



OLF NETWORKING

LFN Developer & Testing Forum

**Control and management of media
bridging for online meetings applications
using FDio/HICN VNF**

**Luca Muscariello
Principal Engineer, Cisco**

18th of Nov 2022, Seattle, WA, USA

Outline

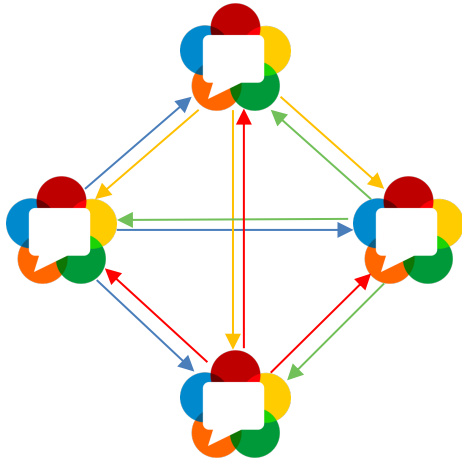
- Motivation
- System architecture
- Media-edge
- Control and management
- Conclusion and work in progress

Motivation

- The quality of online-meetings applications like Webex, Zoom, Teams strongly depends on the network.
- Current production deployments leverage global-scale DC and the public Cloud.
- Cloud-based real-time media switching implies non negligible costs.
- Edge-based deployments bring cost reduction and may also help to reduce latency.

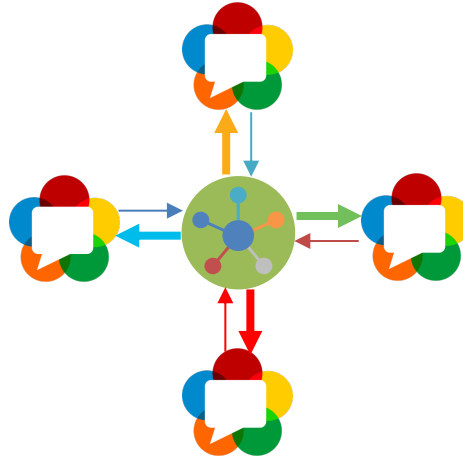
System Architecture

P2P



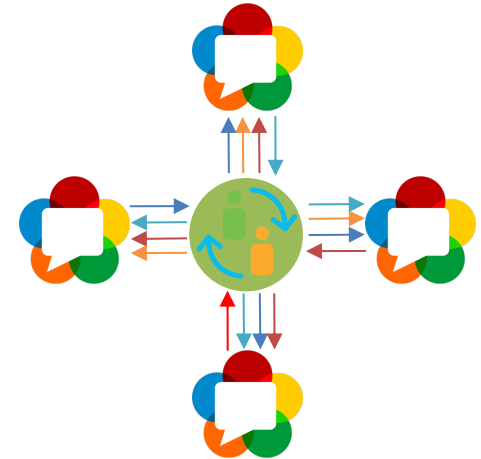
Low scalability

Multipoint Conferencing Unit (MCU)



Heavy load on MCU

Selective Forwarding Unit (SFU)



Scales with the number of users

Turn the media-bridge into an L3 switch

- The SFU is an application-level switching function that is hard to deploy to the edge
- If we could turn the SFU into an L3 switching function it could be possible to deploy at the edge at scale
- In order to do that we need to do two things
 - introduce a novel transport protocol at the client and at the server
 - Introduce a novel L3 VNF

The IRIS media relay architecture

- Media traffic is carried by RTP unidirectional streams
- Uplink streams are unicast from the client to the server
- Downlink streams can be shared across meeting's participants
- We want enable downlink sharing of the RTP streams across all meeting' members.
- To do that we enable RTP distribution using the HICN transport protocol library at client and server.
- HICN implements a pub/sub like communication model where a stream is identified with an IP adress + UDP port and the subcsiber by the host IP address + UDP port.

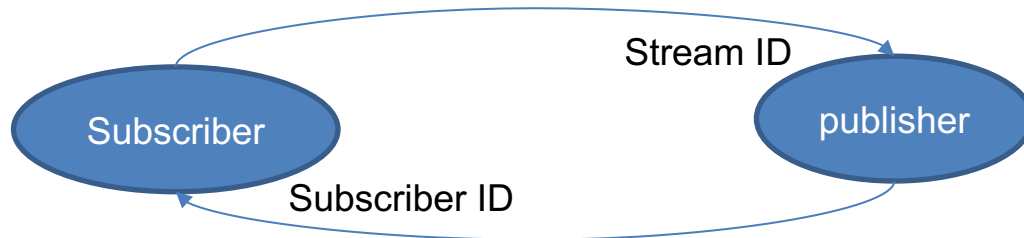
Payload

RTP

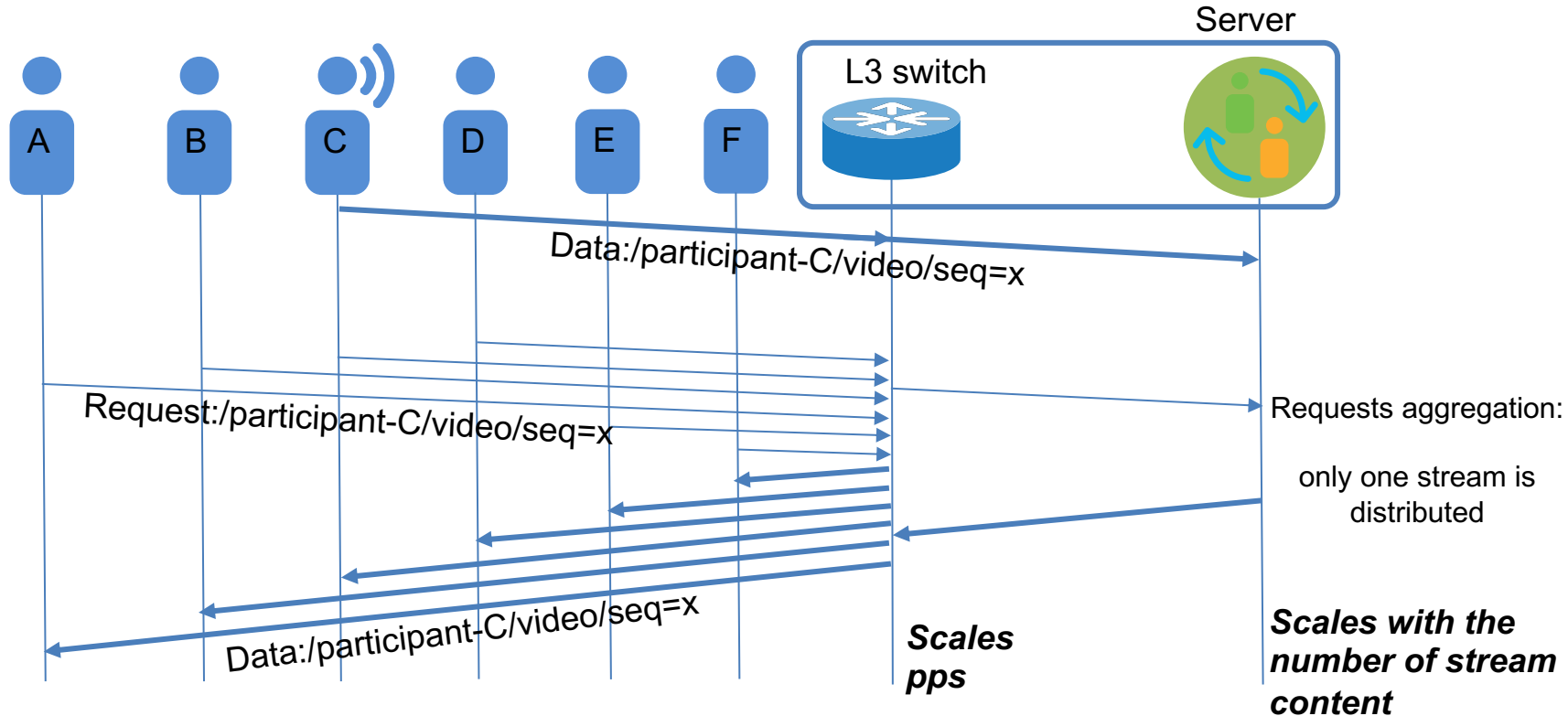
HICN

UDP

IP

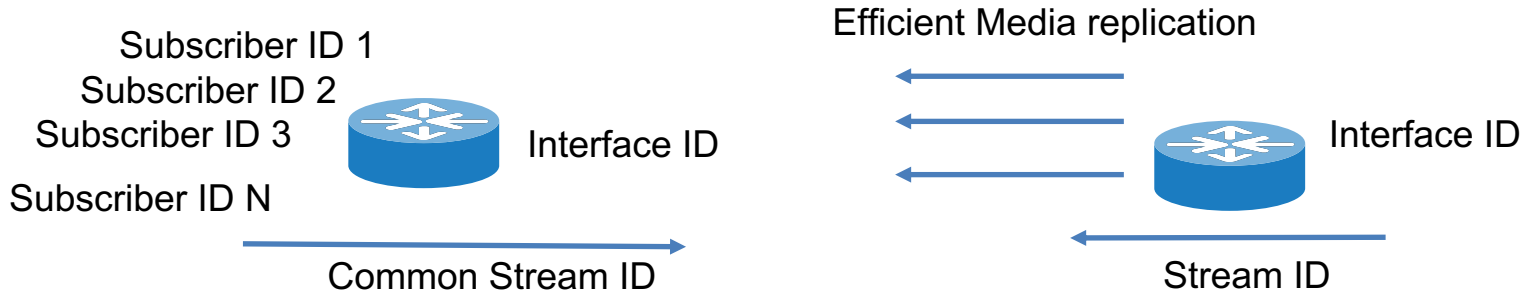


The pub/sub interaction model



The L3 media relay node

- The media relay makes use of the HICN L3 switch, which can interpret media subscriptions requests and data streams and replicate to all interested parties.
- It's a VPP forwarder, with a out-of-tree plugin, HICN, that does three main operations
 - Stores packet requests (subscriptions)
 - Replicates packet replies (publications)
 - It performs NAT operations on subscriber IDs, which is an IP address, stream IDs are immutable (HICN messages are signed)

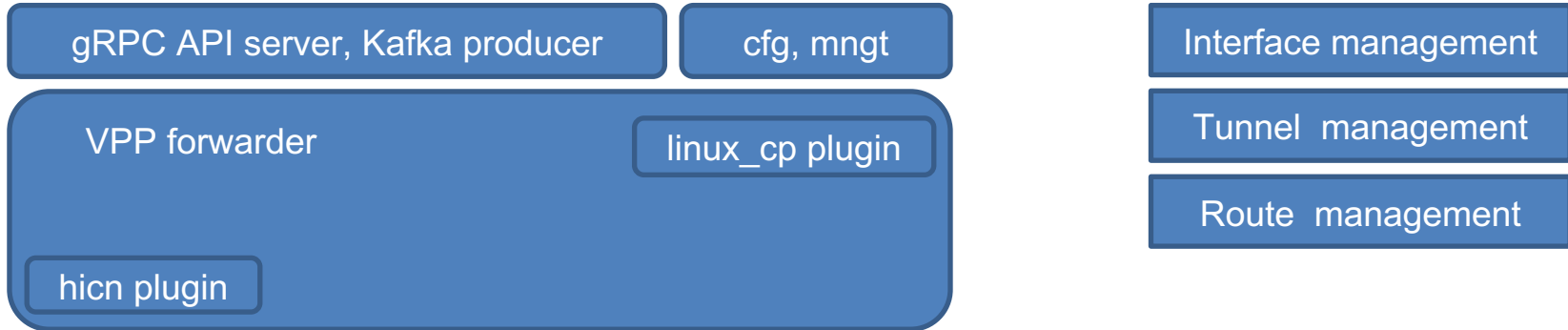


L3 media relay performance

- Comparable performance of a NAT, 6Mpps per core
- At 5Mbps per HD video/content-sharing streamx/audio, 1kB typical packet size, 625pps per stream
- 9600 media streams per core for the distribution channel.
- A scalable SFU can manage 150 streams per core in the best case (using WebRTC livekit open source implementation)
- This makes it feasible to deploy media-relay nodes at the edge and far-edge at little cost.

L3 switch control and management

- The L3 switch is managed by via a gRPC API server, based on the goVPP project which creates Golang bindings of the VPP+plugins binary API.
- The linux_cp can also be used for cfg/management.
- Control and management of the media bridge becomes affectively network control and management.
- The SFU becomes a control plane function.
- Media-relay interconnection is as simple as managing IP routing.



Open source components



All components under Apache 2.0

<https://github.com/cisco-open/hicn-sdk>

<https://github.com/pion>

<https://github.com/livekit>

Conclusion and work in progress

- By introducing HICN pub/sub transport in the client and the server we can offload the application from doing media switching at scale and delegate fully to the network as an L3 function
- The L3 media relay scales with traffic and not with the number of media streams
- It becomes possible to deploy media relays at the edge at scale.
- Control and management becomes a network control and management and can be integrated in networking workflows
- The application can focus on application operations.
- A PoC is in progress which integrates the WebRTC Pion project (github.com/pion) written in Golang.
- WebRTC can potentially scale to an extremely large number of participants w/o requiring costly Cloud deployments.



LF NETWORKING

LFN Developer & Testing Forum

Q&A