UNIVERSITÄT BERN

Providing a Native IP Multicast Interface to Application Layer Multicast

Dragan Milic, Marc Brogle, Torsten Braun

Overview

- Why is there no Internet-wide IP-Multicast?
- Overlay Multicast / Application Layer Multicast (ALM).
- Multicast Middleware: ALM with an IP-Multicast interface.
  - Providing a native IP-Multicast interface to applications.
  - Distributing Multicast traffic using ALM.
  - Prototype implementation.
- Example of an ALM Integration: NEMO.
- Outlook.

Why is there no Internet-wide IP-Multicast?

- At the moment, the Internet consists of IP-Multicast islands:
  - ISPs are not always interested in mutual cooperation.
  - Forwarding IP-Unicast traffic is “easier” (stateless) than forwarding IP-Multicast.
  - ISPs fear the additional traffic induced trough IP-Multicast (easy DoS attacks trough joining high traffic volume Multicast groups).
- Multicast islands are interconnected with tunnels: MBONE [1].
  - Manual setup of tunnels.
  - Administrative overhead.
- Consequence: most of the Internet end-users cannot profit from IP-Multicast services.

Overlay Multicast

- An alternative to relying on IP-Multicast availability is to approximate Multicast data distribution using only Unicast communication (routing trough an overlay network).
  - Advantages:
    - Can minimize (but not eliminate) redundant transmission of data trough the links.
    - Does not require modification of network elements (routers).
    - Can be implemented on the application layer – no need for special operating system support.
  - Disadvantage: topology of an overlay network is almost always less-optimal than the topology of the underlying network → the ALM routing is less efficient than a native IP-Multicast routing.

Application Layer Multicast (ALM)

- User applications (e.g. a video-player) build a dynamic P2P overlay network for forwarding Multicast traffic.
- Advantages:
  - Simple deployment.
  - No infrastructure is needed.
- Disadvantages:
  - Existing applications have to be modified.
  - Each application builds its own overlay network.
  - Existing IP-Multicast based protocols cannot be used.

Multicast Middleware

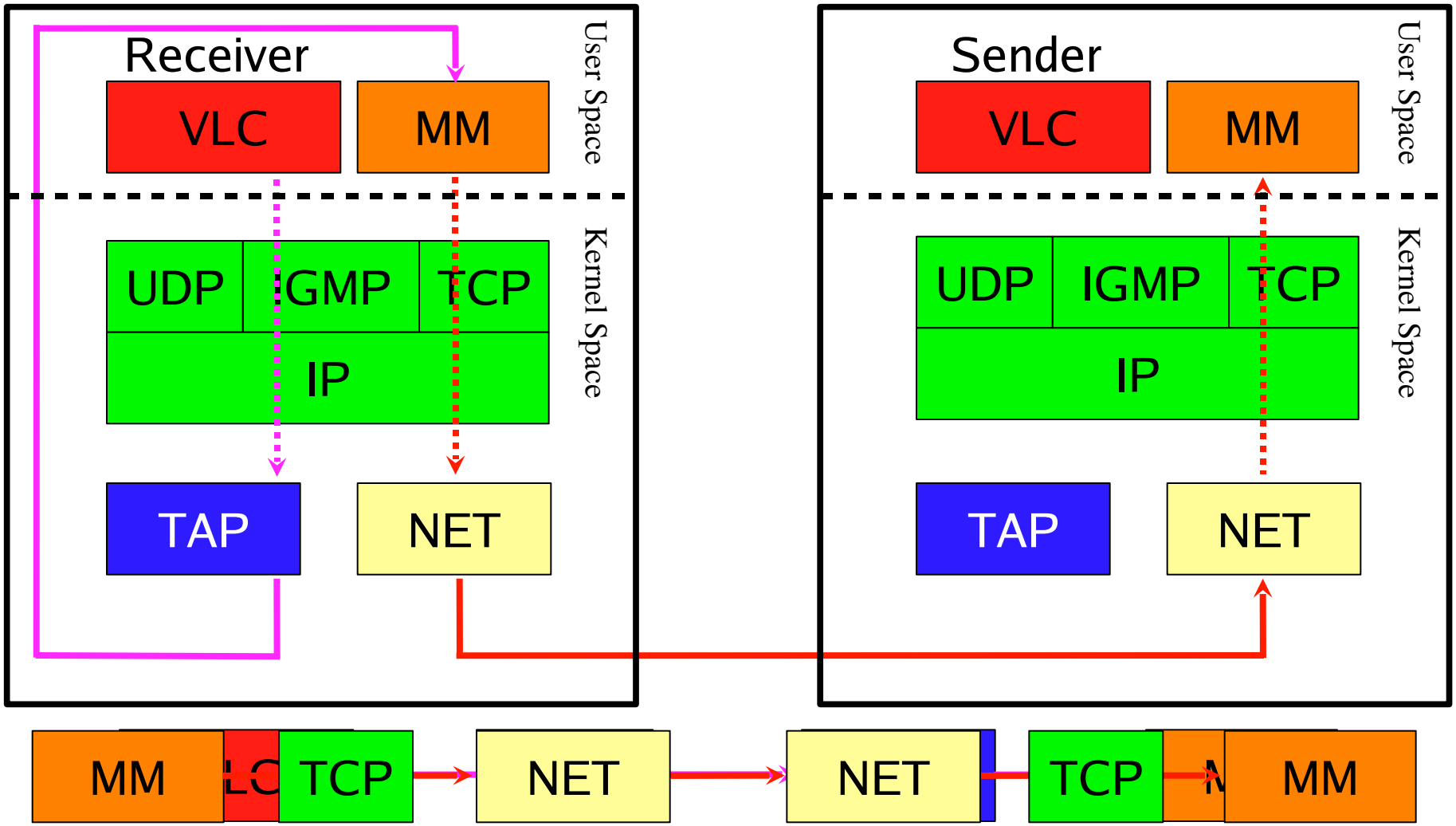
- Basic idea:
  - Using ALM for transporting Multicast traffic.
  - Make ALM usage transparent to applications by providing a native IP-Multicast interface.
- Virtual network interface (TUN/TAP).
  - TUN/TAP is a special kind of network interface that communicates with a user space process instead of communicating with a physical network.
  - From the operating system's point of view, a TUN/TAP interface is just a normal network interface.

Multicast Middleware (2)

- Providing native IP-Multicast trough a TUN/TAP interface:
  - A user space application (the Multicast Middleware) is attached to a TUN/TAP interface and simulates a network with an attached Multicast router (implements IGMP[IPV4] and MLD[IPV6]).
  - The operating system sends the outgoing IP-Multicast traffic trough the TUN/TAP device.
  - The TUN/TAP device forwards the outgoing IP-Multicast traffic to the Multicast Middleware. The captured traffic is routed to all receivers trough ALM.
  - The traffic received by the Multicast Middleware trough ALM is forwarded to the TUN/TAP interface as IP-Multicast datagrams.

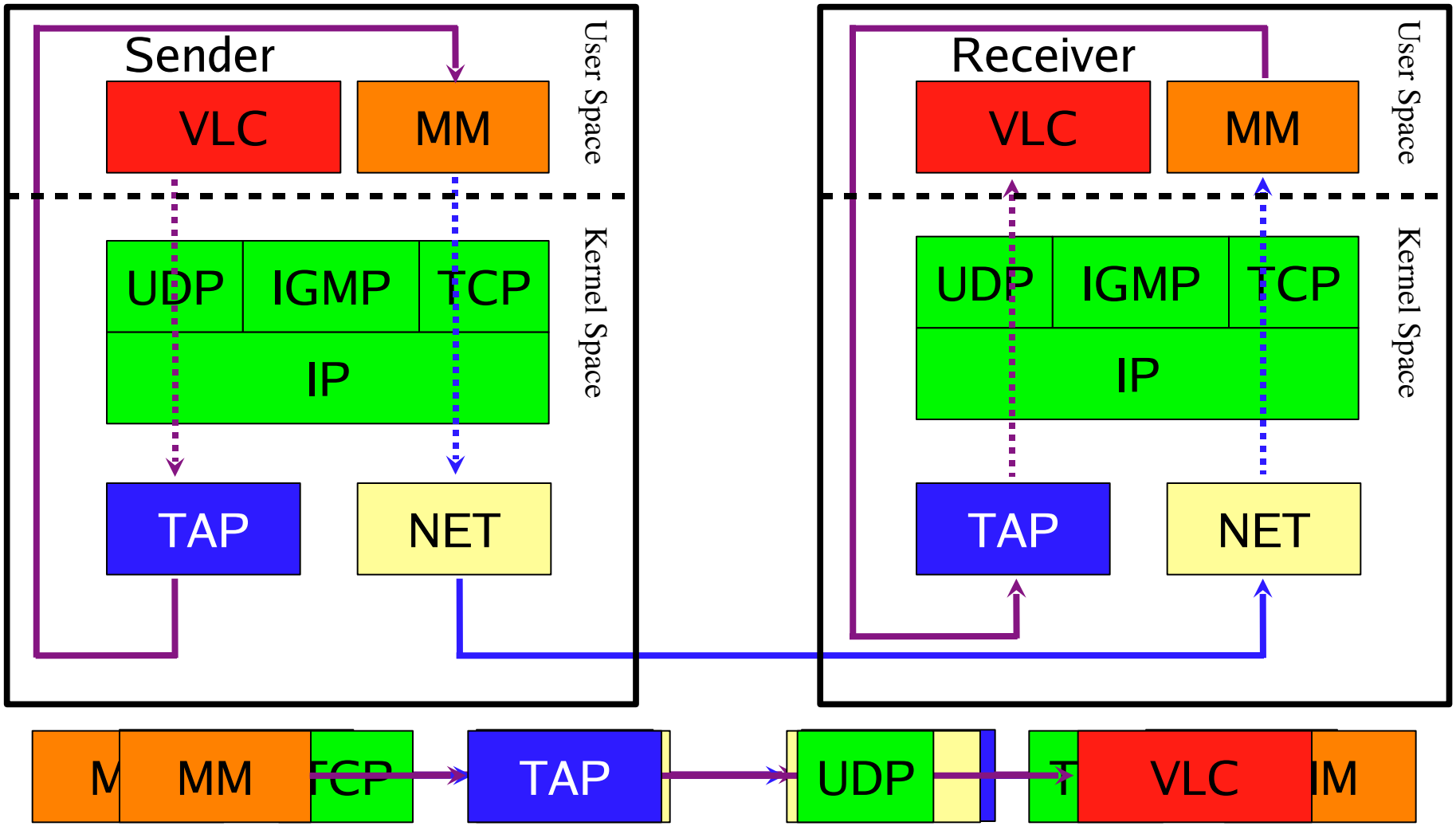
Multicast Middleware: Group Join

(IGMP) Overlay Route Setup



Multicast Middleware: Transporting Data

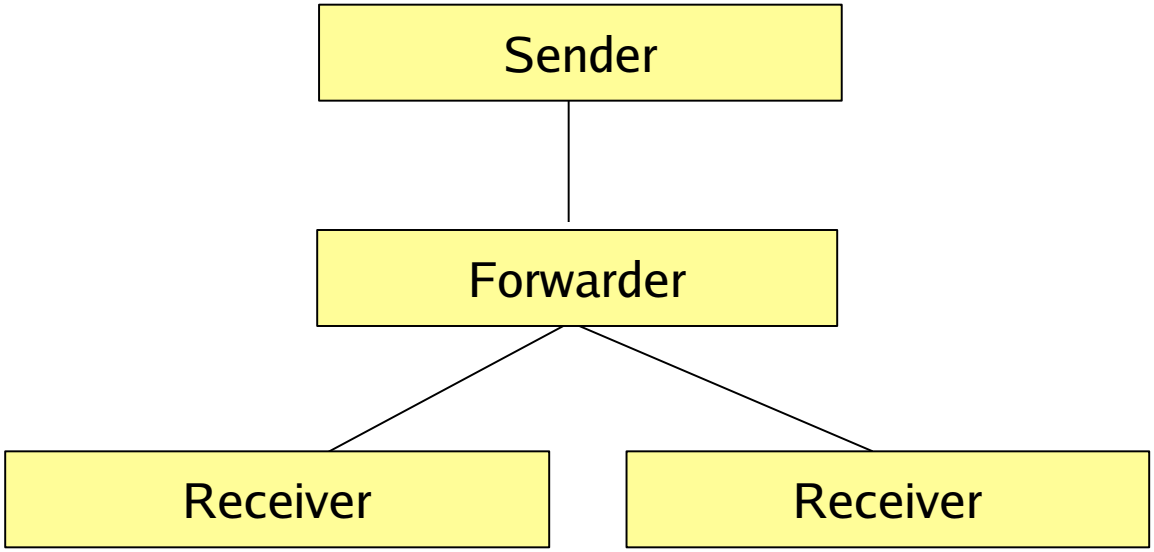
UDP Data Message



Multicast Middleware: Prototype Implementation

- Most of the implementation done in JAVA™.
- A thin layer for the data transfer to/from the TUN/TAP interface is written in C (JNI interface to JAVA™).
- Currently supported operating systems: Linux, Mac OS X.
- Custom overlay network protocol.
- Very good performance (forwarding of ~80 Mbit/s with 3 Ghz P4-CPU resulted in 20-25% CPU usage).
- To be released under GPL license.

Testing the Multicast Middleware Prototype



- Media: 7x DVD MPEG-TS ~11 Mbps (VBR)
- Applications: VLC (built-in IP-Multicast support)
- Results: no dropped packets, no perceived jitter

Example of an ALM Integration: NEMO

- NEMO [2] is a resilient ALM developed at CS department of Northwestern University, Evanston, USA.
- Why NEMO?
  - It promises to provide a high-performance ALM.
  - There is an open-source (GPL) example JAVA-implementation of the protocol [3].
- Integration of NEMO
  - Each state change of IP Multicast subscriptions is mapped to subscribing/canceling subscriptions in NEMO.
  - For the first packet that is sent to an IP-Multicast group, a publisher is registered with NEMO. Consequent packets to the same group are sent using this publisher.

Experience gathered using NEMO

- The available NEMO implementations seems to be performing very well with low bandwidth applications:
  - SAP announcements.
  - Chat application.

- Applications with high bandwidth requirements do not perform well:
  - Streaming MPEG1 1.5 Mbit/s CBR stream is impossible (very high deterioration of the picture quality).
  - Possible cause: NACKs/retransmissions.

Conclusion & Outlook

- It is possible to provide a transparent IP-Multicast interface to almost any ALM.
- Our prototype implementation shows one way of providing an native IP-Multicast interface to an existing ALM (NEMO).
- Pending tasks for the prototype implementation:
  - Integrating other ALMs.
  - Adding support for more operating systems (win32, solaris, open/free BSD).
  - Optimizing the NEMO ALM code (e.g. removing retransmissions)
  - Adding support for “best effort” (measurement based) QoS.

Questions

?

References

[1] H. Eriksson. Mbone: the multicast backbone. Communication of the ACM, 37(8):54–60, 1994.

[2] S. Birrer and F. E. Bustamanteb. Resilient Peer-to-Peer Multicast without the Cost. In *Proceedings of the 12th Annual Multimedia Computing and Networking Conference (MMCN'05)*, San Jose, CA, USA, January 2005.

[3] Nemo. http://www.aqualab.cs.northwestern.edu/projects/nemo/