ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and
Networks Laboratory

Analysis of EADS Case Study: Distributed Heterogeneous Communication System

Simon Perathoner (ETH Zurich)

COMBEST Technical Meeting

EADS Innovation Works, Ottobrunn, Germany, 09 July 2009

Content

PART 1: Glimpse of employed analysis methods

- **Modular Performance Analysis (MPA) based on Real-Time Calculus (RTC)**
  [Thiele et al., 2000]
- **Combination of RTC and Timed Automata (TA): A Hybrid Analysis Method**
  [Lampka, Perathoner, Thiele, 2009]

PART 2: Analysis of EADS Case Study

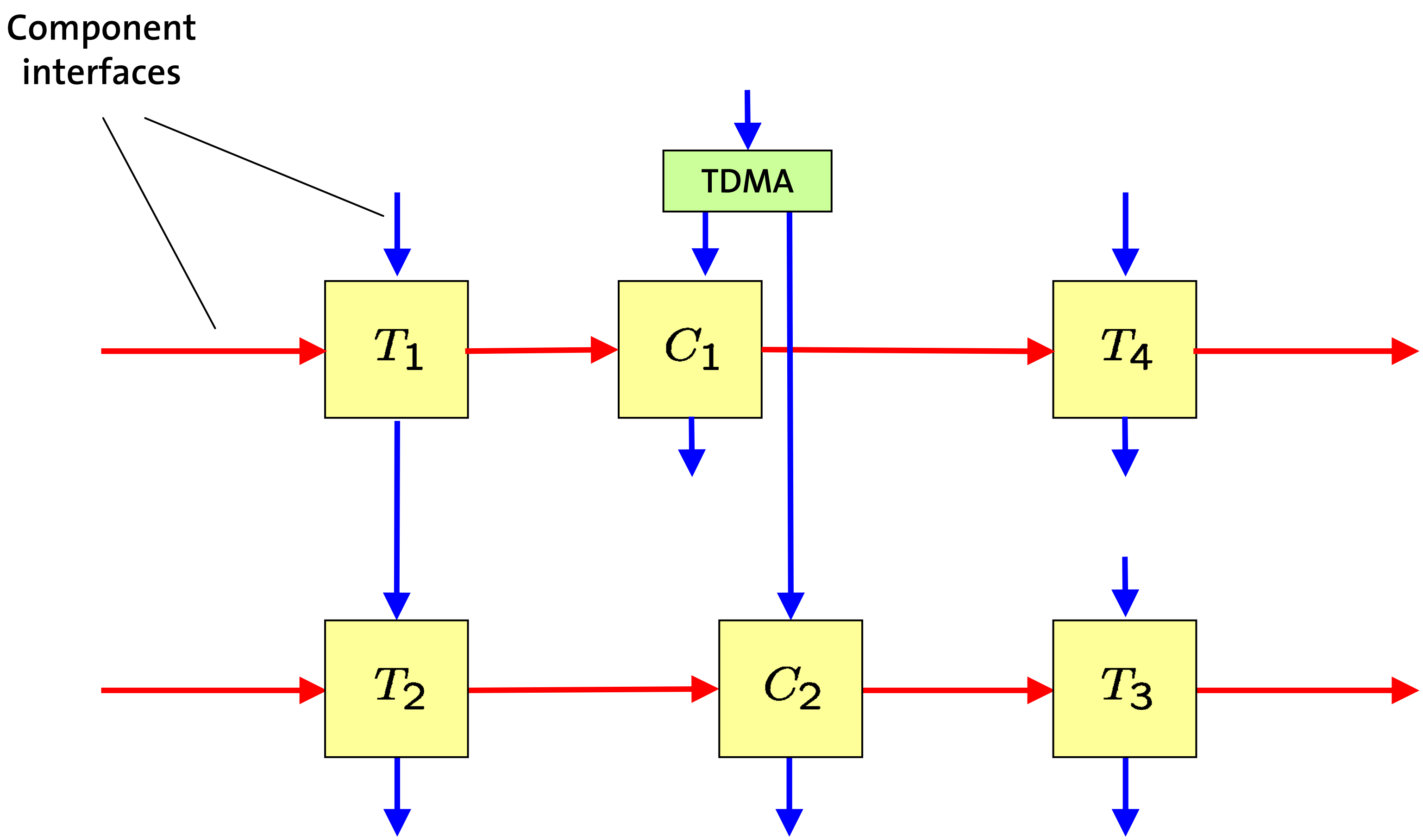
- **Distributed Heterogeneous Communication System by EADS**
- **Traffic characterization**
- **Model and Analysis for simple architecture**
- **Model and Analysis for extended architecture**
- **Remarks on GPS / WFQ**
- **Conclusions, Questions**

PART 1

Analysis methods

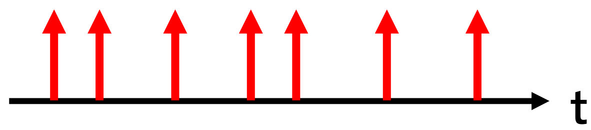
Modular Performance Analysis (MPA)

Analytic approach for performance analysis of distributed real-time systems

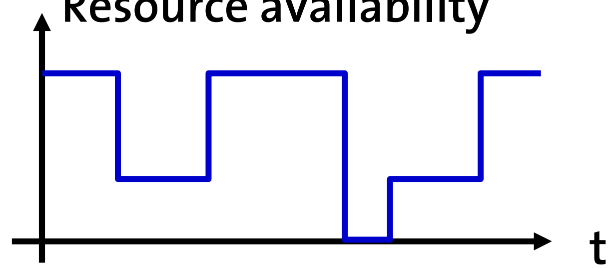


Real-Time Calculus

Event stream

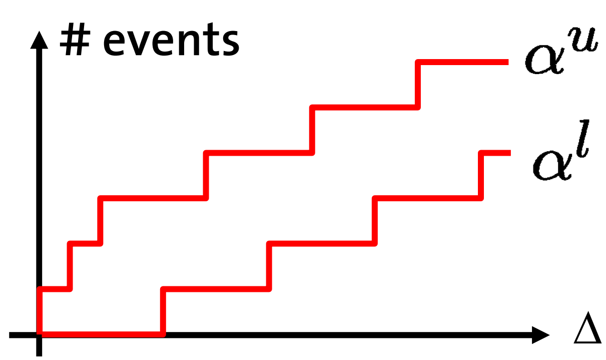


Resource availability

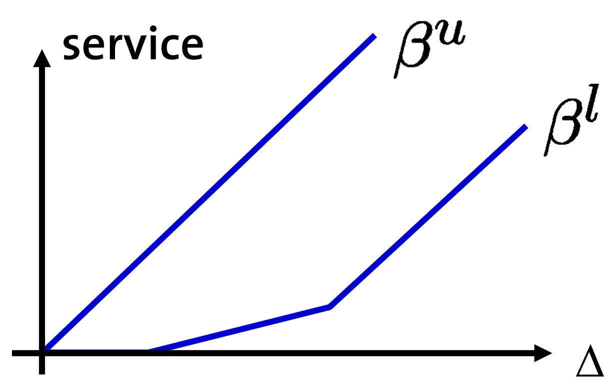


Load model

Service model

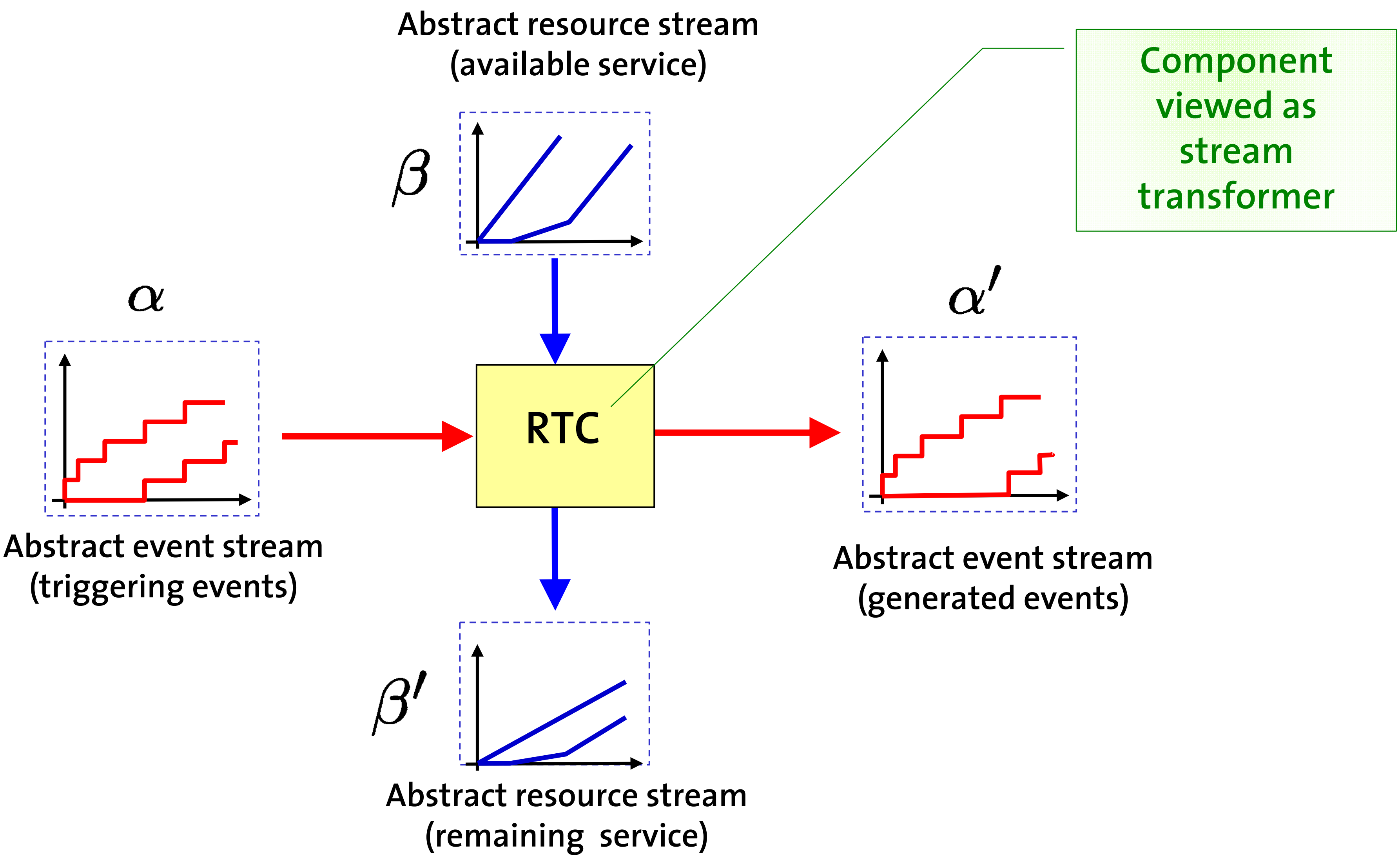


Arrival curves



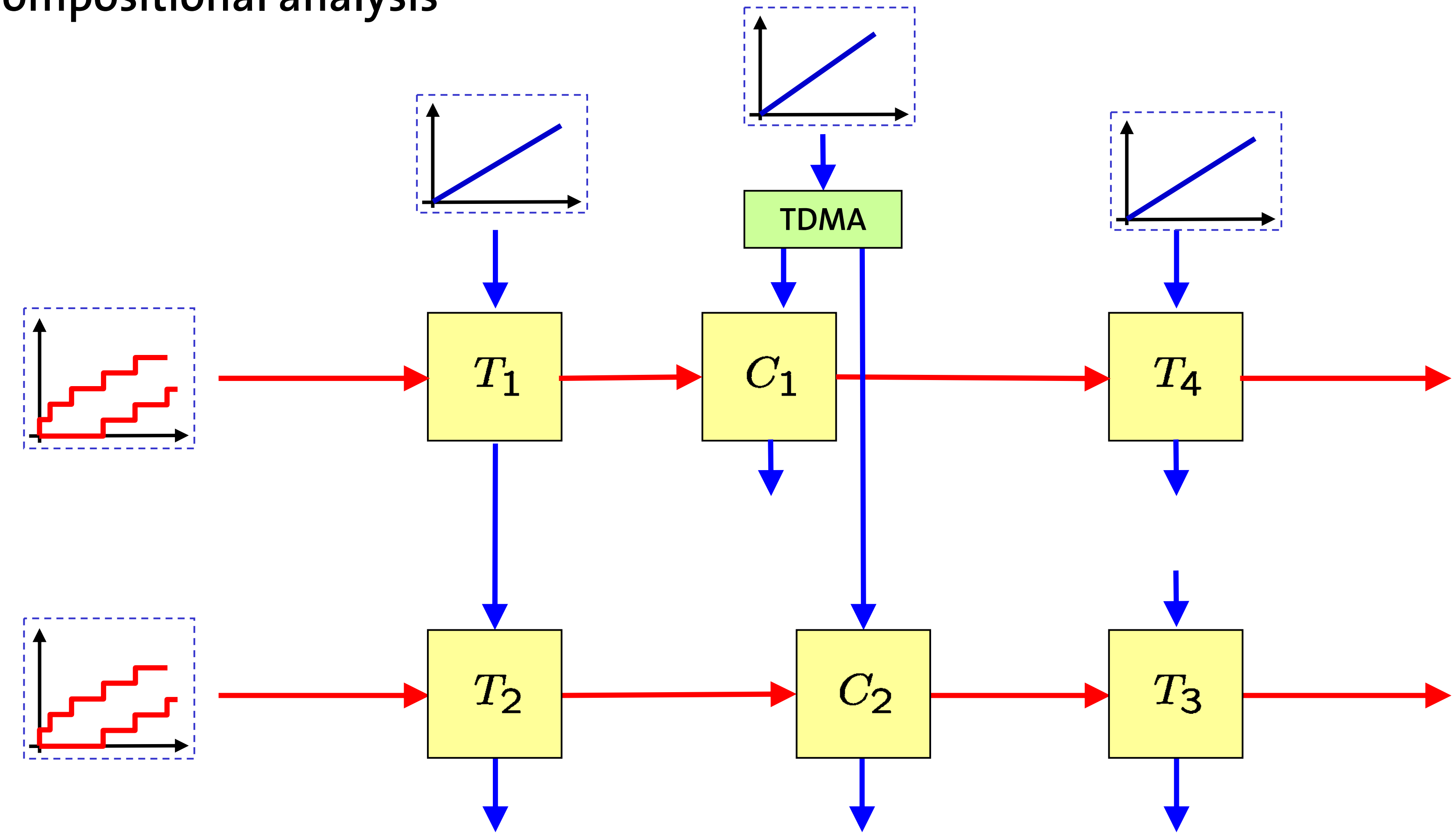
Service curves

Real-Time Calculus



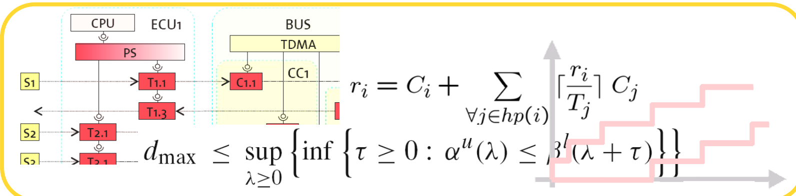
Real-Time Calculus

Compositional analysis



Analytic vs. State-based Approaches

Analytic Real-Time Analysis



Solution of closed form expressions

Examples: RTC, SymTA/S, MAST, ...

+ Good scalability

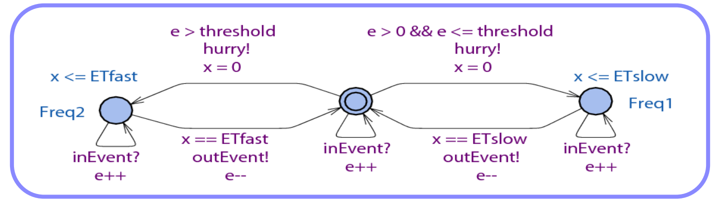
+ Fast analysis

– Limited to few specific measures (e.g. delays, buffer sizes)

– Systems restricted to specific models

– Overly conservative results

State-based Real-Time Analysis



Model checking of properties

Examples: Timed Automata (TA), FSM, ...

– Poor scalability

– Slow verification

} State space explosion

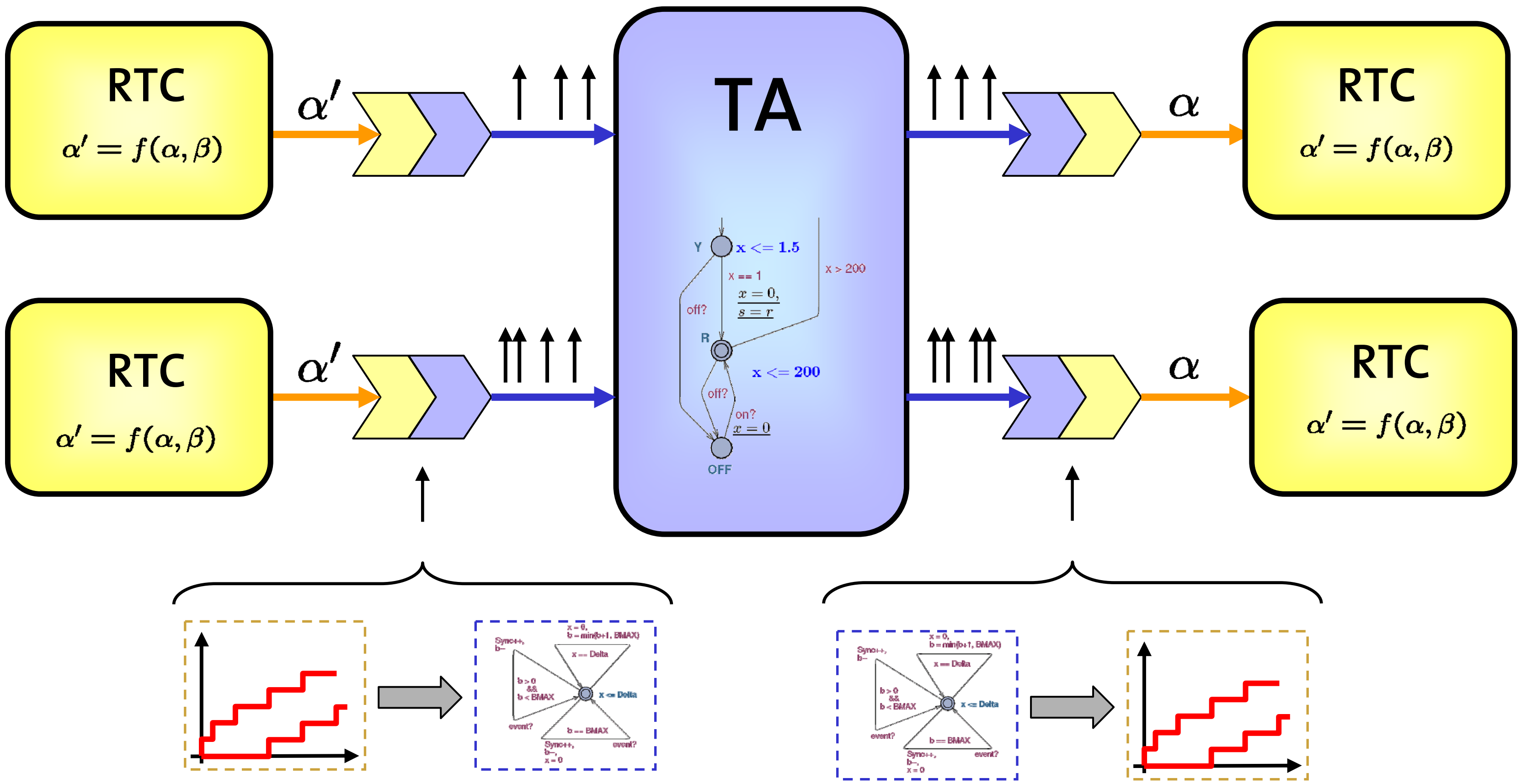
+ Verification of functional and non-functional properties

+ Modeling power

+ Exact results

A Hybrid Analysis Approach

Interfacing Real-Time Calculus (RTC) and Timed Automata (TA)

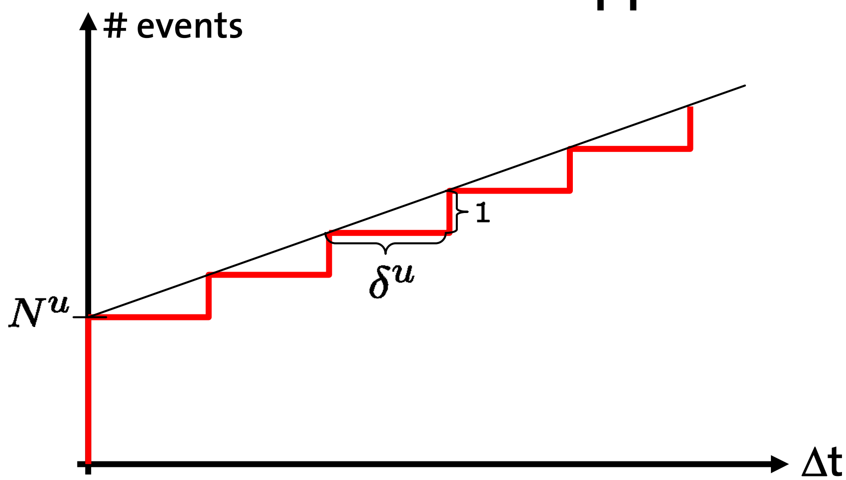


How to represent arrival curves as TA?

How to derive arrival curves from TA subsystem models?

Representation of linear arrival curves as TA

Upper arrival curve

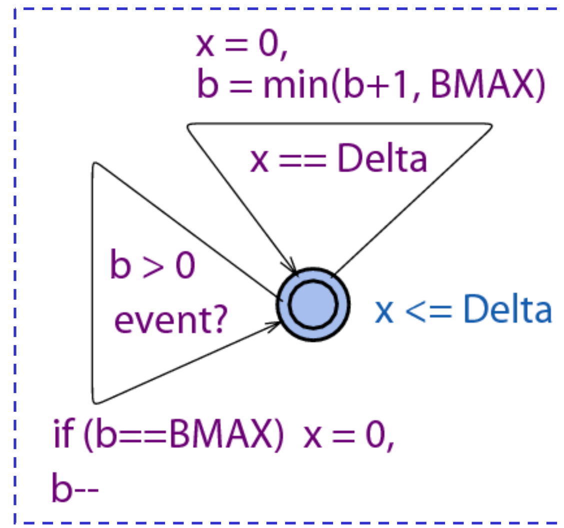


$$\alpha^u(\Delta) = N^u + \left\lfloor \frac{\Delta}{\delta^u} \right\rfloor$$

Max fill level: N^u

Fill rate: $1/\delta^u$

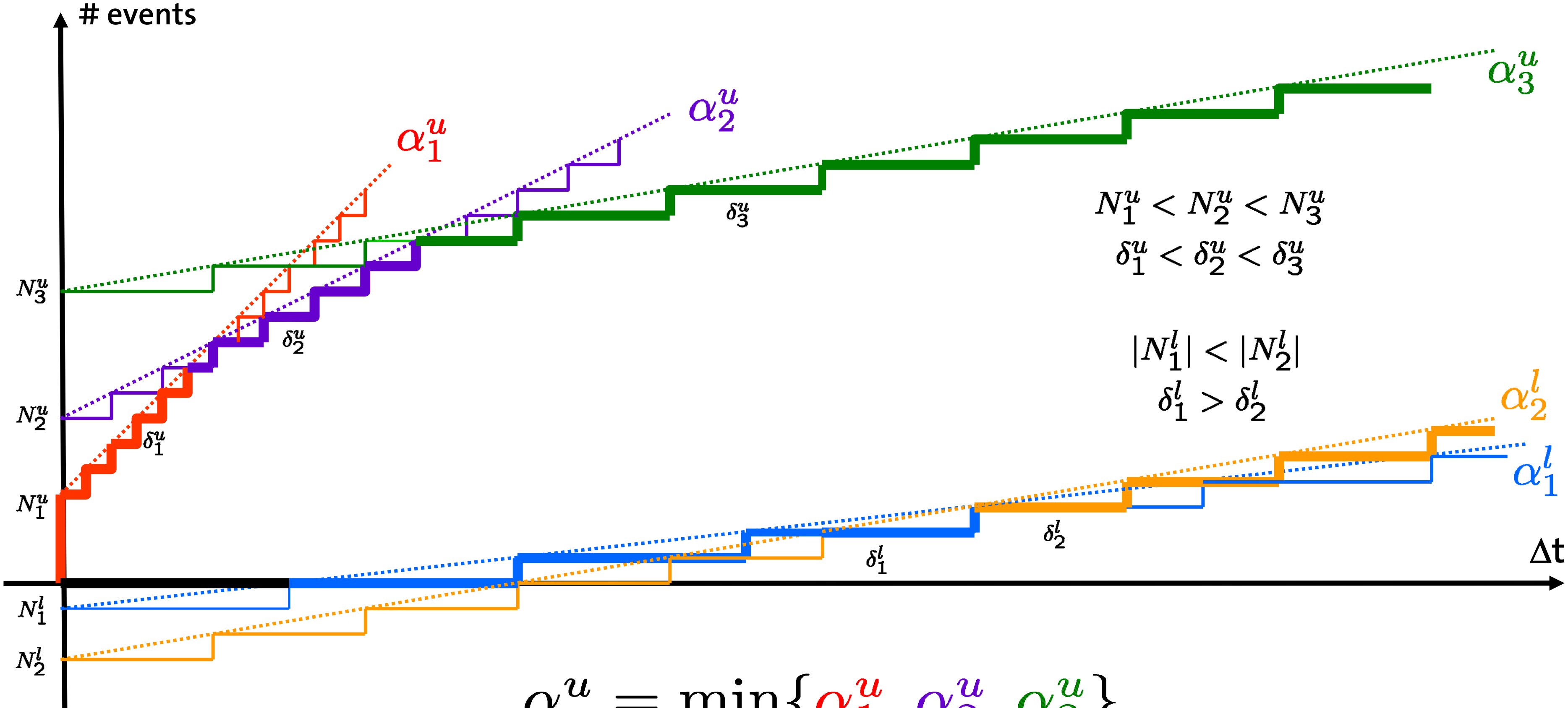
Event emission allowed if fill level > 0



Automaton for linear upper arrival curve (UTA)

Convex and concave patterns

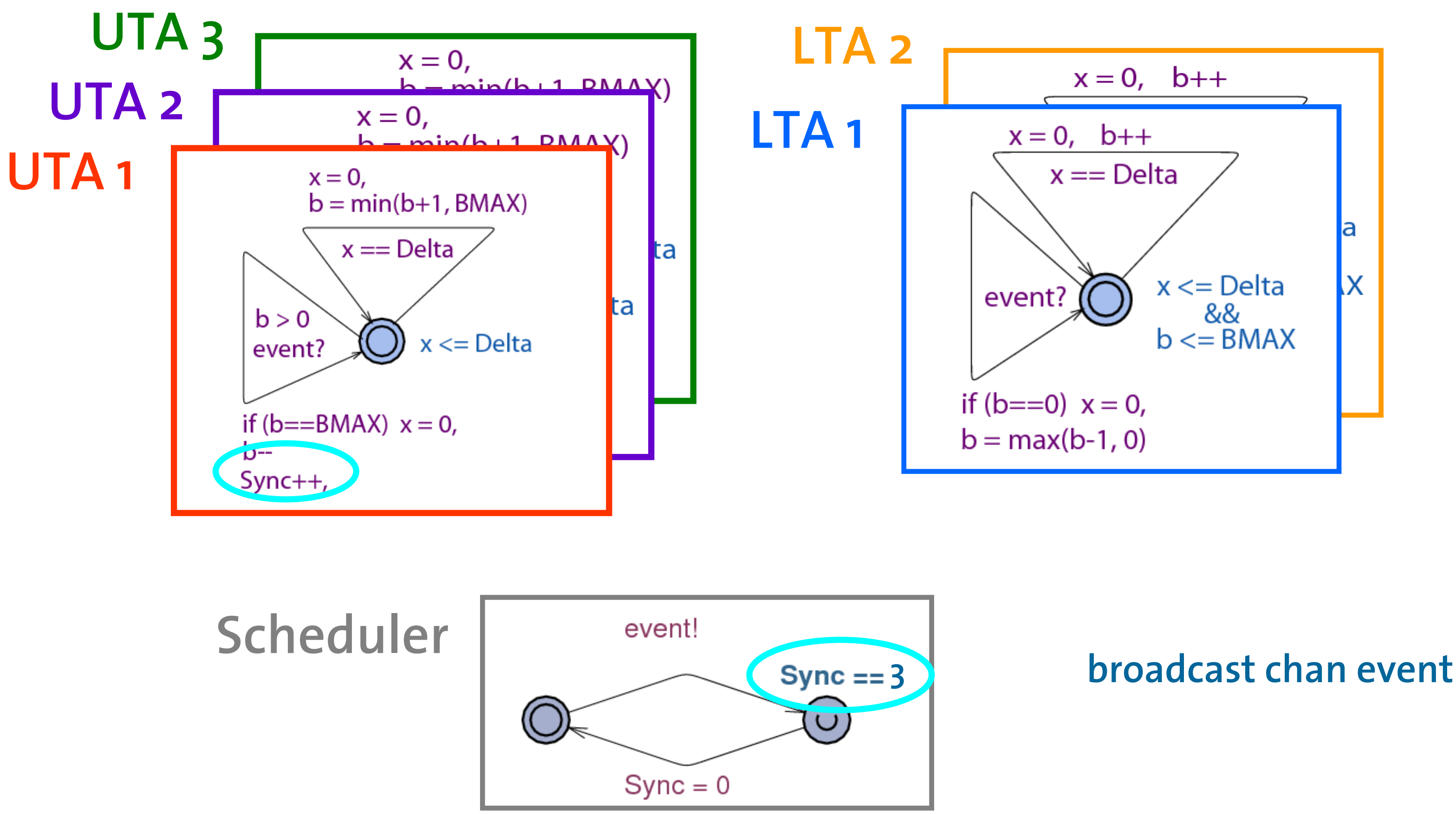
Composition of linear staircase functions



$$\alpha^u = \min\{\alpha_1^u, \alpha_2^u, \alpha_3^u\}$$

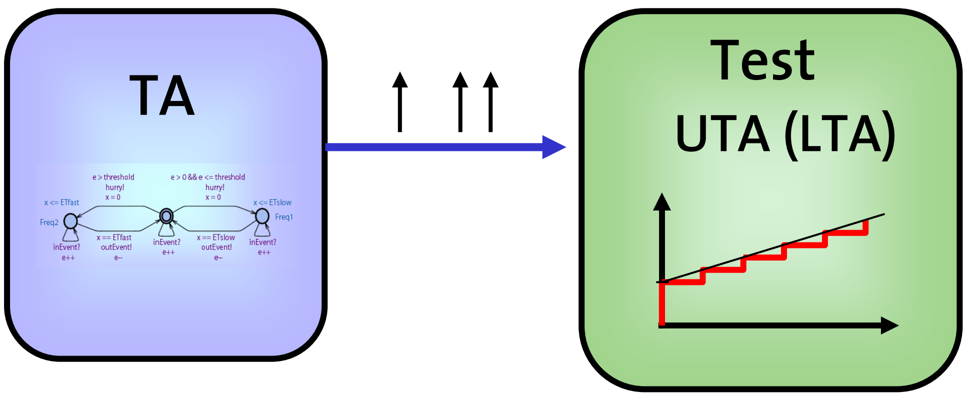
$$\alpha^l = \max\{0, \alpha_1^l, \alpha_2^l\}$$

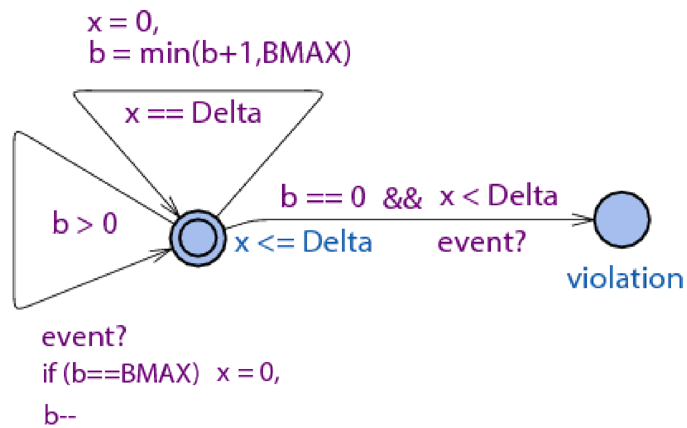
Convex and concave patterns



- Event generation only if all UTA permit it (AND composition)
- Single LTA can enforce event generation (OR composition)

Deriving Arrival Curves from TA





Verify

(A[] (not violation))

- Verify compliance of system output with a number of UTA (N_i,δ_i) and LTA (N_i,δ_i) (Search strategy: Fix one parameter and modify the other by binary search)
- Combine obtained linear staircase functions by min and max operators

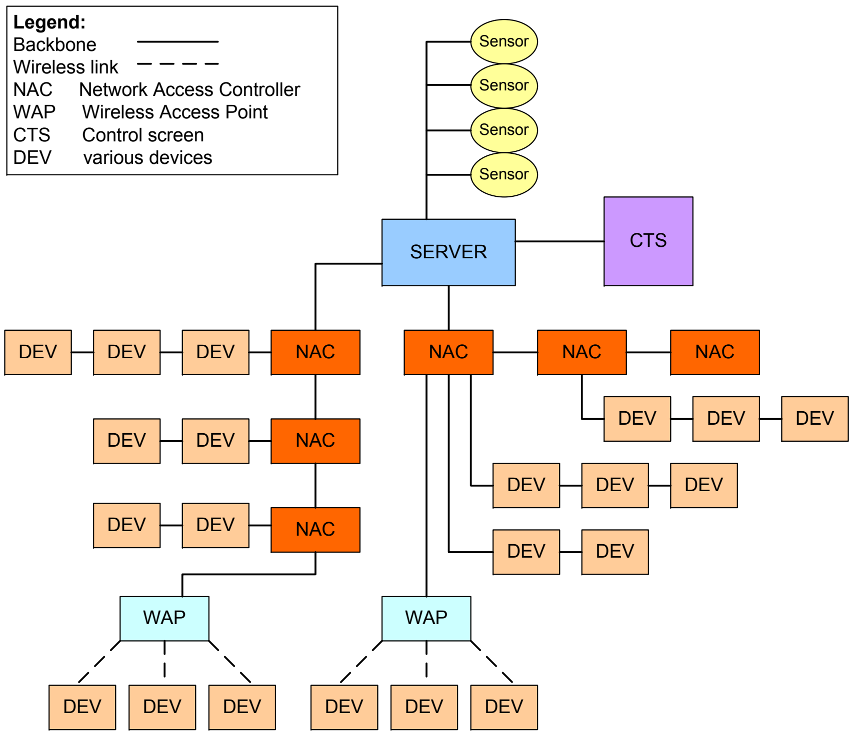
→ Yields convex/concave approximation of system output

PART 2

Analysis of EADS Case Study

Distributed Heterogeneous Communication System (HCS)

System Architecture



1 Server, up to 192 Devices

Switched Ethernet Network (100 Mb/s)

Applications

- Clock synchronization (PTP)
- Audio streaming (announcements + up to 10 music streams)
- Events (e.g. illumination)
- Signaling
- Video surveillance (up to 10 cameras)

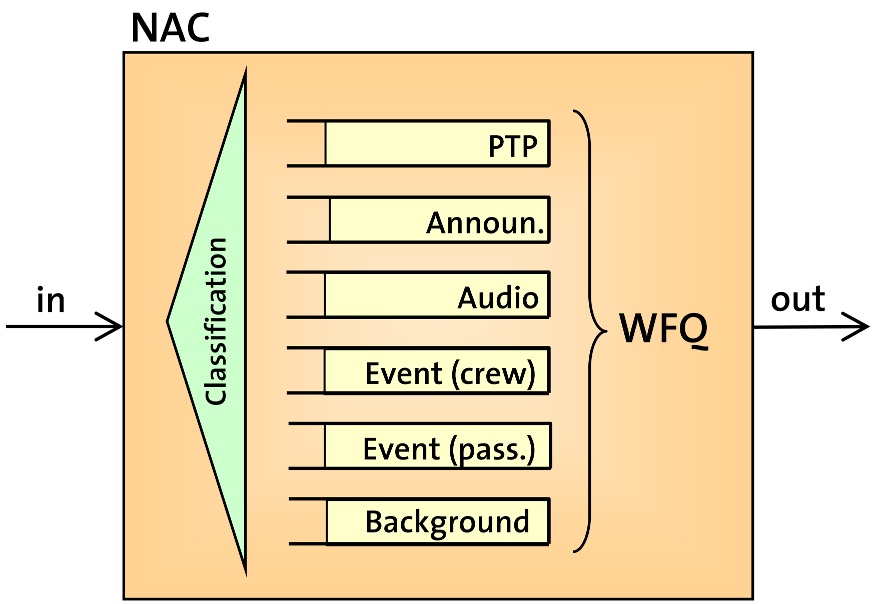
Requirements

- Synchronization precision at least 0.1 ms [R3]
- Max end-to-end delays (e.g. delay microphone-speaker < 0.1 s) [R1]
- Max jitters (e.g. < 0.1 ms for audio playback at different speakers) [R2]
- No buffer over-/underflow [R4]

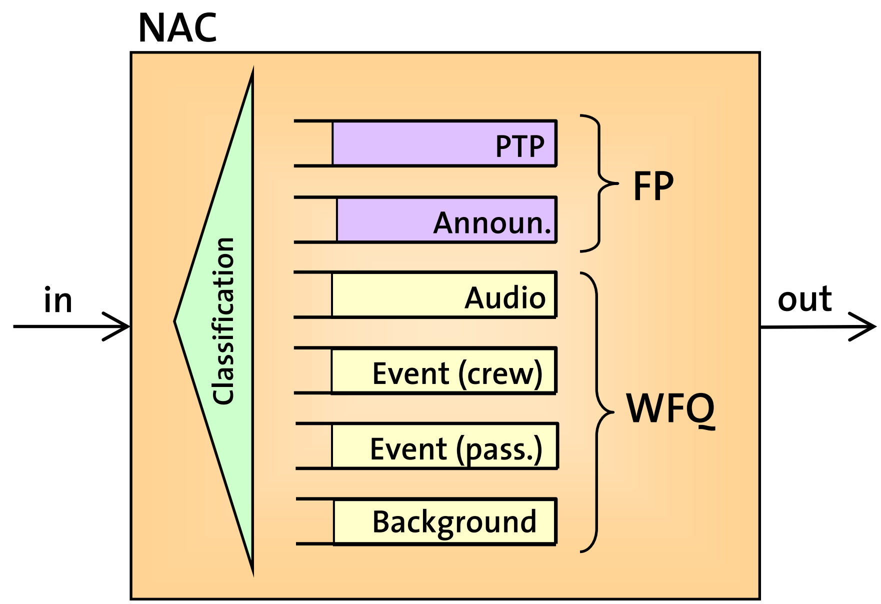
Scheduling inside NACs

Two configurations:

A. Class-based WFQ
(Weighted Fair Queuing)



B. Hybrid configuration



Assumptions for Analysis (1)

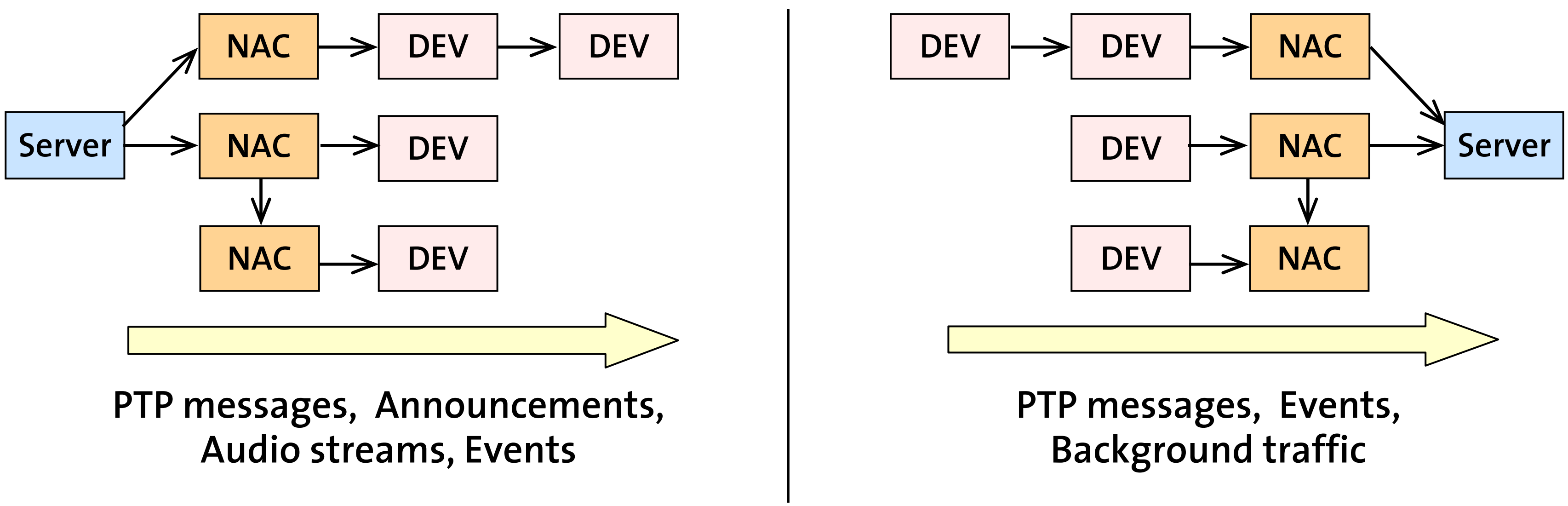
- Traffic from server to devices is sent in multicast

  (No unnecessary duplication of frames)

- Full duplex ethernet links and NACs

  (In the network the traffic SERV→DEV and DEV→SERV is completely independent and handled in different queues inside the NACs)

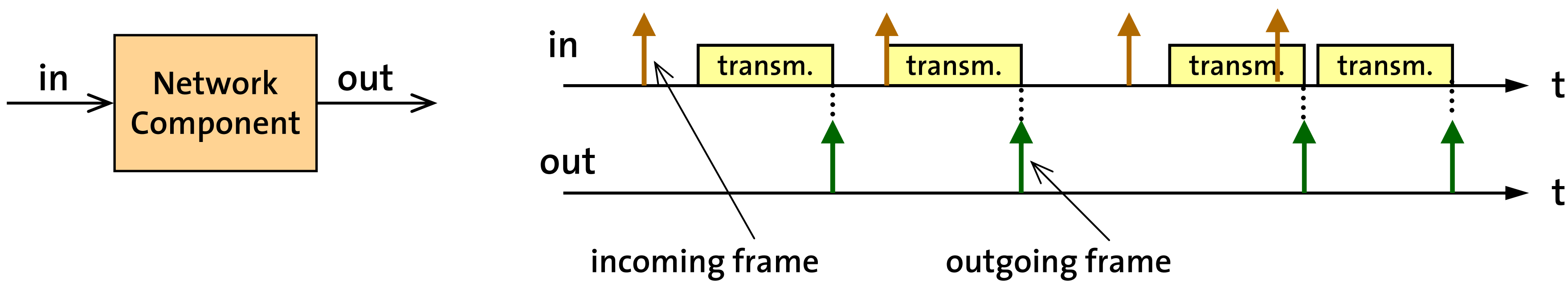
→ We can decompose the problem into two distinct instances:



Assumptions for Analysis (2)

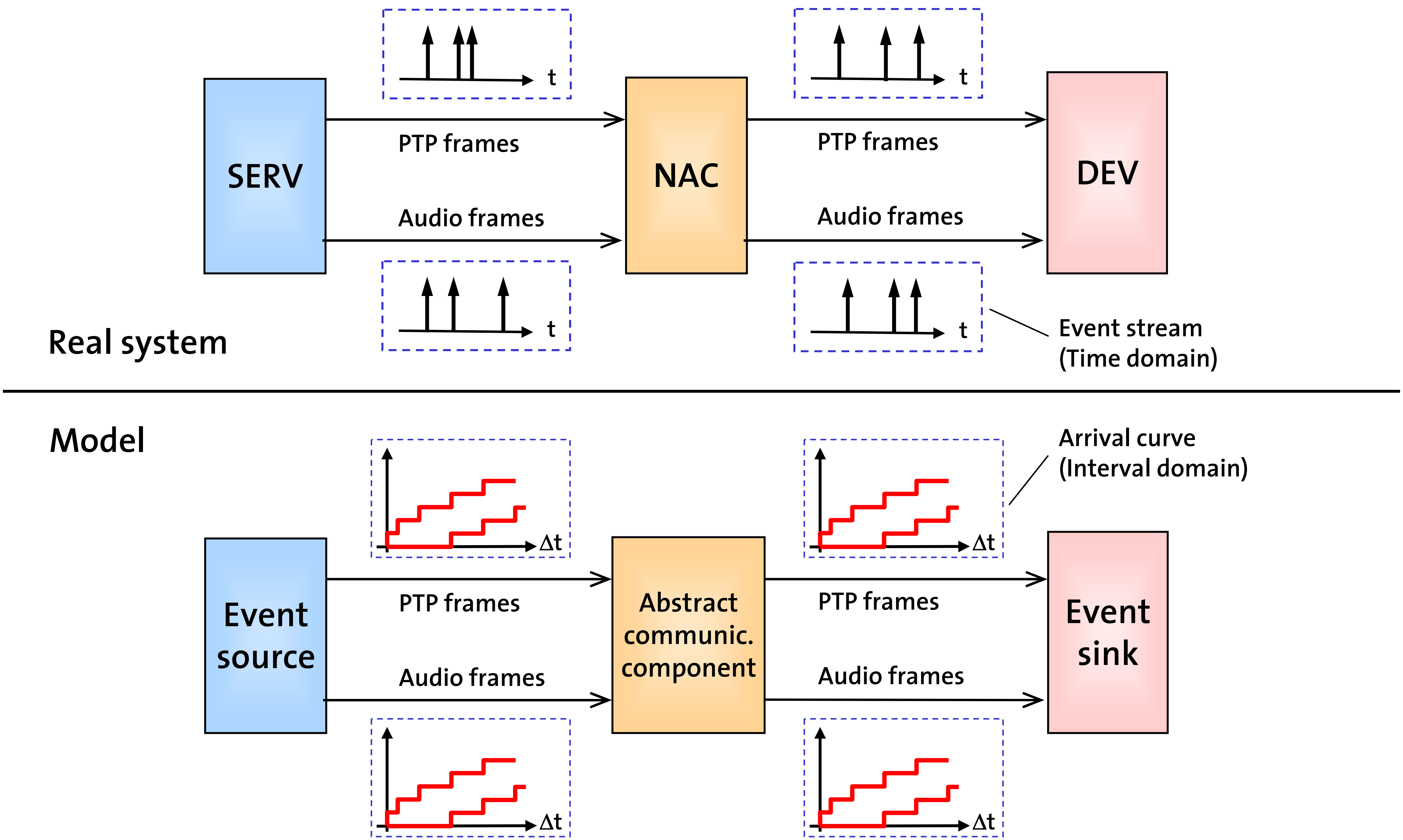
- Only <u>communication</u> among components is considered in the model
  (Execution times of processes on SERV and DEV can be neglected)

- Frame traffic in the network abstracted by <u>timed event streams</u>



The transmission of a frame is modeled as the processing of an event

Abstract Event Streams

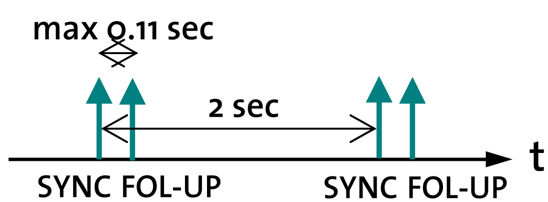


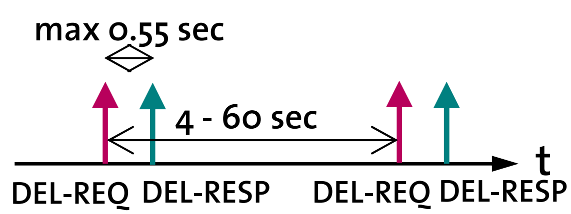
Traffic Characterization (PTP Synchronization)

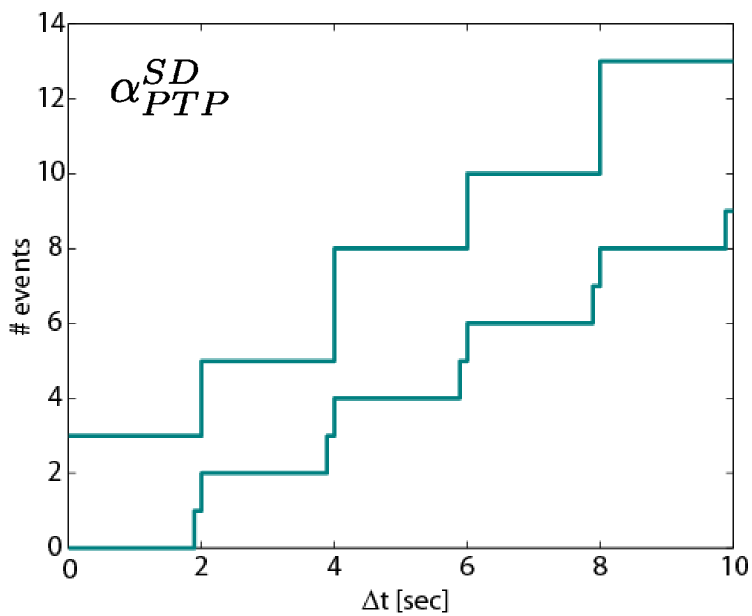
- **Direction Master → Slave**
  - 1 Sync message every 2 sec
  - 1 Follow-up message every 2 sec
  - 1 Delay-response message every 4-60 sec

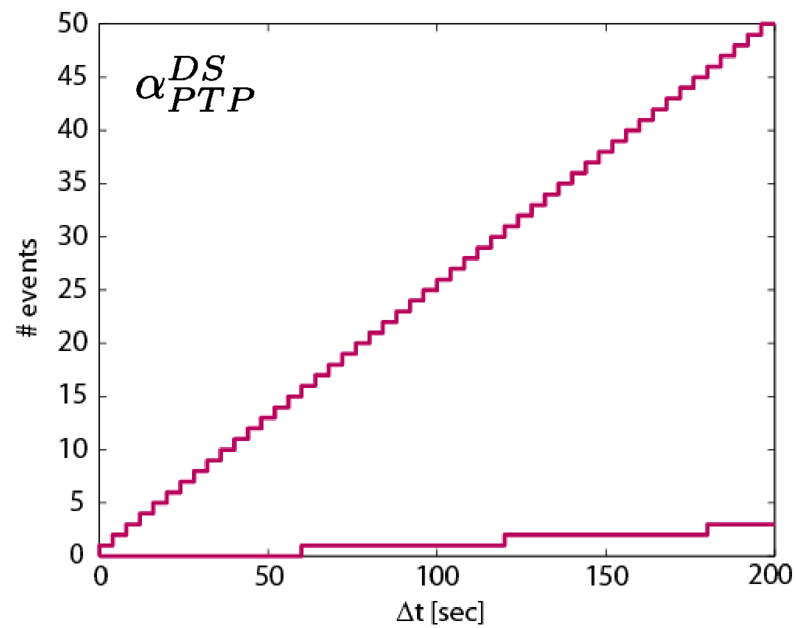
- **Direction Slave → Master**
  - 1 Delay-request message every 4-60 sec

Frame size: 172 Bytes
(for all PTP frames)
⇒ Transm. time = 13.8 μs







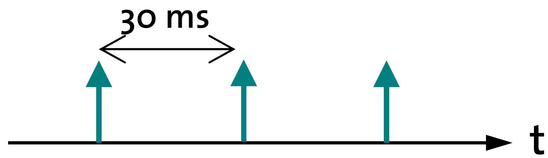


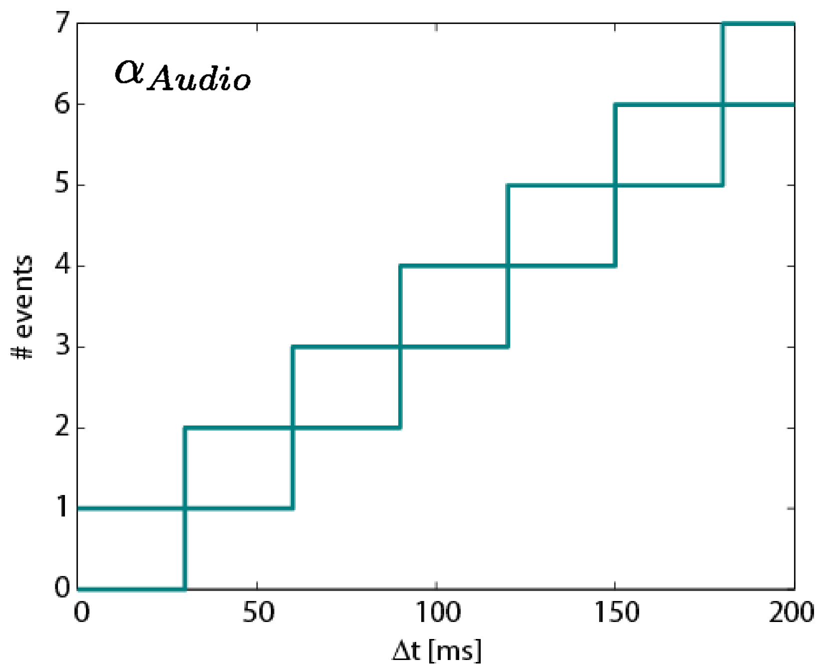
Traffic Characterization (Audio streaming)

Samples of 12 bit at frequency of 32kHz ⇒ Total data rate of 384 kbps

Frame size: 1518 Bytes
⇒ Transm. time = 121 μs

Frame rate: 33 frames/sec





Traffic Characterization (Event-based traffic)

Illumination, VOIP, …

Not modeled, spec missing

Traffic Characterization (Background traffic)

- Video surveillance

  5 Mbit/s per DEV (max. 10 Video DEVs)

  Spec of frame size missing

  Assumption: 1 Frame of 18750 Bytes every 30 ms

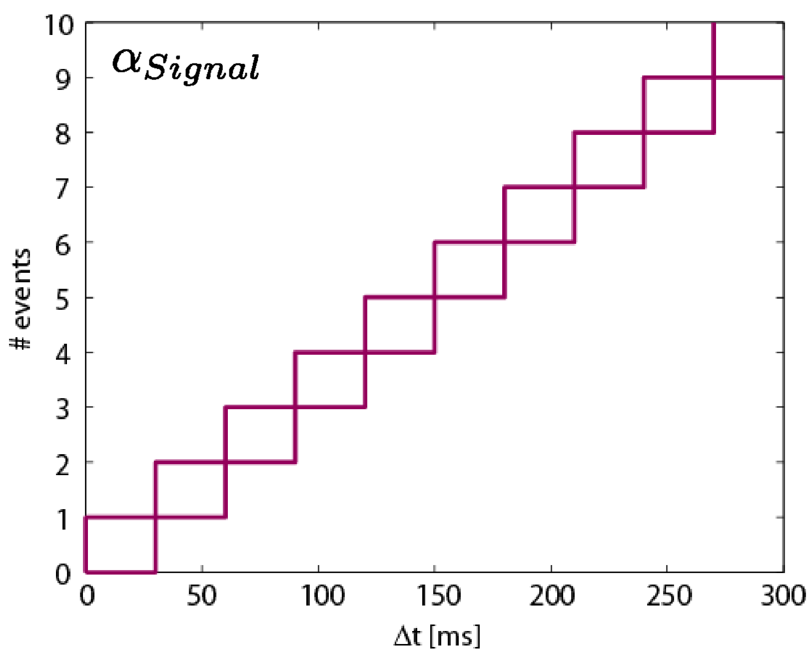
  ⇒ Transm. time = 1.5 ms

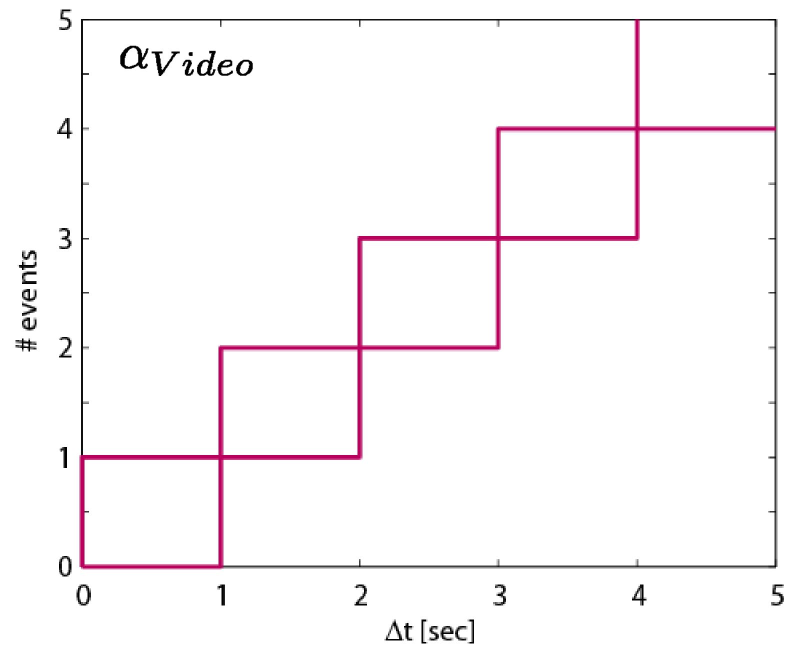
- Signaling

  1 Frame every second for each DEV

  Frame size: 102 Bytes

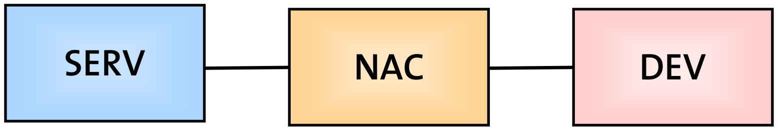
  ⇒ Transm. time = 8.2 μs





Scenario 1

Simple architecture with 1 NAC and 1 DEV



Purpose:
Understand how to model NAC components

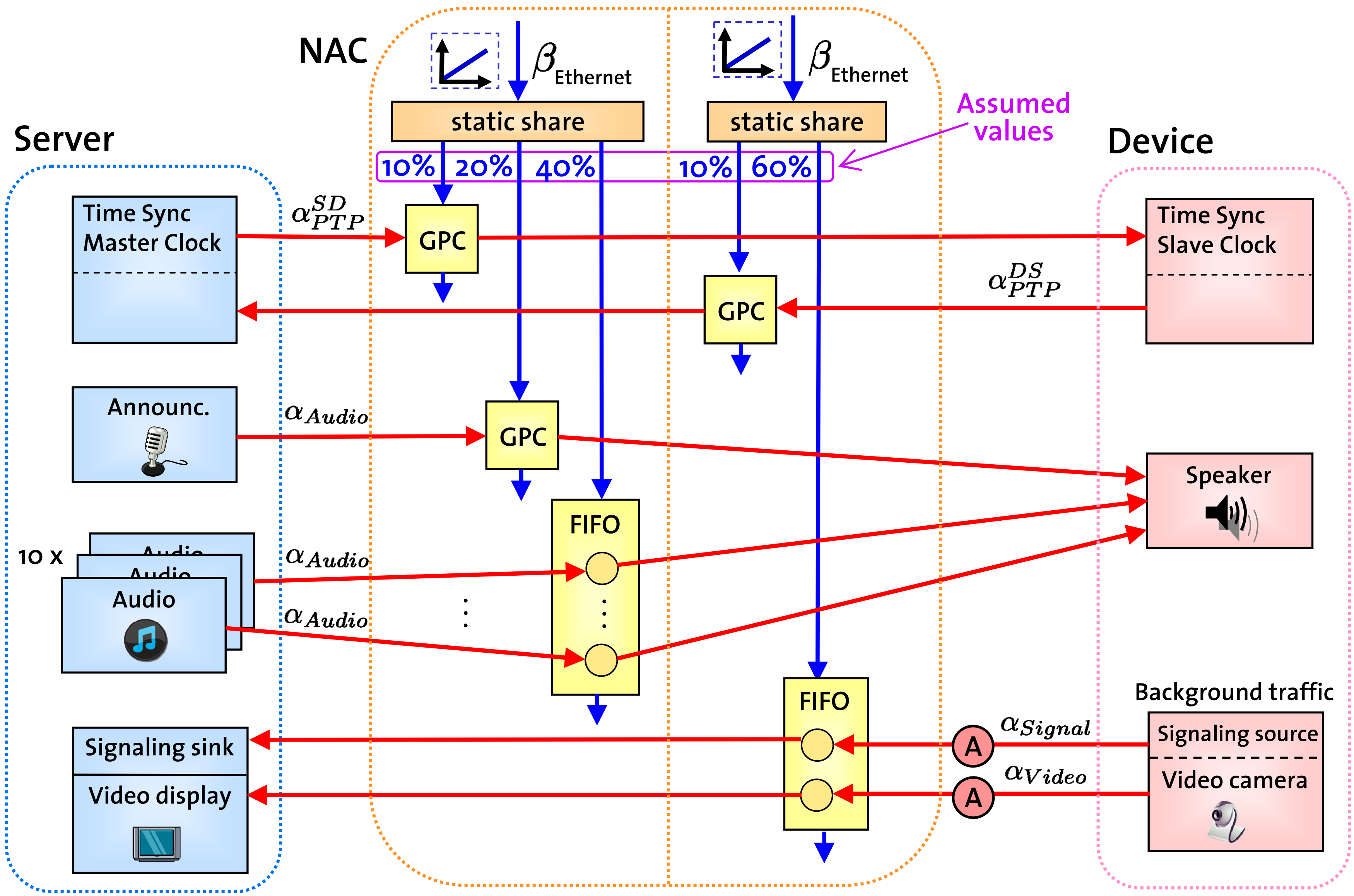
- PTP traffic SERV→DEV and DEV→SERV
- Announcements
- 10 Audio streams
- Background traffic (signaling + 1 video stream)

→ Compute worst-case end-to-end delays and buffer sizes

→ Check requirements

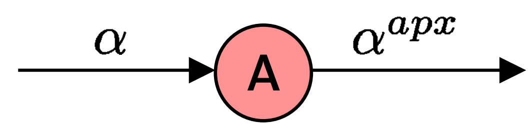
→ Compare the two different scheduling policies (WFQ, FP+WFQ)

MPA Model (WFQ Scheduling Policy)

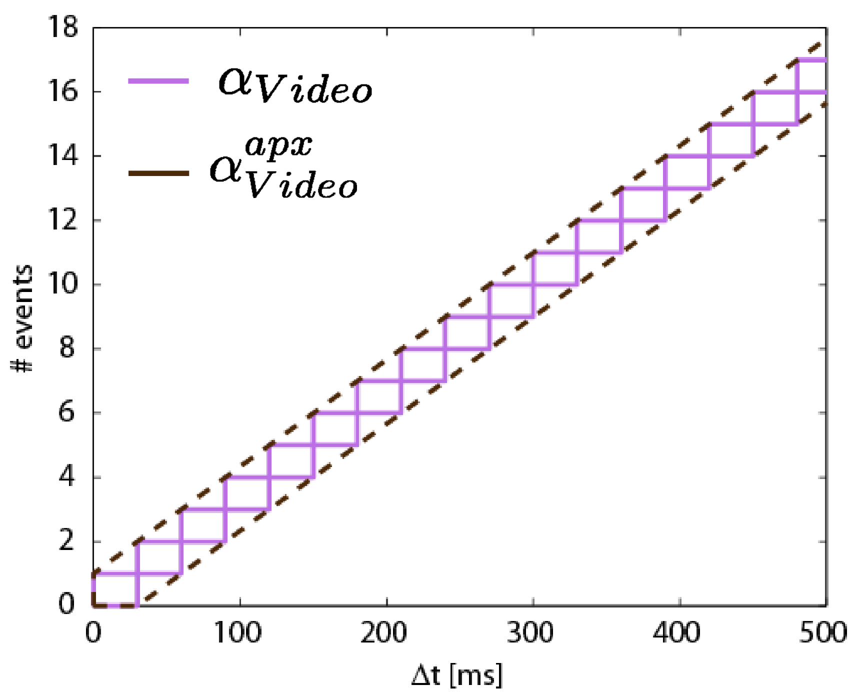


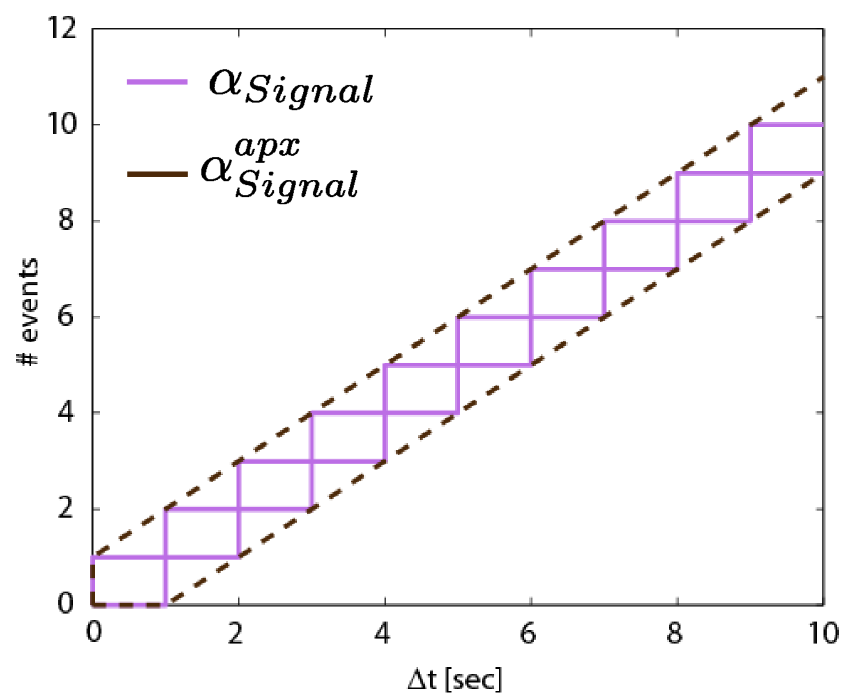
Approximation

- Approximation of arrival curves where needed in order to speed up analysis

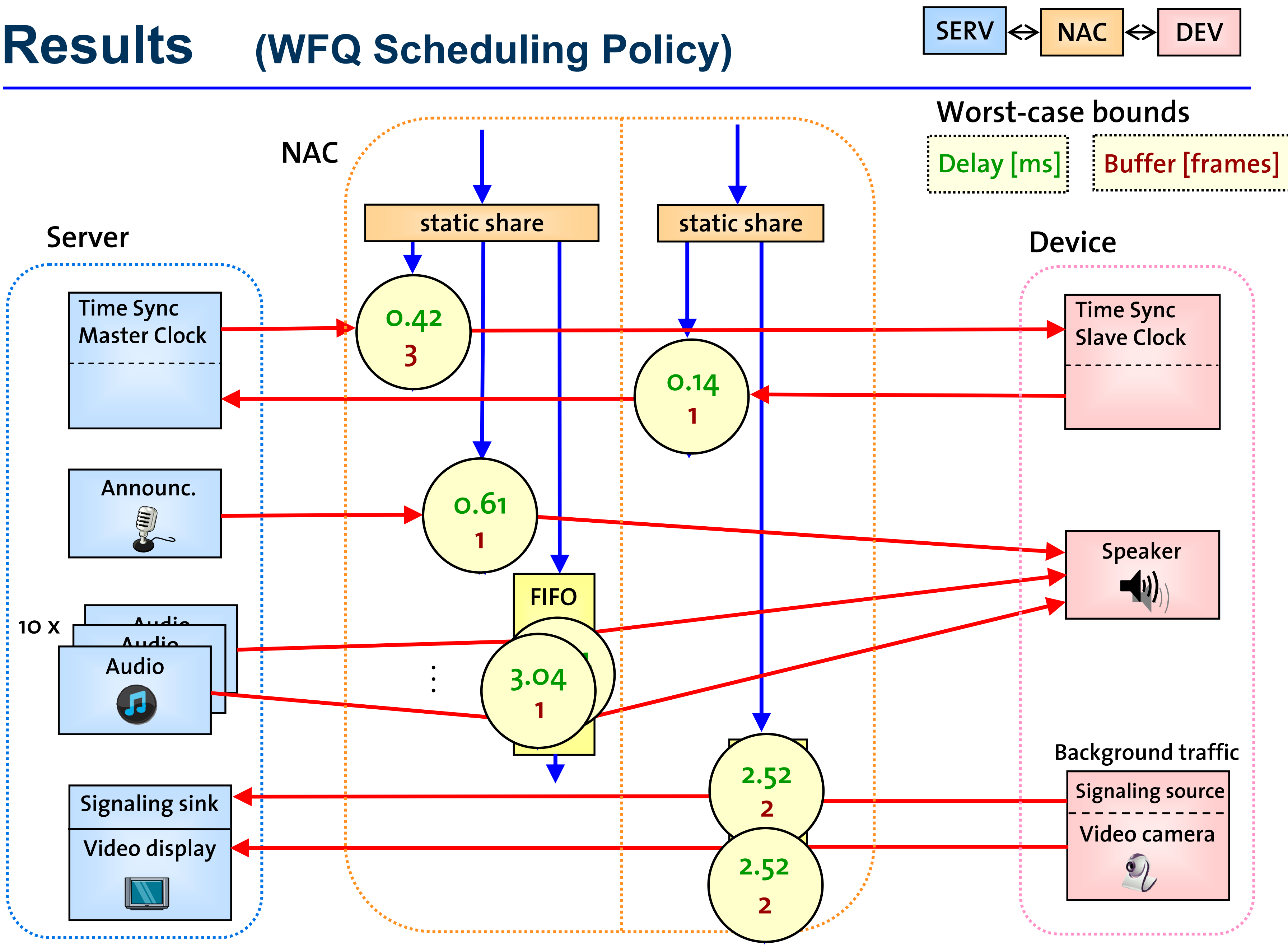


- The approximations are conservative
  $\rightarrow$ They introduce pessimism but analysis results are still safe

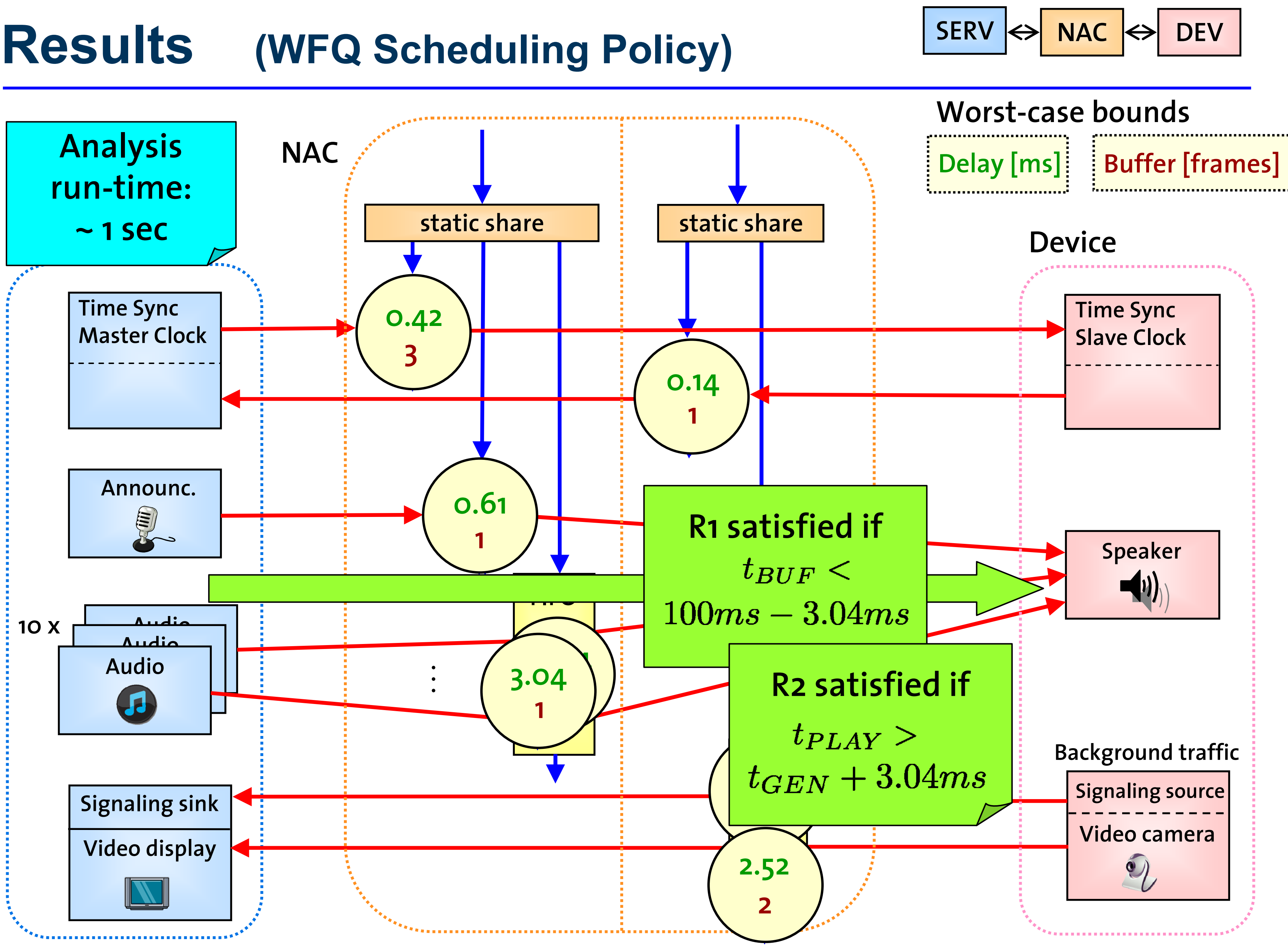




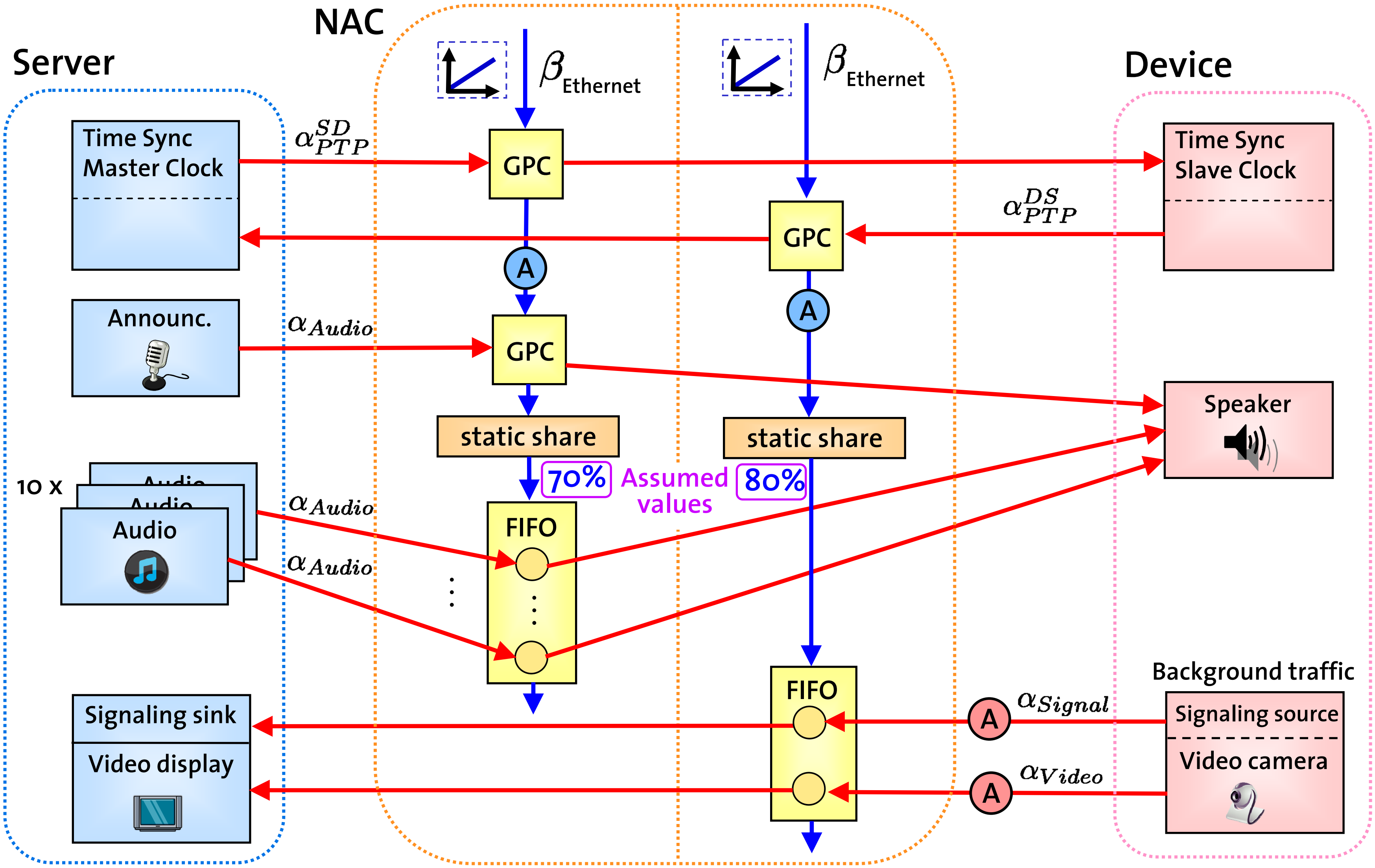
Results (WFQ Scheduling Policy)



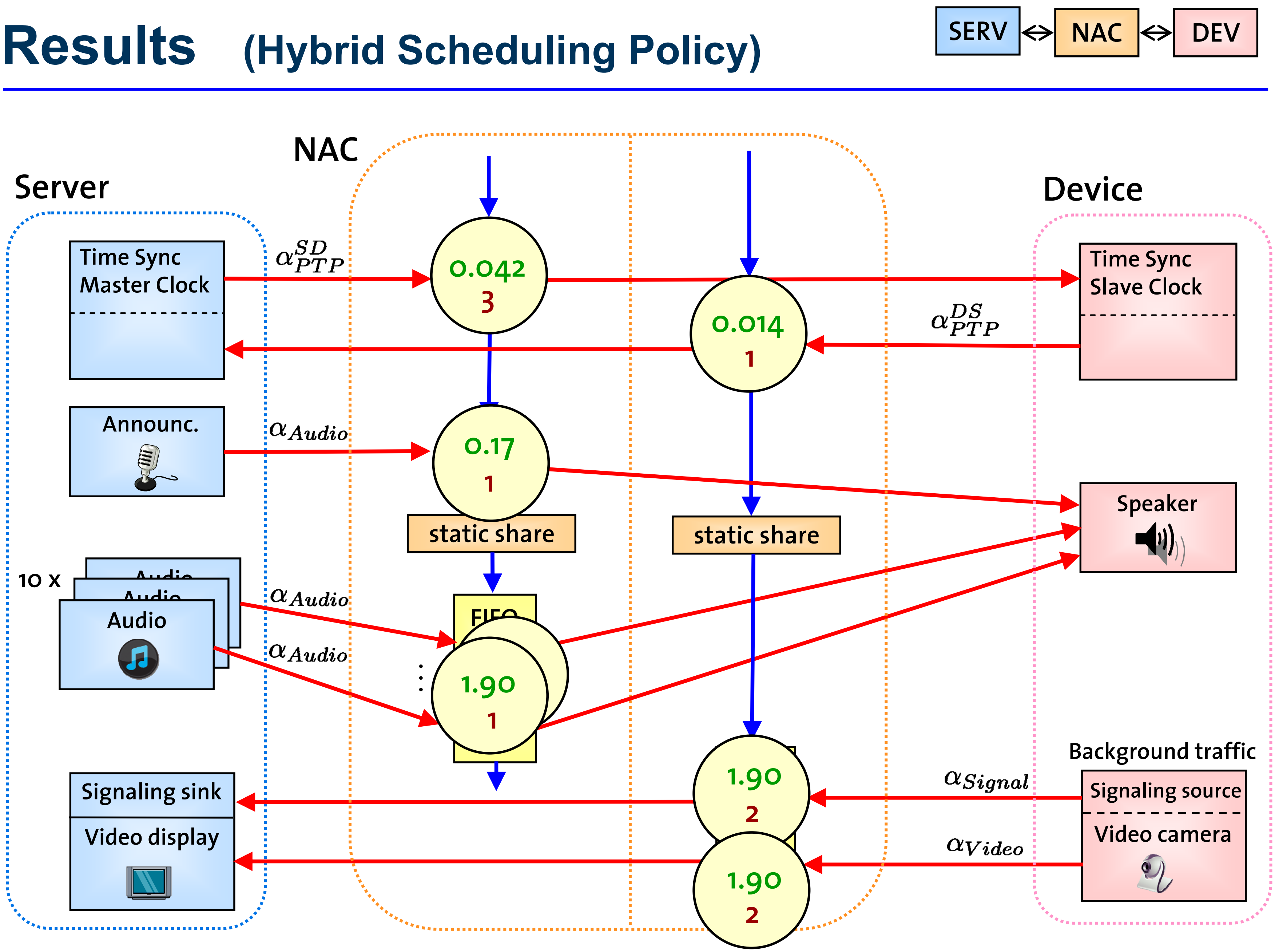
Results (WFQ Scheduling Policy)



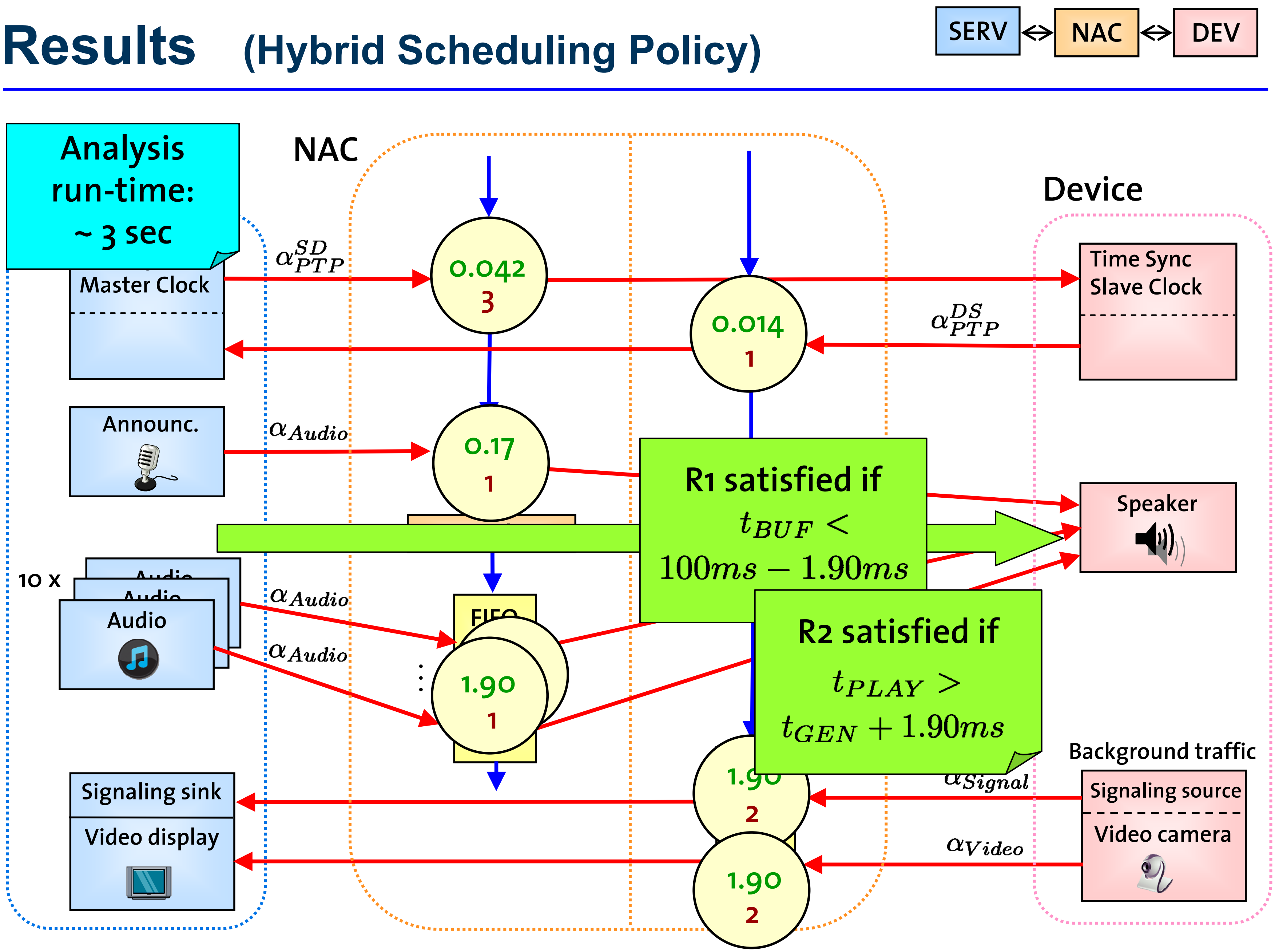
MPA Model (Hybrid Scheduling Policy)



Results (Hybrid Scheduling Policy)

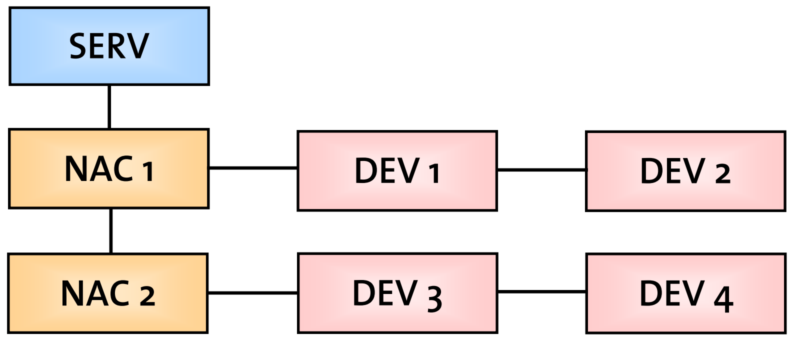


Results (Hybrid Scheduling Policy)



Scenario 2

Extended architecture with 2 NACs and 2 DEVs



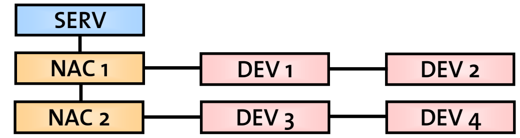
Purpose:
Understand how to model large system configurations

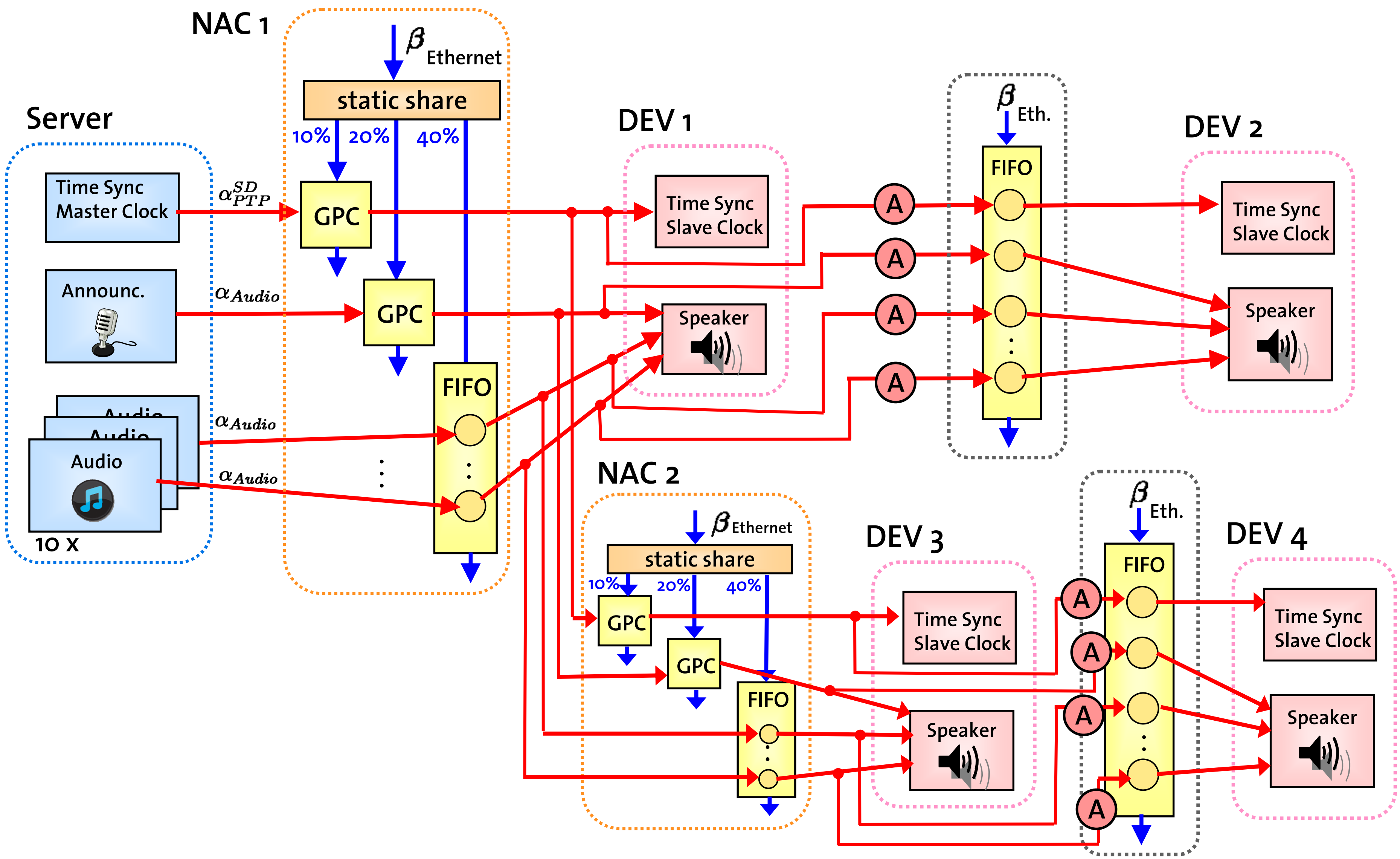
- PTP traffic SERV→DEV and DEV→SERV
- Announcements
- 10 Audio streams
- Background traffic (signaling + 1 video stream for each device)

→ Compute worst-case end-to-end delays and buffer sizes

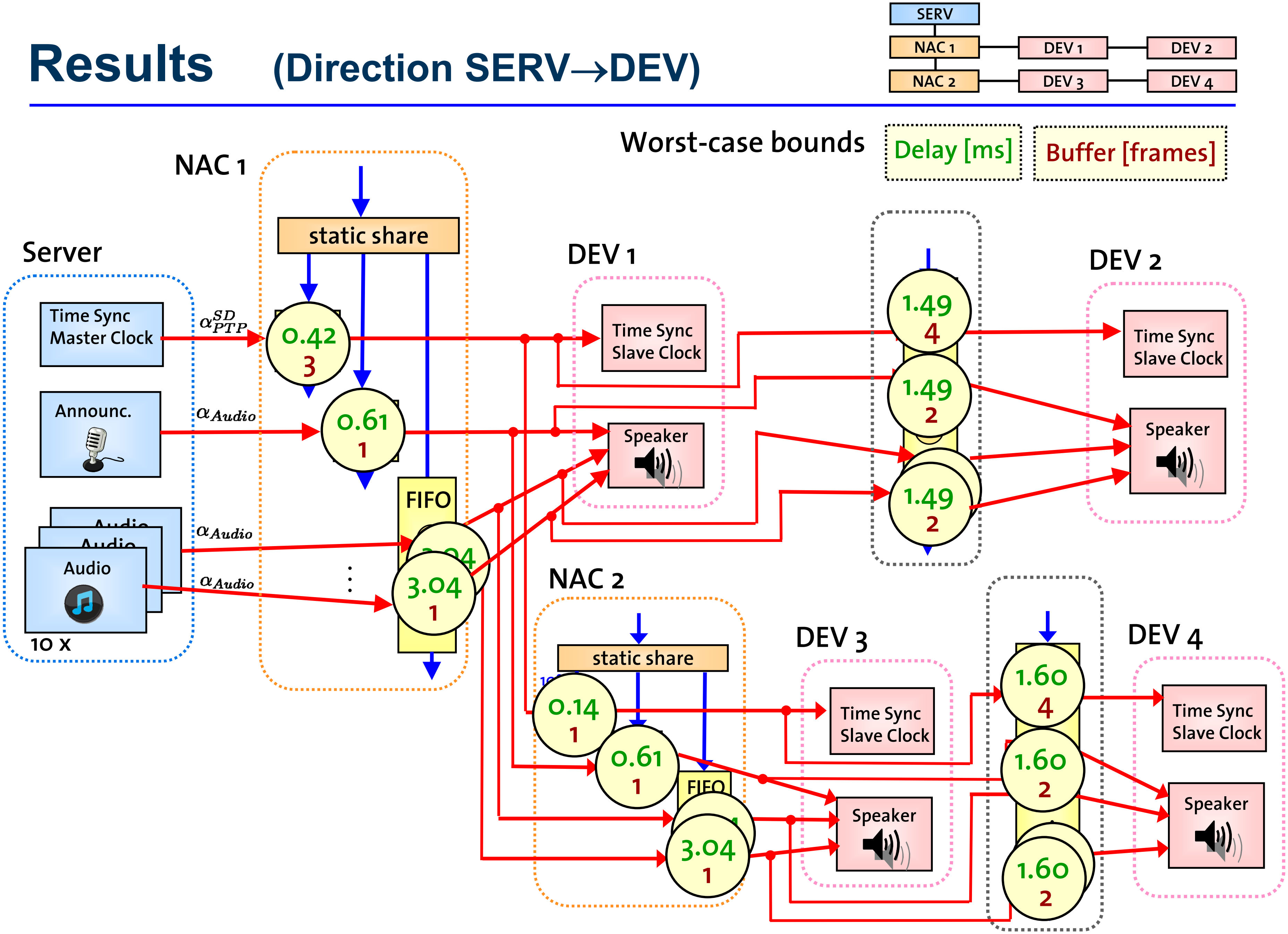
→ Check requirements

MPA Model (Direction SERV→DEV)

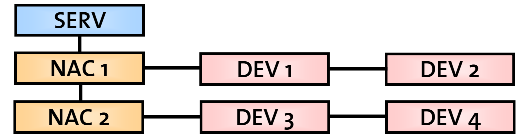


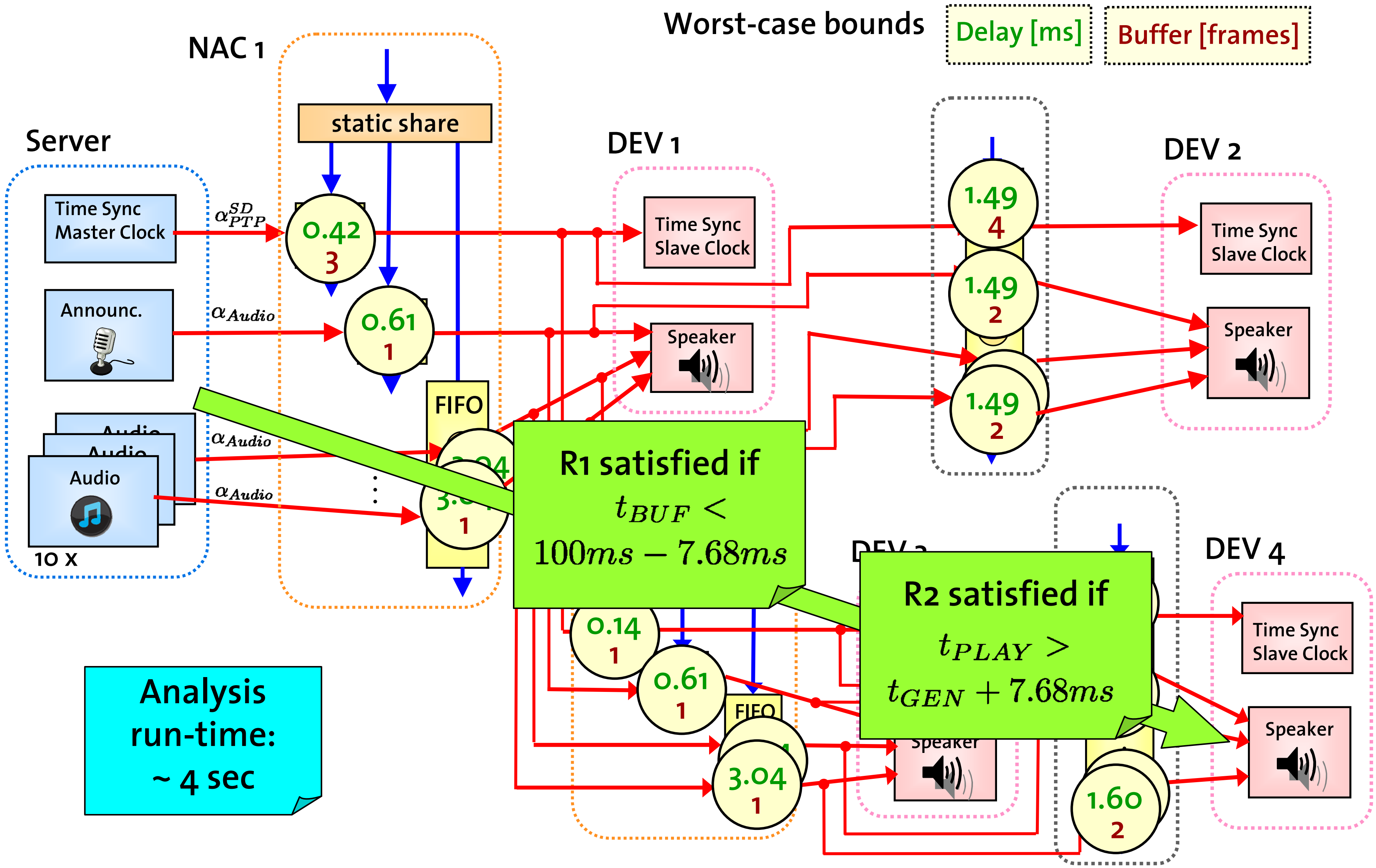


Results (Direction SERV→DEV)

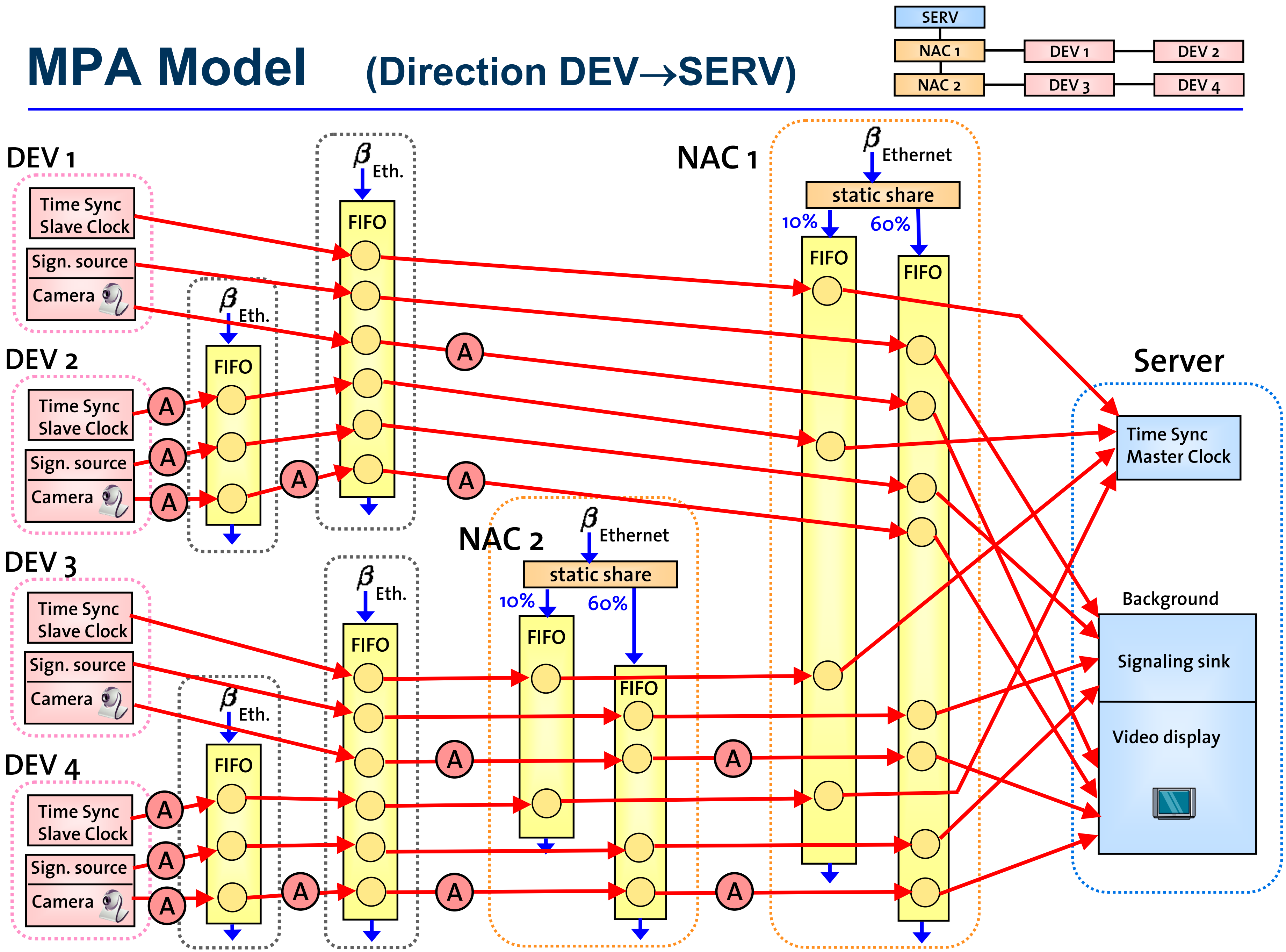


Results (Direction SERV→DEV)

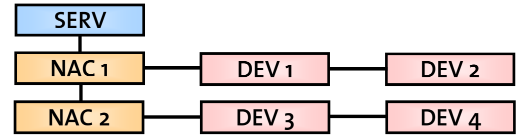


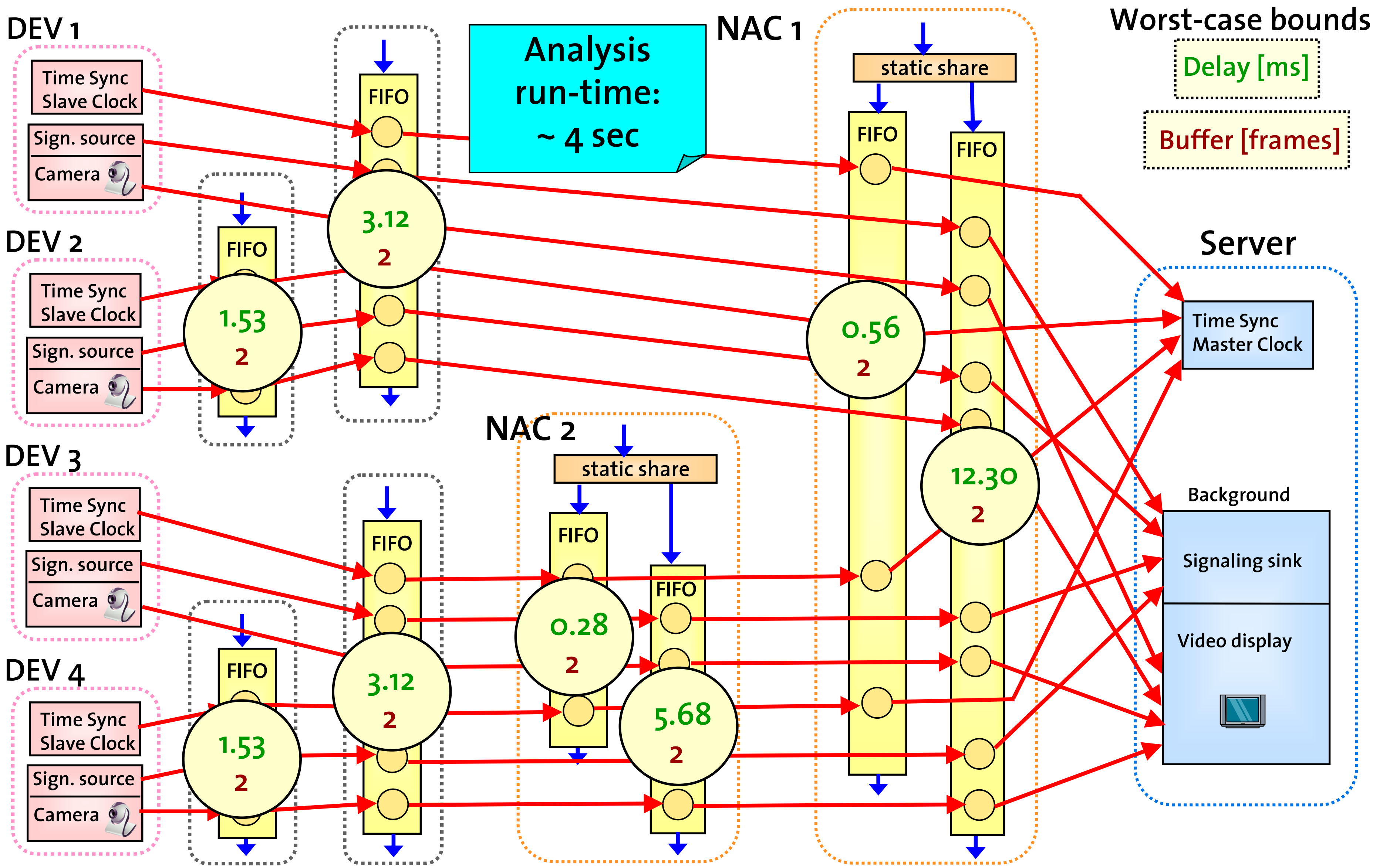


MPA Model (Direction DEV→SERV)



Results (Direction DEV→SERV)





Remarks

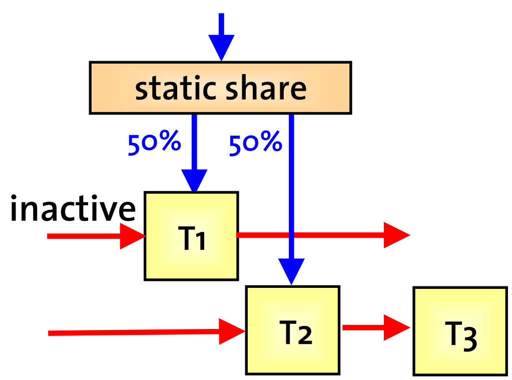
1. General Processor Sharing (GPS) and Weighted Fair Queuing (WFQ) are not exactly the same

   - GPS is an ideal scheduling algorithm that cannot be implemented in practice (requires fluid traffic, i.e. infinitesimal packet sizes)
   - WFQ is a packet approximation of GPS
   - It has been proven that the delay bounds provided by WFQ and GPS differ at most by the transmission time of one packet

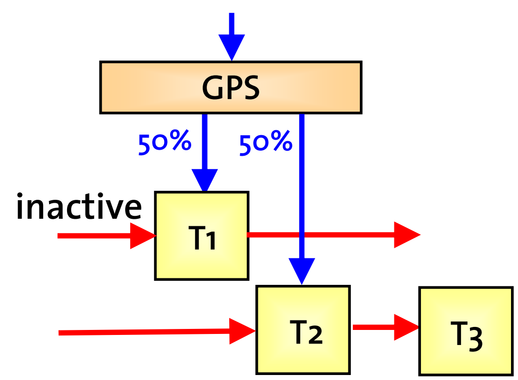
$\Rightarrow$ The delays determined assuming GPS need to be adapted accordingly

Remarks

2. The static bandwidth sharing employed in the MPA model is different from GPS scheduling!
   - Under congestion both policies concede the same guaranteed bandwidths to the different streams
   - However, in the presence of inactive streams the behavior of the two policies is different: Under GPS scheduling the non-utilized bandwidth is available to the remaining streams whereas it is lost under static bandwidth sharing.



Max bandwidth for T2: 50%

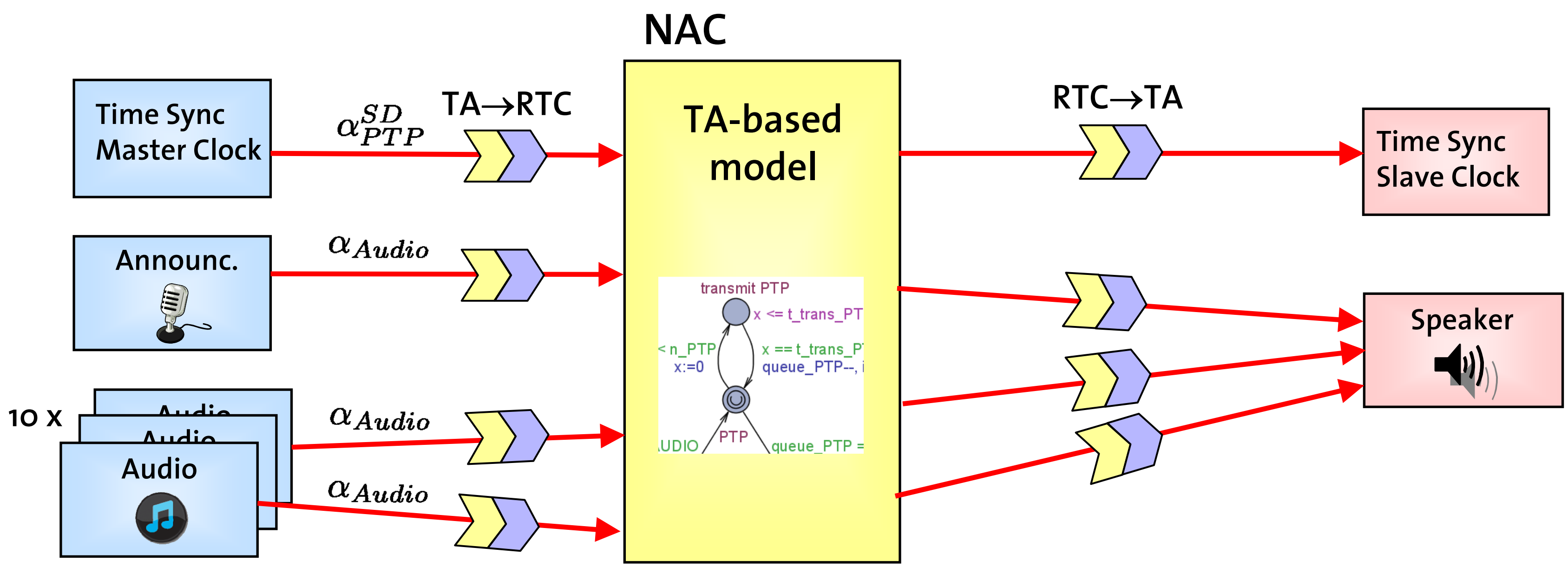


Max bandwidth for T2: 100%
→ Higher load for T3

⇒ The bounds derived assuming static bandwidth sharing are in general not conservative for a GPS system!

Proposed Solution (Work in progess)

- Construct a state-based (timed automata) component model for the NAC that better reflects the behavior of a WFQ scheduler
- Use the introduced hybrid analysis approach (RTC+TA) to interface RTC and TA components and derive conservative performance bounds



Weighted Round Robin

- Precise modeling of the WFQ algorithm with TA is cumbersome

  (For each packet in each queue a predicted finishing time of transmission needs to be stored and updated at any new packet arrival)

- Start with a simpler approximation of GPS: Weighted Round Robin (WRR)

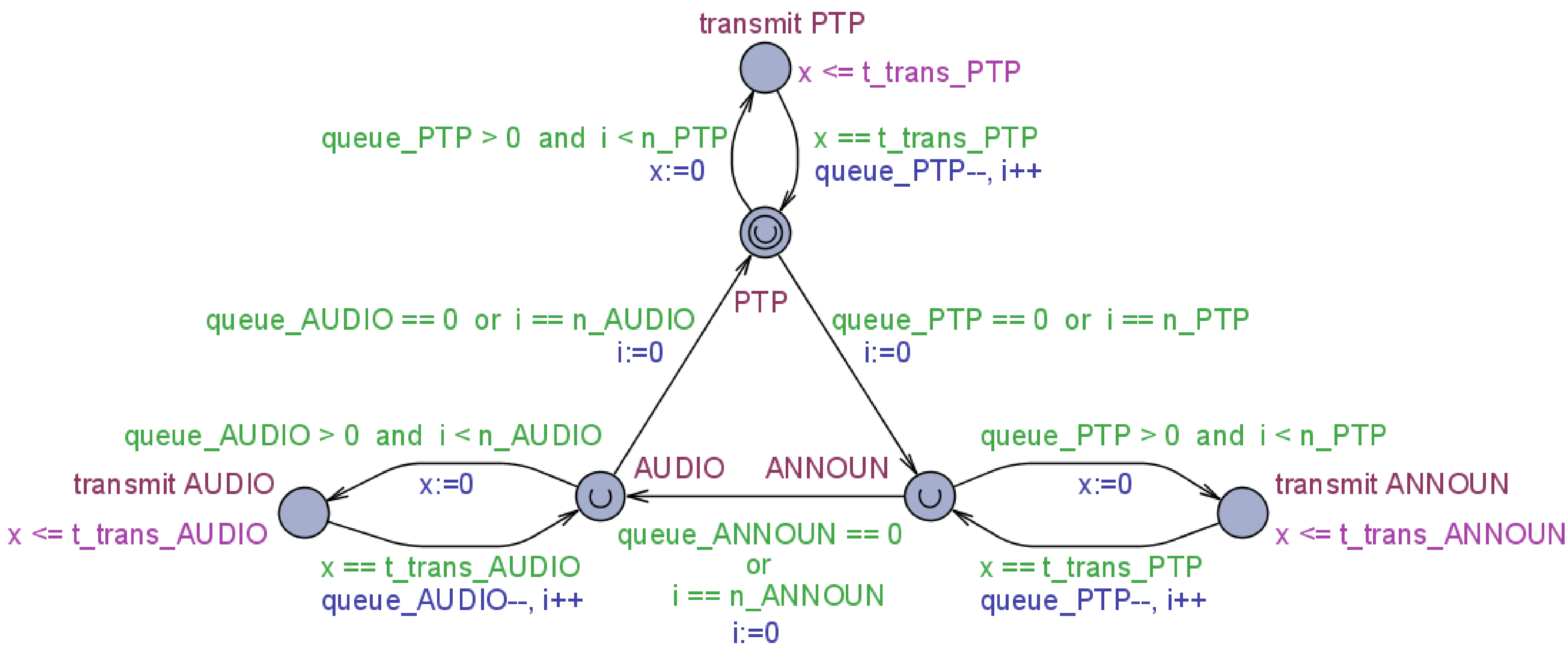
- WRR cycles over all queues and serves a number n_i of packets from each nonempty queue, where

$$n_i = \text{normalized}(w_i / s_i)$$

$w_i =$ weight assigned to stream i

$s_i =$ mean packet size of stream i

TA model of NAC implementing WRR



Conclusions

- Modeling and Analysis of EADS case study in MPA-RTC
- Ready for the MPA analysis of larger system architectures
- Work in progress: Better model for WFQ scheduling in NACs
- Next steps:
  - Apply hybrid analysis approach (RTC+TA) to case study in order to get better results
  - MPA analysis: Evaluate how well the quality of the results scales with the system size.

    (For larger systems we need more approximations in order to keep analysis time short → We have to expect less accurate results)

Questions

1. The PTP protocol assumes uniform delay for the transmissions SERV→DEV and DEV →SERV. However, if we transmit SYNC and FOLLOW-UP messages in multicast and DELAY-REQ and DELAY-RESP in unicast the delays can be considerably different.

2. Input load not fully specified: Upper bound for PTP messages? Size of video frames? Event-based traffic?

3. Topology: How many DEVs at most per NAC? 4 in each daisy chain or 4 in total?

4. More information about the synchronized playback of audio frames at different speakers is needed. Where and how is the playback timing decided?