

Integrating Dynamic-TDMA Communication Channels into COTS Ethernet Networks

Overview and Case Studies

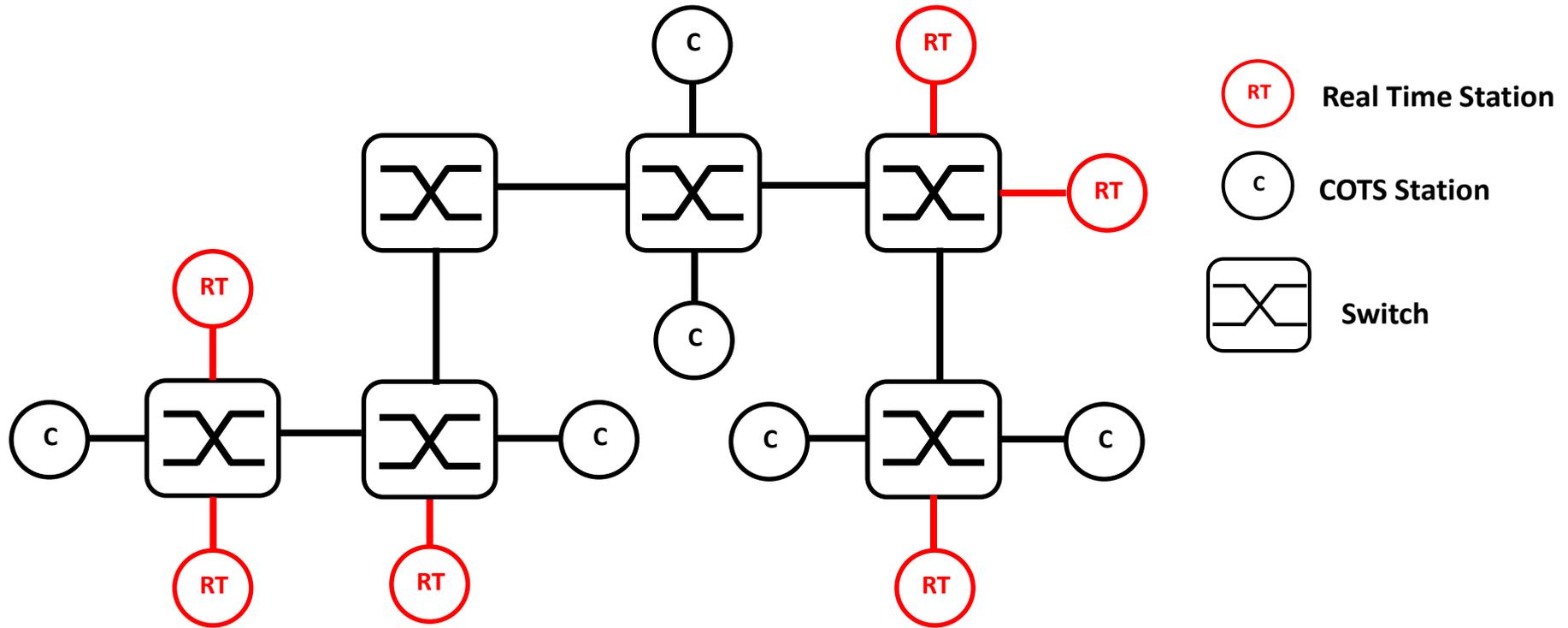
G. Carvajal^{1,2}, L. Araneda², A. Wolf²,
M. Figueroa², and S. Fischmeister¹

¹ University of Waterloo, ON, Canada

² Universidad de Concepcion, Chile

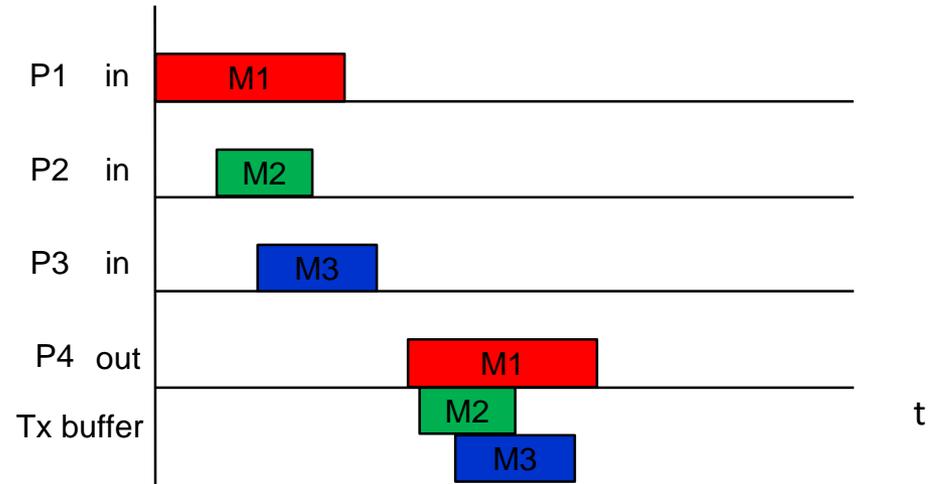
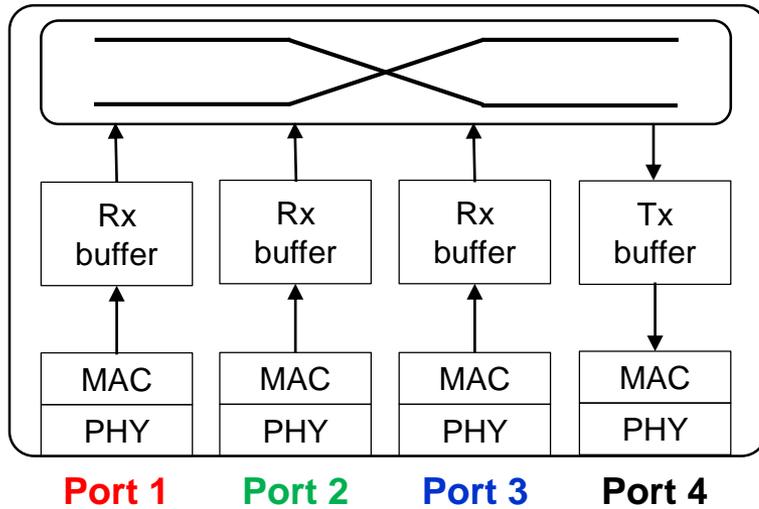
March 2016

Real-Time Ethernet



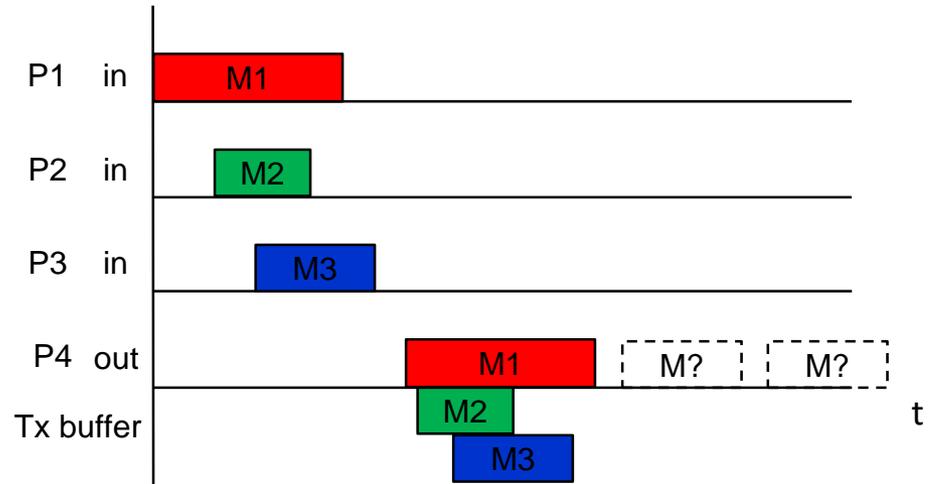
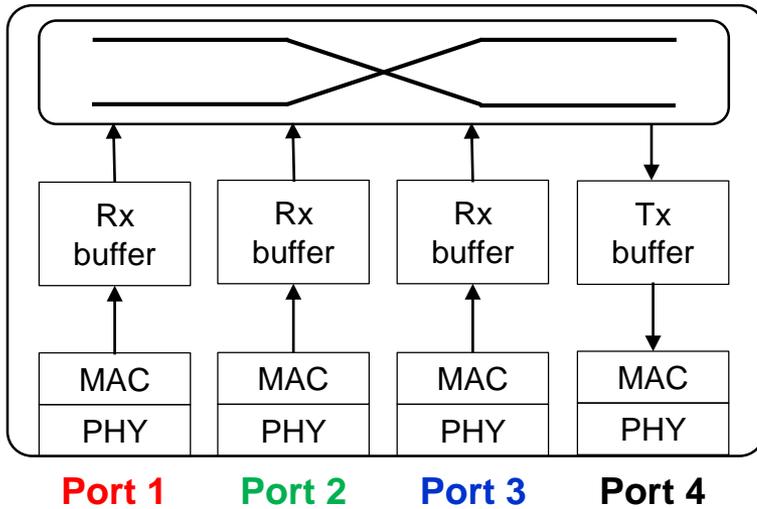
- Unified backbone for heterogeneous distributed systems
 - Real-time traffic: regular patterns, strict timing guarantees
 - Best-effort (COTS) traffic: bursty, high-bandwidth, no timing guarantees

Latency in Standard Ethernet



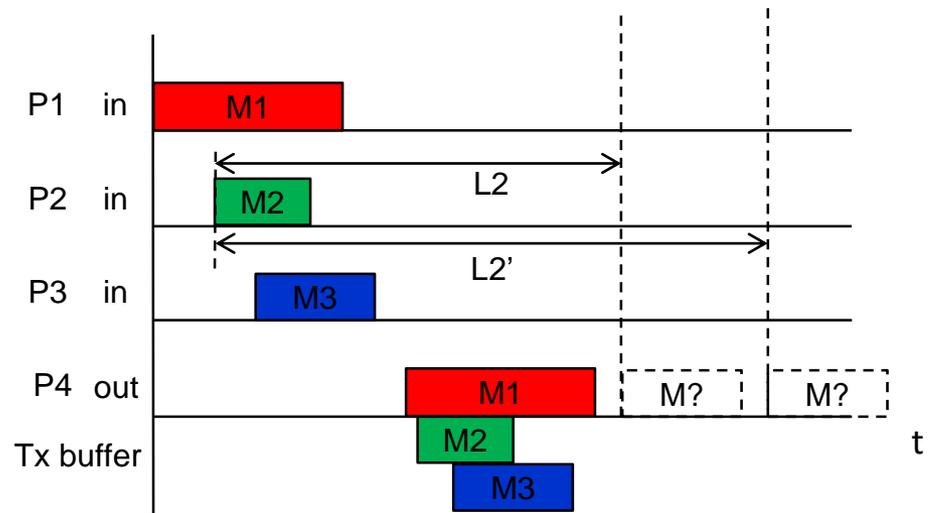
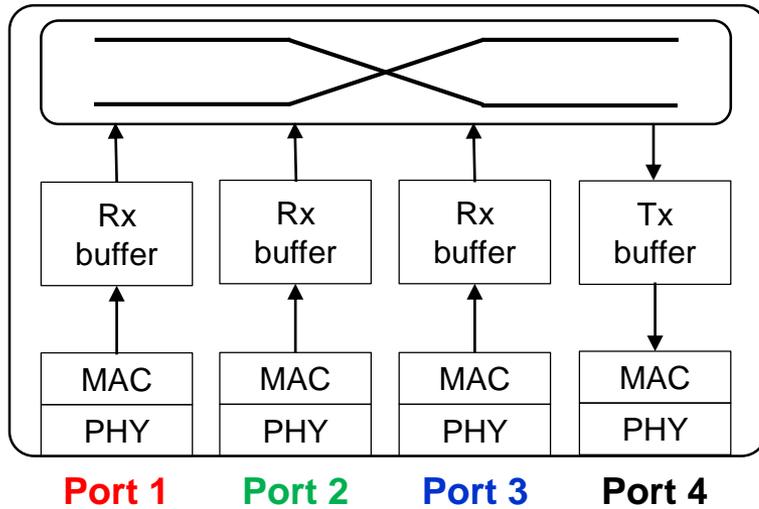
- Stations can transmit data of any size at any time

Latency in Standard Ethernet



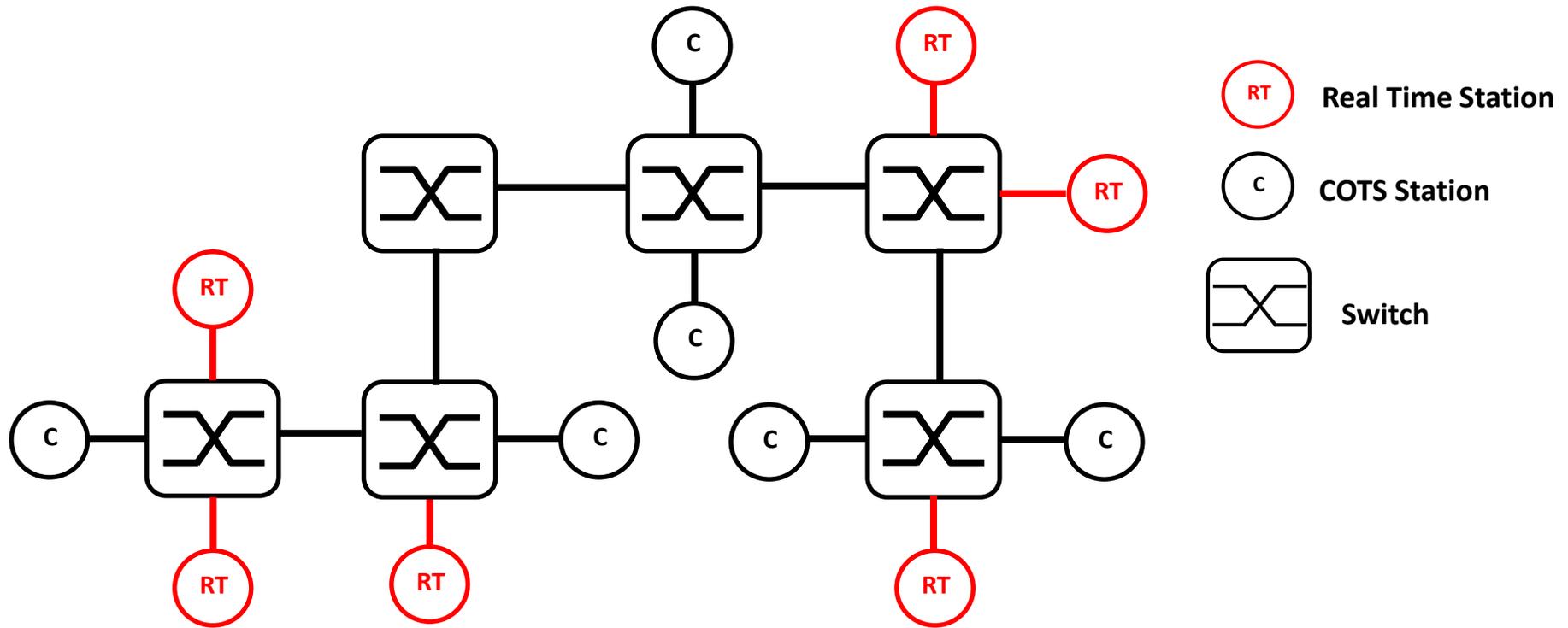
- Stations can transmit data of any size at any time
- What is the latency of M2?

Latency in Standard Ethernet



- Stations can transmit data of any size at any time
- What is the latency of M2?
- Arbitration based on statistical multiplexing (max. throughput)
 - Latency depends on implementation and instantaneous network load
 - Out of order delivery
 - High jitter
 - Latency is a **probabilistic** parameter

Real-Time Ethernet: Main Driving Forces



- Guaranteed communication latency between real-time stations
 - Require new paradigms and hardware support
- Compatibility with standard devices and protocols (best-effort)
- Minimize the intervention of legacy COTS stacks
 - Leverage economy of scale and compatibility with legacy settings

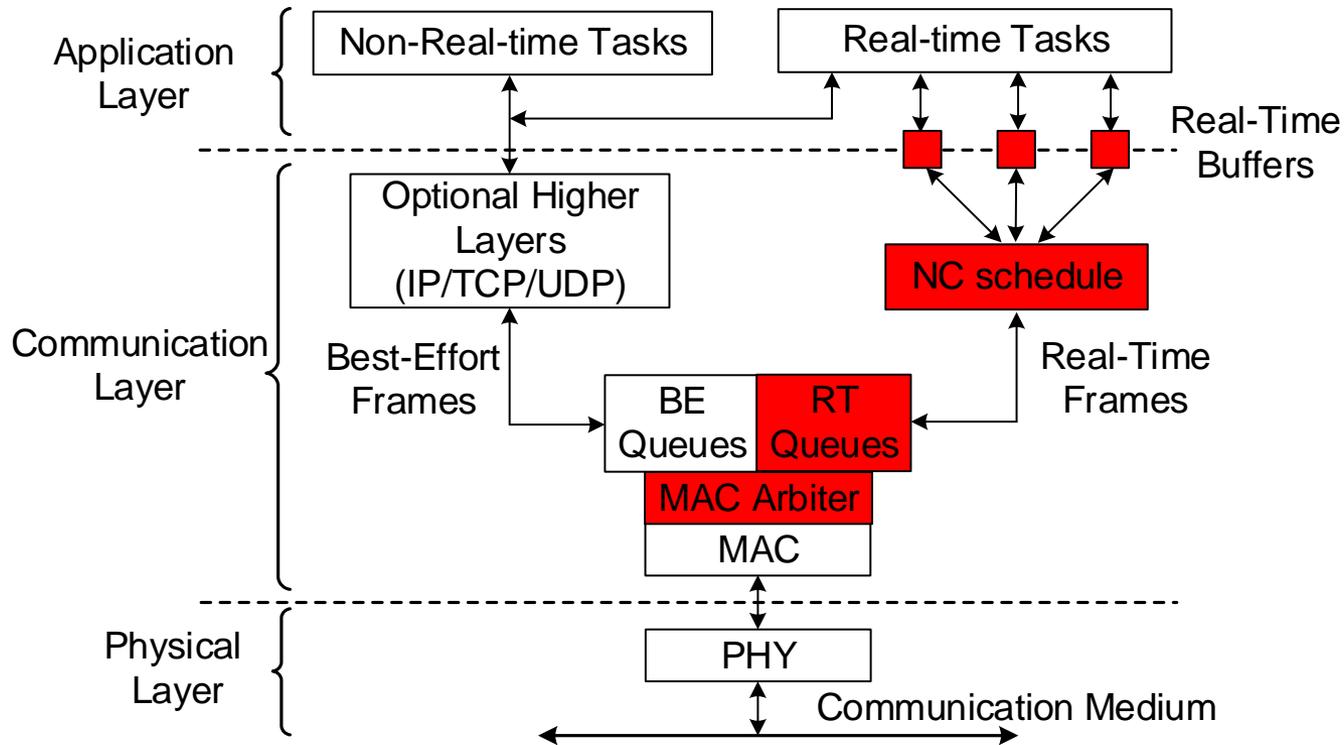
Real-Time Ethernet: Open Problems

- Ultra-low jitter
- Precise synchronization
- Efficient response-time analysis
- Efficient use of bandwidth with heterogeneous traffic profiles
- Segmentation of real-time domains in large networks
- Multiple-operational modes and online bandwidth management

Atacama Framework

- Custom extensions tailored for time-triggered communication
- Based on **dynamic-TDMA** arbitration
 - Off-line schedule design considering worst-case scenarios
 - Conditional allocation of time slots at runtime
 - Efficient bandwidth utilization and **multi-mode** systems
- Hand-optimized hardware modules for NICs and switches
 - Programmable ASIP for deterministic execution of schedules
 - Deterministic end-to-end communication latency for time-sensitive data
 - Logical segmentation of real-time domains in large networks
 - Seamless integration in COTS devices
- Fully open-source

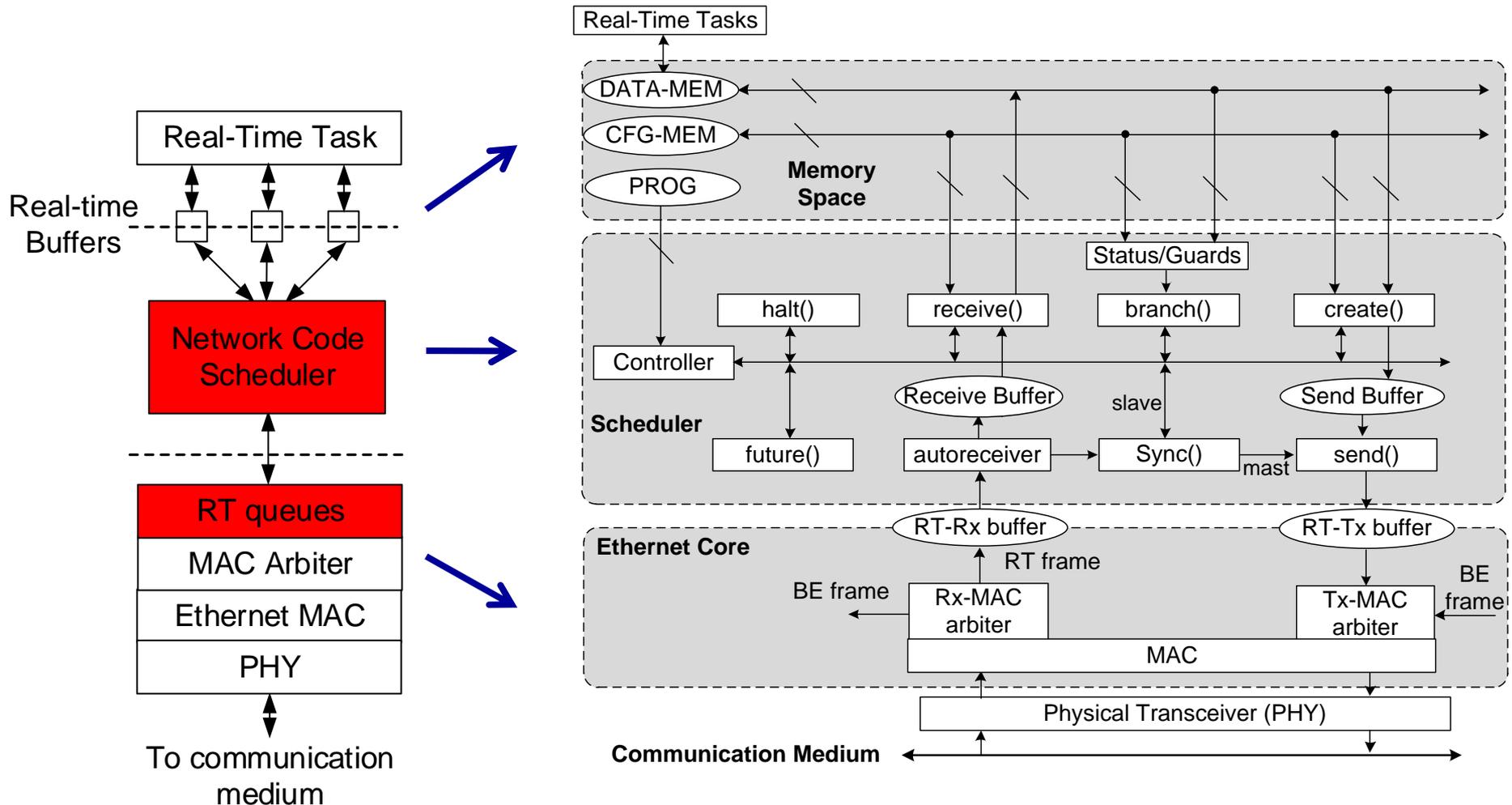
Real-Time Capable NICs



- Real-time extensions (red blocks) provide a parallel stack
- Pre-programmed schedules specify precise points in time to move data between real-time buffers and the medium

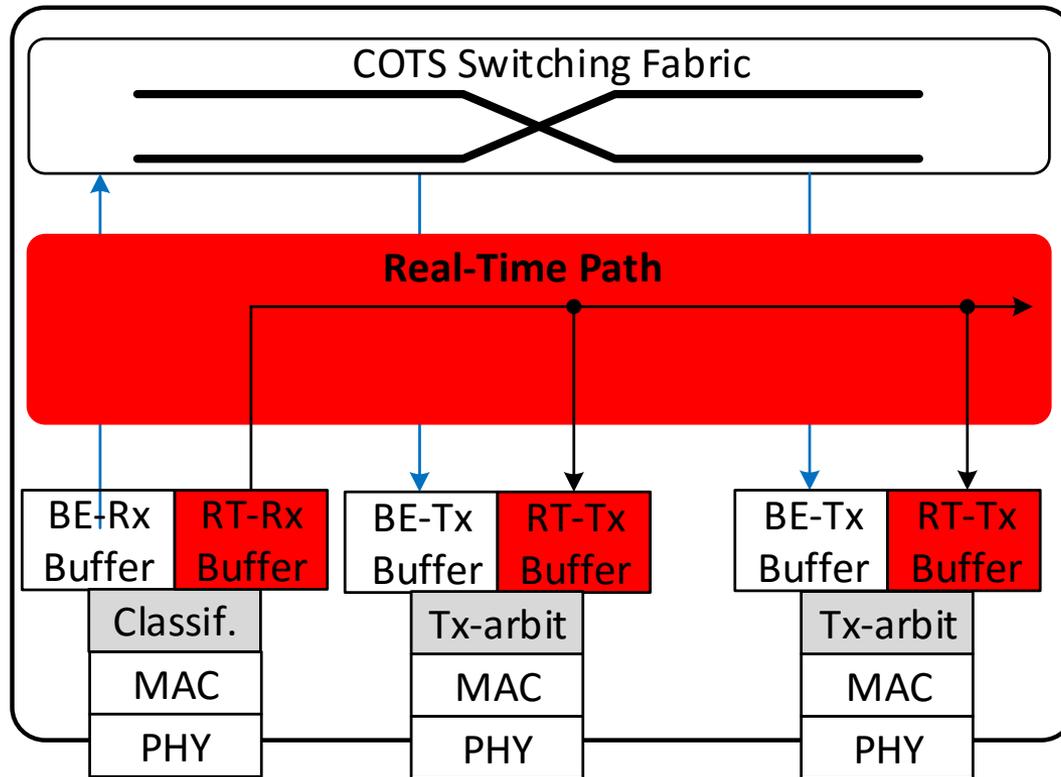
Real-Time Capable NICs

- **Full hardware** implementation of Network Code instructions
- **Fast and deterministic** transactions between buffers and medium

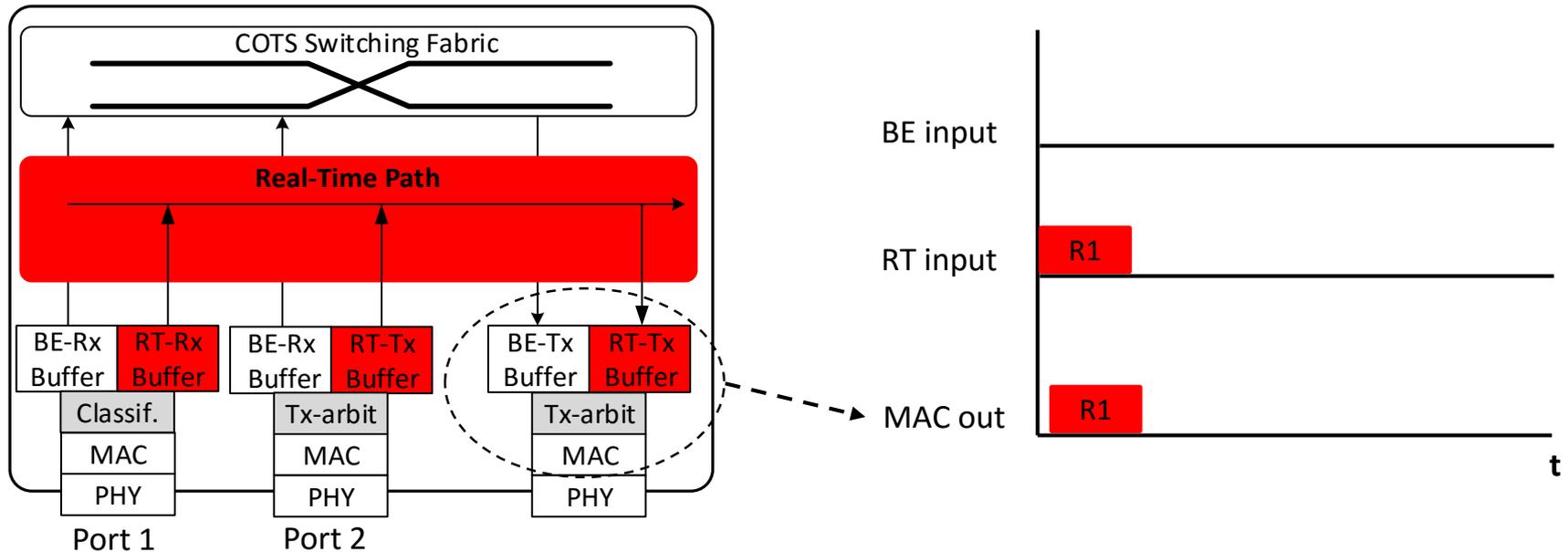
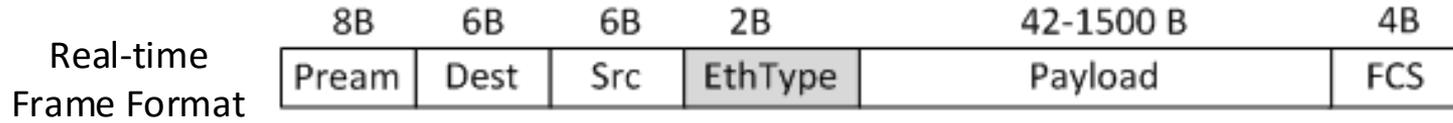


Real-Time Capable Switch

- Dedicated **bus-like path** for scheduled frames
 - On-the-fly frame classification and processing
 - Coordination of real-time frames prevents arbitration overhead
 - Default broadcast prevents address processing
 - Cut-through forwarding and strict priority

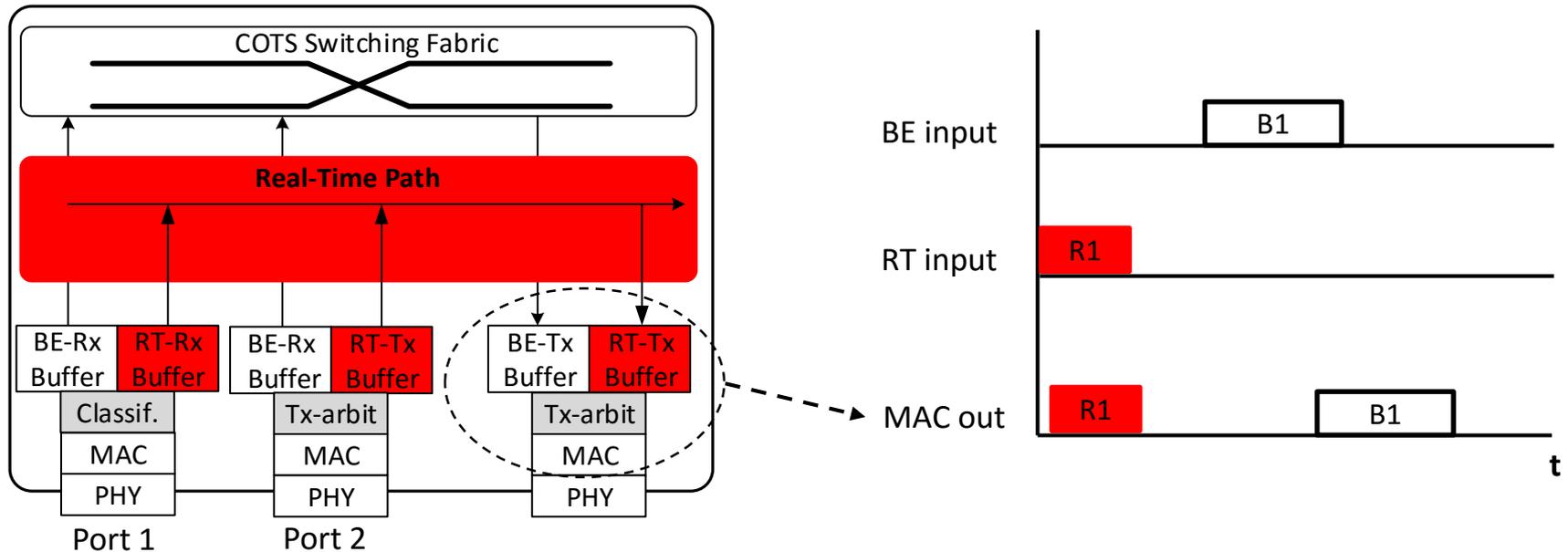
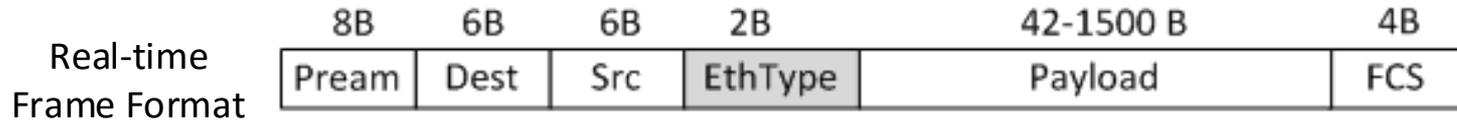


Real-Time Capable Switch



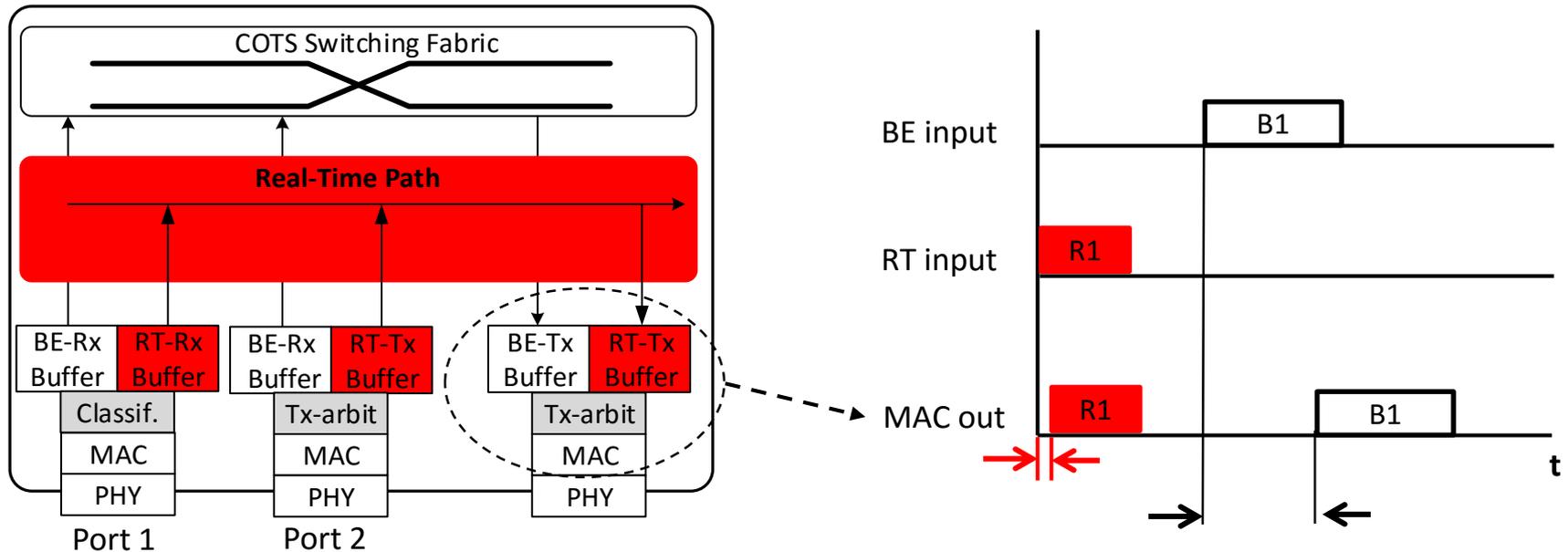
- On-the-fly classification of incoming frames

Real-Time Capable Switch



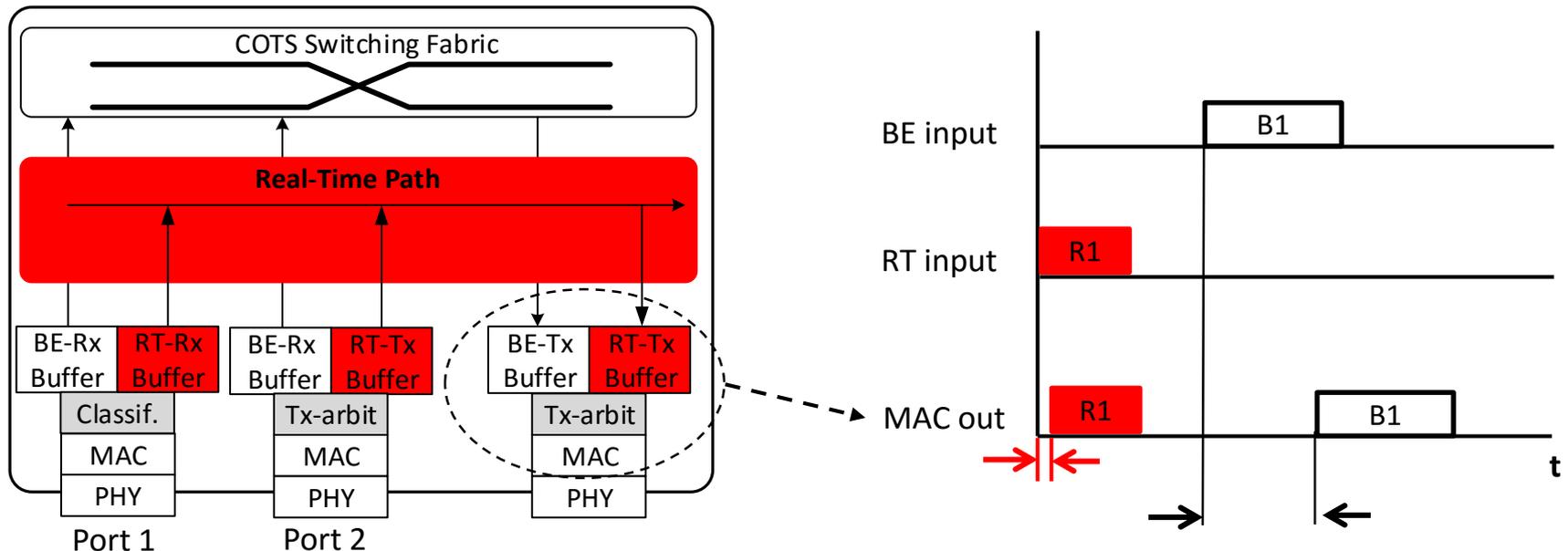
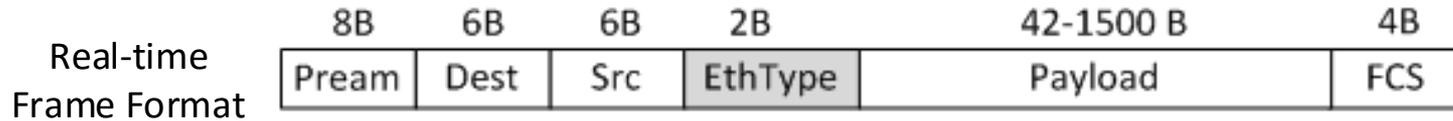
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency

Real-Time Capable Switch



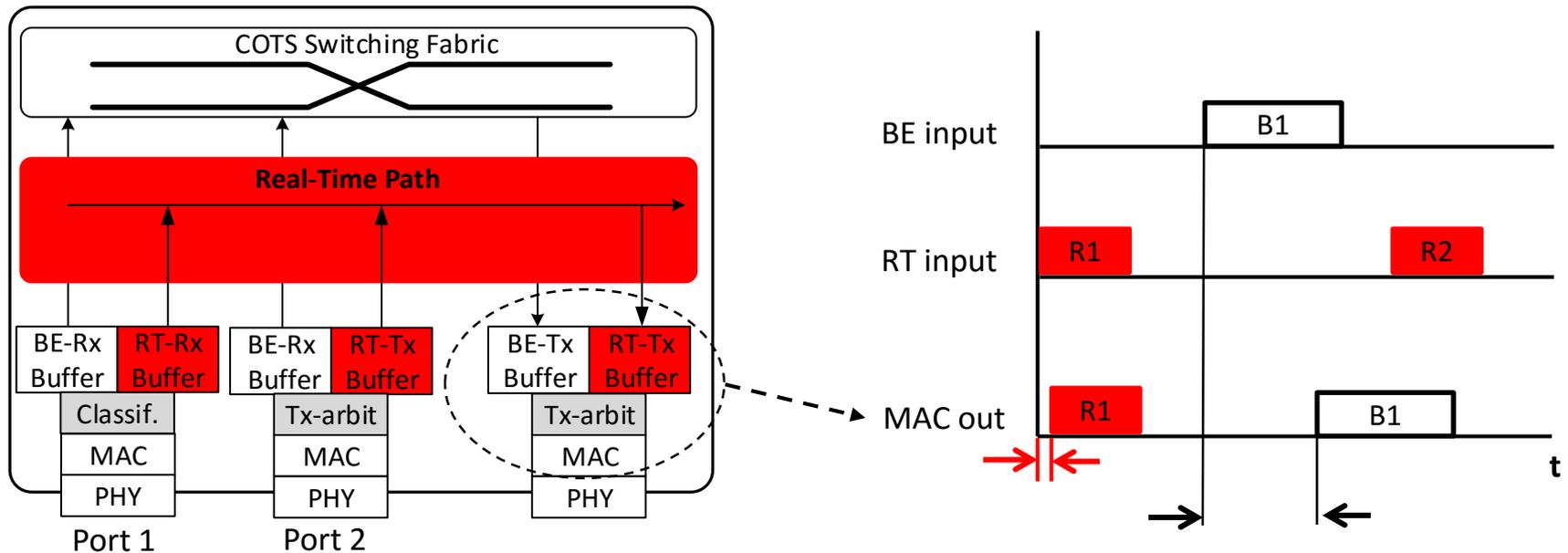
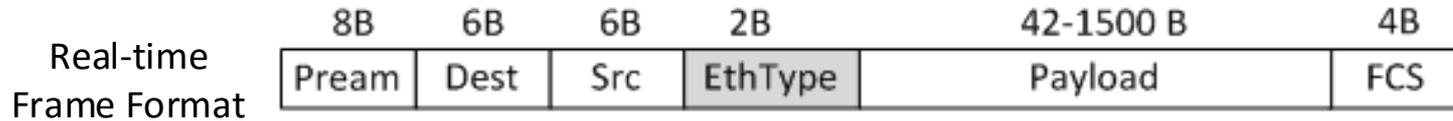
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency

Real-Time Capable Switch



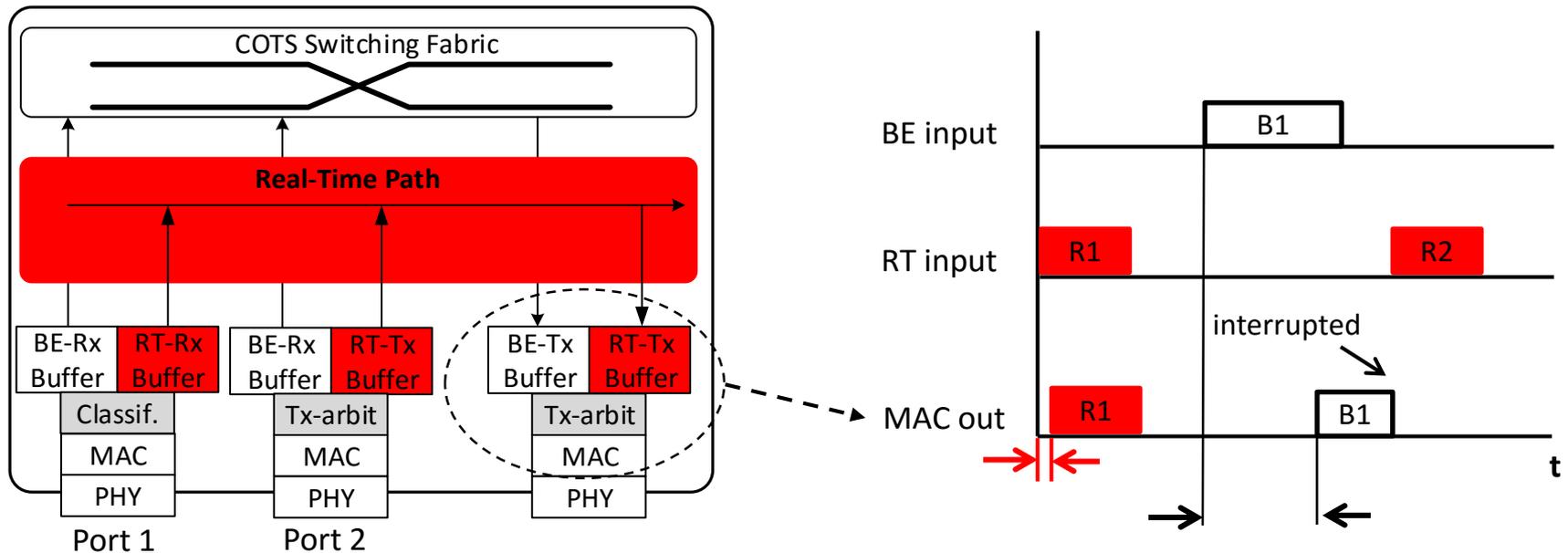
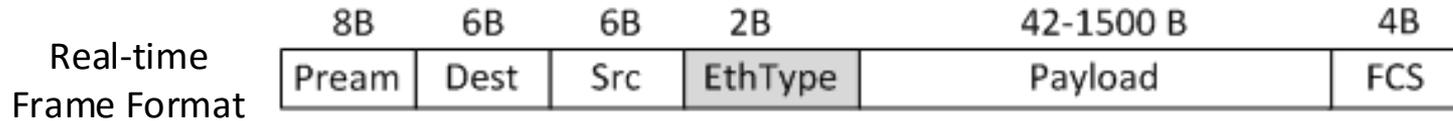
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency
 - Real-time frames receive strict priority access to output ports
 - Best-effort frames communicate if there are no real-time frames
 - Interruption/preemption of conflicting best-effort transactions

Real-Time Capable Switch



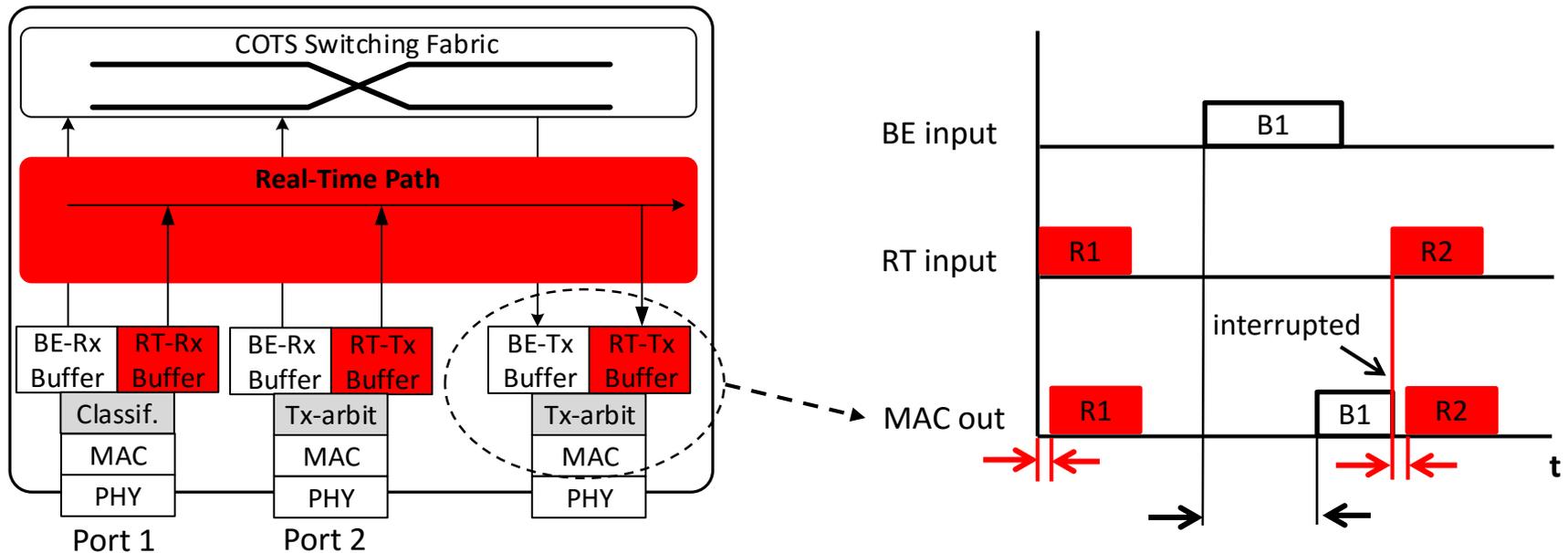
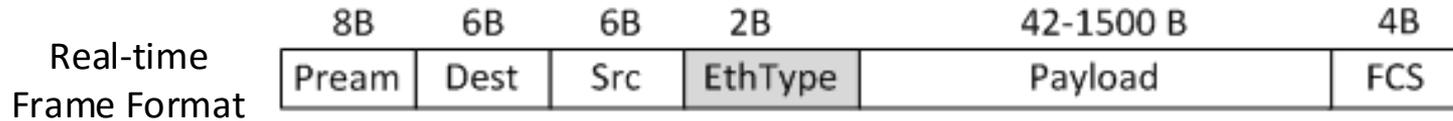
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency
 - Real-time frames receive strict priority access to output ports
 - Best-effort frames communicate if there are no real-time frames
 - Interruption/preemption of conflicting best-effort transactions

Real-Time Capable Switch



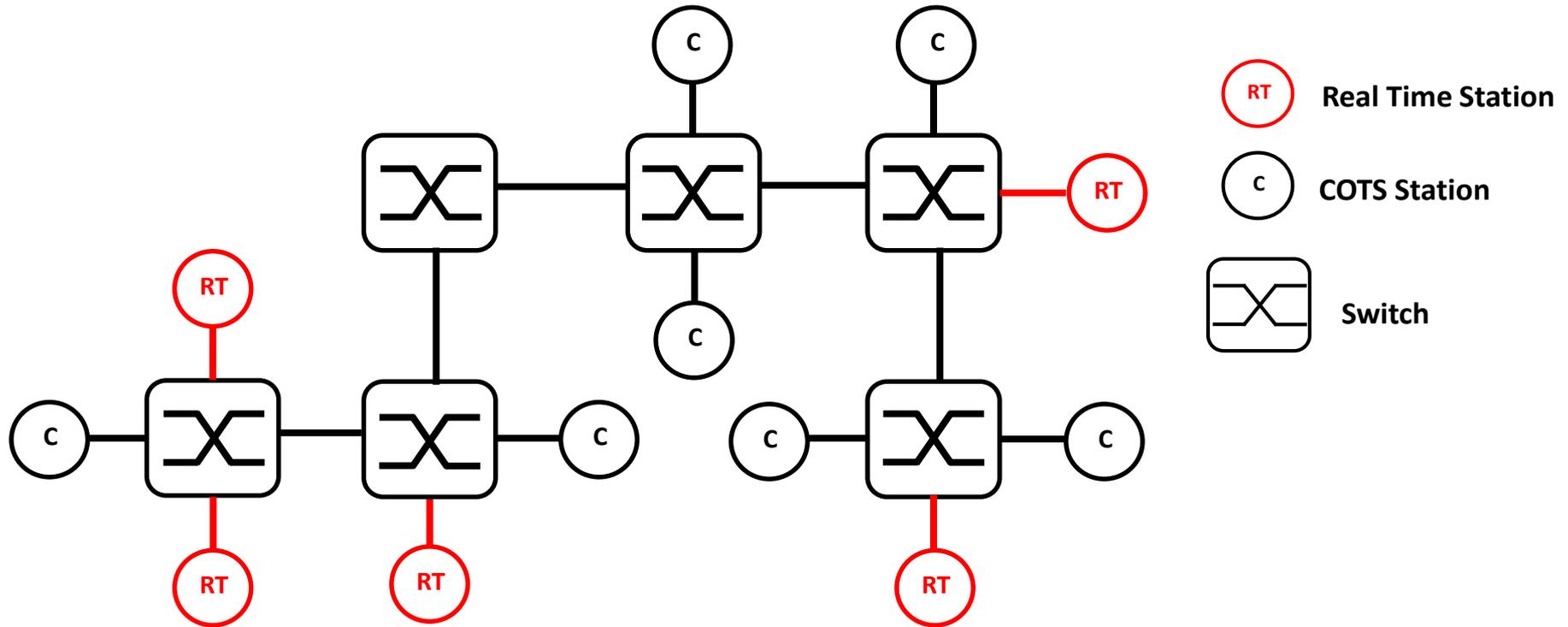
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency
 - Real-time frames receive strict priority access to output ports
 - Best-effort frames communicate if there are no real-time frames
 - Interruption/preemption of conflicting best-effort transactions

Real-Time Capable Switch



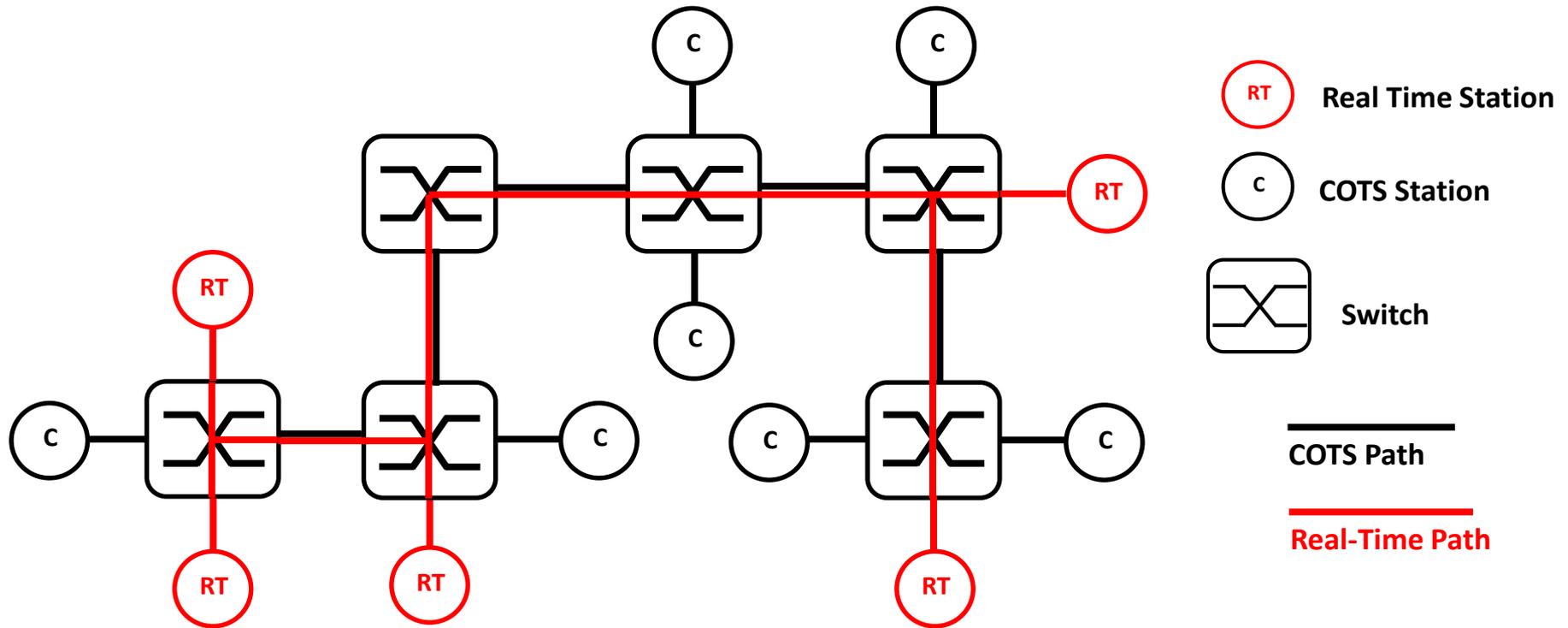
- On-the-fly classification of incoming frames
- Low-overhead of real-time path => low and fixed latency
 - Real-time frames receive strict priority access to output ports
 - Best-effort frames communicate if there are no real-time frames
 - Interruption/preemption of conflicting best-effort transactions

Automatic Integration of RT Devices



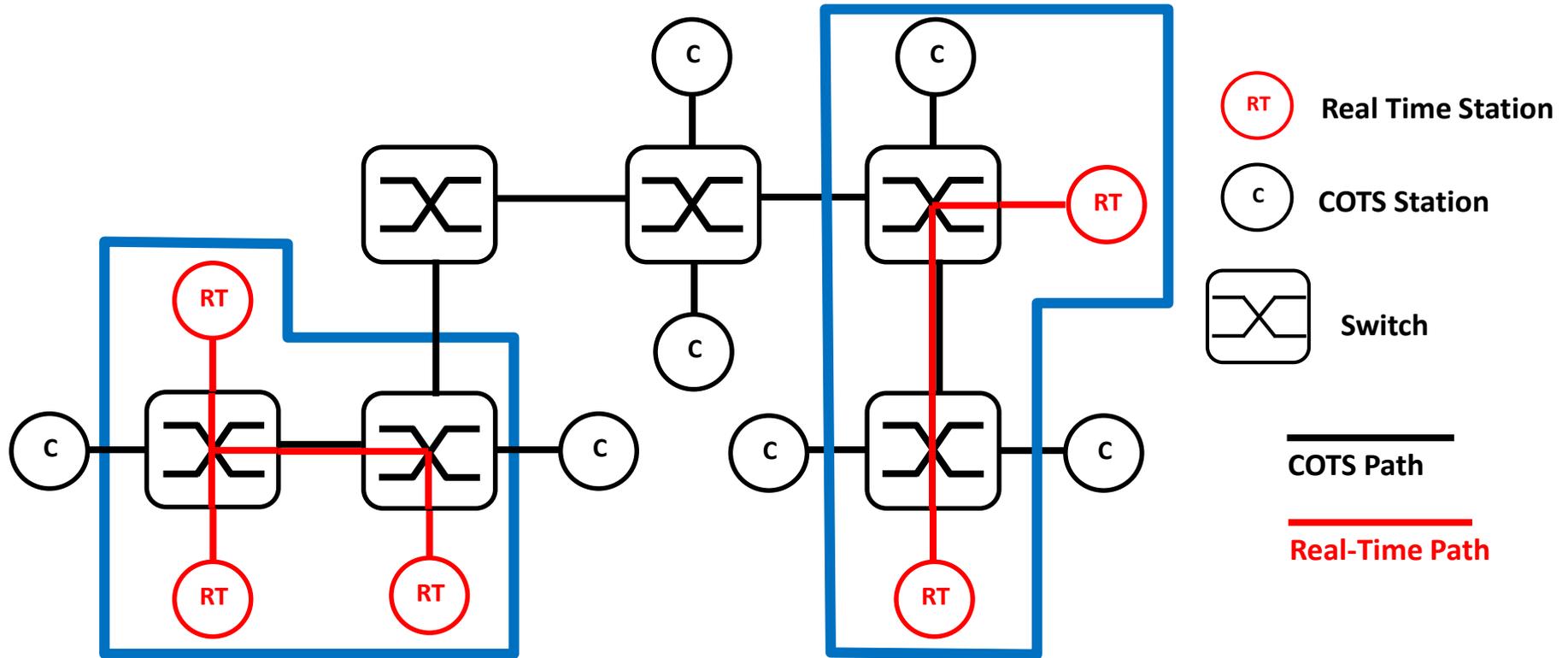
- Acknowledge mechanism to discover real-time capable devices
 - Triggered-during synchronization cycle
 - Generates a logical-bus between real-time capable stations
 - Switches do not forward real-time frames to COTS stations
 - Mitigates penalization over best-effort traffic

Automatic Integration of RT Devices



- Acknowledge mechanism to discover real-time capable devices
 - Triggered-during synchronization cycle
 - Generates a logical-bus between real-time capable stations
 - Switches do not forward real-time frames to COTS stations
 - Mitigates penalization over best-effort traffic

Segmentation of RT domains



- Segmentation of real-time domains in large networks
 - Keep real-time traffic within predefined scopes
 - Stations in real-time domains can still exchange best-effort frames
 - Mitigates penalization over best-effort traffic

Summary

- Comprehensive RTE framework
 - Formal communication model for dynamic-TDMA
 - Programming environment for describing and verifying schedules
 - Hardware infrastructure for testing and validating in the field
- Experimentally validated properties
 - Deterministic end-to-end latency for real-time frames
 - Accurate latency models in multi-hop settings
 - Online bandwidth management
 - Segmentation of real-time domains in large networks
 - Seamless integration with COTS stacks and legacy infrastructure
- Open-source
 - All software and hardware description code is available
 - Low-cost and easily extensible
 - Rapid prototyping

Demo: Buffer-less Video Streaming

- **Setup:**
 - A camera sending reference video lines
 - A set of remote displays receiving video lines
 - A multi-hop network with real-time capable devices
- **Run:**
 - The camera periodically sends video lines to remote displays
 - Displays periodically read and show the most recent line
 - **No buffering** of video lines
 - Lines delivered out of time are automatically discarded

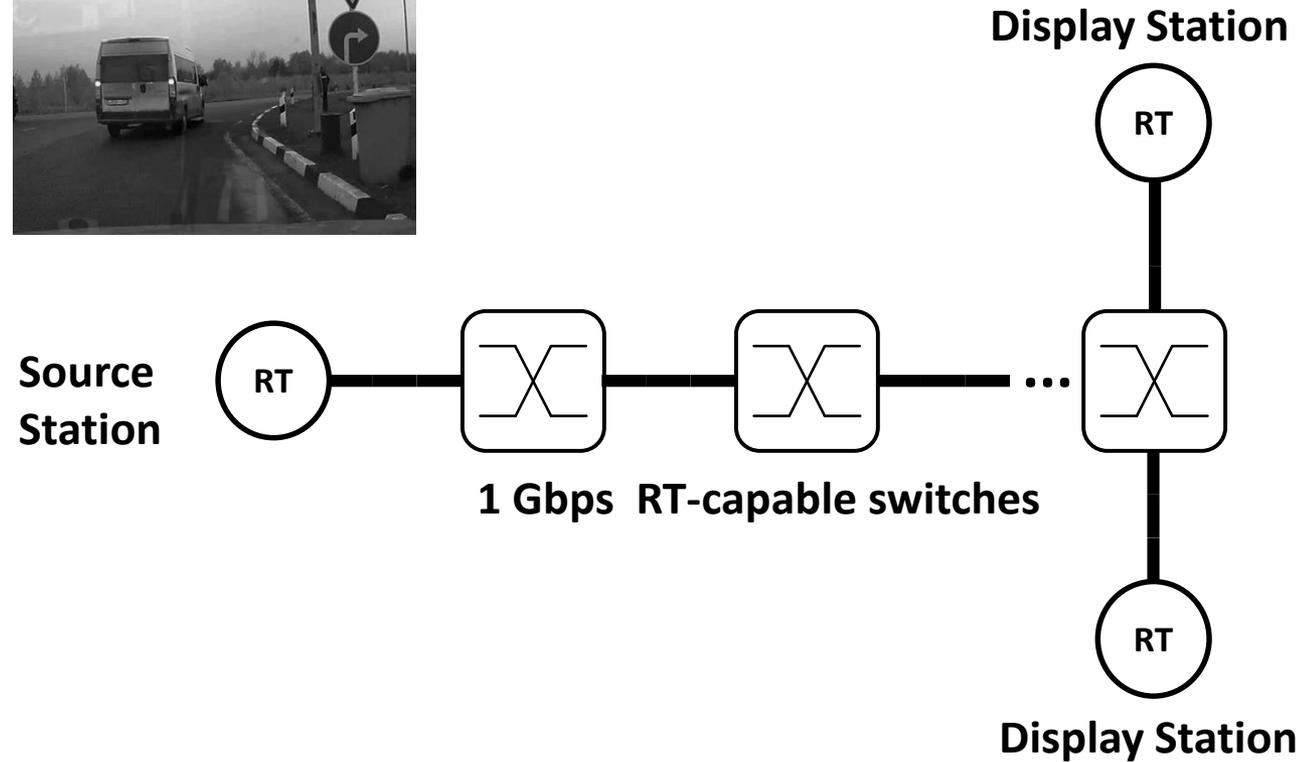
Experiment 1

Timing Properties of Real-Time Capable NICs and Switches

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz



Buffer-less Video Streaming

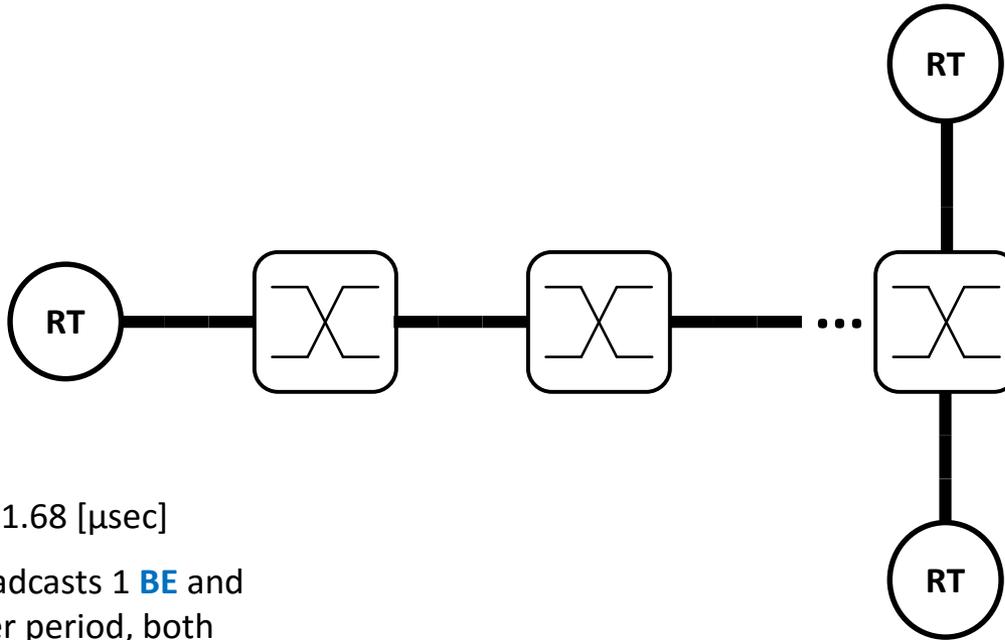
❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz



Display Station

Source Station



Display Station

❑ Line-Period: 31.68 [μ sec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Total Video Bandwidth \approx 340 Mbps

Buffer-less Video Streaming

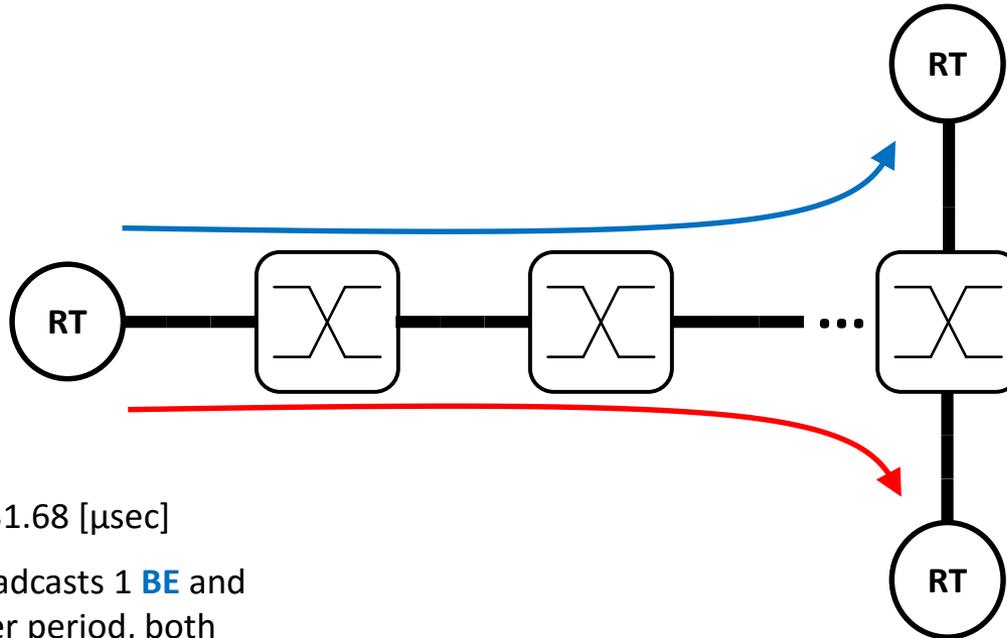
❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz



Display Station

Source Station



Display Station

❑ Line-Period: 31.68 [μsec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Total Video Bandwidth ≈ 340 Mbps

Buffer-less Video Streaming

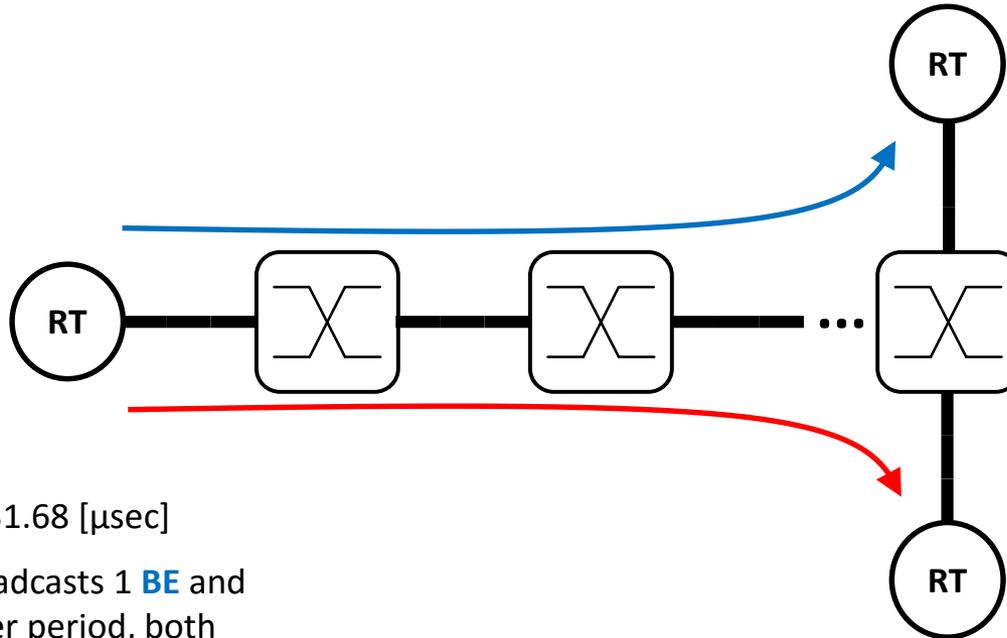
❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz



Display Station

Source Station



❑ Line-Period: 31.68 [μsec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Total Video Bandwidth ≈ 340 Mbps

Display Station

❑ Line-Period: 31.68 [μsec]

❑ On each period, receivers display the received line immediately (**No buffer!**)

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

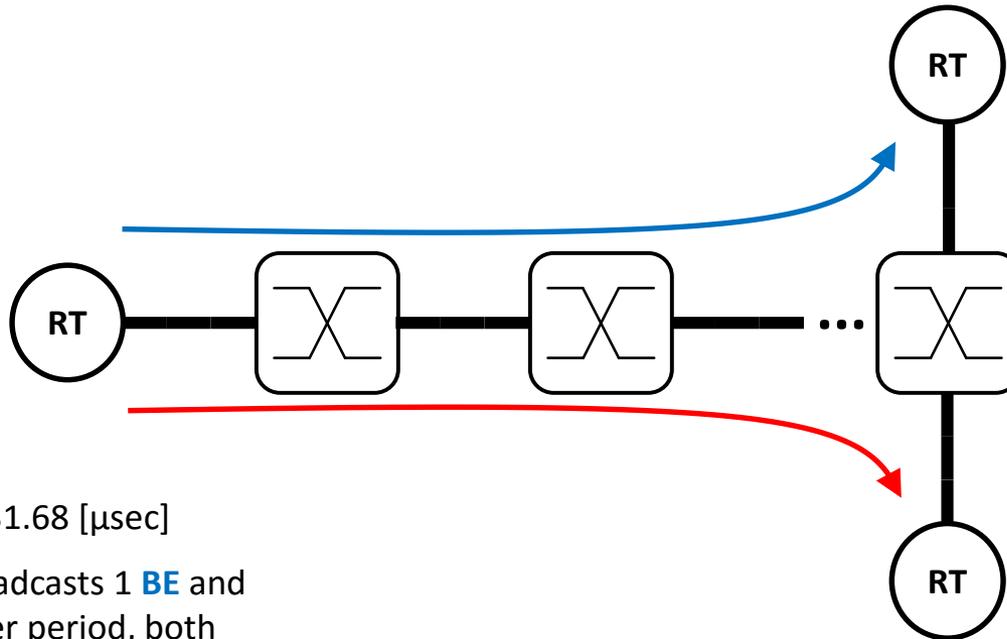


Best-Effort Video



Display Station

Source Station



Real-time Video



Display Station

❑ Line-Period: 31.68 [μsec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Total Video Bandwidth ≈ 340 Mbps

❑ Line-Period: 31.68 [μsec]

❑ On each period, receivers display the received line immediately (**No buffer!**)

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

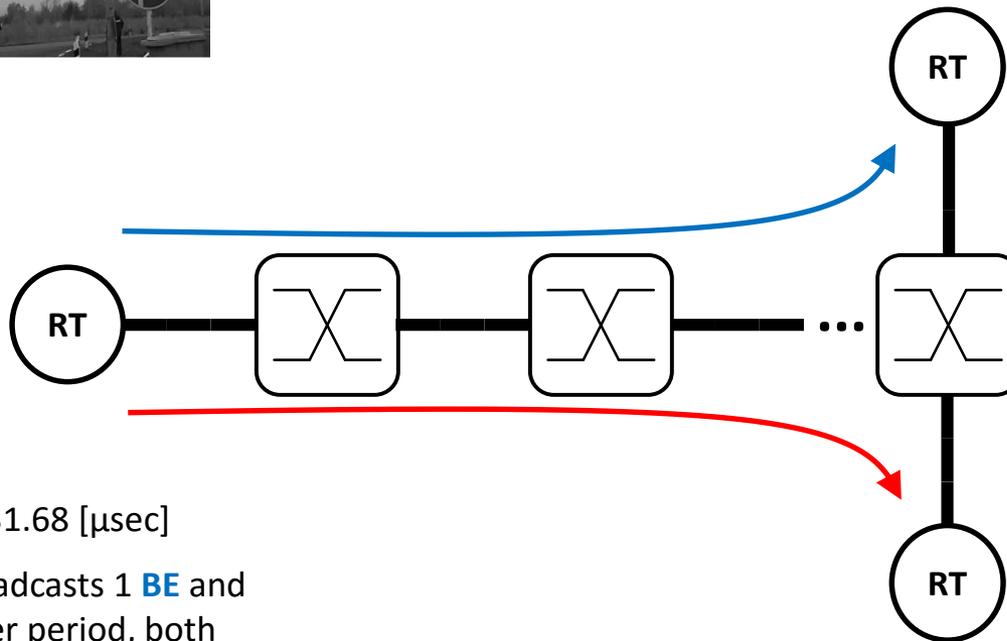


Best-Effort Video



Display Station

Source Station



Real-time Video



Display Station

❑ Line-Period: 31.68 [μsec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Total Video Bandwidth ≈ 340 Mbps

❑ Line-Period: 31.68 [μsec]

❑ On each period, receivers display the received line immediately (**No buffer!**)

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

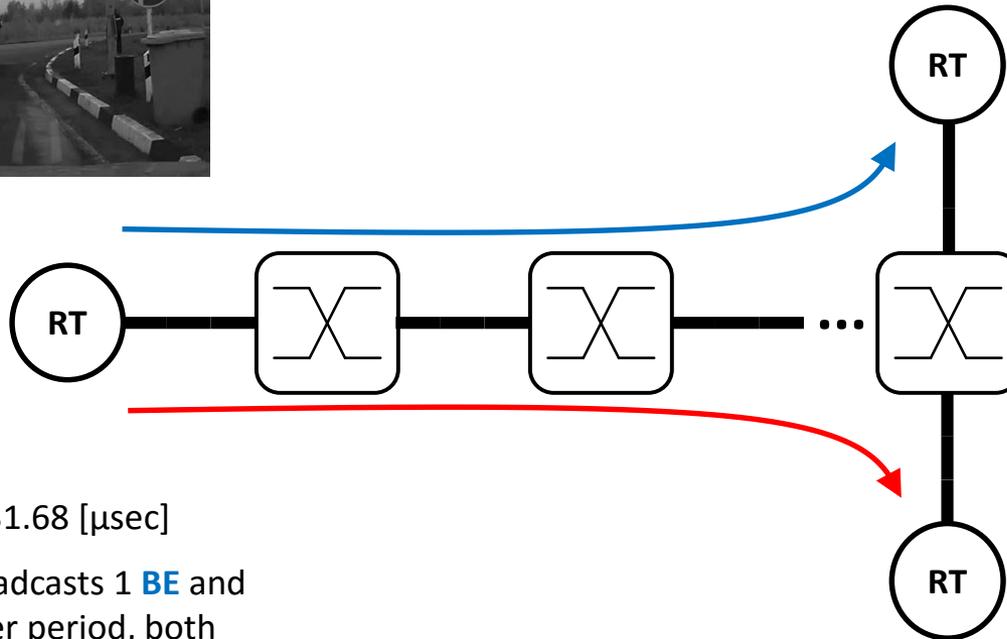


Display Station

Best-Effort Video



Source Station



Display Station

Real-time Video



❑ Line-Period: 31.68 [μsec]

❑ Schedule broadcasts 1 **BE** and 1 **RT** frame per period, both containing the same line

— BE video lines

— RT video lines

❑ Line-Period: 31.68 [μsec]

❑ On each period, receivers display the received line immediately (**No buffer!**)

❑ Total Video Bandwidth ≈ 340 Mbps

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

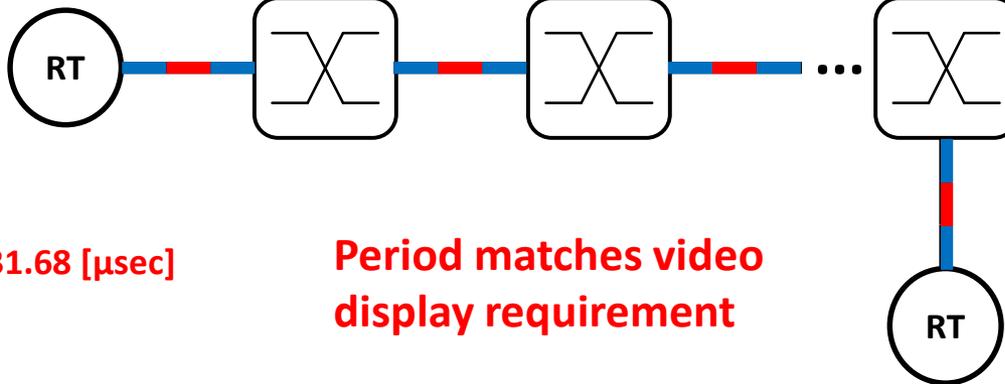


Display Station

Best-Effort Video



Source Station



❑ Line-Period: 31.68 [μsec]

Period matches video display requirement

Display Station

Real-time Video



❑ Line-Period: 31.68 [μsec]

Delivery of video lines requires strict periodicity for correct displaying

Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

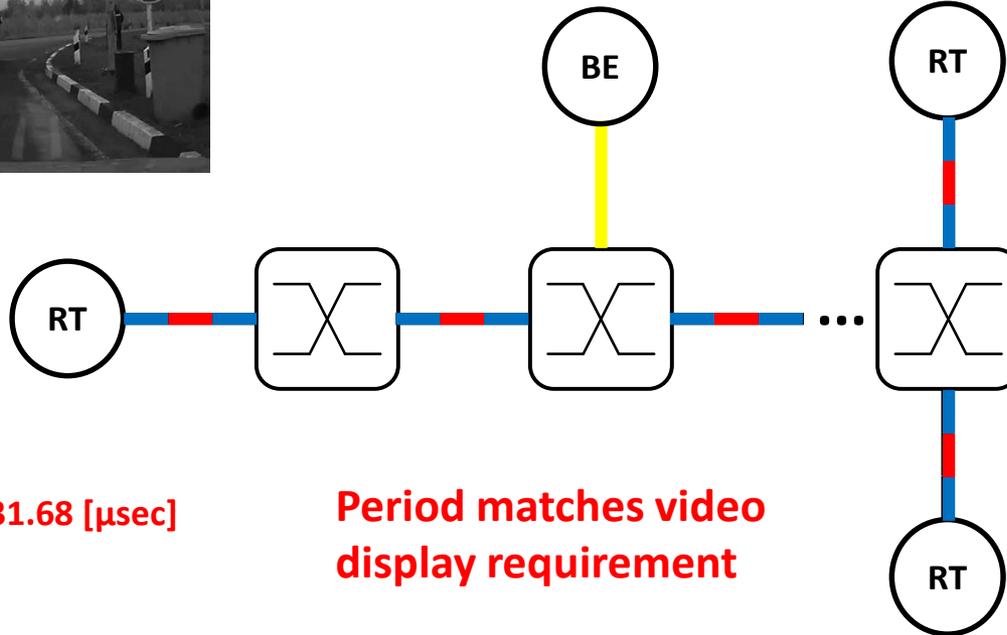


Display Station

Best-Effort Video



Source Station



❑ Line-Period: 31.68 [μsec]

Period matches video display requirement

Display Station

❑ Line-Period: 31.68 [μsec]

Real-time Video



Buffer-less Video Streaming

❑ Grayscale Video: 640x480 @60fps

❑ Pixel clock: 25 Mhz

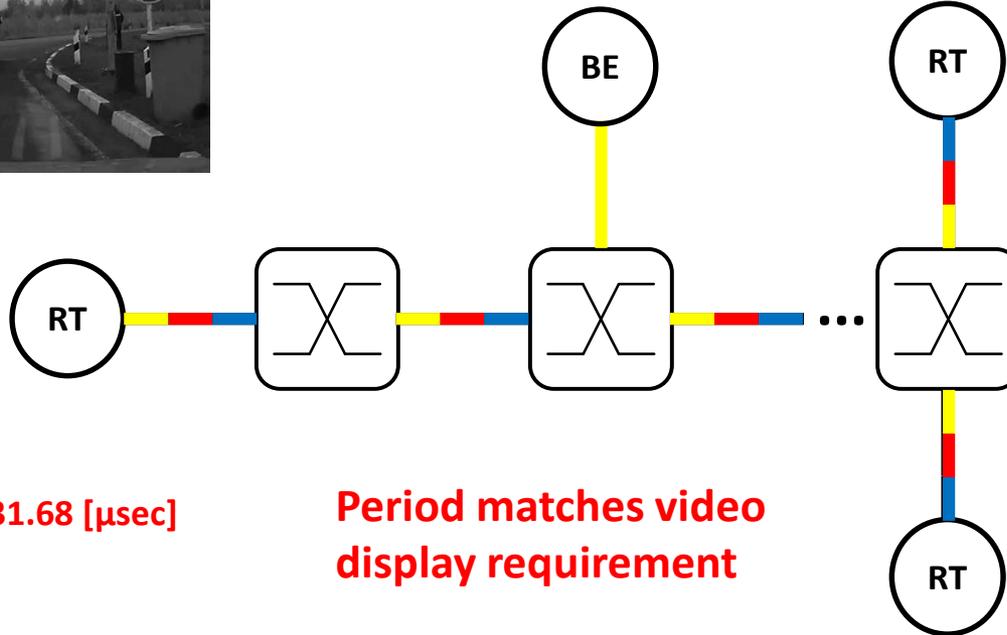


Display Station

Best-Effort Video



Source Station



❑ Line-Period: 31.68 [μ sec]

Period matches video display requirement

❑ Line-Period: 31.68 [μ sec]

Additional best-effort traffic generates congestion in the COTS fabrics

Real-time Video

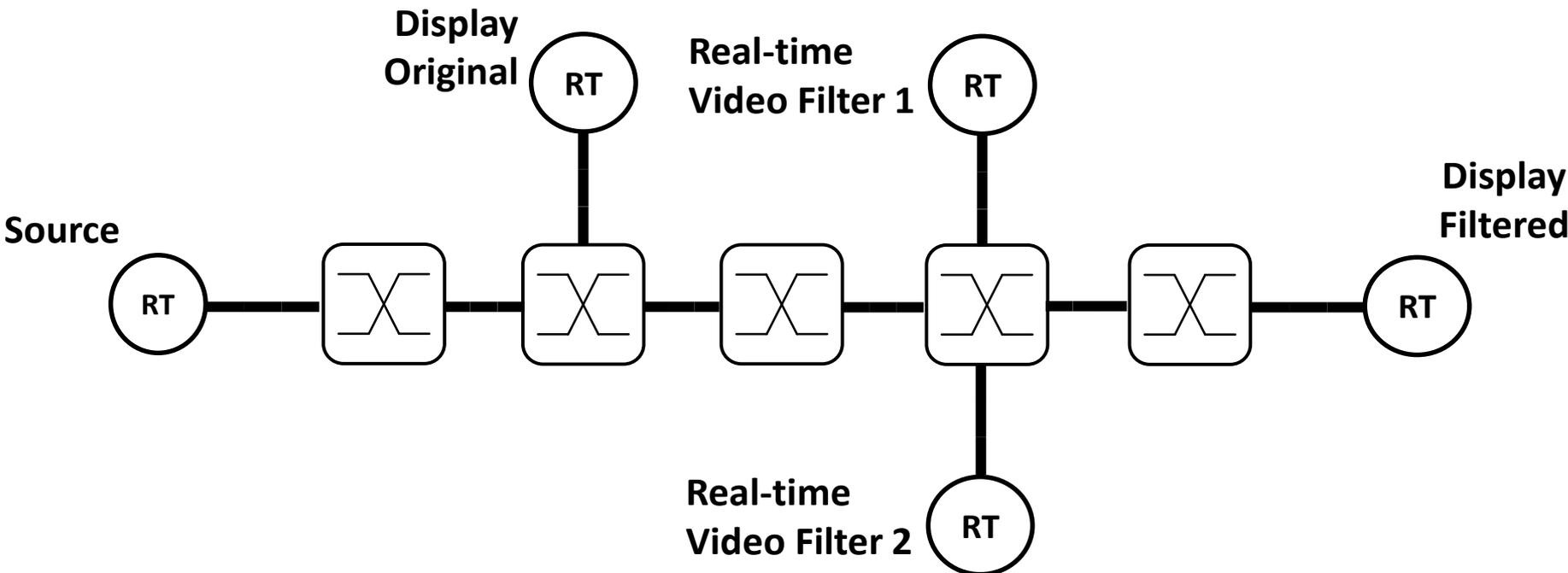


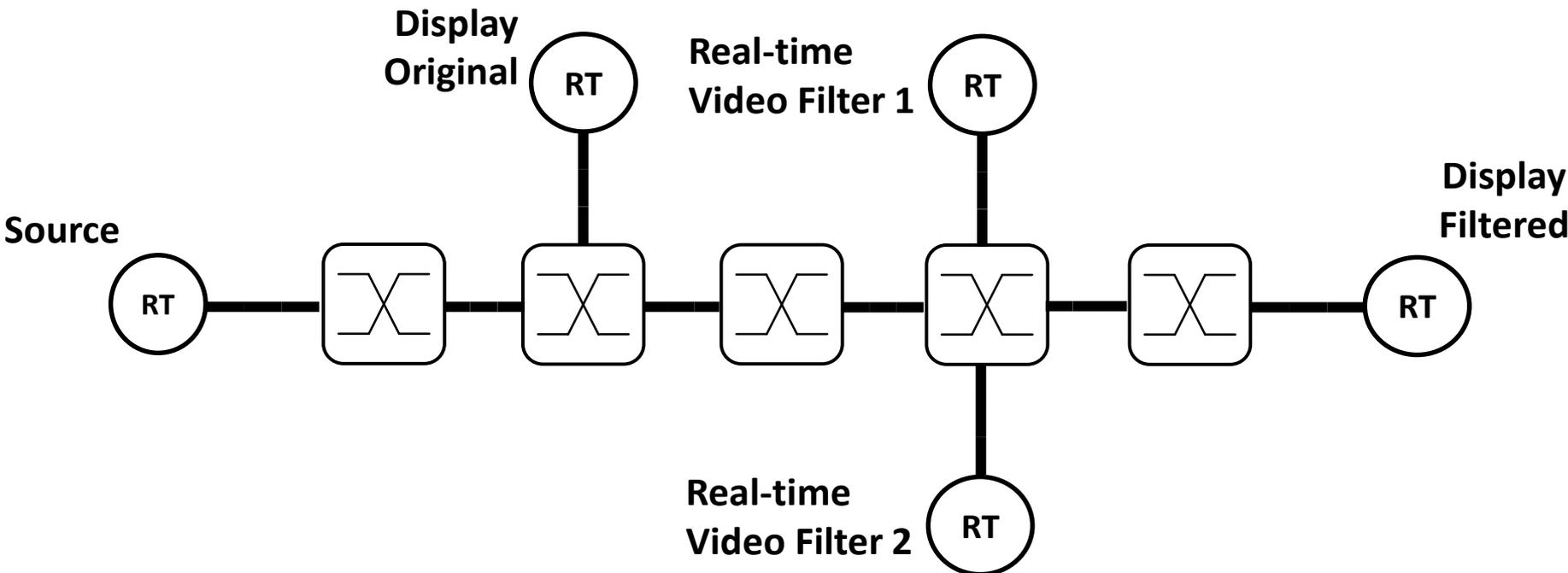
Buffer-less Video Streaming (Video Demo)

Play file demo1-timing.mp4

Experiment 2

Distributed Video Processing and Dynamic Allocation of Real-time Bandwidth



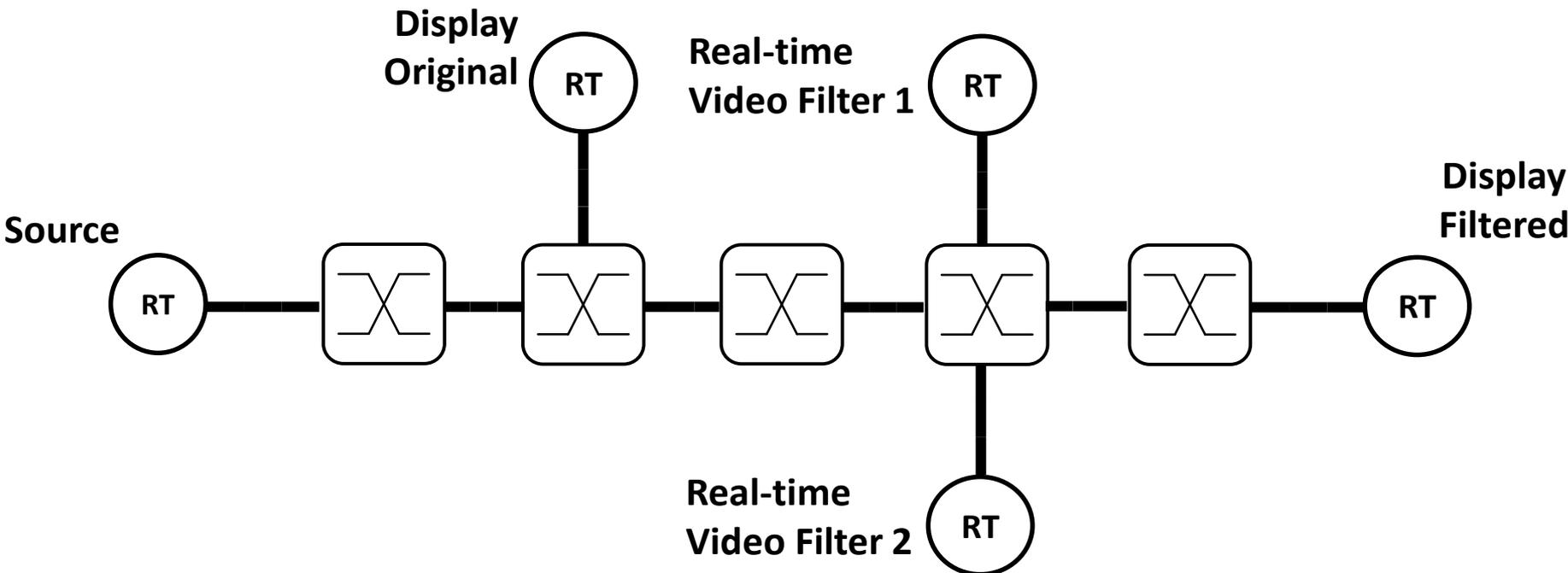


Original



On-the-fly processing

Assembled Processed

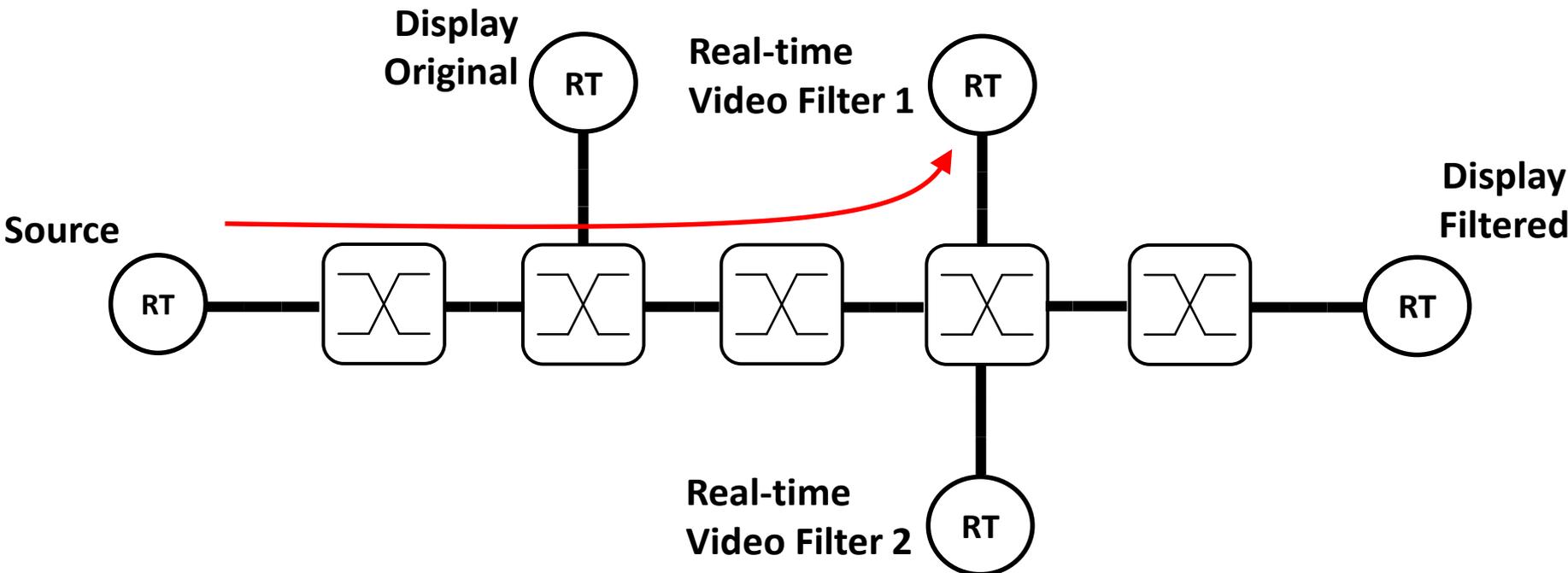


Original



On-the-fly processing

Assembled Processed

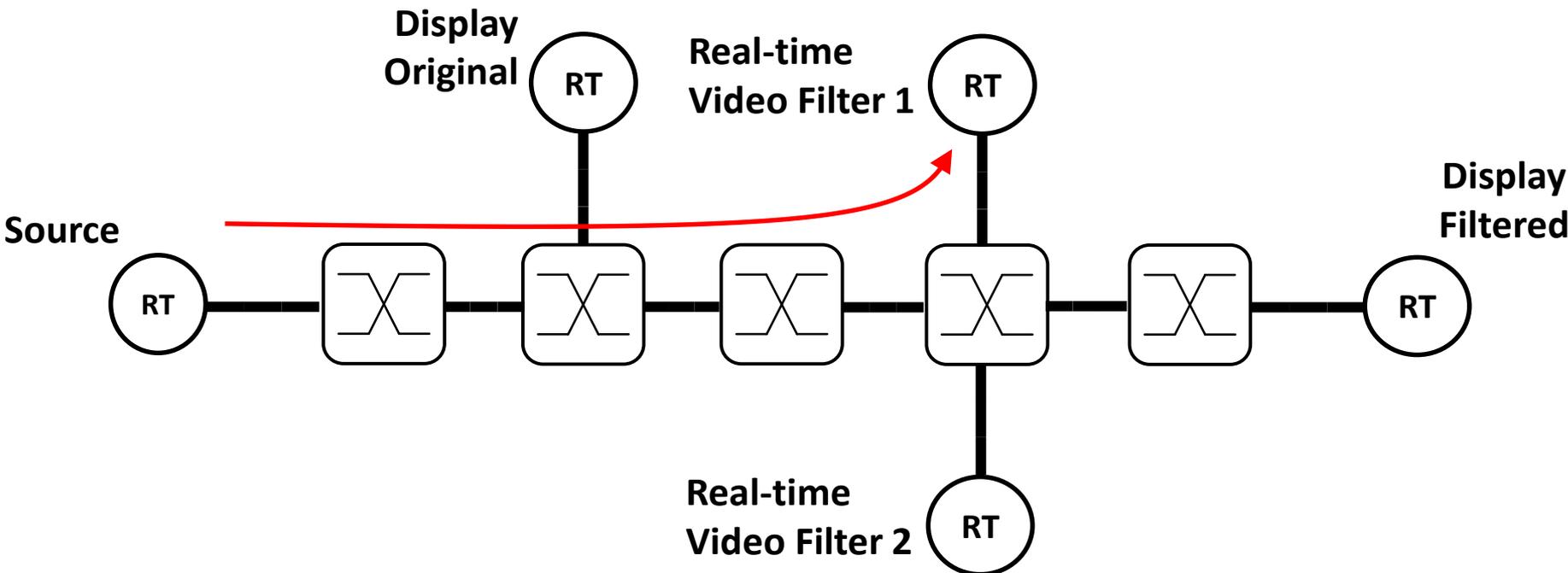


Original



On-the-fly processing

Assembled Processed



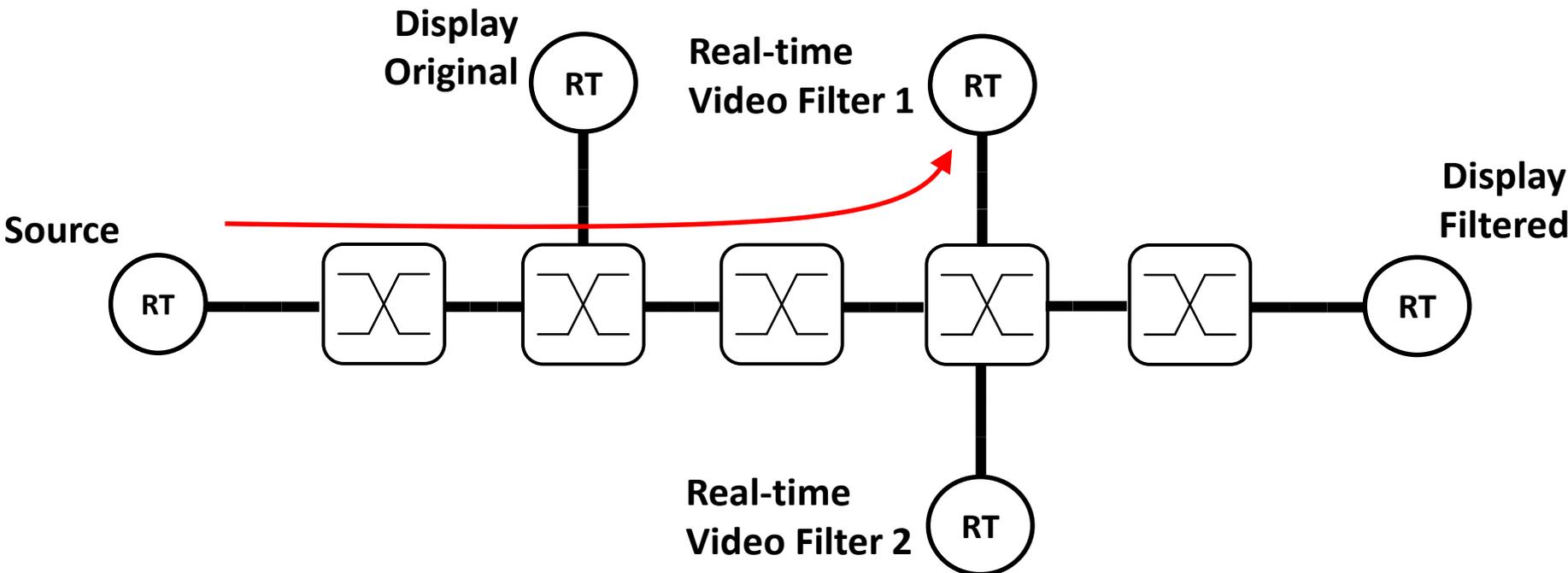
Original



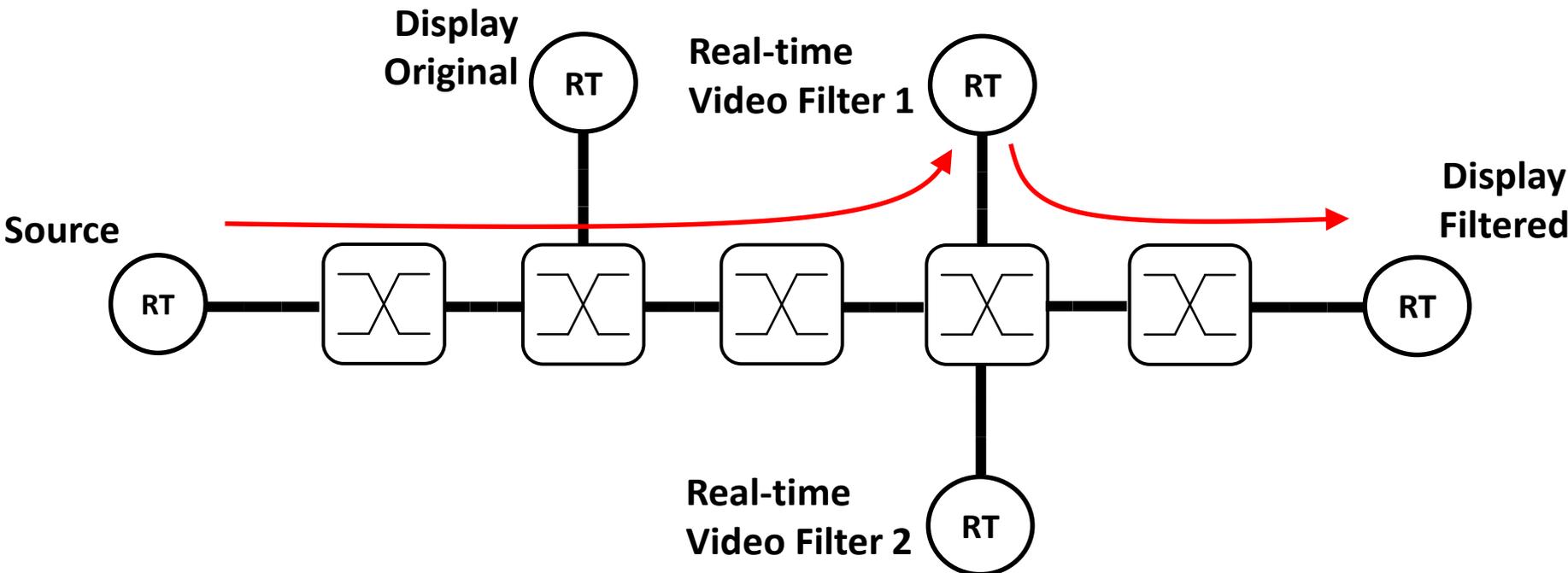
On-the-fly processing



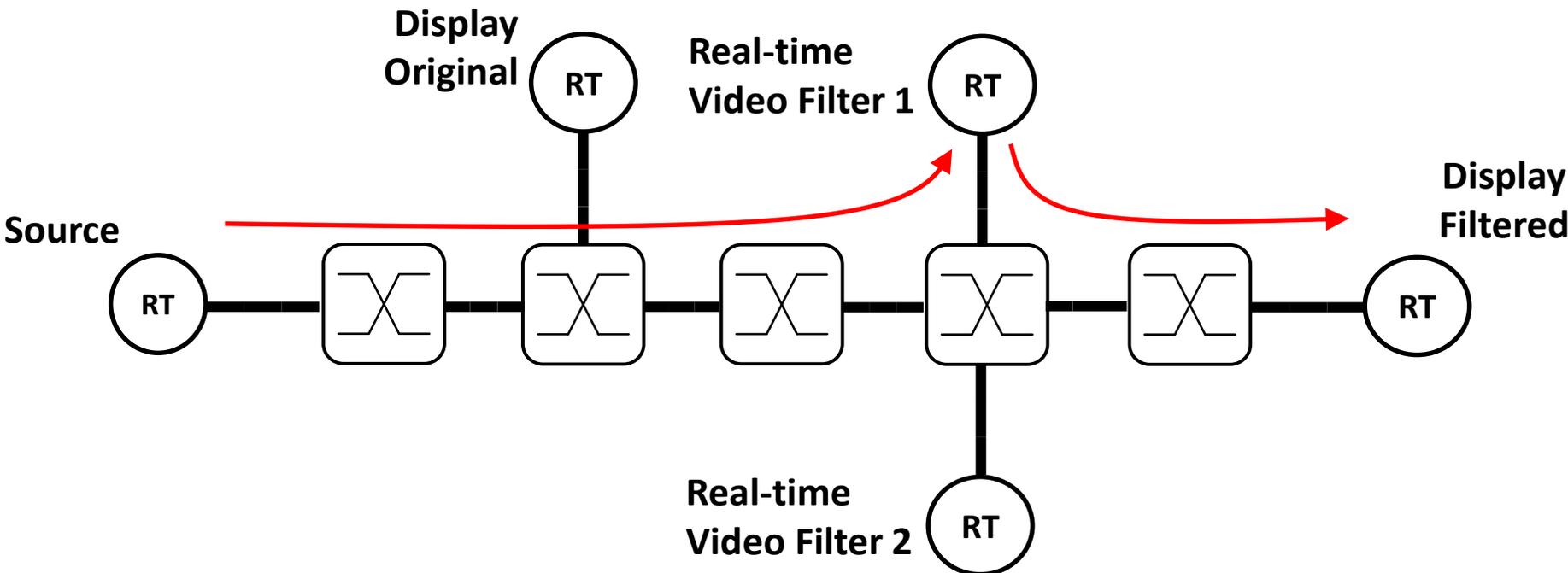
Assembled Processed



Assembled Processed



Assembled Processed



Original

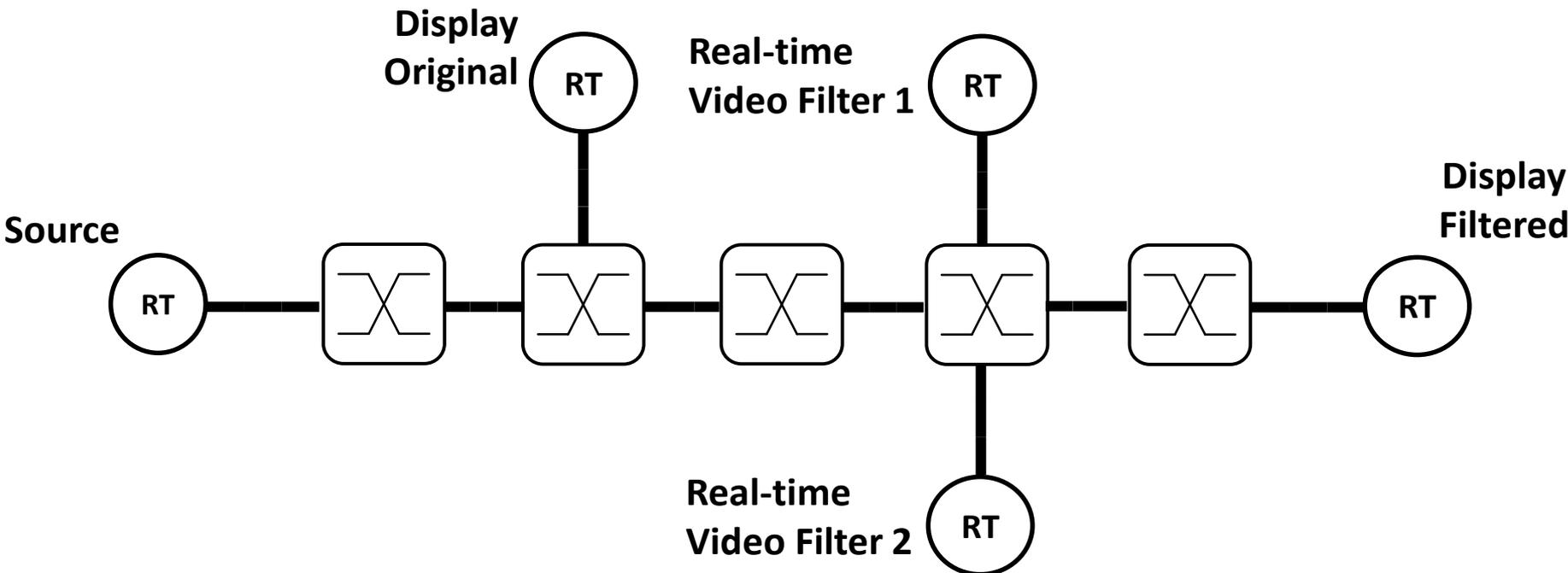


On-the-fly processing



Assembled Processed





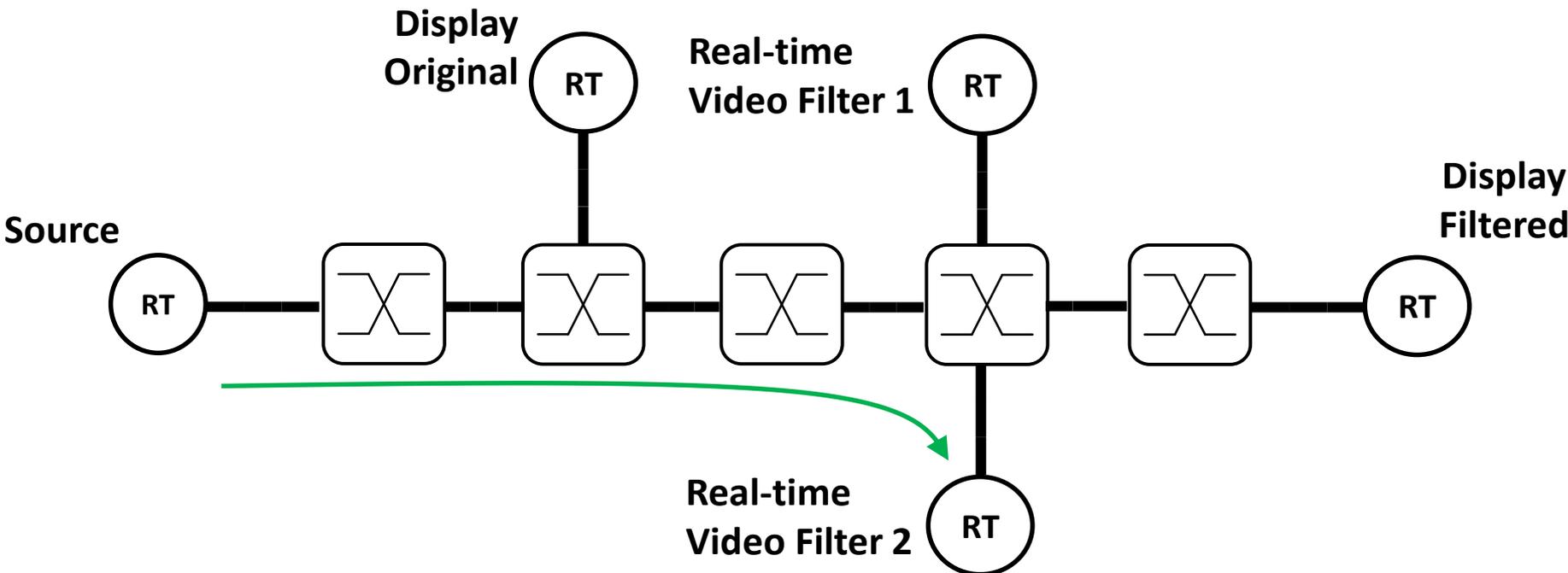
Original



On-the-fly processing

Assembled Processed





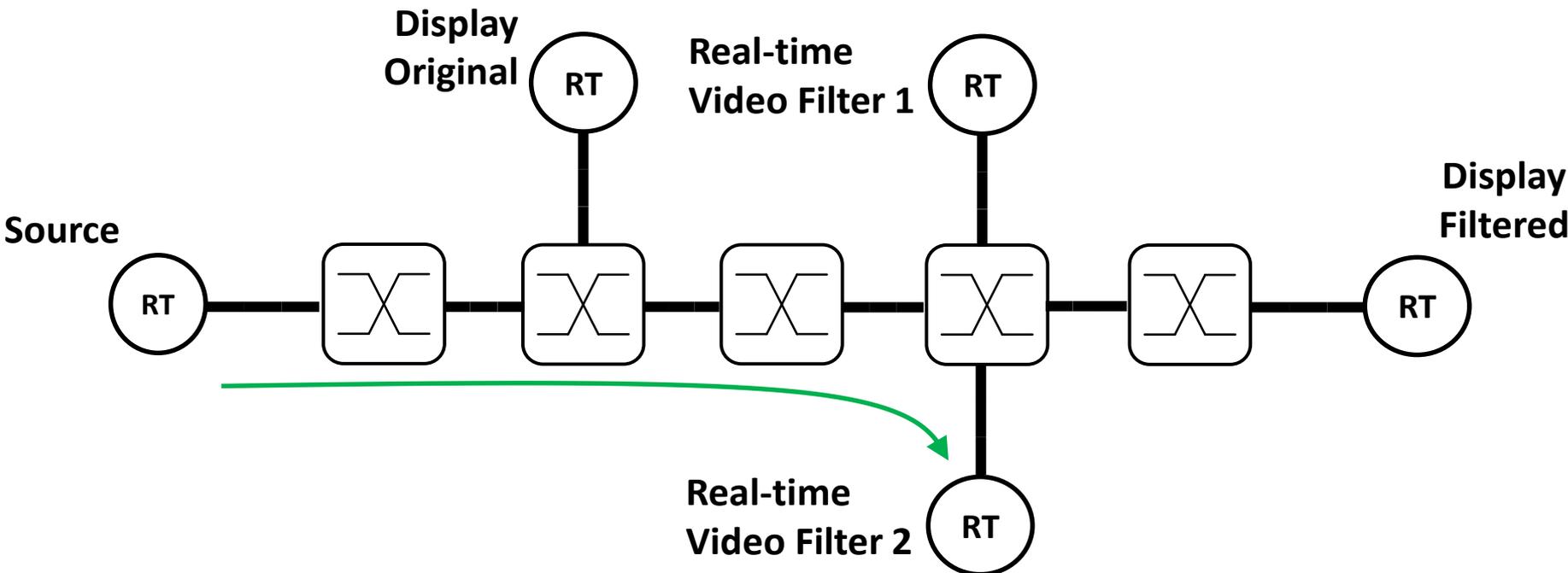
Original



On-the-fly processing

Assembled Processed





Original

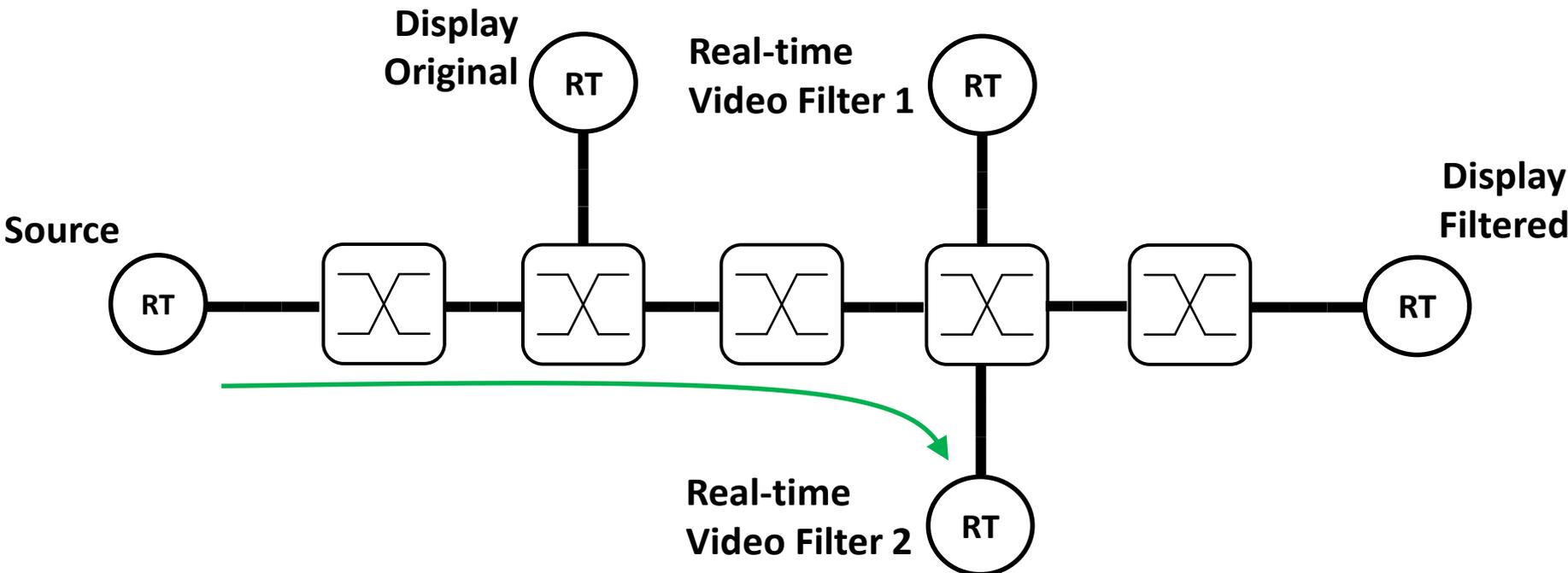


On-the-fly processing



Assembled Processed





Original

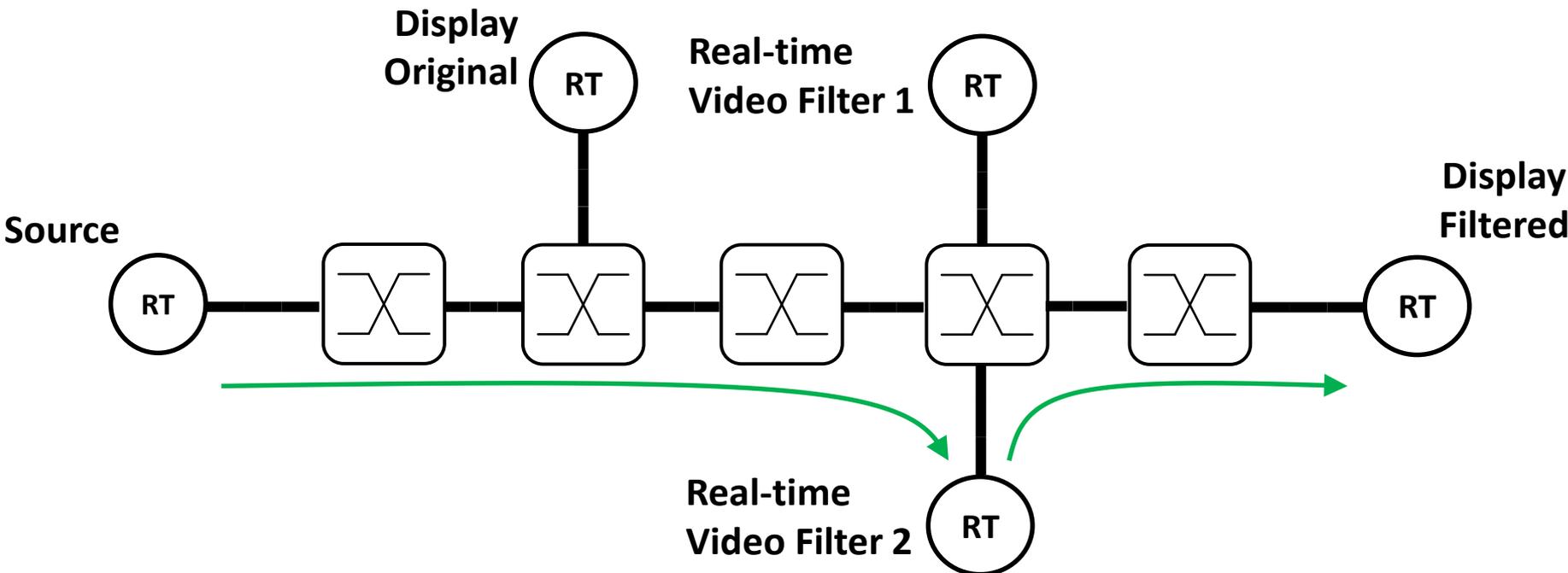


On-the-fly processing



Assembled Processed





Original

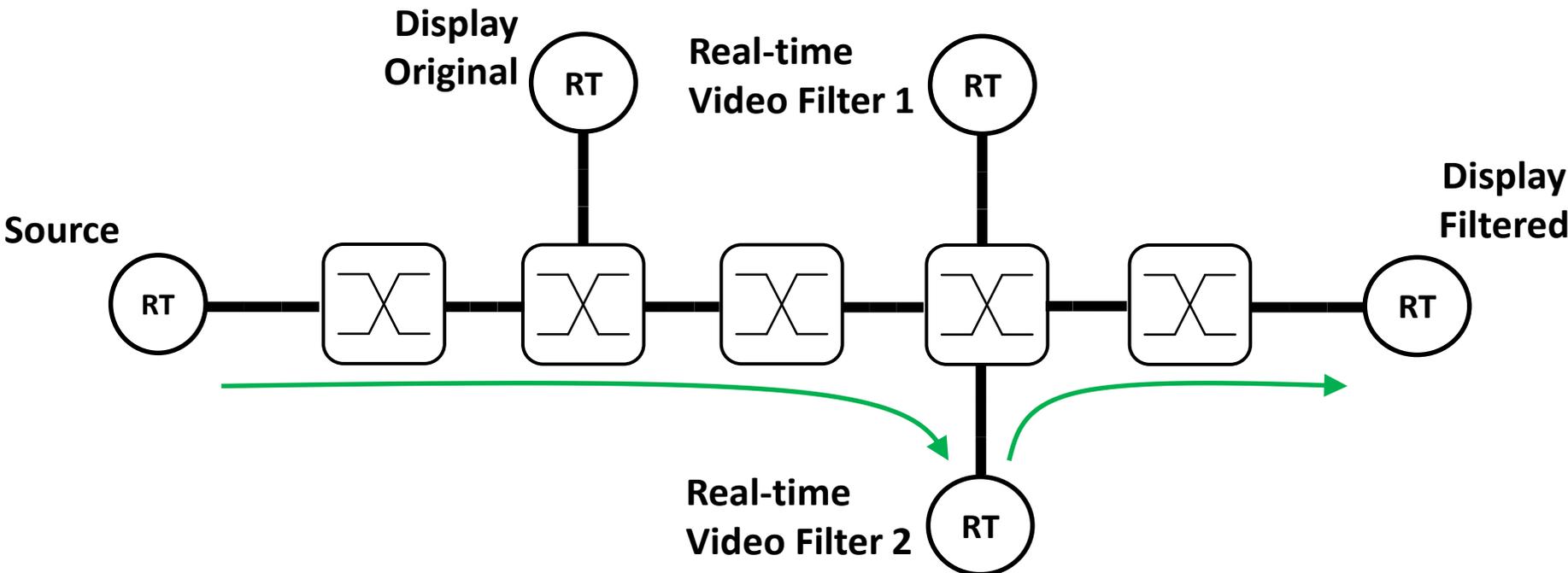


On-the-fly processing



Assembled Processed





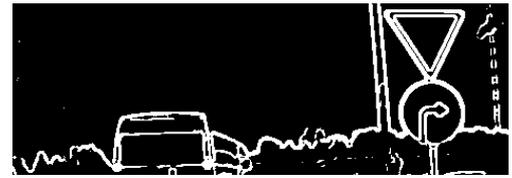
Original

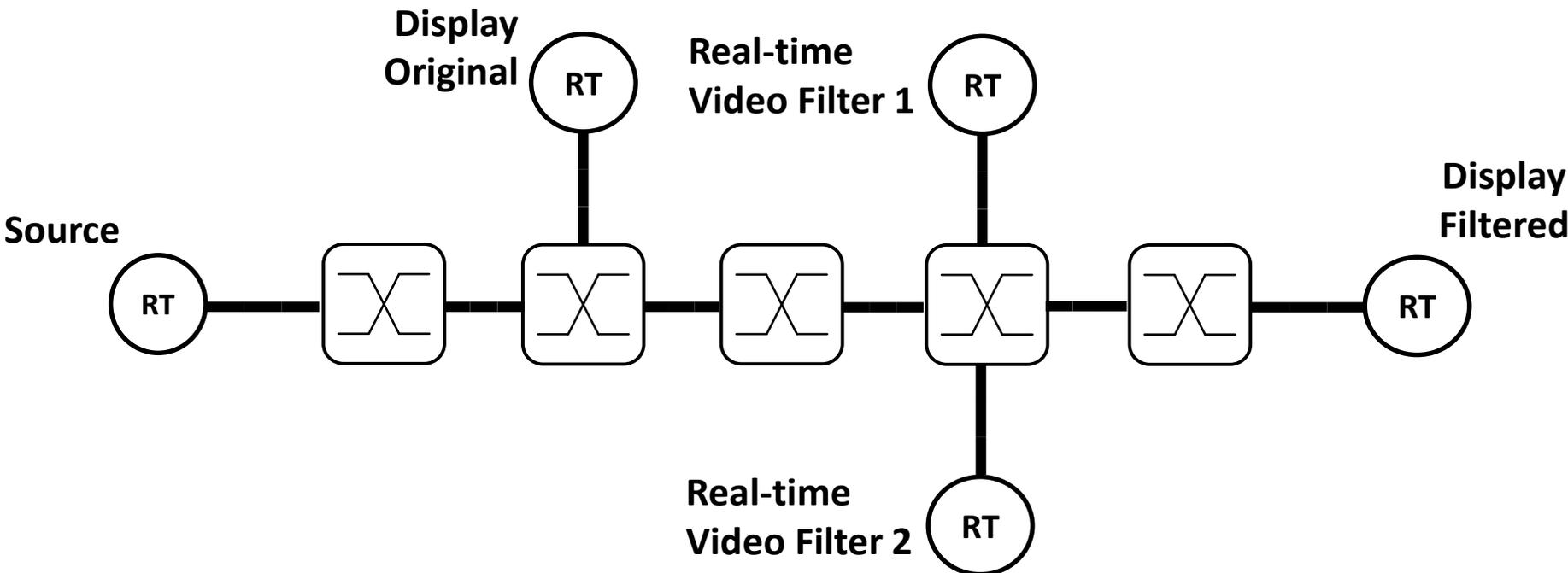


On-the-fly processing



Assembled Processed





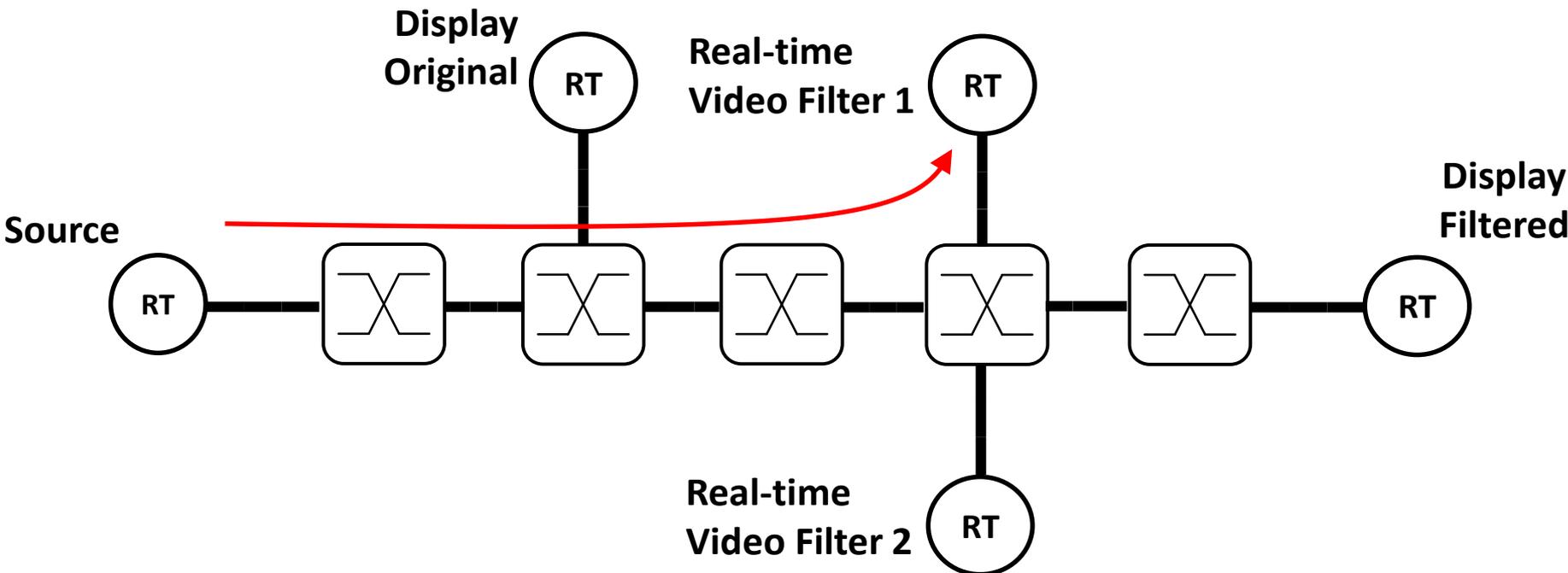
Original



On-the-fly processing

Assembled Processed





Original

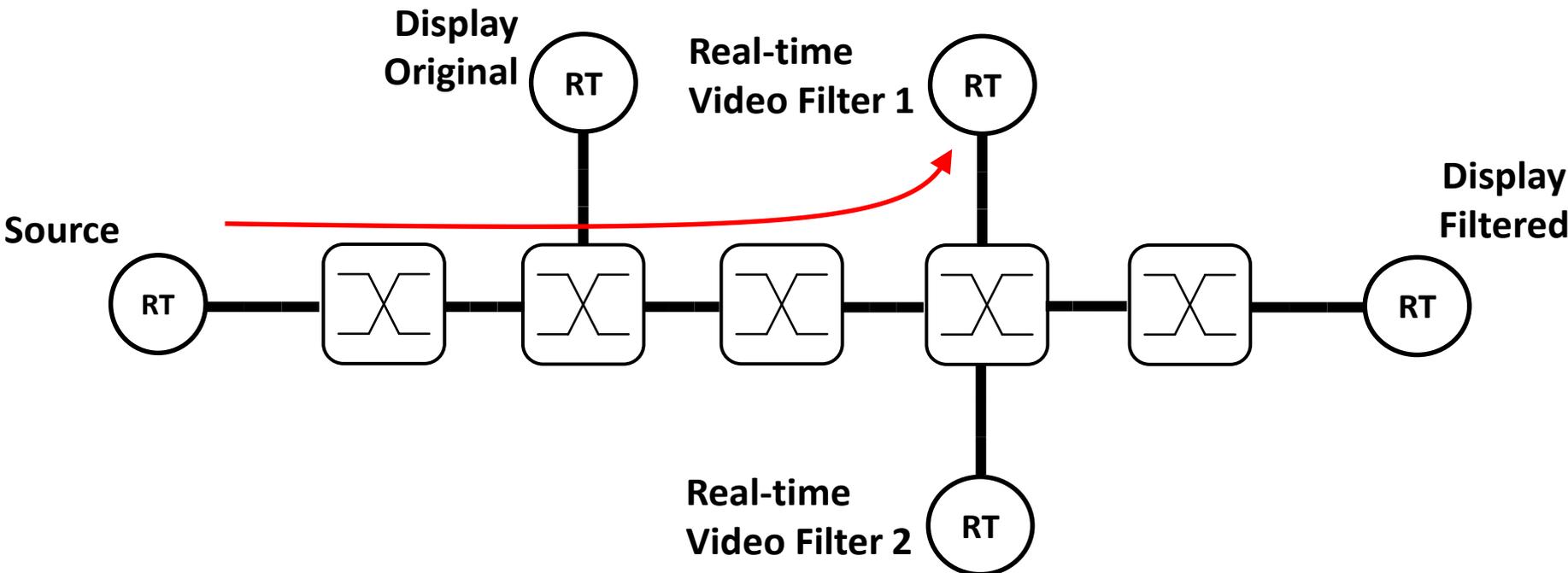


On-the-fly processing



Assembled Processed





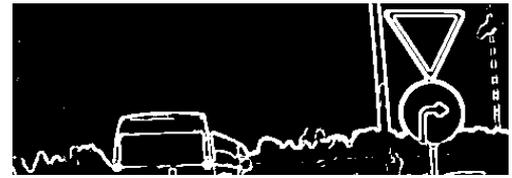
Original

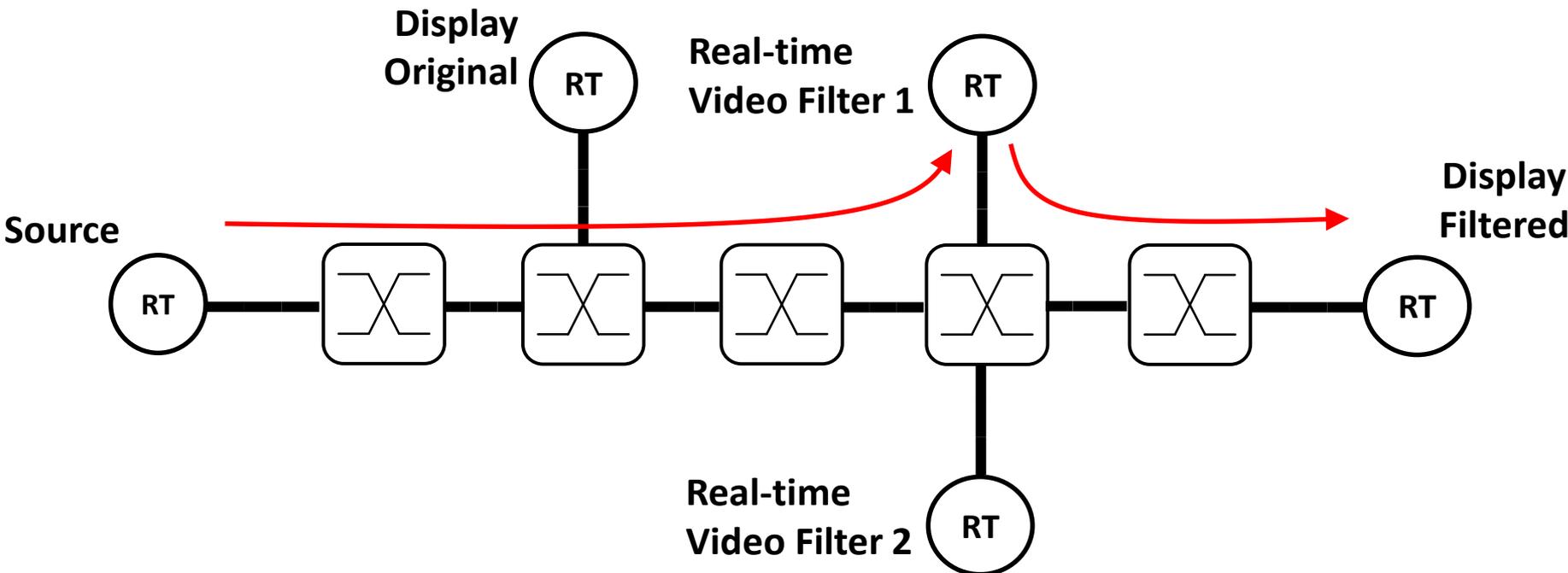


On-the-fly processing



Assembled Processed





Original

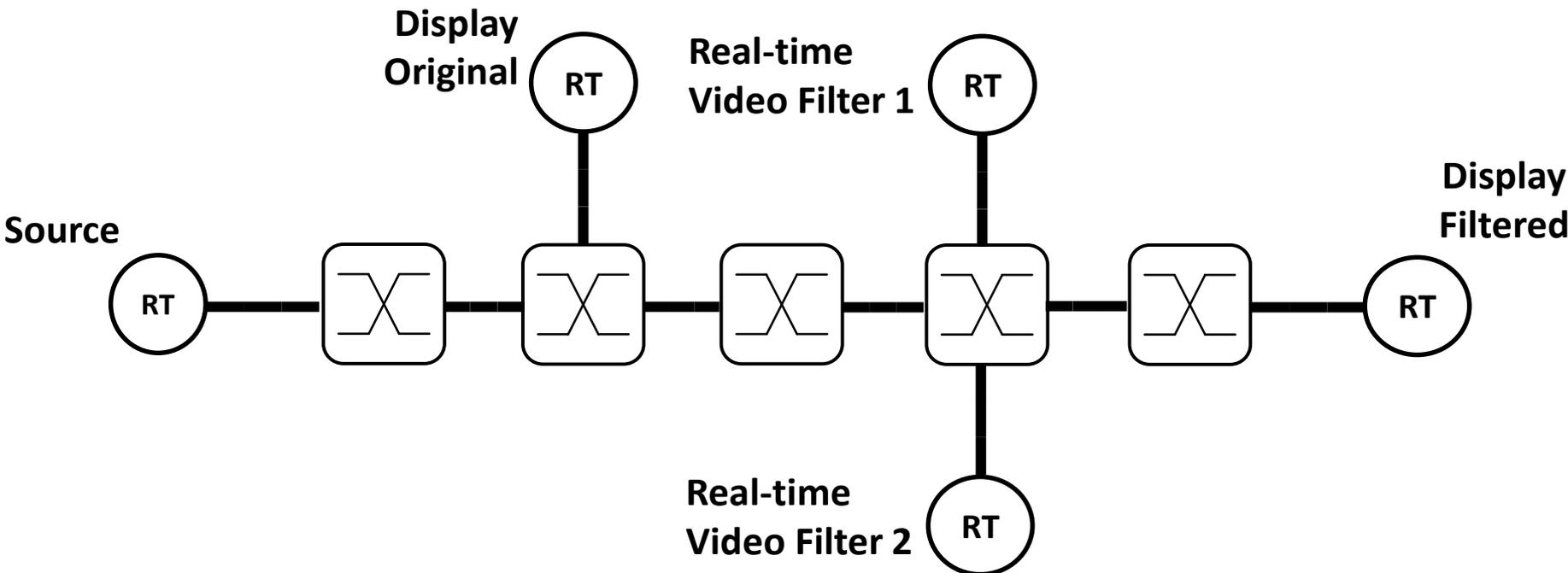


On-the-fly processing



Assembled Processed





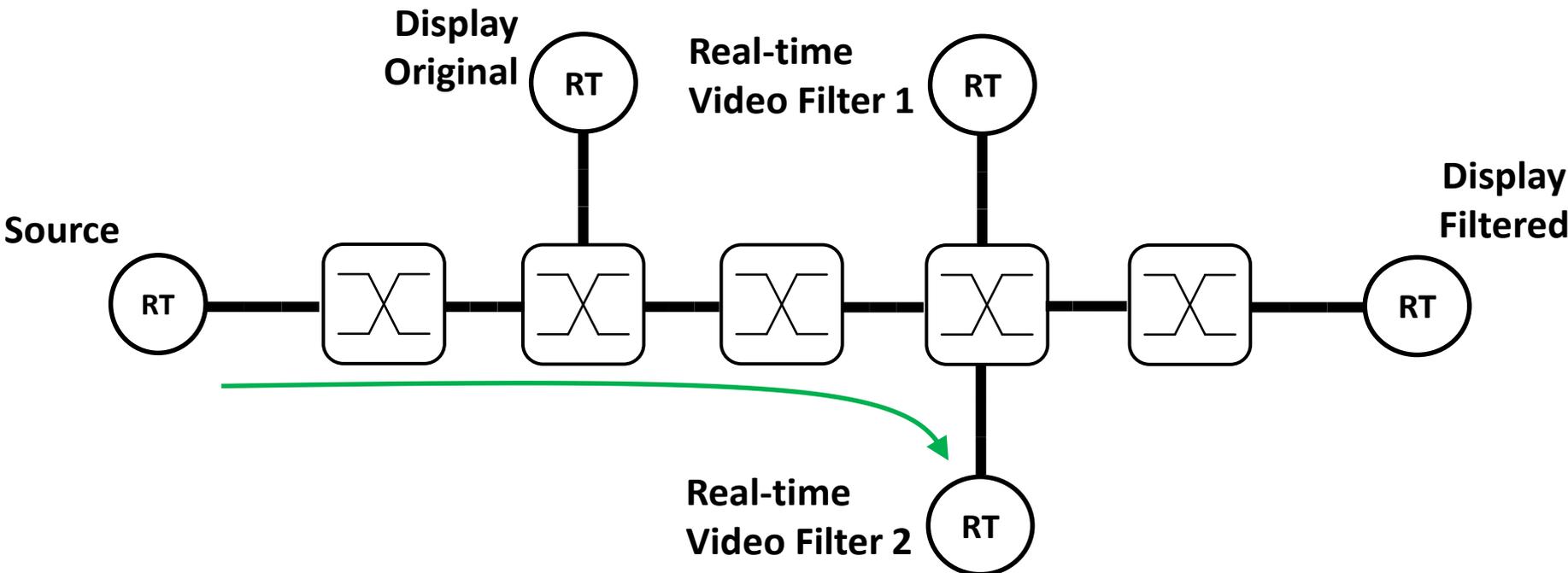
Original



On-the-fly processing

Assembled Processed



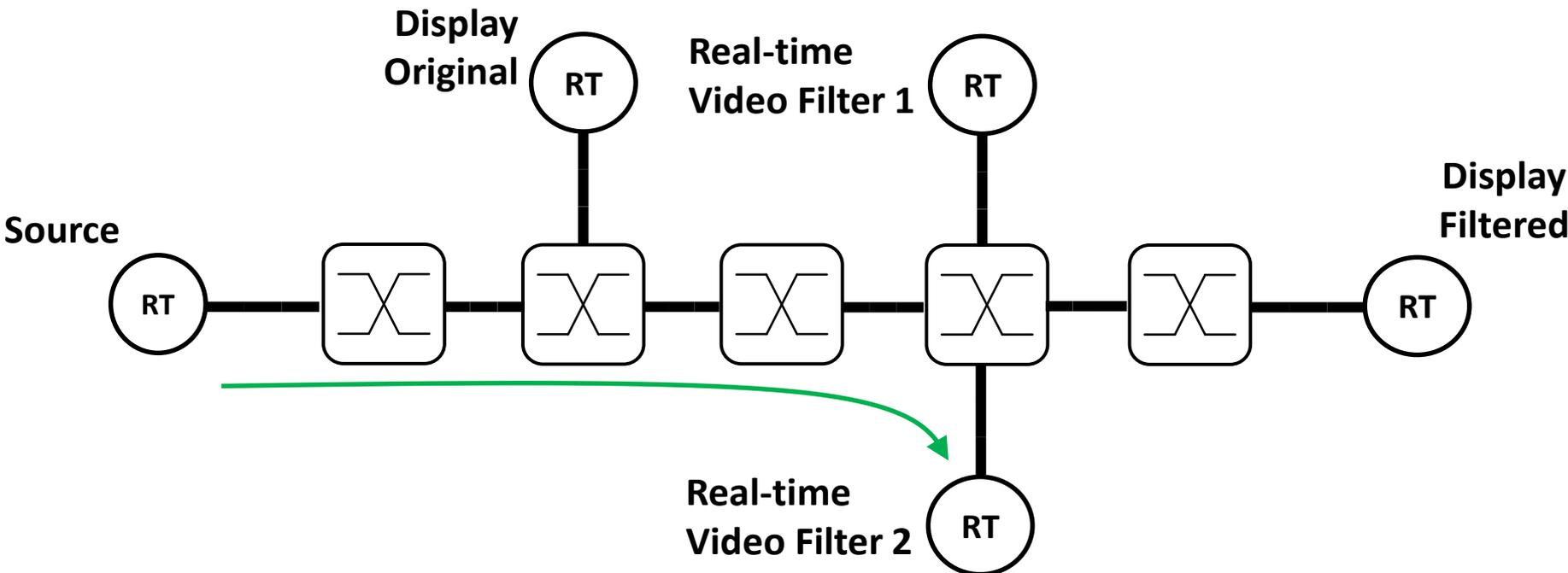


Original

On-the-fly processing

Assembled Processed



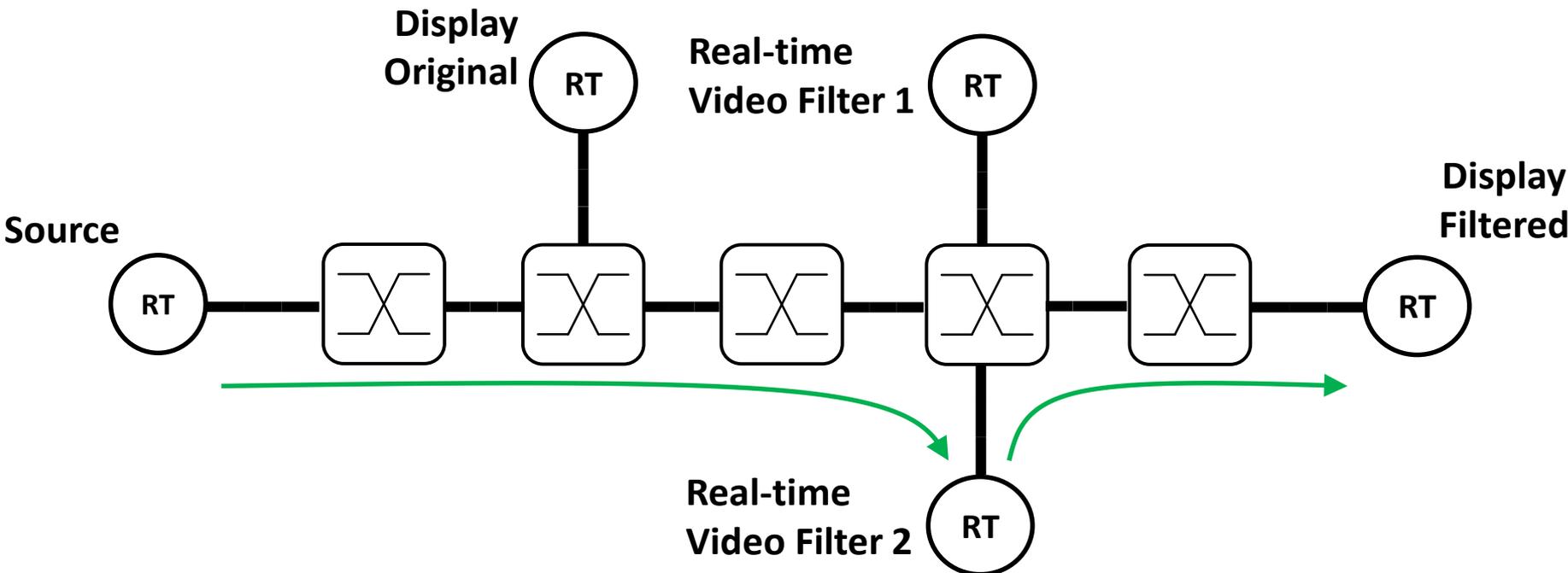


Original

On-the-fly processing

Assembled Processed

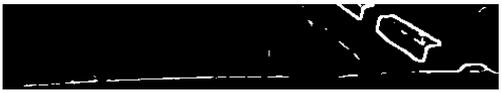




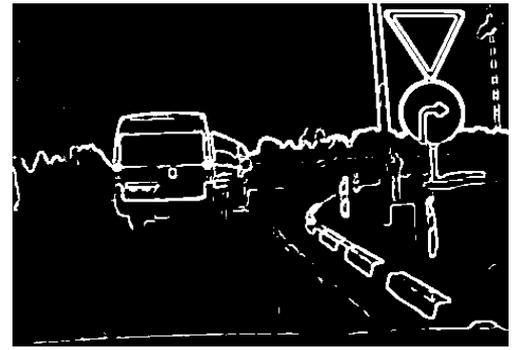
Original

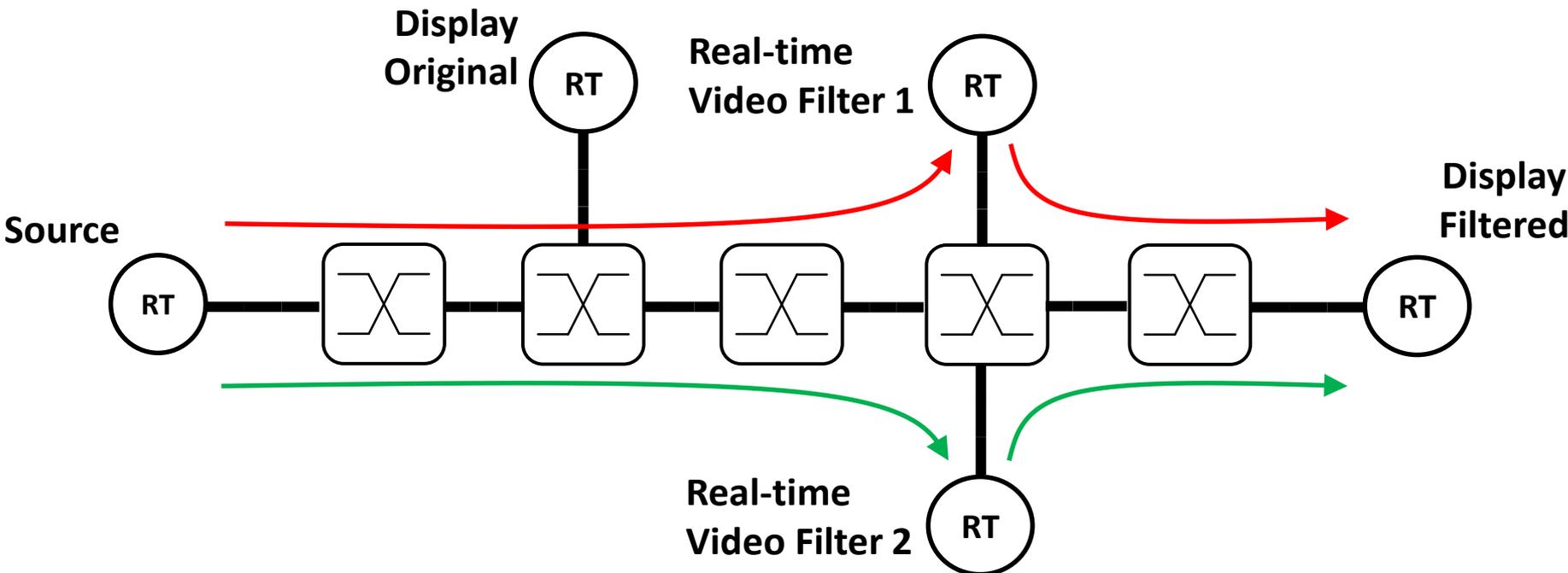


On-the-fly processing



Assembled Processed



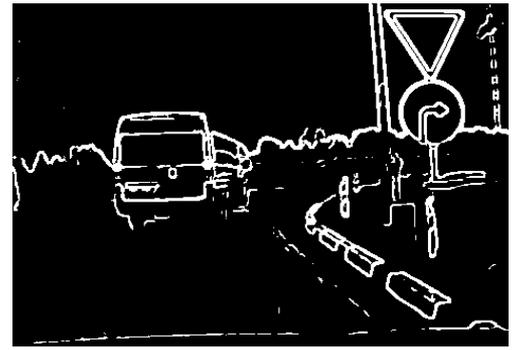


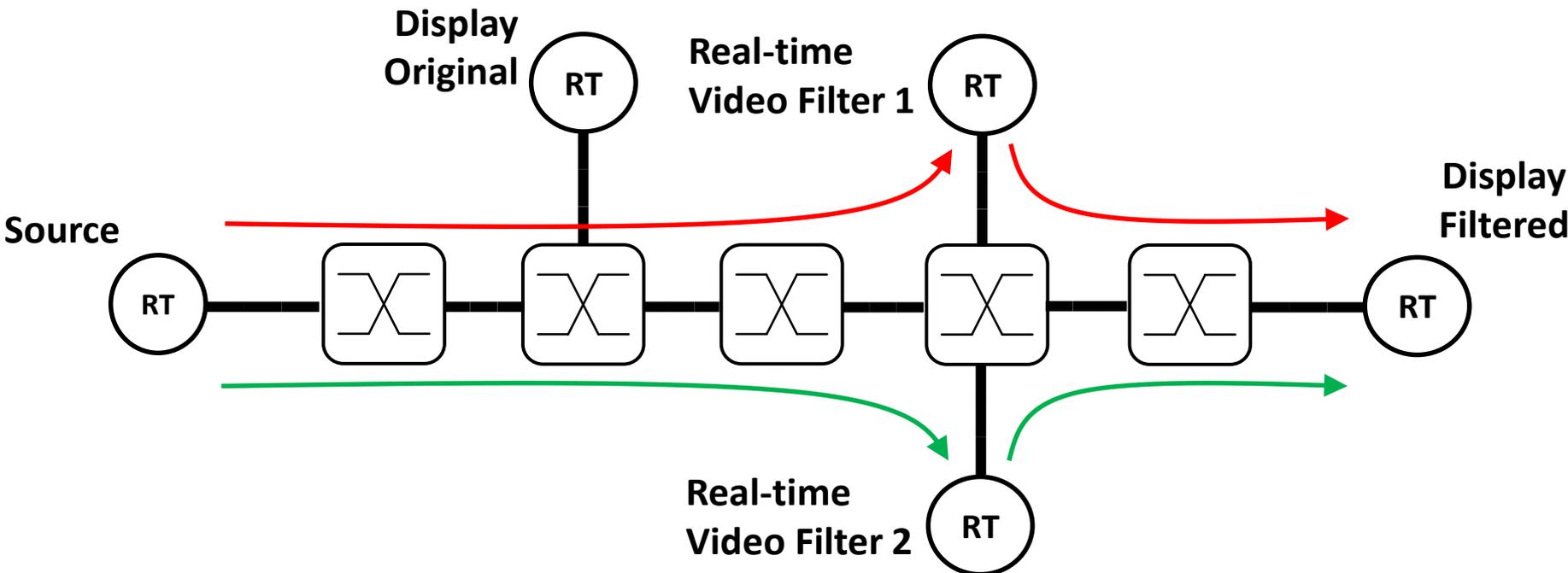
Original



Dynamic-TDMA schedules in real-time filters prevent broadcasting processed lines containing only black pixels

Assembled Processed



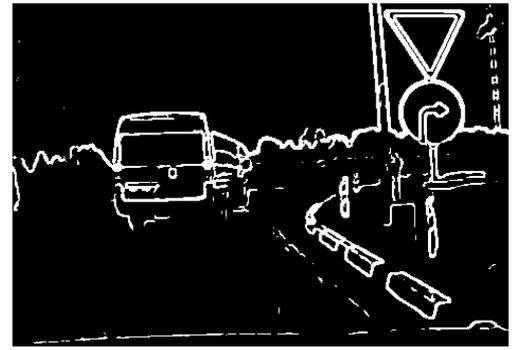


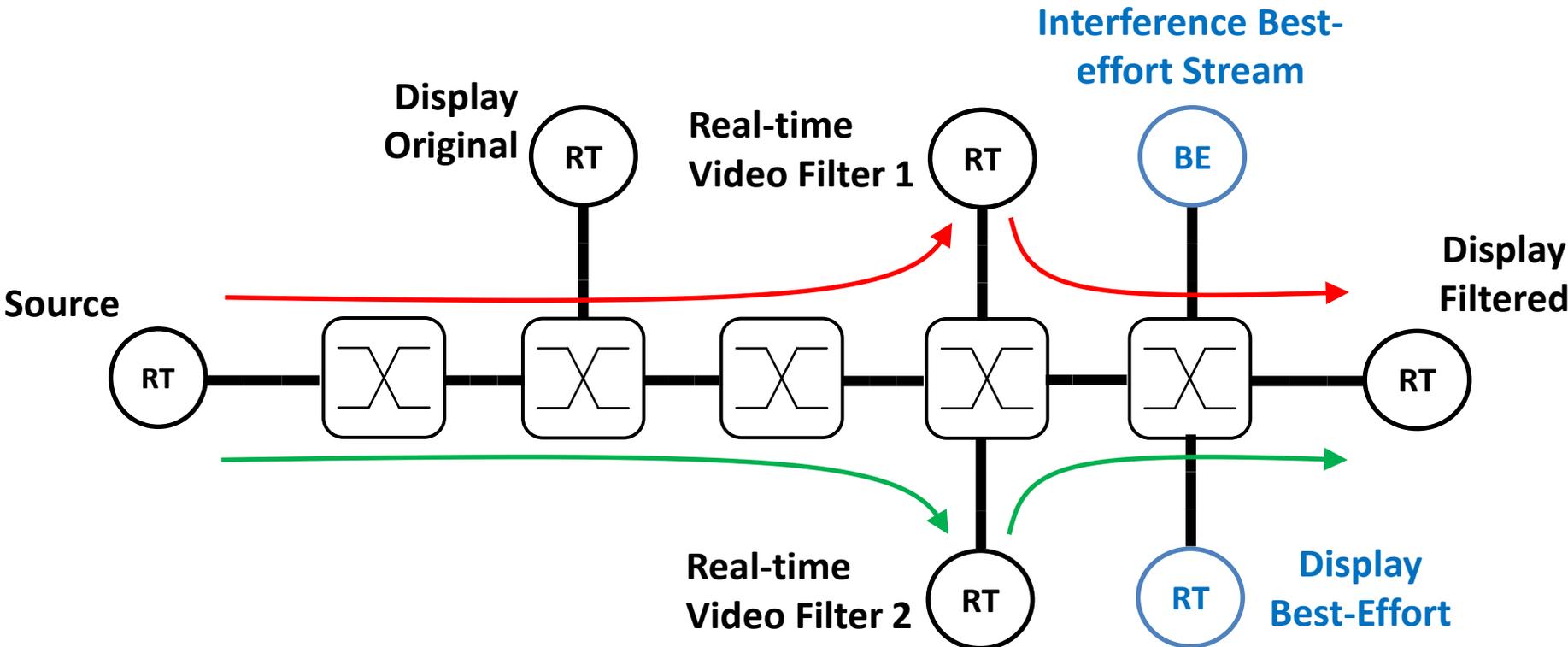
Original



The required bandwidth for transmitting the filtered video will depend on the content of the original video source

Assembled Processed



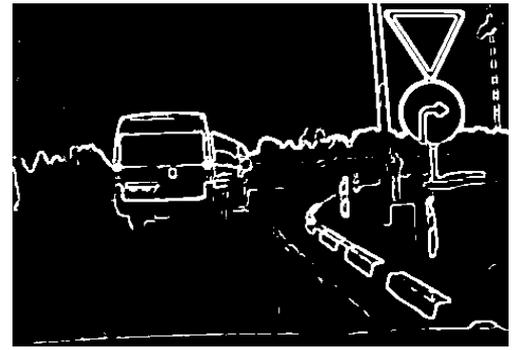


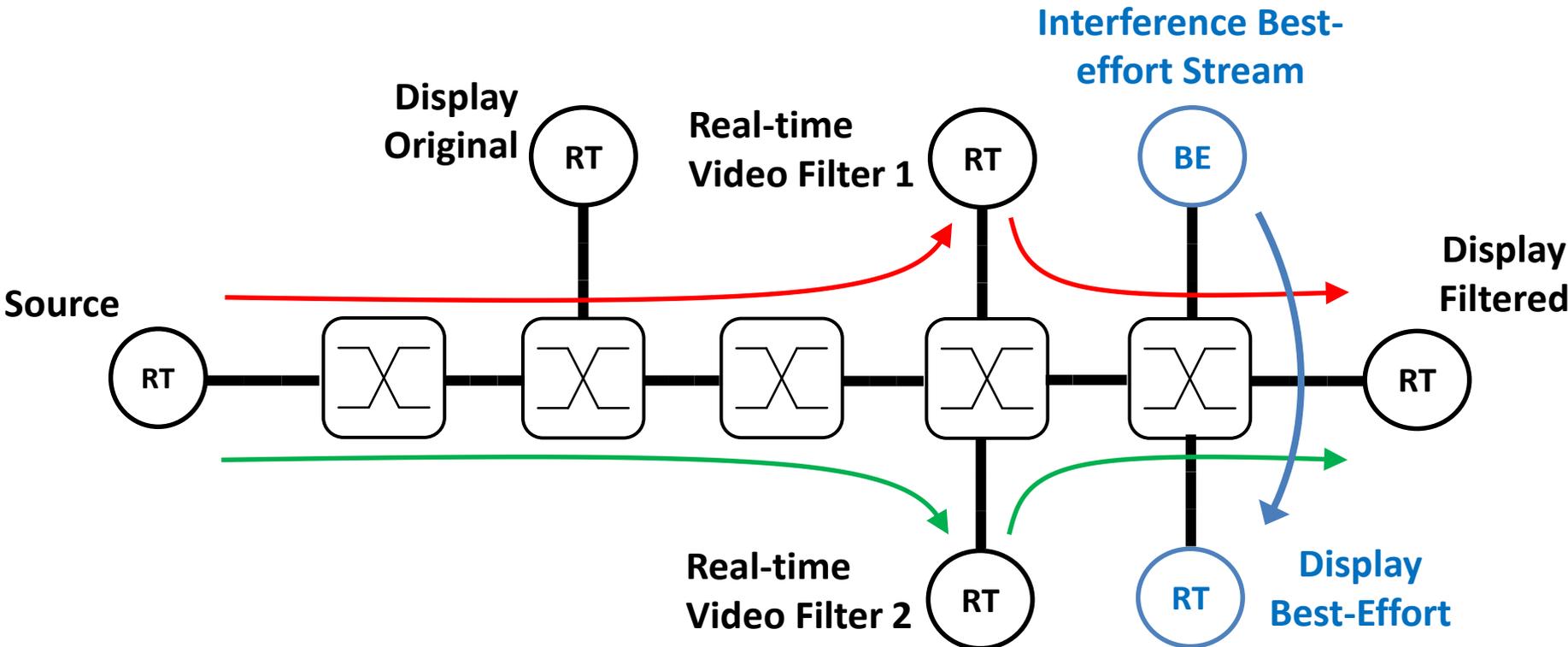
Original



A coexisting best-effort video stream propagate in the gaps between real-time frames flowing between filters and display

Assembled Processed



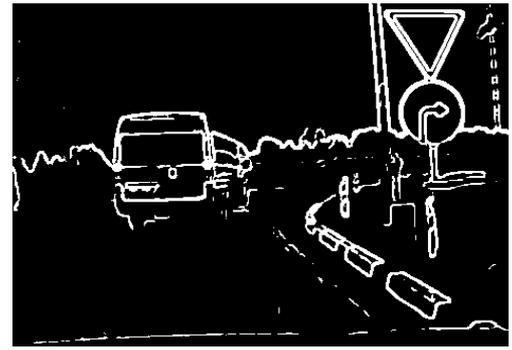


Original



The following example illustrates the effects of dynamic bandwidth management over the best-effort video

Assembled Processed



Experimental Setup (Video)

Play file `demo2-full-dynamic.mp4`

Closing Remarks

- *Atacama* is the first comprehensive open-source framework for Real-time Ethernet supporting dynamic-TDMA arbitration
- *Atacama* is an open-source research platform and easily extensible
- The framework intends to accelerate the development of new networking solutions for next-generation distributed systems

Credits

- Gonzalo Carvajal - Postdoctoral Researcher (UW, Canada)
- Luis Araneda - M.Sc. Student (UdeC, Chile)
- Alejandro Wolf - M. Sc. Student (UdeC, Chile)
- Miguel Figueroa – Assoc. Professor (UdeC, Chile)
- Sebastian Fischmeister – Assoc. Professor (UW, Canada)

For more information, related documentation, technical details, and access to the code, visit

<http://esg.uwaterloo.ca>

Thanks for Watching