Security Level:

ICN based Scalable Video Conferencing on Virtual Edge Service Routers (VSER) Platform

Asit Chakraborti, Aytac Azgin, Ravi Ravindran, G. Q. Wang

www.huawei.com

Version: V1.1

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Agenda

- Motivation
- VSER Platform
- Conferencing over VSER Platform
- More on the Conferencing application
- Results
- Improving the VSER Platform

Motivation

- **Service from the Edge [1]**
  - Service-centric Compute, Storage and Bandwidth scaling
  - Tailor services to locality and user context (mobility, social parameters)
  - Minimize latency and jitter
  - Avoid backbone bottlenecks
- **ICN Deployment [1]**
  - Caching and aggregation at the Edge has already been shown to be effective
  - Names for service/content/device enable context aware network
  - Potential for new business models for network operators
- **NFV/SDN programmability**
  - Enables service and network virtualization
  - Allows management of services as well as ICN network

[1] Xuan Liu at al “Towards software defined ICN based edge-cloud services“ 2013 IEEE CloudNet

Virtual Service Edge Router Platform

Objective:

- A Virtualized ICN Edge Router to host several ICN Services
- Service Orchestration layer for Service Control + Management

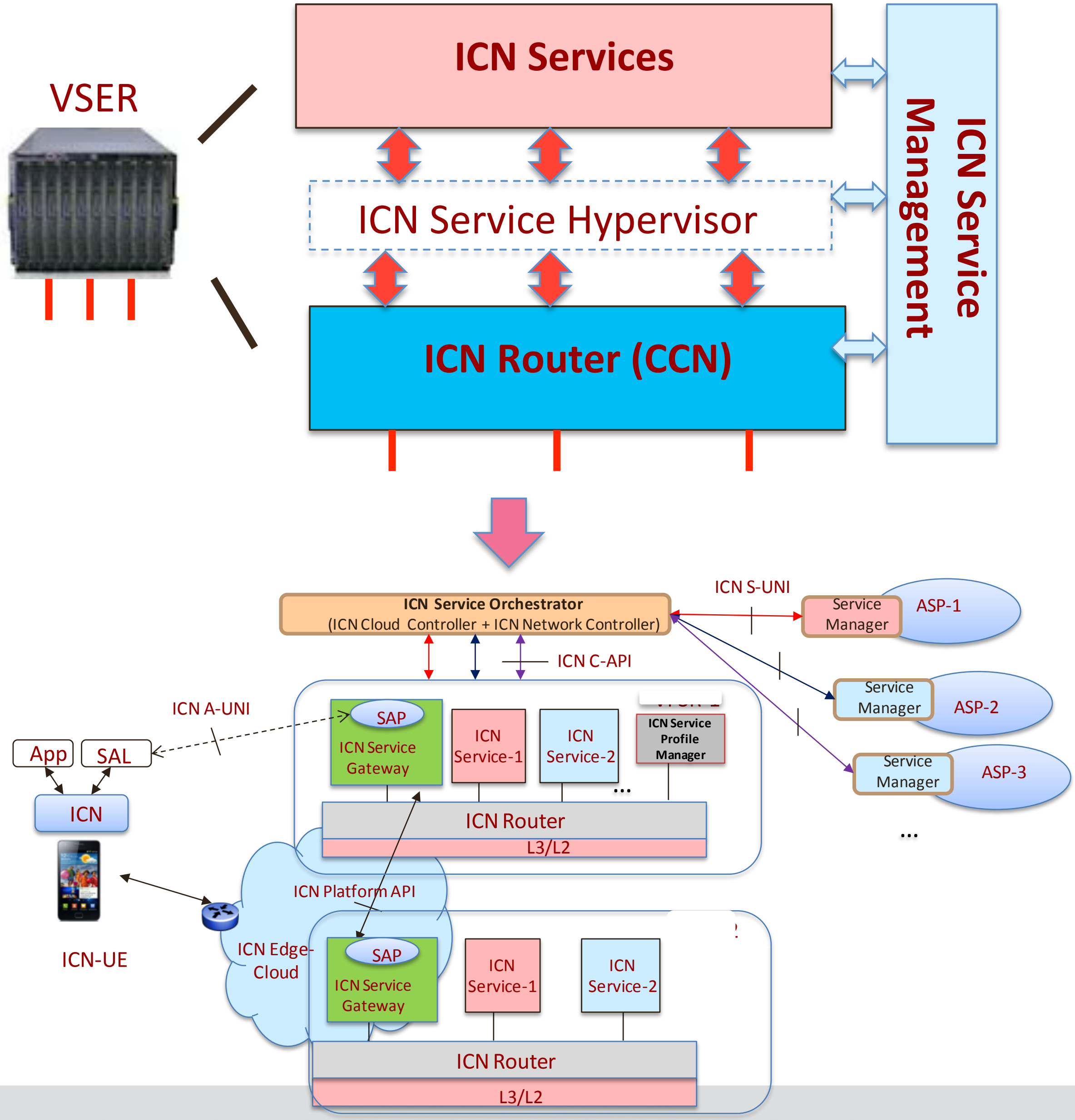
Implementation:

- CCN based Software Router, with Virtualized Service Plugins
- Service controller applications manage service logic.
- CCN service layer components extends to the User Entity for service interaction.

Usage:

- Supports both real-time and non-real time services

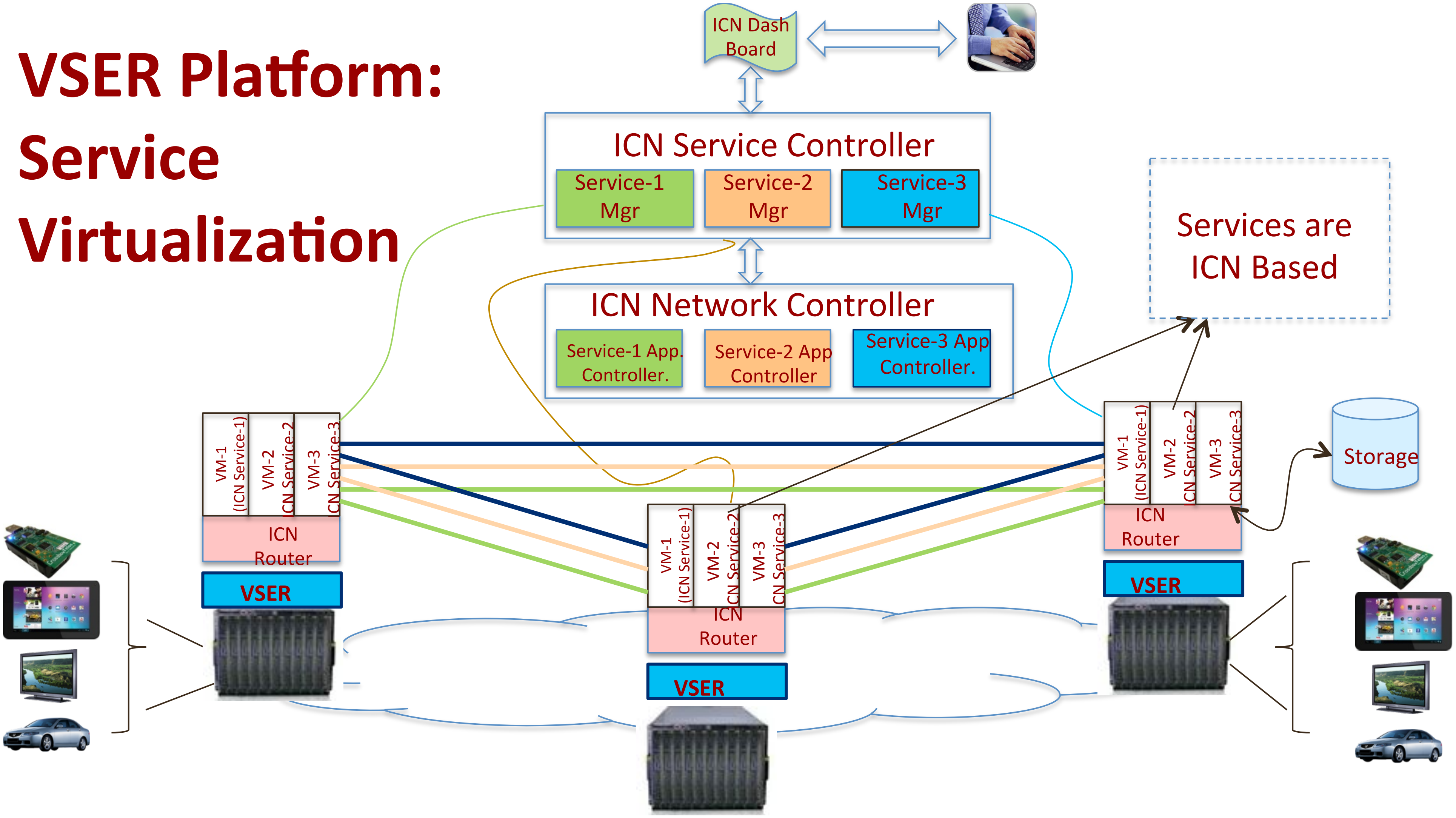
Virtual Service Edge Router High Level View



VSER Platform Highlights

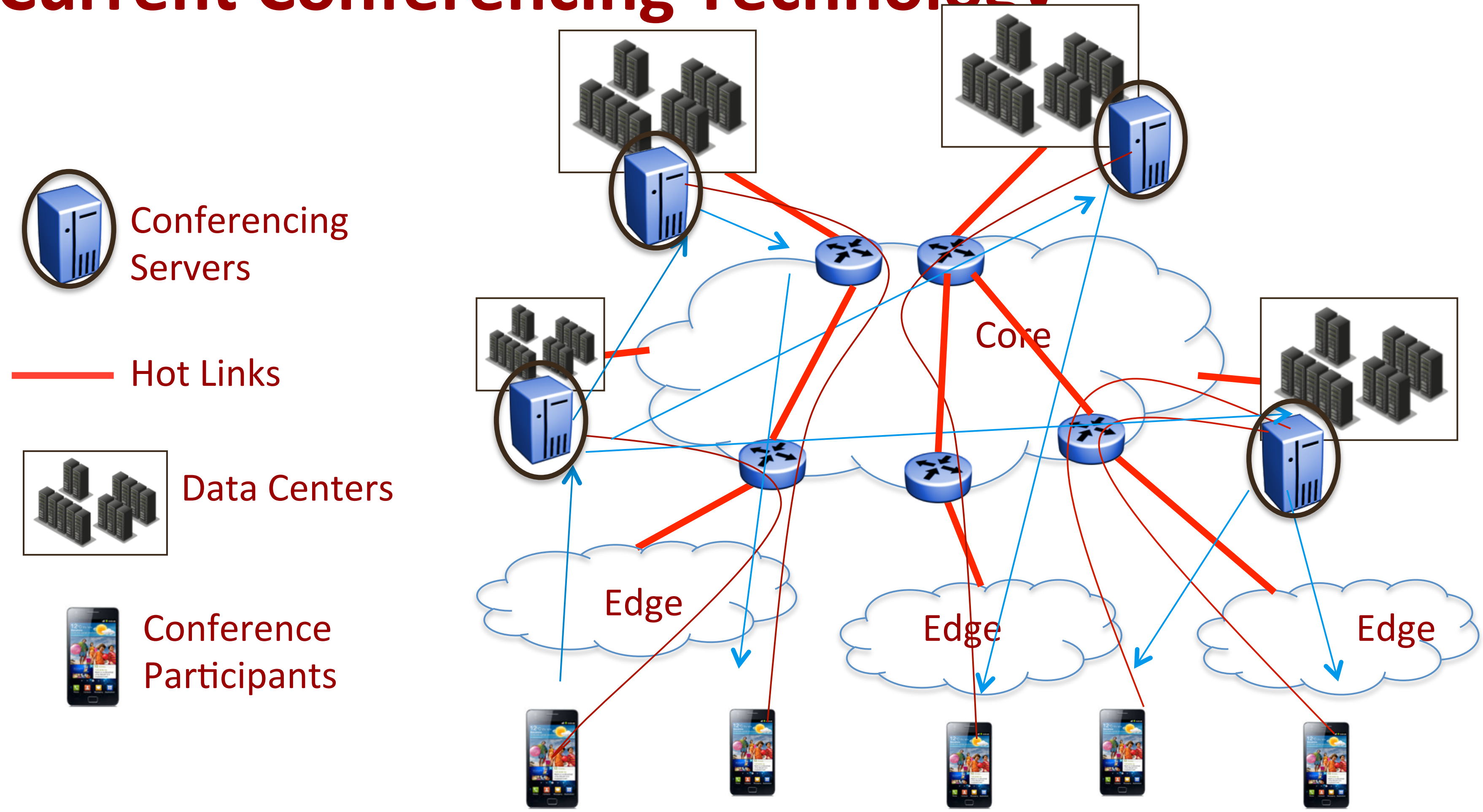
- Service Edge Router
- Non-proprietary Platform
- Overlay deployment of ICN
- Optimized software stack including Multi-threaded CCNx
- Service Management by OpenStack and FloodLight
- Service Discovery, Service/ Network Programmability
- Generalized to any service, real-time (conferencing, IOT) or non real-time (content delivery)

VSER Platform: Service Virtualization



- **Services can be anything: Real-time (E.g. Conferencing), Non-Real time (VoD, M2M)**
- **The VM's interconnect at specific application/service level, ICN Router helps with Name based Routing, Caching, Multicasting**

Current Conferencing Technology



Conferencing Servers

Hot Links

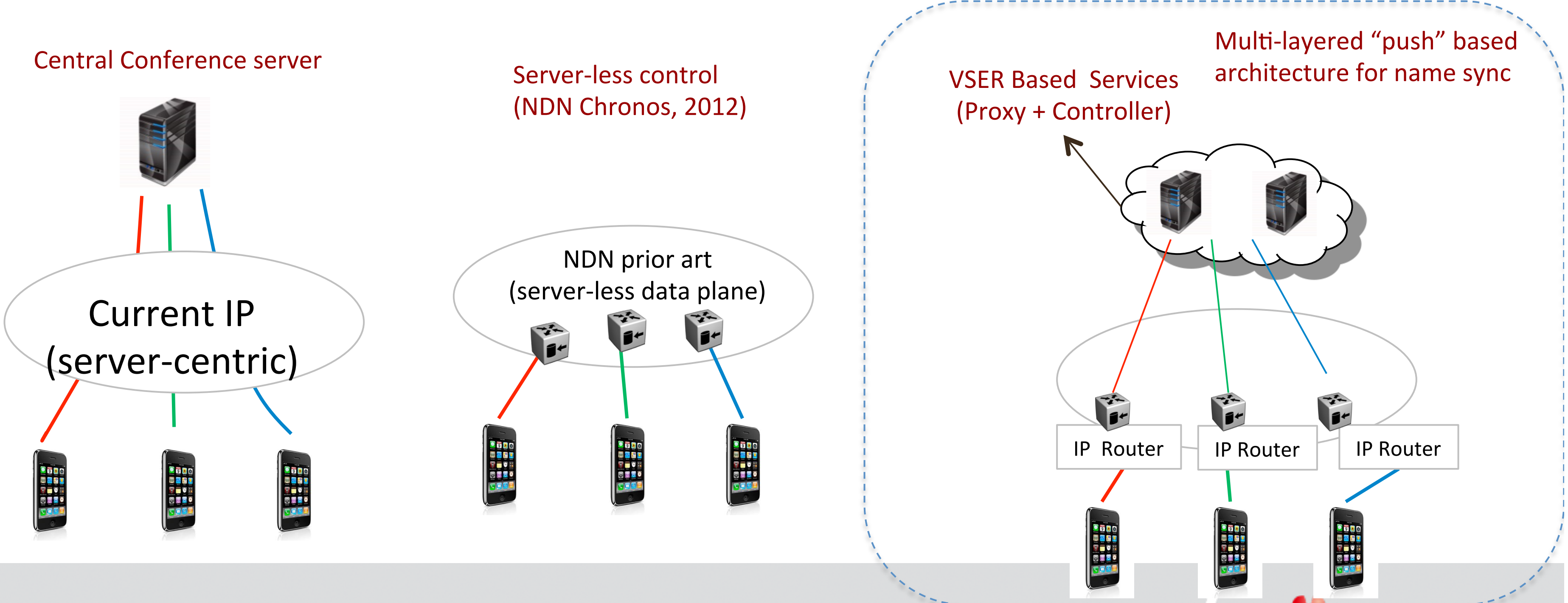
Data Centers

Conference Participants

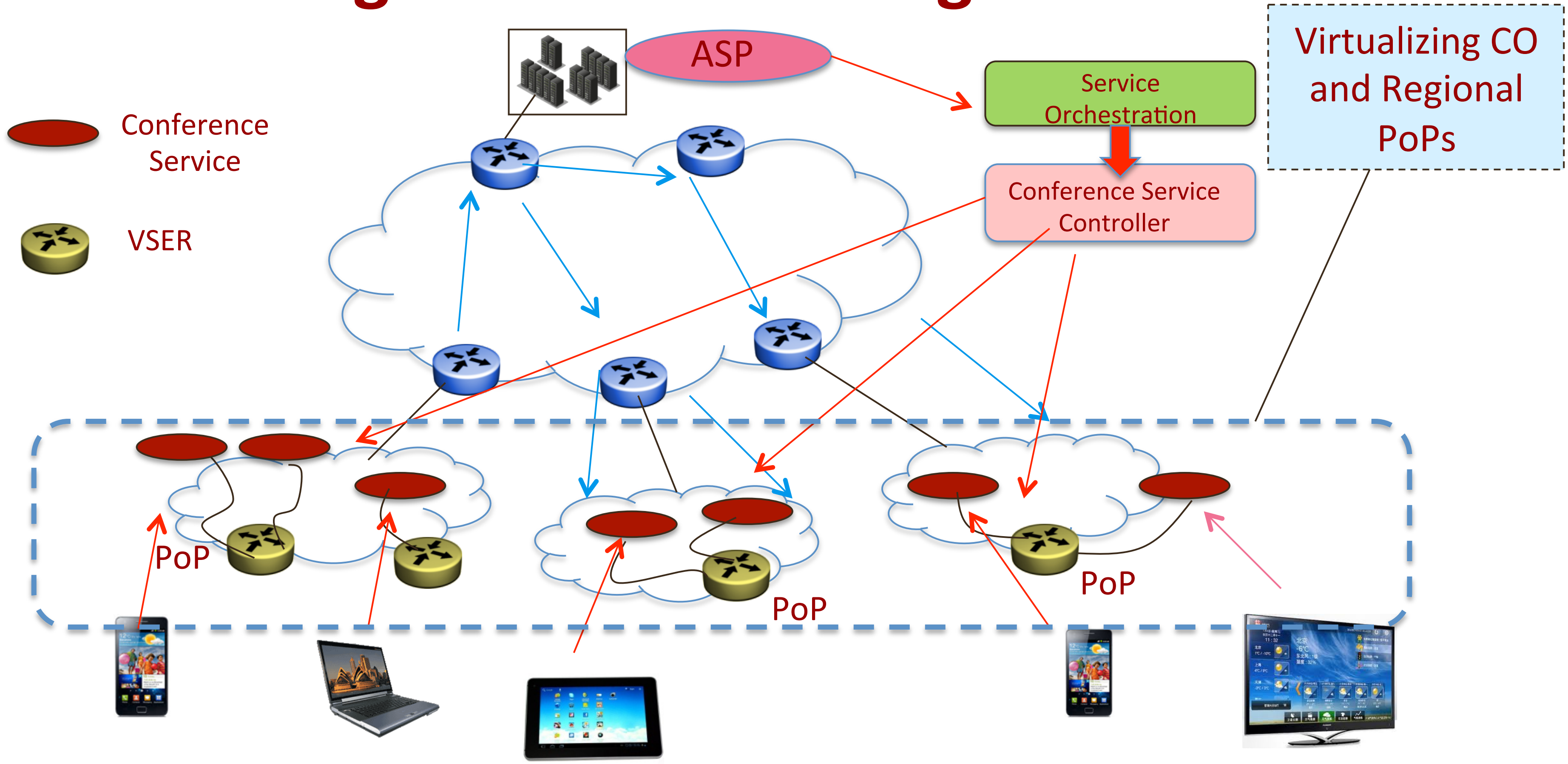
- **Today a conferencing is typically scaled using a network of servers in the network.**
- **Unicast - Number of streams in the network is still O(N^2) and potentially traverse many bottleneck links.**

Large Scale Multi-Media Conferencing

- Large scale conferencing is a problem due to session and state complexity.
- We realize a network based conferencing system on the VSER platform.
- Includes conference framework: Client-Agent, Proxy, and Conference Controller.
- The conference framework modules only synchronize digests among participants, the data is multicast in the data plane

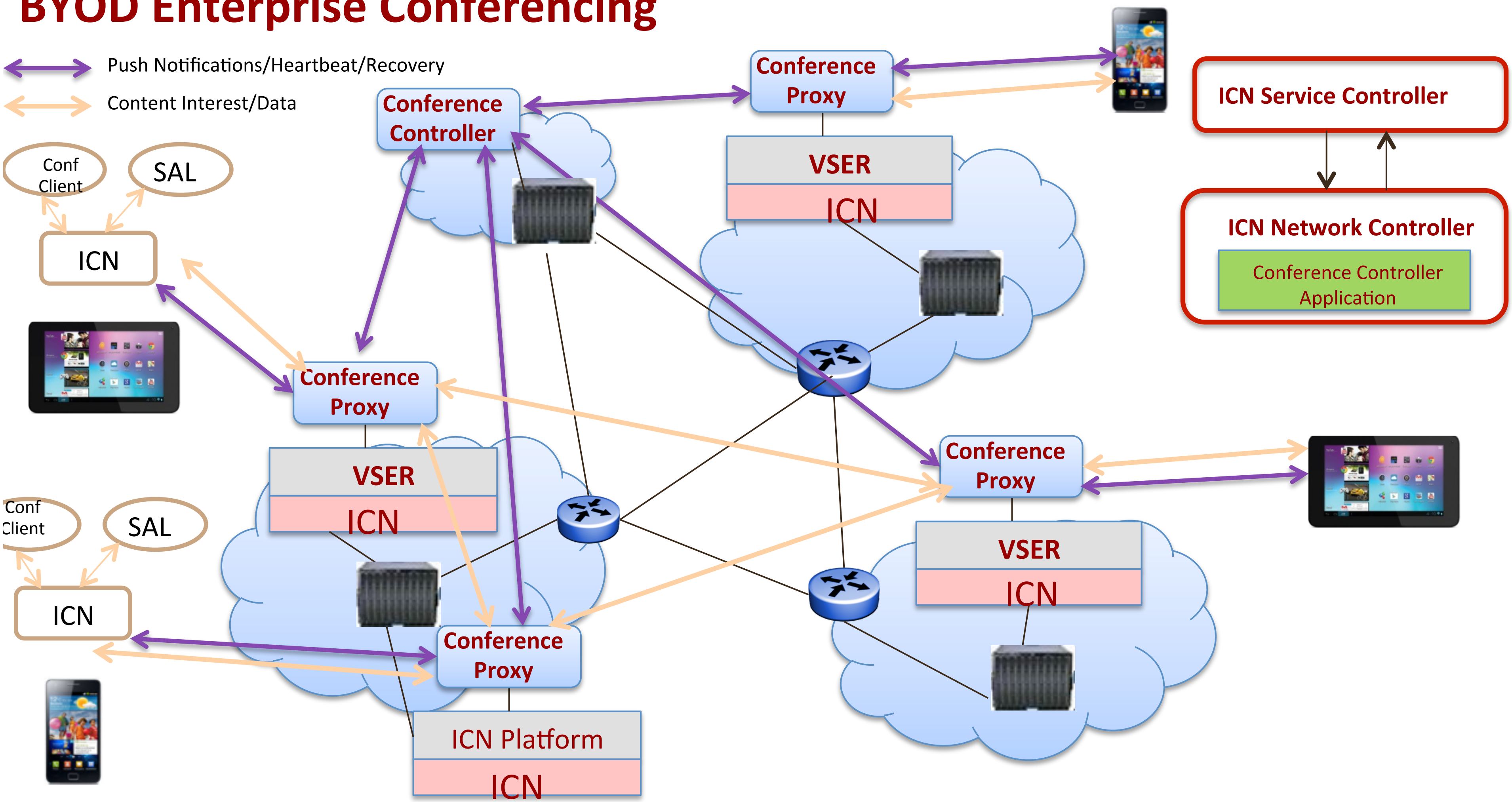


Conferencing as Virtualized Edge Service


ASP
Service Orchestration
Conference Service Controller
Virtualizing CO and Regional PoPs
Conference Service
VSER
PoP
PoP
PoP
PoP

- **Provision conference service points at the network edge – Scales Computing/Bandwidth**
- **More bandwidth saving without expensive/proprietary MCU**
  - Data Streams are aggregated by CCN
- **Virtualize Regional PoP or Central Offices (CO)**

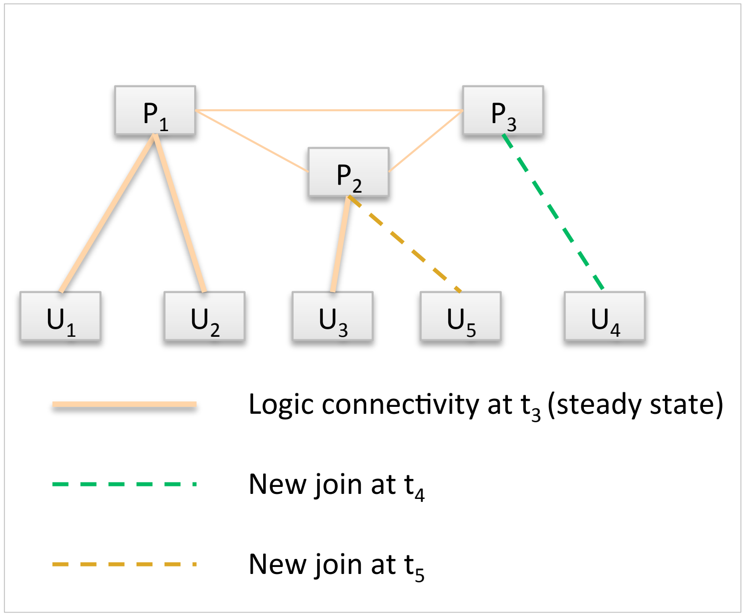
BYOD Enterprise Conferencing



- The conference framework can handle multiple conference instances simultaneously.
- Handling session disruption of UE should be easy using digest logs at Proxy and Controller
- The Conf. Controller Application handles events due to Participant Join/Leave, Load balancing etc.

[1] Ravi Ravindran et al, "Towards Software Defined ICN based Edge Cloud Service", IEEE, CloudNet, 2013

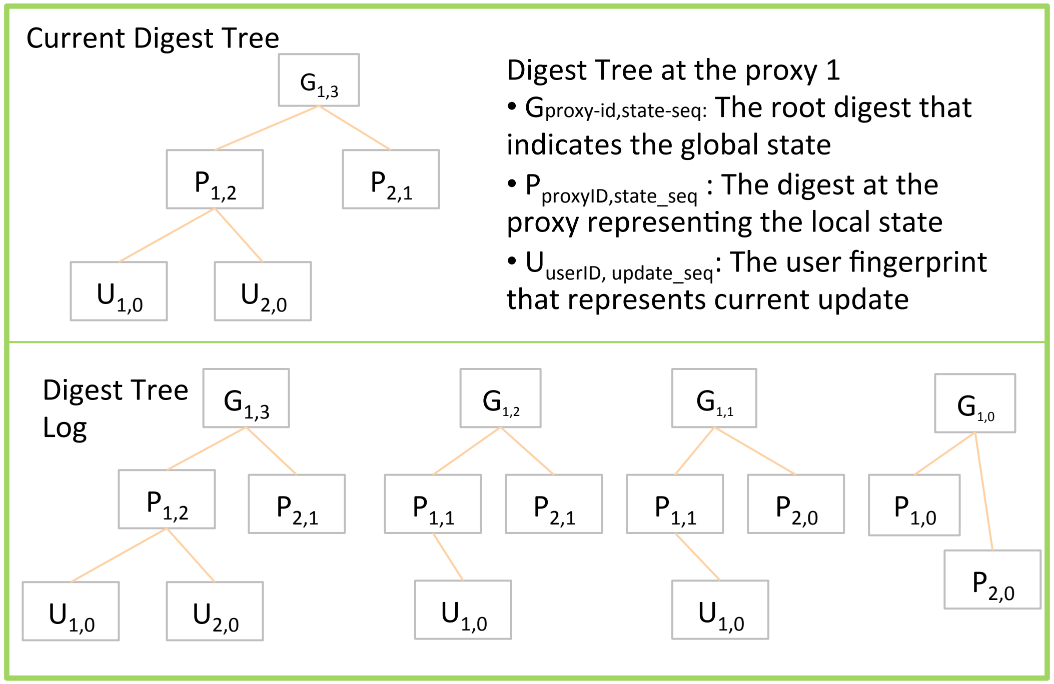
Conerence state management



Logic connectivity at t_3 (steady state)

New join at t_4

New join at t_5



Digest Tree at the proxy 1

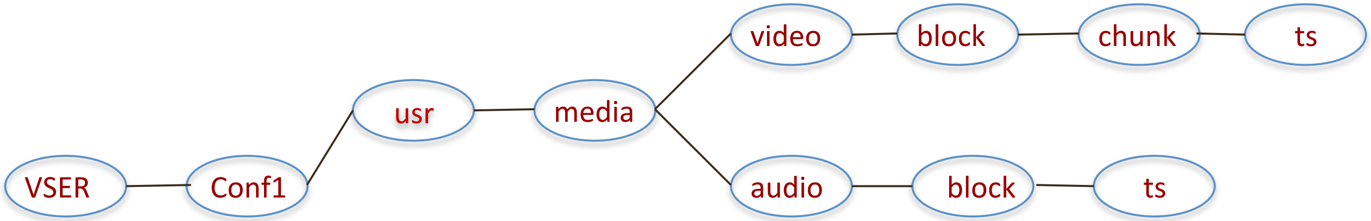
- $G_{\text{proxy-id,state-seq}}$: The root digest that indicates the global state
- $P_{\text{proxyID,state\_seq}}$: The digest at the proxy representing the local state
- $U_{\text{userID, update\_seq}}$: The user fingerprint that represents current update

Current Digest @ U_1	Log @ U_1
$<G_{1,3}>, U_{2,0}$	t_4: $G_{1,3}$: $U_{2,0}$ t_3: $G_{1,2}$: $U_{3,0}$ t_2: $G_{1,1}$: $U_{1,0}$ t_1: $G_{1,0}$: $U_{1,0}$ t_0: $G_{0,0}$

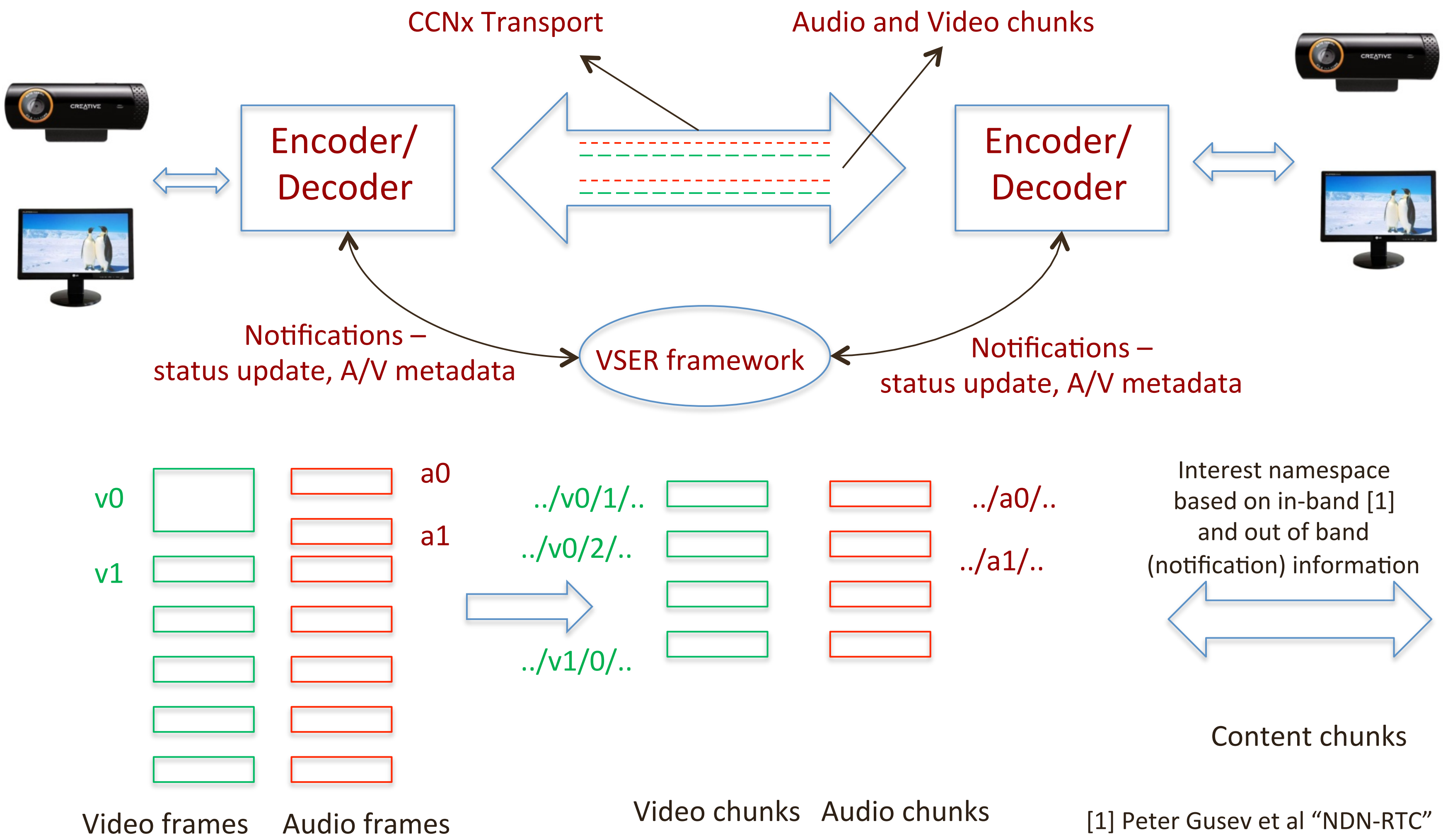
Digest Tree & Log Examples

Notification and namespace

- **Notifications are at the core of the conferencing service**
  - Represents the state change of a participant
  - Pushed by the conferencing service to all participants
  - Useful for recovery and history
  - Can be used at different granularity
    - ❑ A new notification for every text chat entry
    - ❑ A new notification every time an i-frame is available
    - ❑ A new notification indicating audio/video timestamp every few seconds

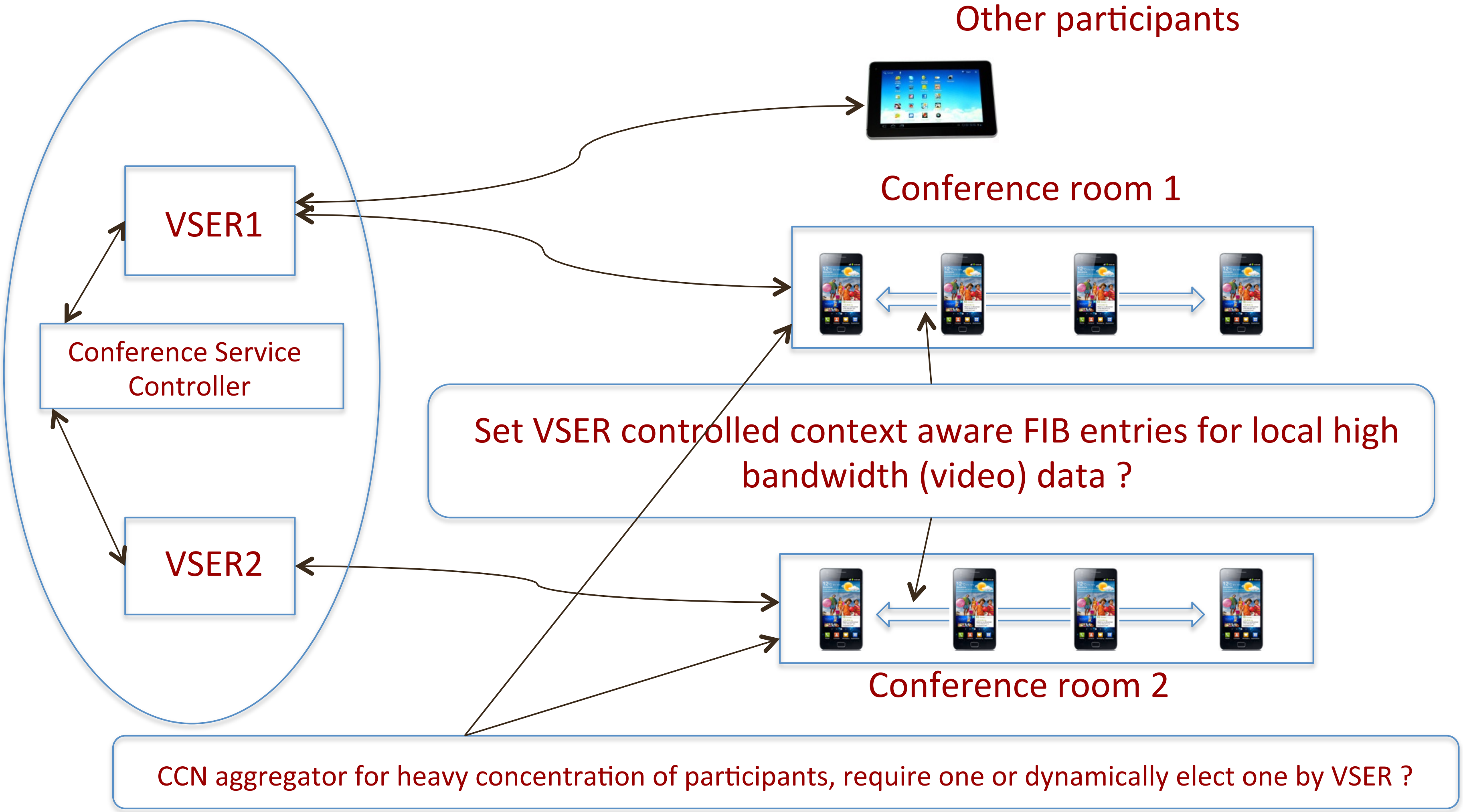


Audio Video Transport



[1] Peter Gusev et al “NDN-RTC”

Intelligent data-path management by VSER: Minimizing bandwidth



Other participants

Conference room 1

VSER1

Conference Service Controller

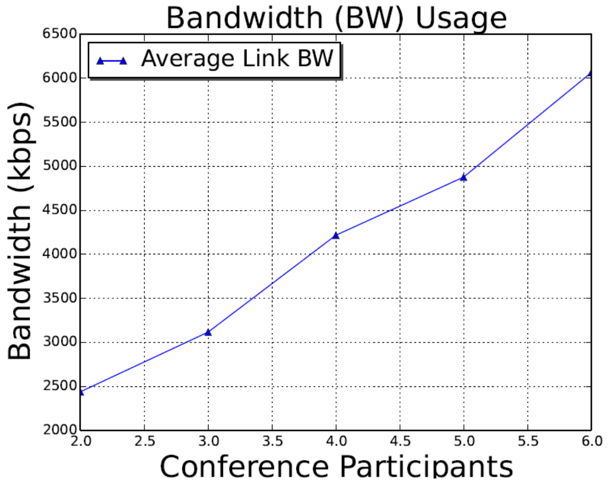
VSER2

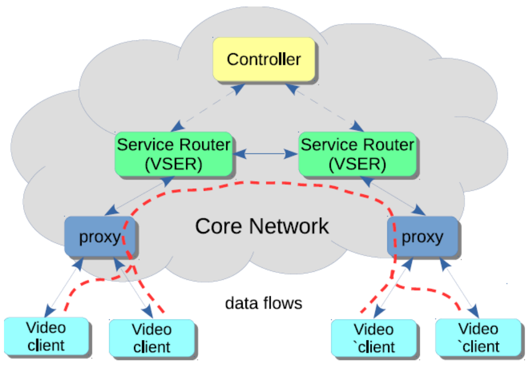
Set VSER controlled context aware FIB entries for local high bandwidth (video) data ?

Conference room 2

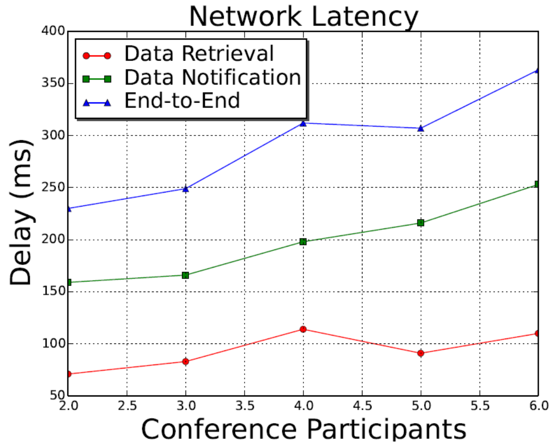
CCN aggregator for heavy concentration of participants, require one or dynamically elect one by VSER ?

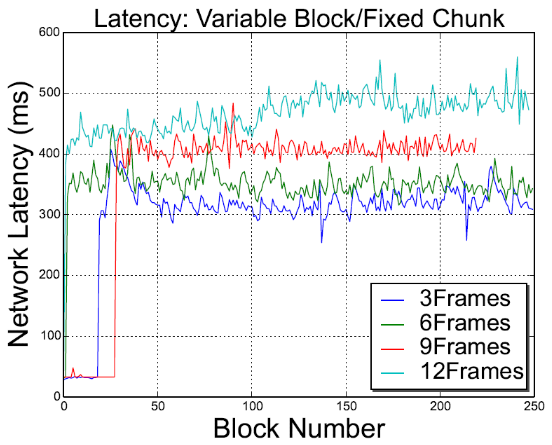
Results for Video Conferencing over VSER Platform [1]





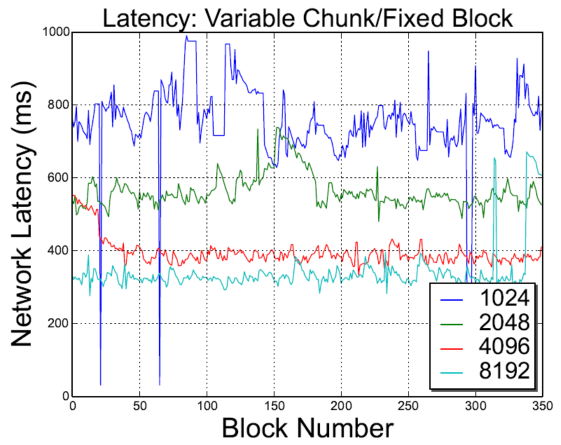
Test Setup





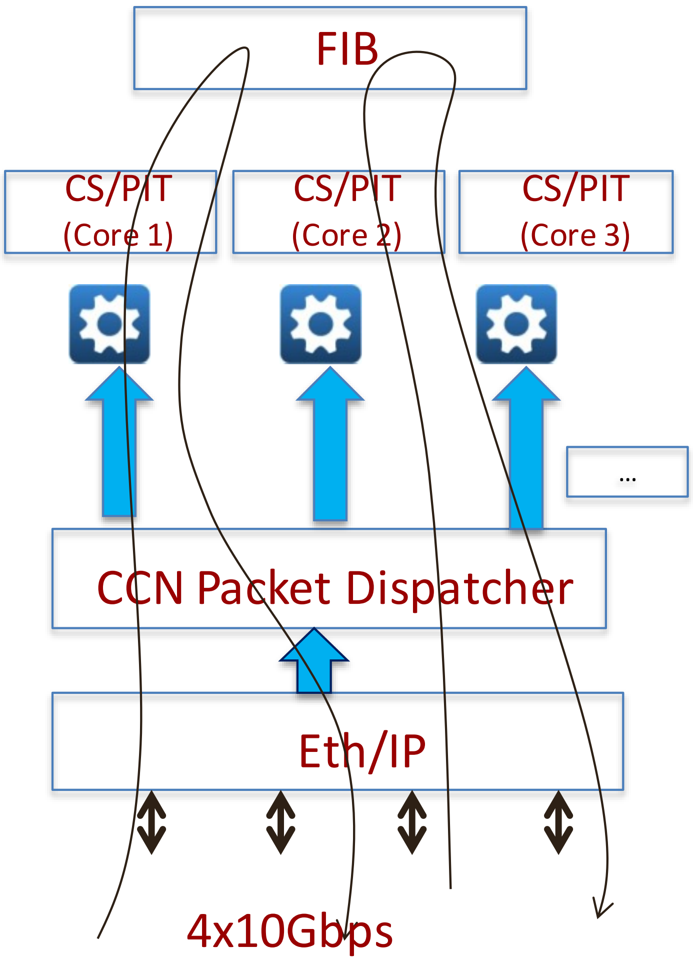
Problems:

1. Audio & Sync
2. Frame Transport
3. Notification dependence
4. Bandwidth management
5. De-jitter buffer



[1] Anil Jangam et al "Real-time Multi-party Video Conferencing Service over Information Centric Network" MuSIC 2015

VSER Platform: Multi-core Software Router


FIB

CS/PIT (Core 1)

CS/PIT (Core 2)

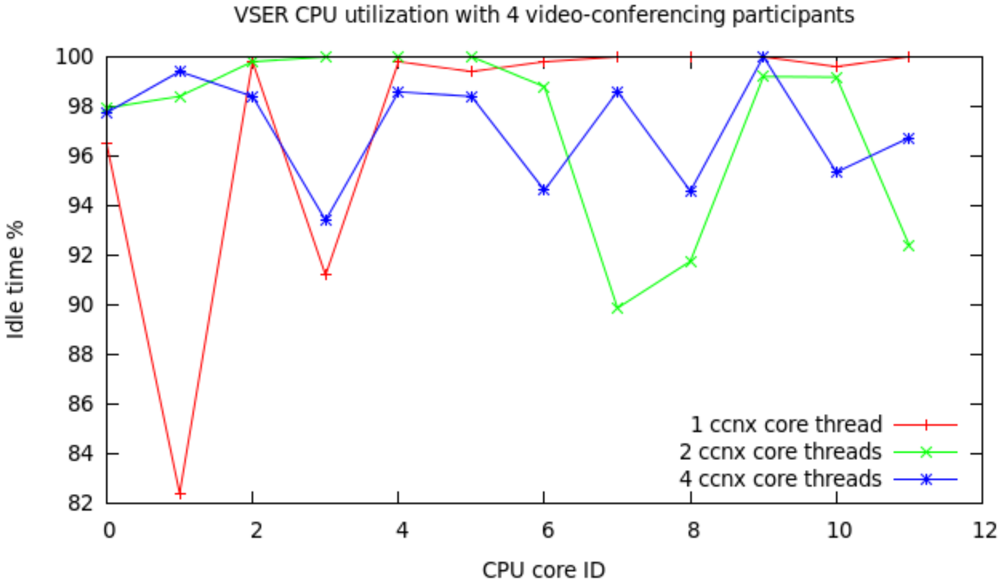
CS/PIT (Core 3)

…

CCN Packet Dispatcher

Eth/IP

4x10Gbps



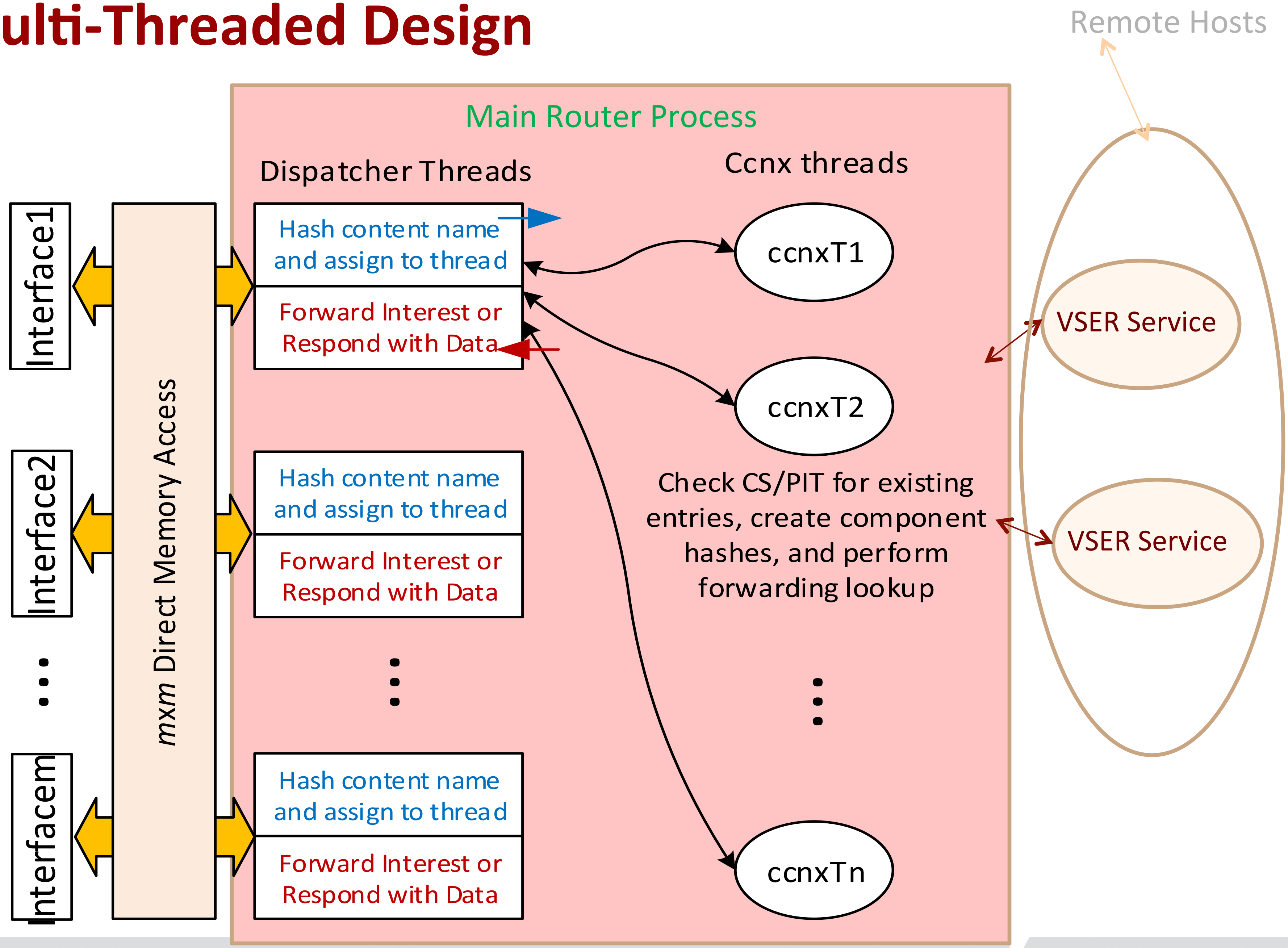
- 2 Processor Xeon (2.9Ghz), SDRAM: 128GB
- Running TLV based implementation

Multi-threaded Software Router

- **Problems with the "current ccnd" implementation**
  - Compute-intensive and runs single threaded, many cores under-utilized
  - Represents significant bottleneck on a high performance router with multiple 10GbE links
- **We need to parallelize "ccnd" by defining subtasks**
  - Dispatcher, responsible for packet I/O, with one thread per interface
  - Core-ccnx (*ccnx*-Threads), which divides entire namespace [1] into sub-namespaces and assigns one thread per sub-namespace
- **Need to optimize *linux* for 10GbE operation**
- **Global FIB, but per core thread PIT/CS**
  - Investigating distributed FIB
- **Investigating impact of queues between threads**

[1] Won So et al, "*Name Data Networking on a Router: Fast and DoS Resistent Forwarding with Hash Tables*", ANCS, 2013

Multi-Threaded Design


Remote Hosts
Main Router Process
Dispatcher Threads
Ccnx threads
Interface1
Interface2
Interfacem
mxm Direct Memory Access
Hash content name and assign to thread
Forward Interest or Respond with Data
Hash content name and assign to thread
Forward Interest or Respond with Data
Hash content name and assign to thread
Forward Interest or Respond with Data
ccnxT1
ccnxT2
ccnxTn
Check CS/PIT for existing entries, create component hashes, and perform forwarding lookup
VSER Service
VSER Service