University of Stuttgart Institute of Parallel and Distributed Systems

# NeSTiNg A Network Simulator for Time-sensitive Networking

**I**PVS

**David Hellmanns** 



- Our Motivation
- Discrete Event Simulation (DES): A primer
- Implementation
- Evaluation of Simulation Results
- Conclusion & Future Work

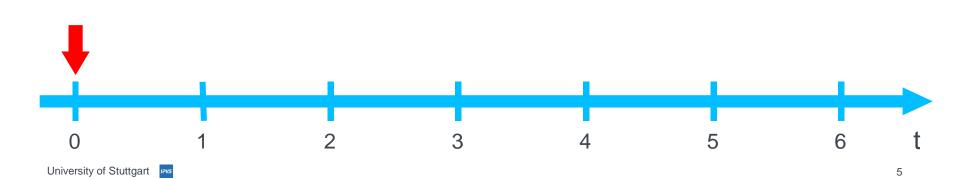
#### **Our Motivation**

- TSN evaluation for our research
  - NeSTiNg started as an internal tool for TSN evaluation
  - $\rightarrow$ Publication of code as a contribution to the <u>research</u> community
  - Simulation of functional behavior, not emulation of real systems

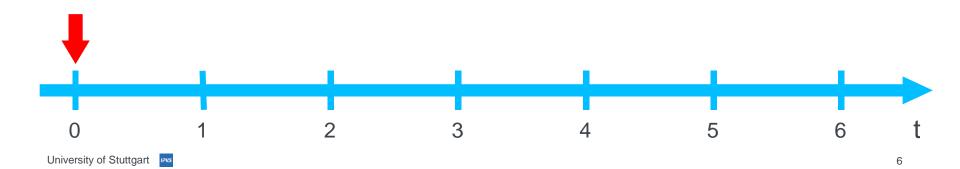
- Additional results of the development
  - Deeper understanding of standards
  - Awareness of edge cases and corner cases

Global simulation clock

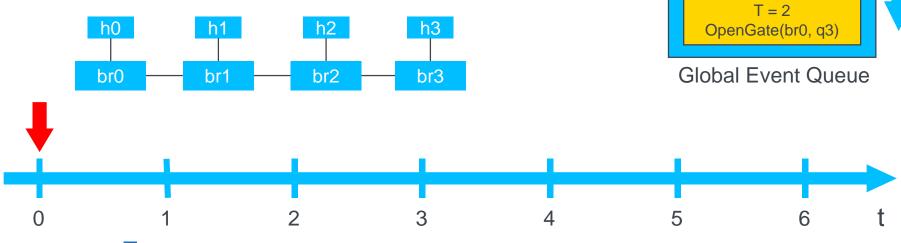
· Global simulation clock: Time domain is discrete



- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue

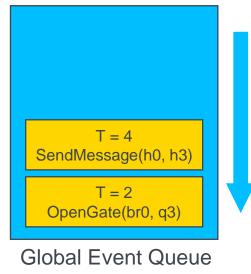


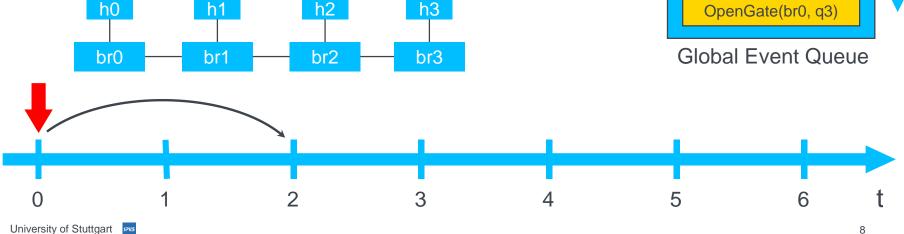
- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue



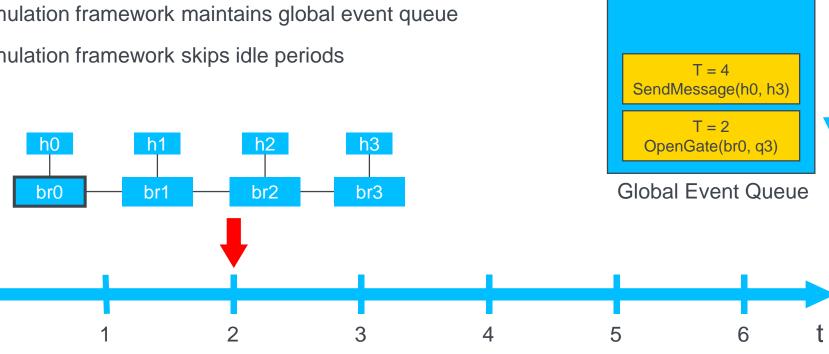
T = 4SendMessage(h0, h3)

- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods

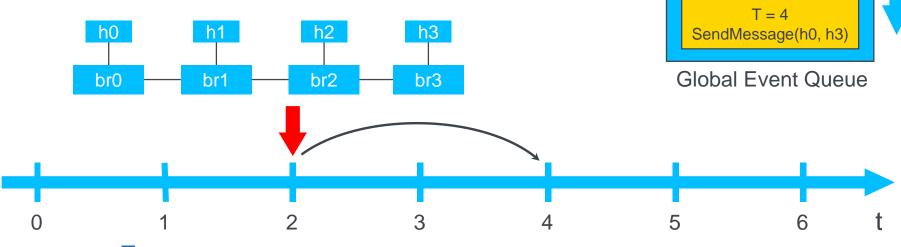




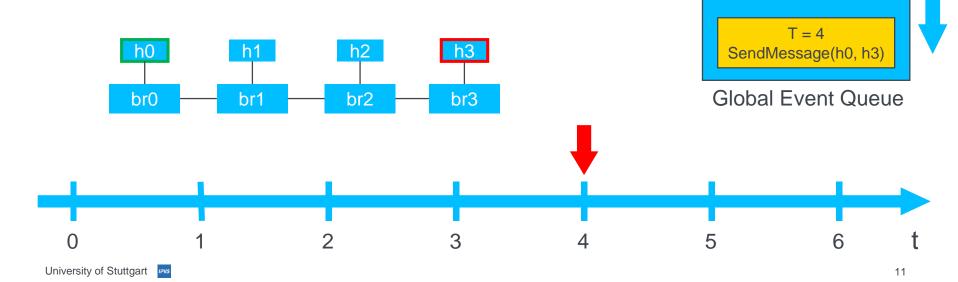
- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods



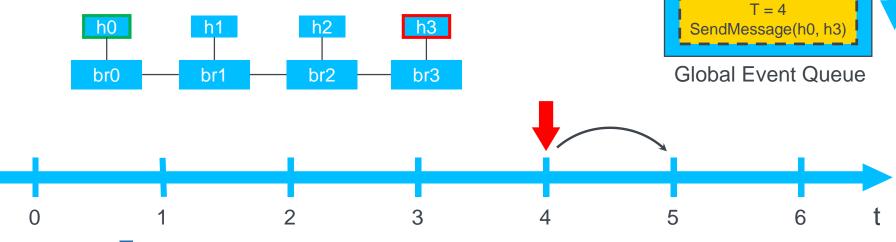
- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods



- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods

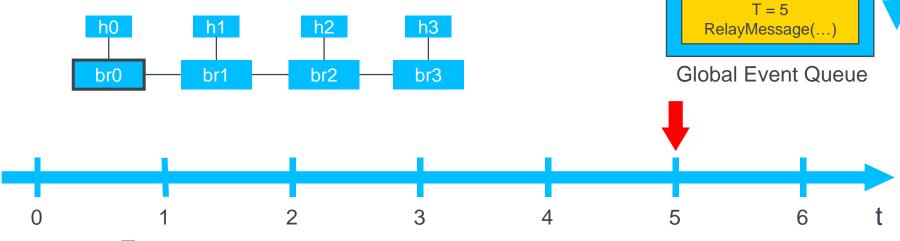


- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods
- Processing an event can create new events



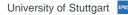
T = 5 RelayMessage(...)

- Global simulation clock: Time domain is discrete
- Simulation framework maintains global event queue
- Simulation framework skips idle periods
- Processing an event can create new events

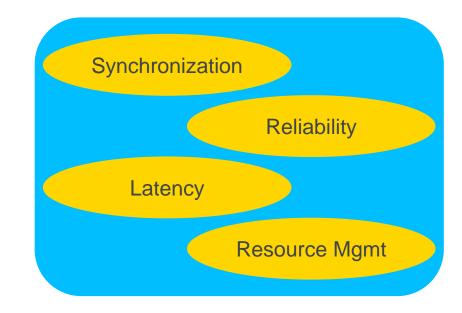


## **Building the Model**

- A simulation model:
  - Abstraction of the real system
  - Reduced to critical parts
- Which TSN components do we want to simulate?
- Which abstractions can we make for our TSN model?
- Which auxiliary components are necessary to simulate our TSN model?



# **TSN Components**

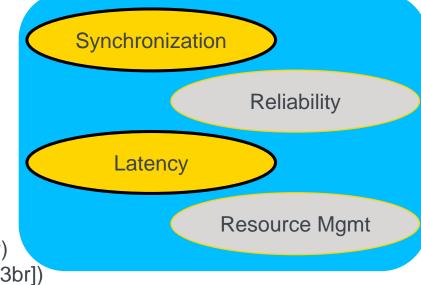


University of Stuttgart

15 Courtesy of János Farkas

#### **TSN Components**

Clock synchronization



- Latency
  - Scheduled Traffic (Qbv)
  - Credit Based Shaper (Qav)
  - Frame-Preemption (Qbu/[.3br])

# **Time-aware Shaping**

- Affected Systems:
  - Bridges
  - End stations
- Required capabilities:
  - VLAN  $\rightarrow$  PCP
  - Queuing
  - Gate Control
  - Scheduling
- Prerequisite: Time synchronization of network

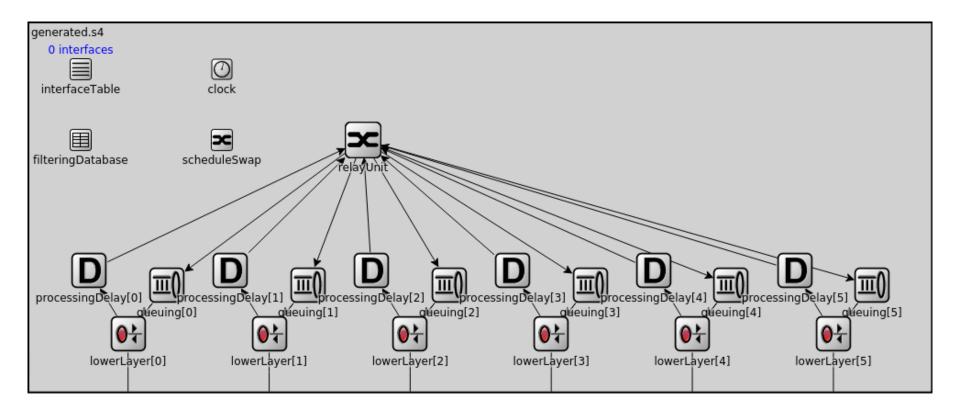
## **Building Block: INET Framework**

- "An open-source OMNeT++ model suite for wired, wireless and mobile networks."
- We do not want to reinvent the wheel
- Already implements
  - Ethernet channels and Ethernet components
  - Higher Layer protocols
- Compatibility to INET allows simulation of converged networks by utilizing its higher layer components

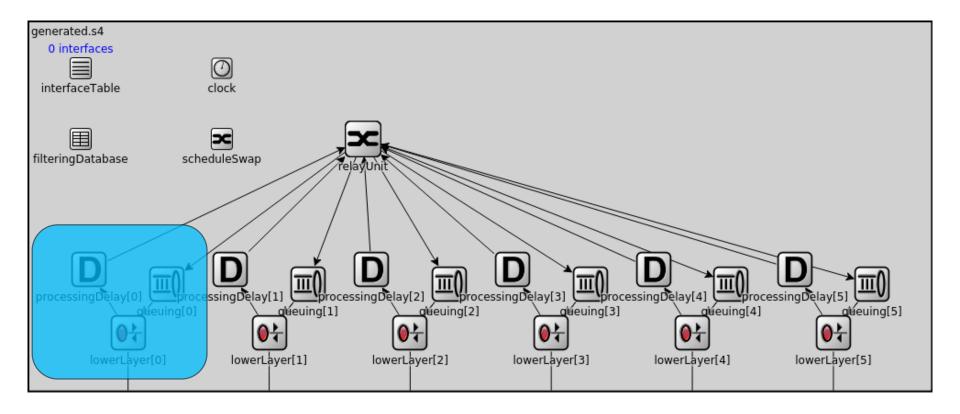
## **Time Synchronization of Network**

- Global simulation clock could be used to synchronize clocks of Bridges and End systems
  - Global simulation clock cannot be used to simulate drift, jitter or PTP
- $\rightarrow$ Interface for a user-defined clock model
  - · Currently only ideal clock is implemented
- User-defined clock may not simulate every tick
  - $\rightarrow$  only interesting time points are being simulated (DES)
- Interested components can subscribe to ticks
  - Priority queue
  - Clocks notifies subscribed components if a tick occurs

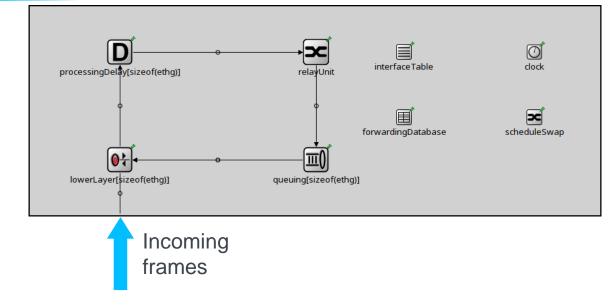
#### Implementation

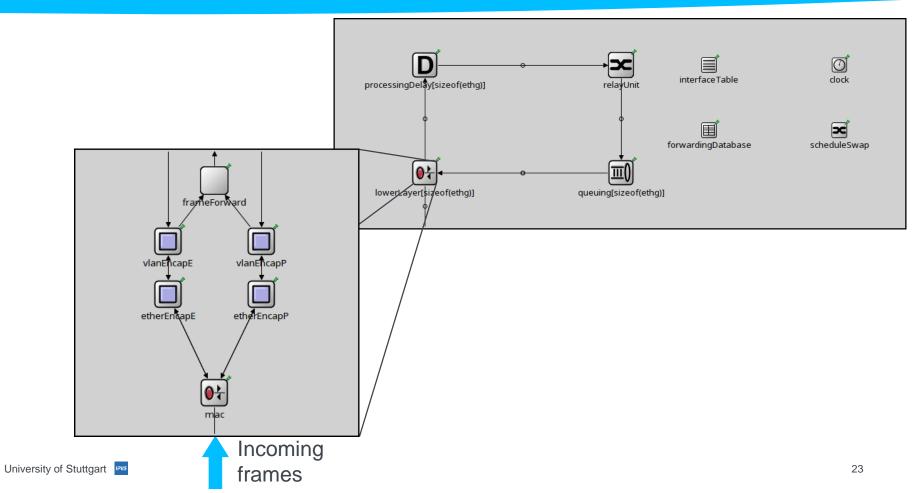


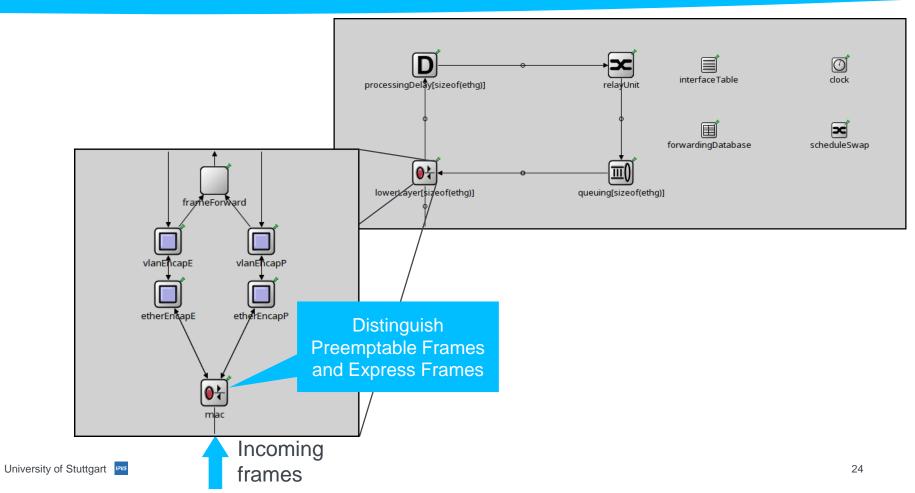
#### Implementation

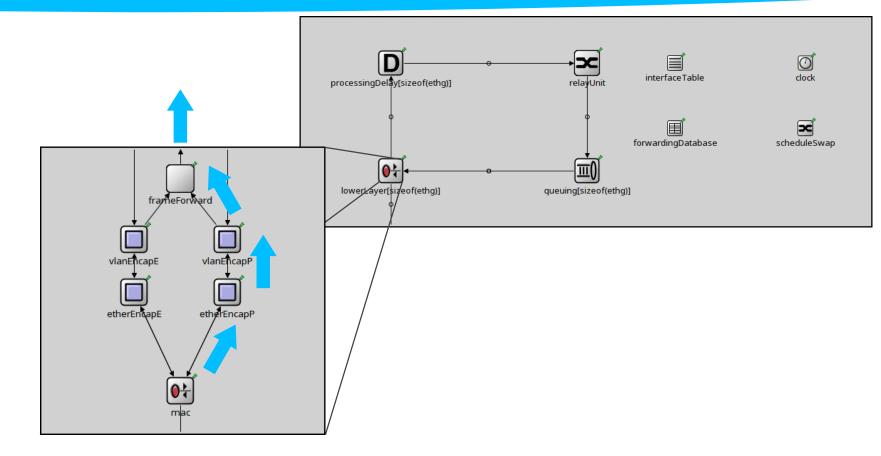


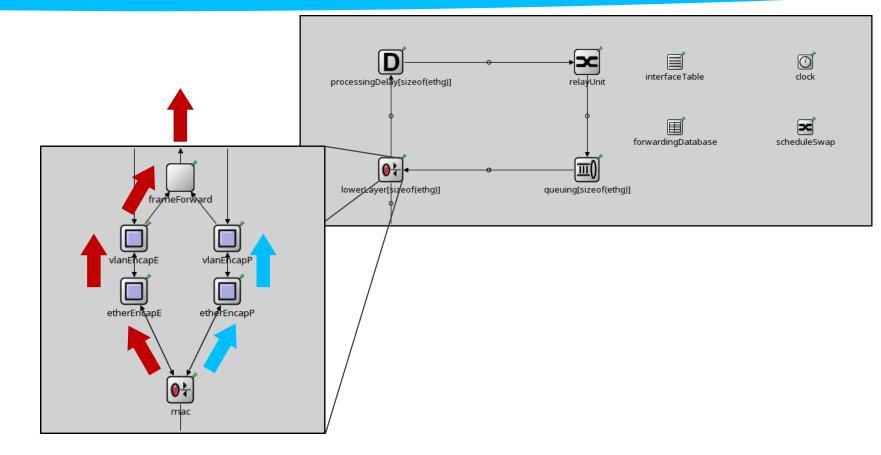
# Implementation

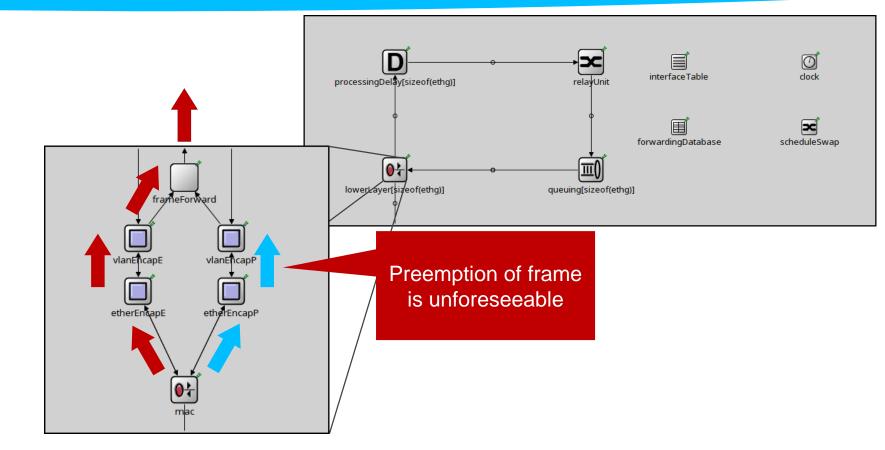


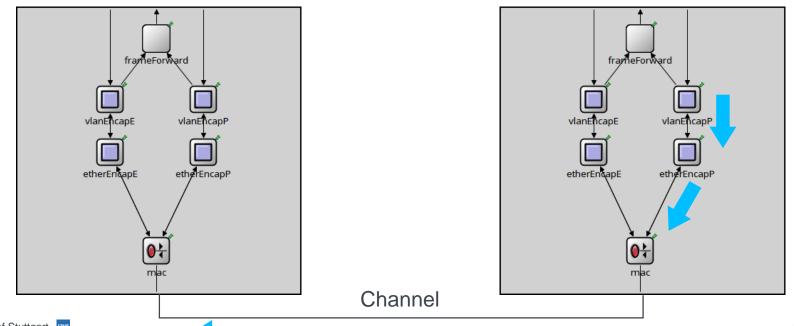


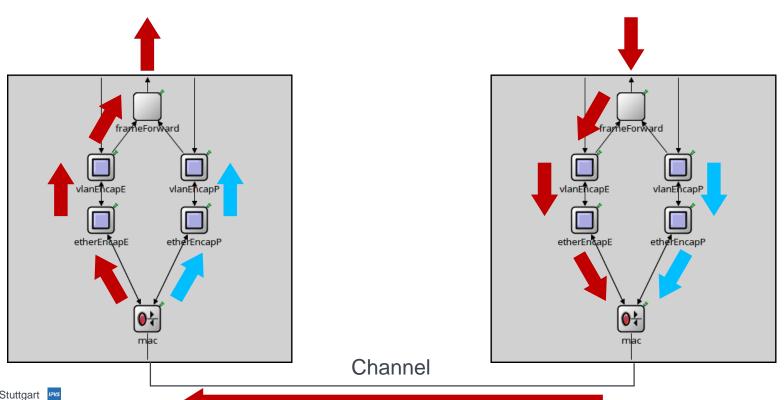


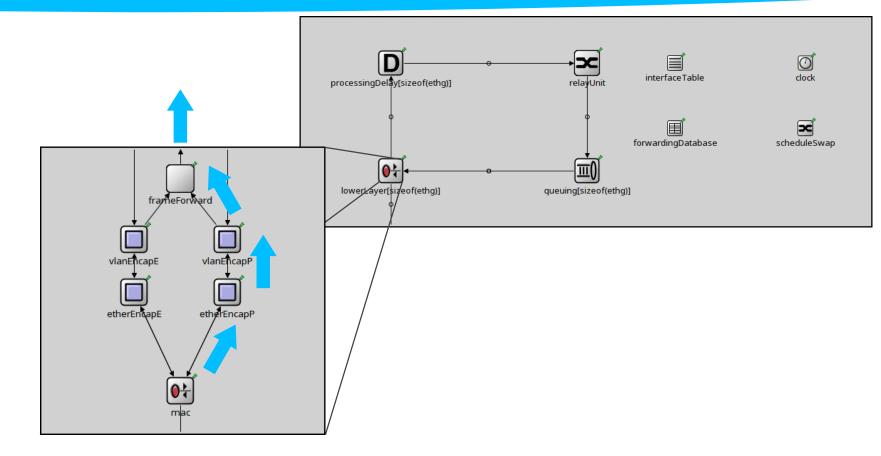




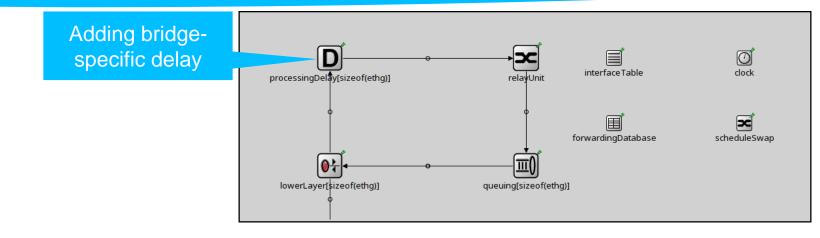




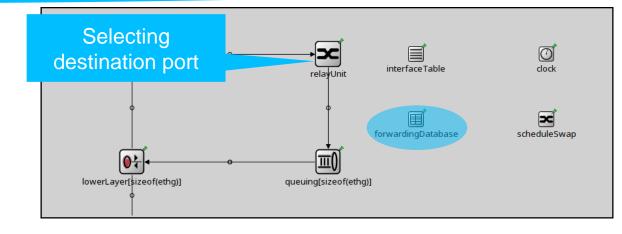




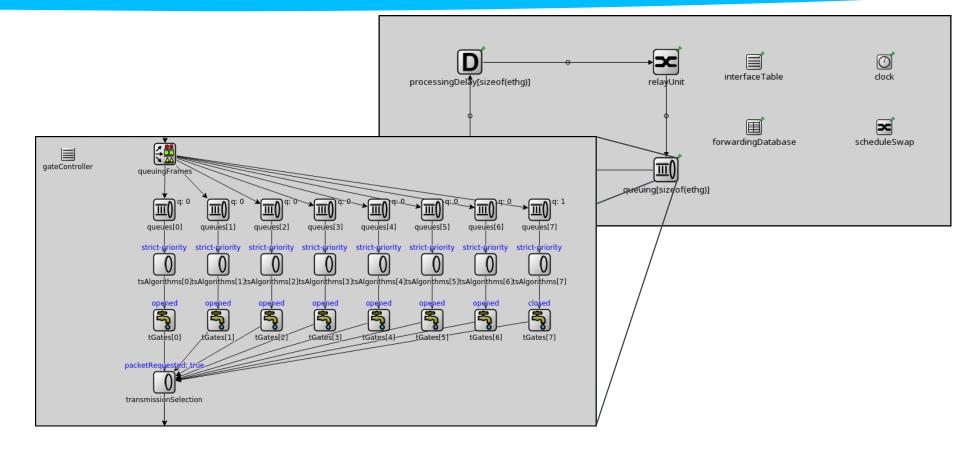
#### **Implementation: Processing Delay**



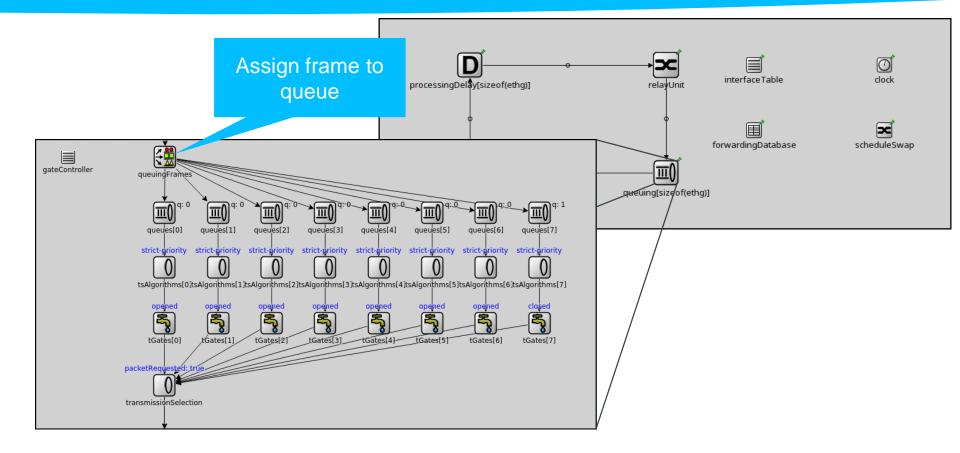
# **Implementation: Relay Unit**



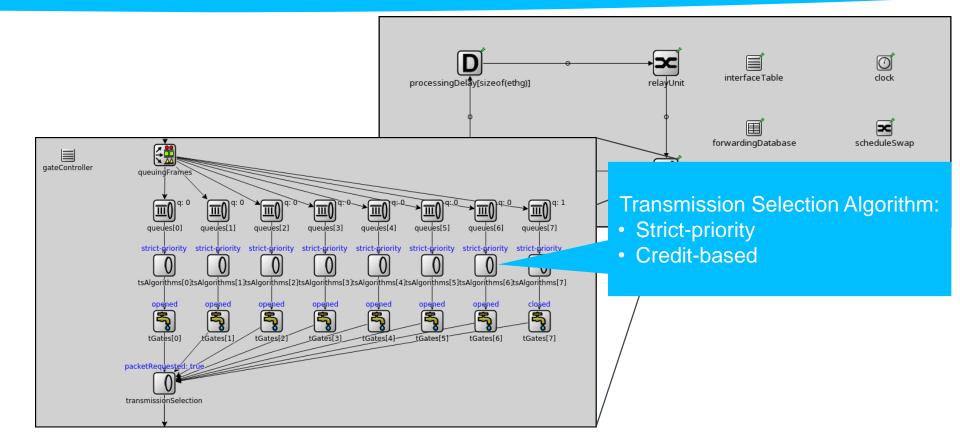
#### **Implementation: Queuing**



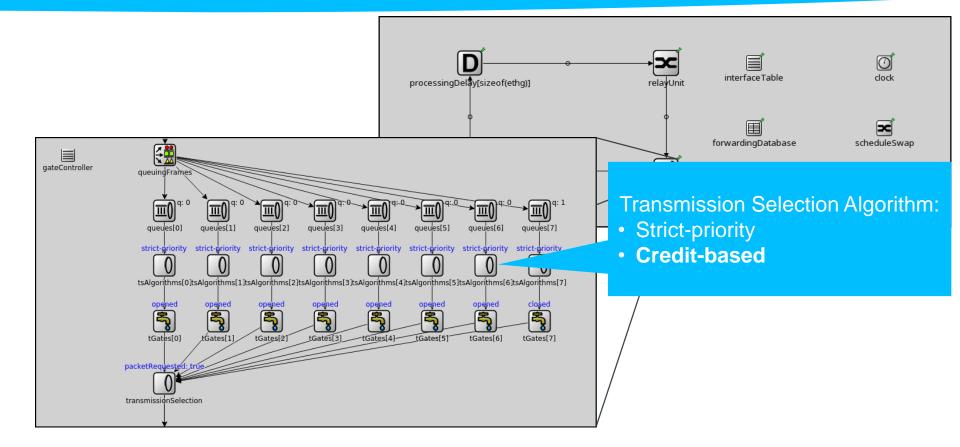
## **Implementation: Queuing**



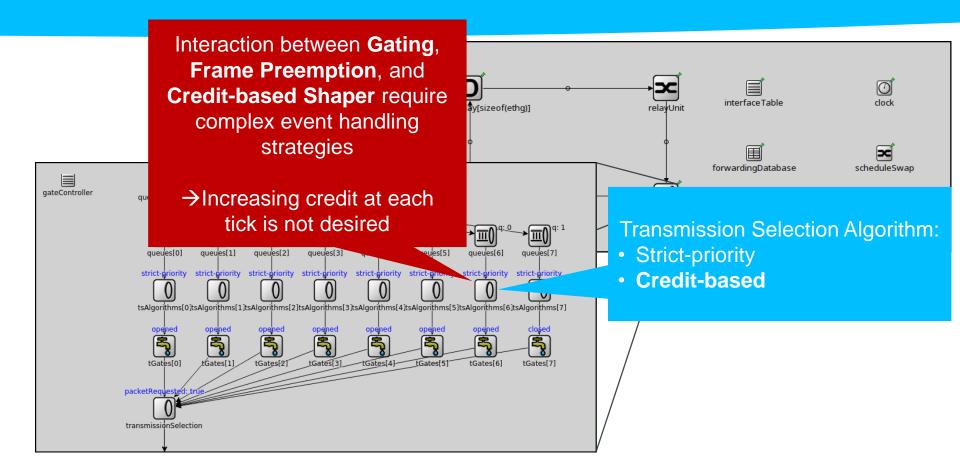
#### **Implementation: Transmission Selection Algorithm**



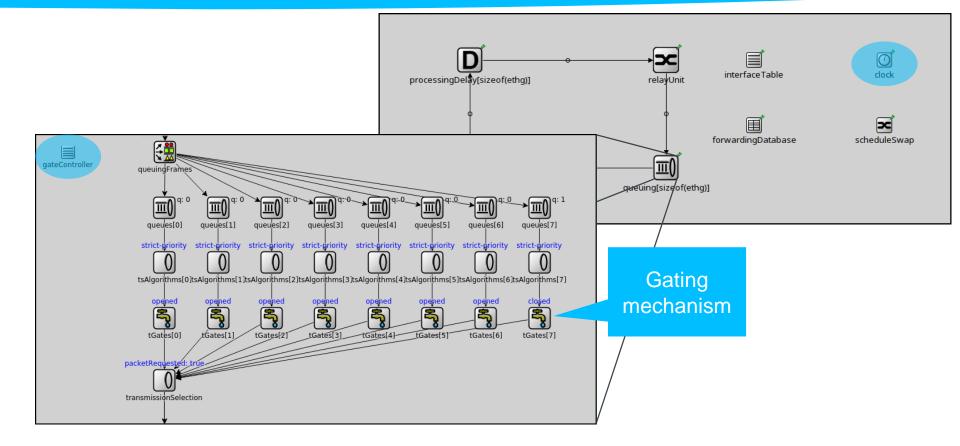
#### **Implementation: Transmission Selection Algorithm**

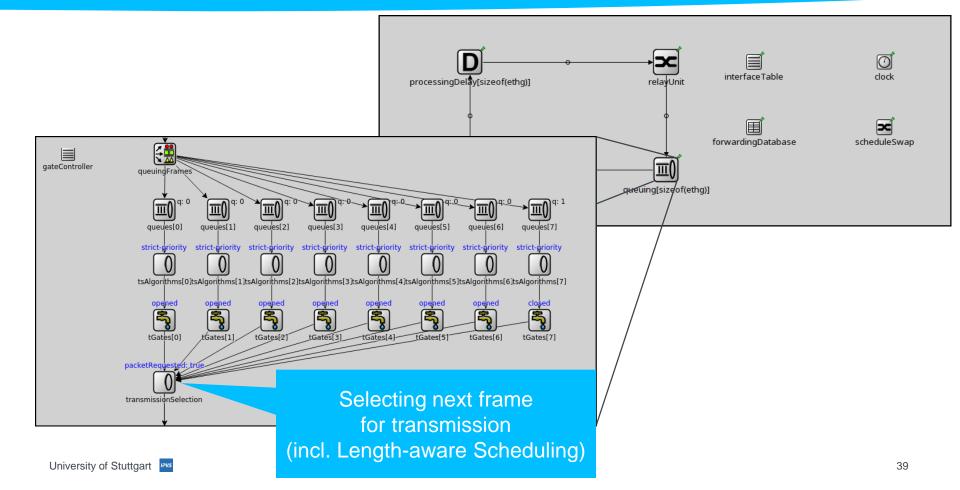


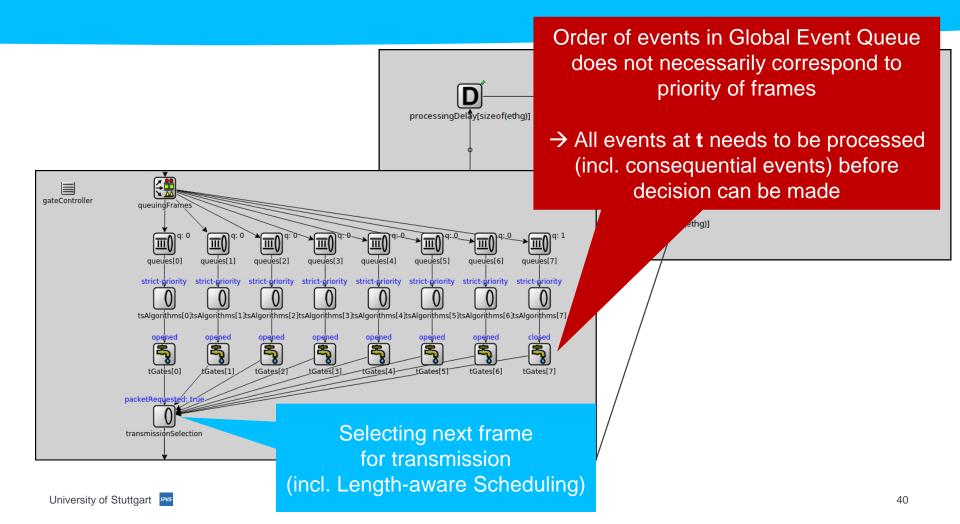
#### **Implementation: Transmission Selection Algorithm**

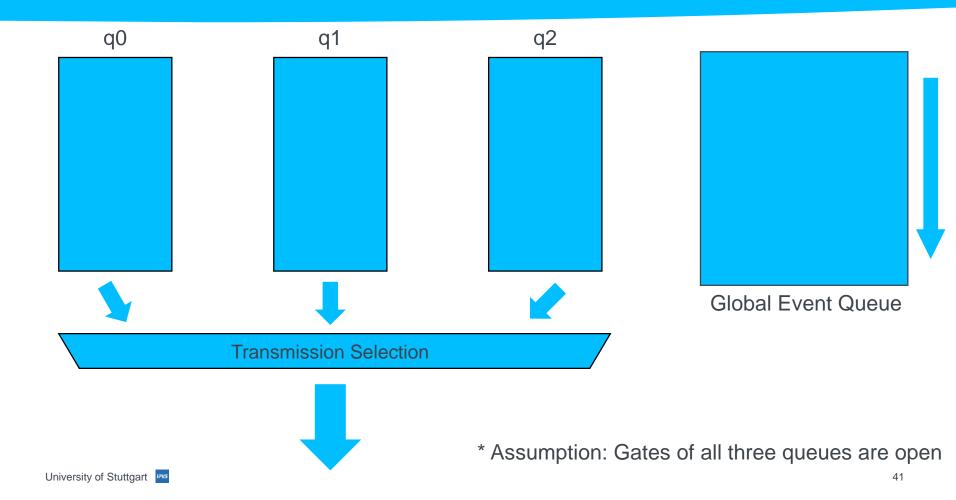


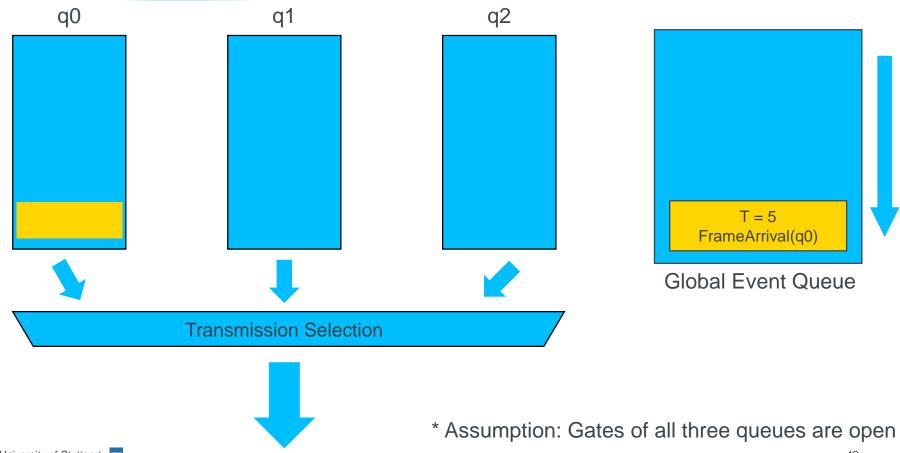
#### **Implementation: Gating mechanism**

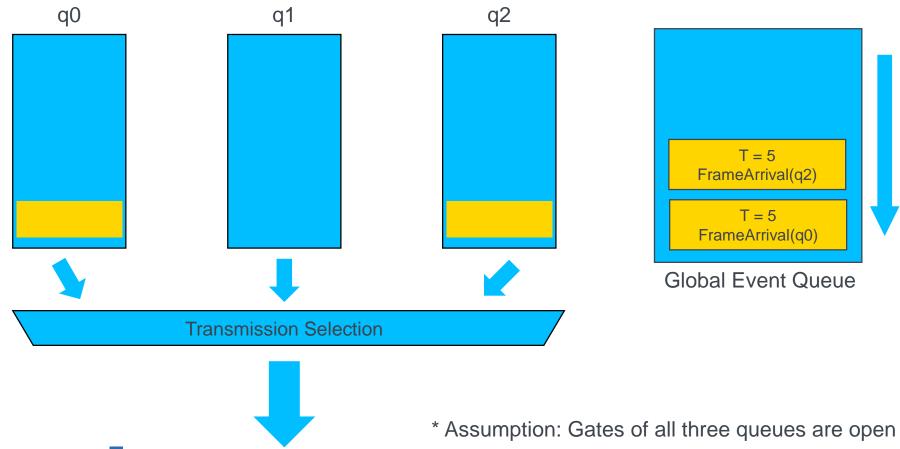


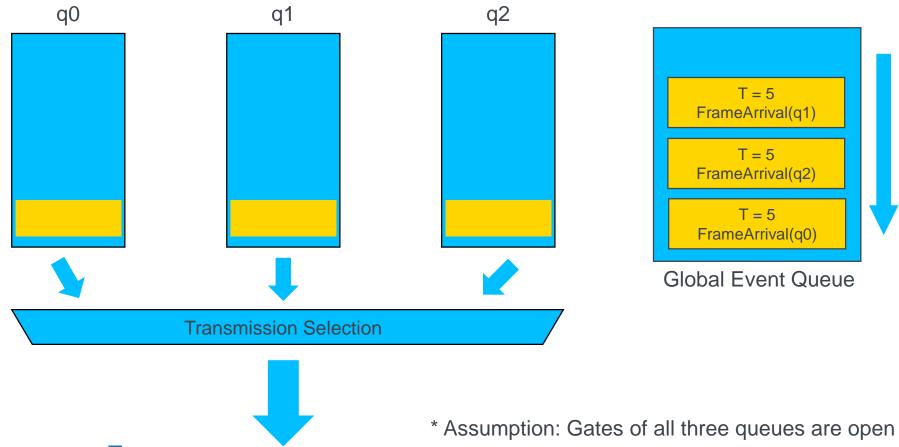




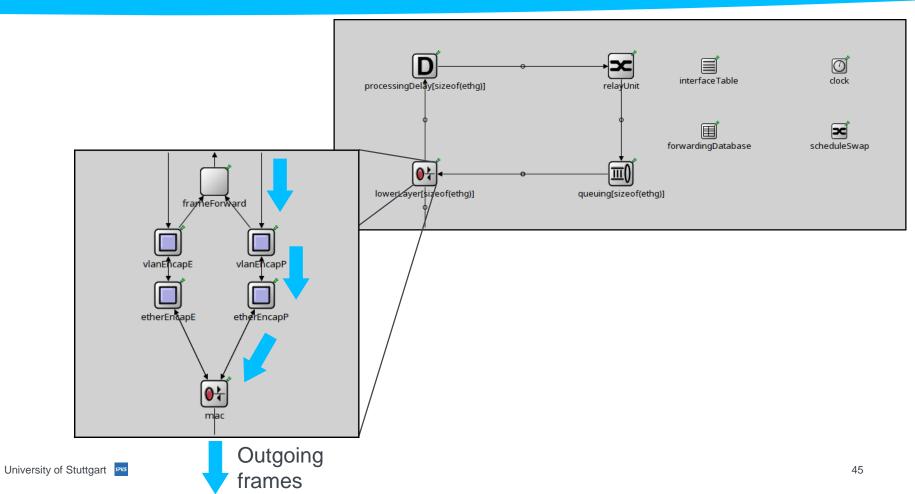




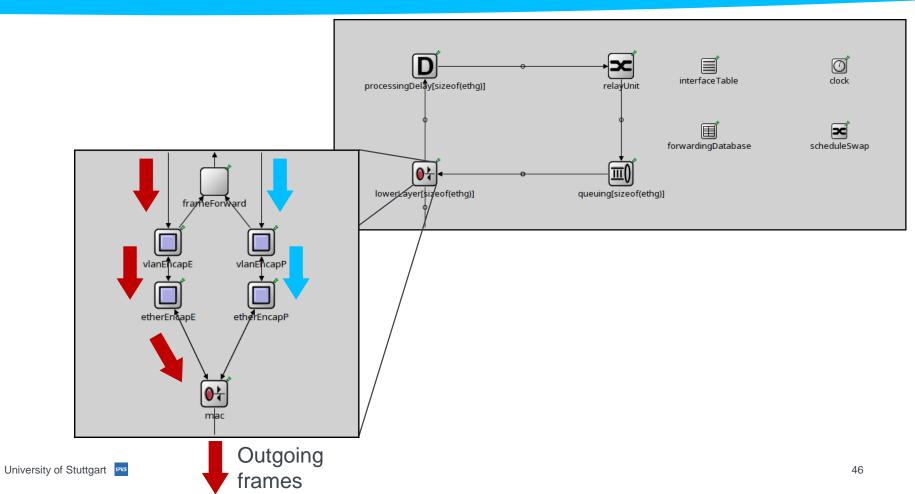




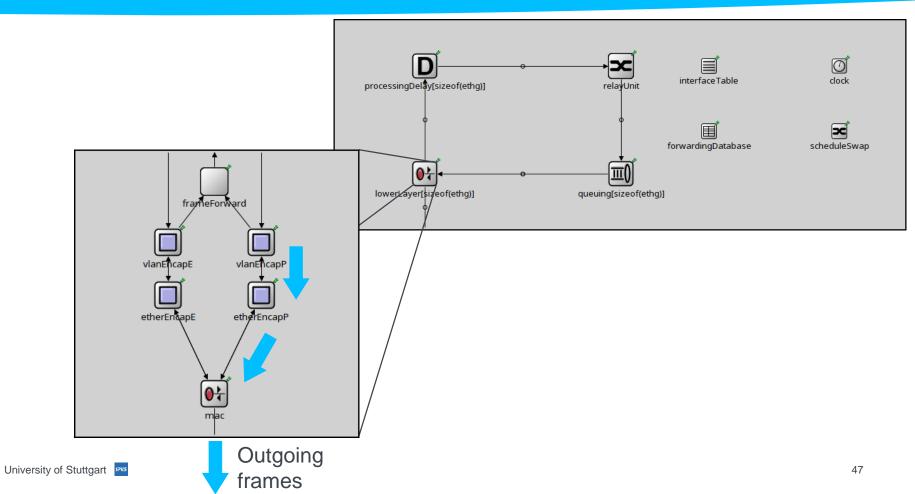
# Implementation: Frame Preemption (Outbound)



# Implementation: Frame Preemption (Outbound)



# Implementation: Frame Preemption (Outbound)



#### **Analysis of Simulation Results**

- OMNeT++ logs general data automatically
  - For special interests (e.g. deadlines) additional code is mandatory
- scavetool: An internal result analysis tool of OMNeT++
  - · Can be used to have a casual view of the data
  - Is capable of filtering, sorting, and joining data to prepare it for export
- Toolchain for extensive evaluation of results
  - scavetool for data export
  - (Excel)
  - Python SciPy Stack / R

## **Analysis of Simulation Results**

Generated.nsz_nii.tsn_app			
Generated.hs2 fl.etherEncap		pk-75-0	
Generated.hs2 fl.vlanEncap		#222st 0 EndTransr	nission EndIFG
Generated.hs2 fl.mac		#220	
Generated hs2 11.clock		#274	#32#330
Generated.hs2 fl3.tsn app		#267	#323/330
Generated.hs2_fl3.etherEncap	ok-81-0	#20/	pk-81-1
Generated.hs2_h3.elner.clcap	44433-0 EndTransrBissbitd		
Generated.ns2_n3.vianchCap			#323 EndTran
	#192		
Generated hs2_f13.clock			#328
Generated hs3_f1_clock	#191		#324
Generated.hs3_fl2.tsn_app			
Generated.hs3_fl2.etherEndae-93-0			
Generated.hs3_f12.vlanEncap_#633.0 EndTransmission EndIFG			
Generated.hs3_fl2.mac \$62			
Generated.hs3_f12.clock #64 #3309			
Generated.hs3_fl0.tsn_app #61			
Generated.hs3_f10.etherEncap		pk-99-0 pk-99-1	
Generated.hs3_fl0.vlanEncap		#249-0 EndTransmist #278-1 EndTra	nsmission EndIFG
Generated.hs3 fl0.mac		#245 #241	
-Generated.hs3 fl0.clock		#247 #25#3#576	#329 #336
Generated.s0.relayUnit		#244 #268	
Generated.s0.processingDelay[3]			#352
Generated.s0.processingDelay[4]			pk-93-0
Generated.s0.processingDelay[5]			ph-93-0
Concrete Society and Concrete Society (S)			#311 #351
Generated.s0.queuinq[0].queuinqFrances. #115 #157		#282	#342
Generated.so.queuing(0).queues(7)		#202	#372
Generated.so.queuing[0].tsAlgorithmsi2BtateChanged gateStateChanged			
Generated.s0.queuing[0].tGates[1] ##6StateChanged gateStateChanged			
Generated.s0.queuing(0).tGates(2) grteStateChanged gateStateChanged			
Generated.s0.queuing[0].tGates[3] gite3QateChanged gat StateChanged			
Generated.s0.queuing[0].tGates[4] gdt&EtateChanged gdt \$128Changed			
Generated.s0.queuing[0].tGates[5] gressateChanged gatestateChanged			
state and so queuing[0].tGates[5] gate State Changed gate state and gate state an			
Generated:s0.queuing[0].tGates[7] #74 #131			01-93-0
Generated s0.queuing[0].transmissionSeleetton #132			
Generated #99queuing[1].queuingframes			
Generated.s0.queuing[1].queues[7]			
Generated.s0.queuing[1].tsAlgorithms[7]		gateStateChanged	gateStateChanged
Generated.s0.queuing[1].tGates[0]		ateStateChanged	ateStateChanged
Generated.s0.gueuing[1]:tGates[1]		##32StateChanged	THE REAL PROPERTY IN THE REAL PROPERTY INTO THE REAL PROPERTY
Generated.s0.gueuing[1].tGates[2]		Contesting of the Changed	w#949KateChanged
Generated.s0.queuing[1]:tGates[3]		re#29AteChanned	A stateChanged
Generated.s0.queuing[1]:tGates[4]		d #233teChange	A the state of the
Cenerated.so.ueuminf11:tottes[5]		1 geoscie Change	detestateChanged
Generated.so.queuing[1].tGates[6]		n #200 eChange	n watechanged
16senerated:s0.queuing(1):tGates(7)		#291	#349
Generated s0.queuing[1].transmissionSelection		#292	#350
Generated.s0-typeuing[2].queuingFrames		#292	#30
Generated.s0.queuing[2].queues[7]			#Stattingueued
500ns 3us 4us 5us 6us 100r 8us	50us 201us 203us	300us 394us 395us 500ns 397us 5	is 430ps 369 398us s 398us 500ns 400us

### **Conclusion & Future Work**

- NeSTiNg implements a basic set of TSN synchronous shaping features
- Standard compliance is a key driver
- Publication of code as contribution to the research community
- Compatibility with INET allows simulation of converged networks
- Future Work
  - Migration to INET 4.0
  - Further extension of the model (e.g. Ingress Policing and Filtering)
  - Extensive evaluation
  - Validation
- <u>https://1.ieee802.org/protocol-simulations/</u>