Trees“ for n:m Group Routing



5. Reliable multicast is difficult

Two schemes:

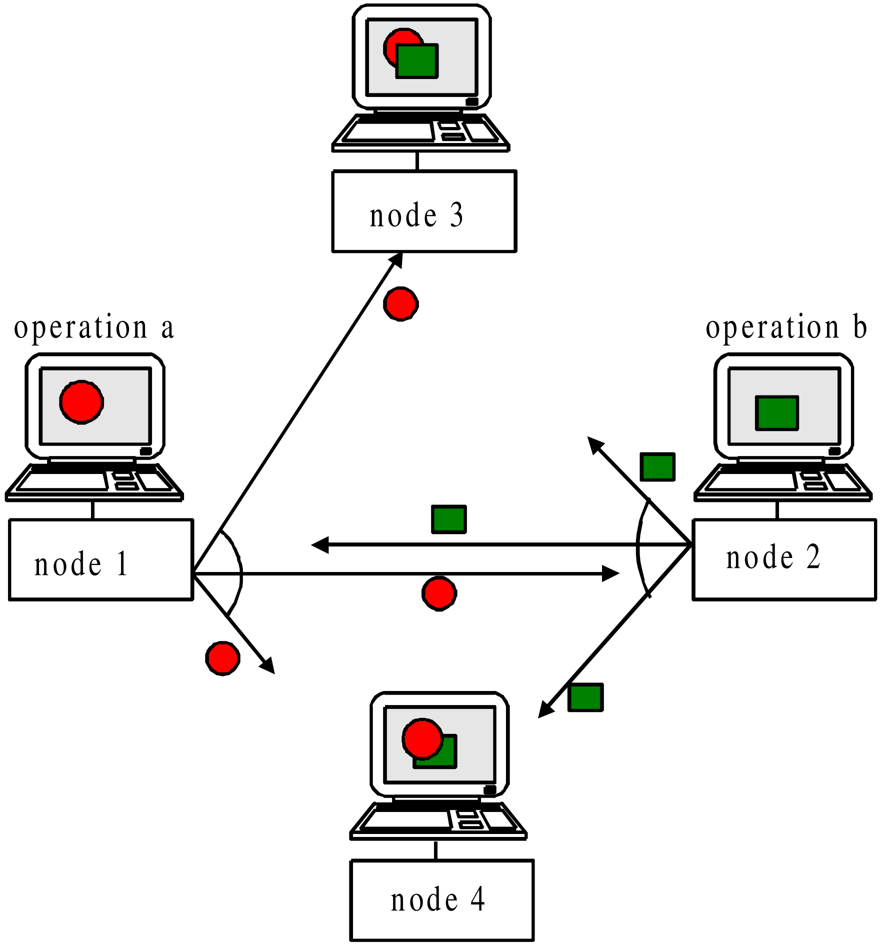
- **ACKs or retransmission:** causes ACK implosion problem
- **FEC:** not very good for burst errors, no 100% guarantees



ACK implosion

6. Multicast Ordering is difficult

The Multicast Ordering Problem



7. Multicast with QoS guarantees is very difficult

Guaranteed Quality-of-Service (QoS):

- ♦ maximum end-to-end delay
- ♦ maximum delay jitter (variance in delay)
- ♦ minimum reliability

for a given traffic load described by a "flowspec".

This requires **resource reservation** on all links and nodes in the network:

- ♦ bandwidth
- ♦ CPU cycles
- ♦ buffer space
- ♦ "schedulability"

- Routes should take QoS parameters and node resources into account.
- QoS parameters of the application must be mapped to physical resources in the network nodes

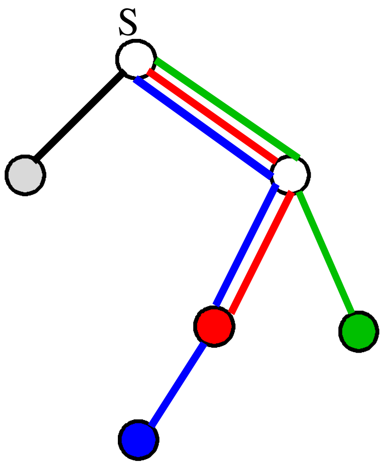
Multicast Filtering

Goal: Optimize subtree traffic to match the processing power of receivers.

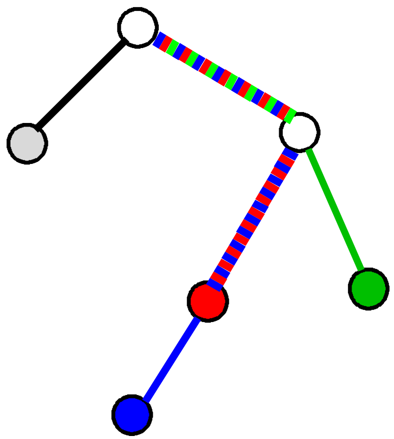
Tree nodes contain adjustable packet filters.

Requires a hierarchical stream encoding.

Requires application-specific knowledge in the network nodes!

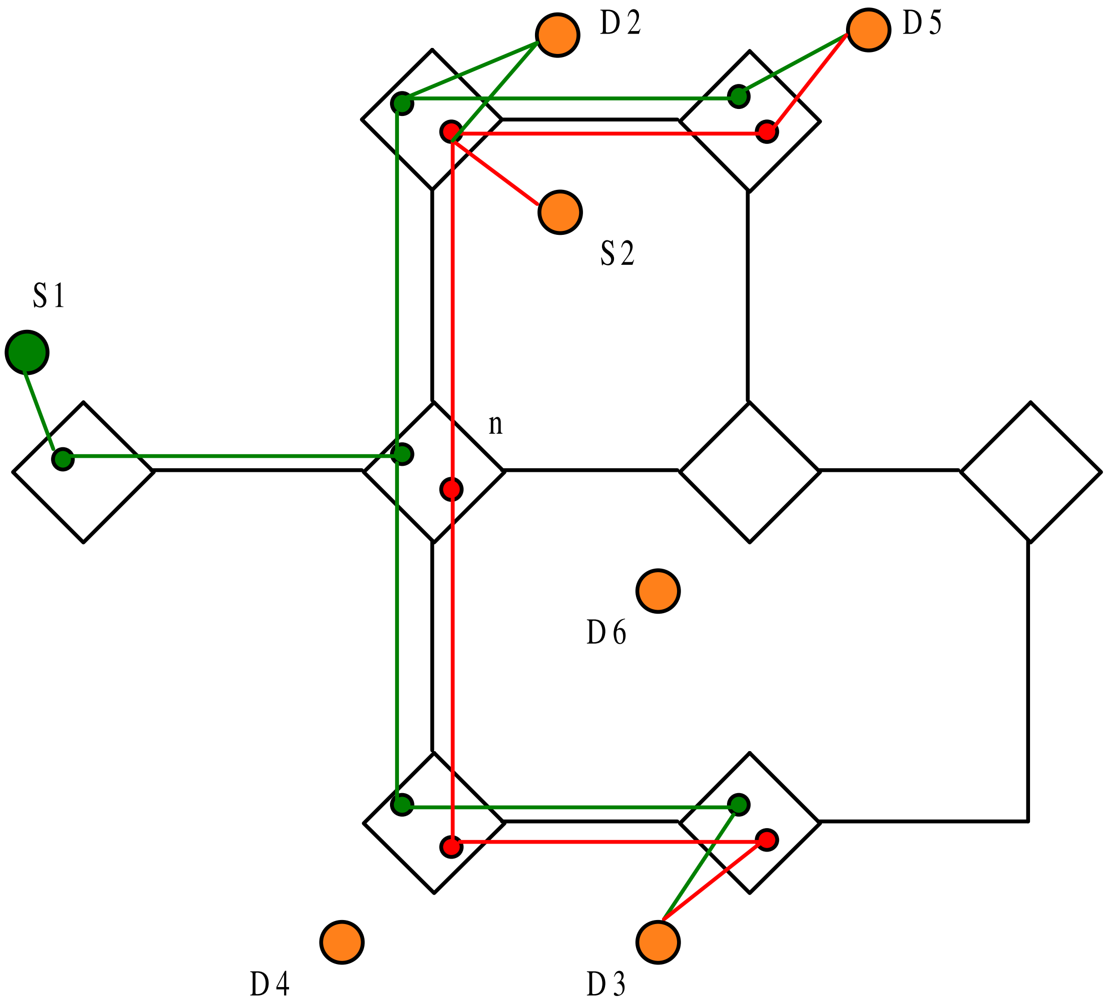


n separate connections



one multicast connection
with substream filters

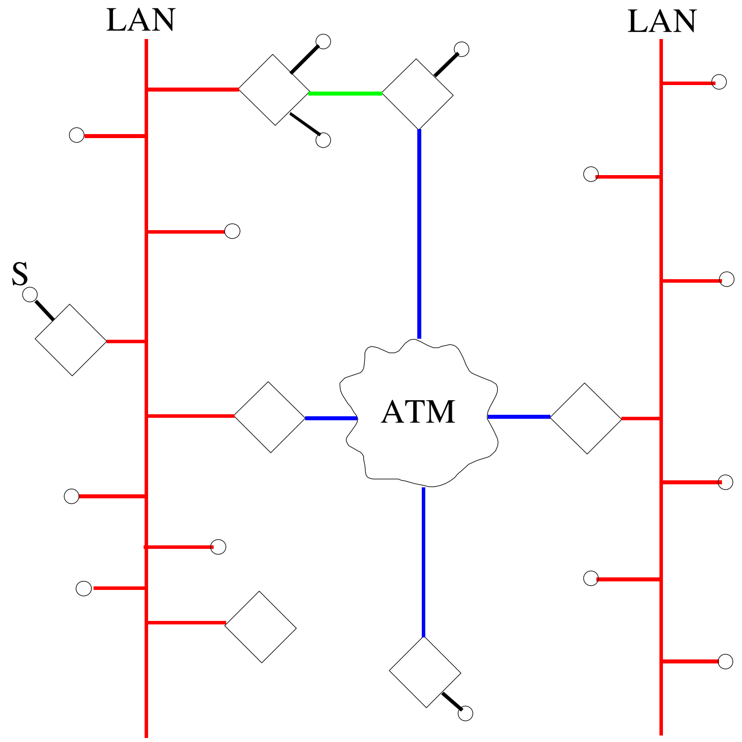
8. Dynamic join and leave with QoS guarantees is even more difficult



9. Heterogeneous networks of subnetworks make things even harder

Local area subnetworks have their own multicast/
broadcast capabilities. Must be treated differently.

Routers are different from end-nodes; we need special router-to-router protocols and end-node-to-router protocols.



network of networks

10. Conclusions

Status

Multicast in worldwide interconnected networks is very important for the efficient support of multimedia applications.

Today, there is no multicast protocol supporting all of the requirements.

Multicast IP and the MBONE protocols are useable, but do not support QoS for continuous media and are not very scaleable.

The next-generation of Internet protocols will support group addressing, multicast routing and resource reservation much better, but neither QoS mapping nor QoS routing.

Open Problems

Should reservation protocols be connection-oriented or connectionless (or „soft-state“)?

How can dynamic join-and-leave for group members be combined with QoS guarantees?

What are the best, scaleable tree routing algorithms for multicast? Should they be sender-based or core-based?

What are the best, scaleable reliable multicast protocols?

How should multicast filtering and media scaling be implemented in the network? Should we allow applications to "inject" filters into the network nodes?

How can multicast ordering protocols be integrated into the overall architecture?