Improvement of Schedulability Analysis with a Priority Share Policy in On-Chip Networks

Zheng Shi, Alan Burns

Real-time system group
The University *of York*

RTNS 2009

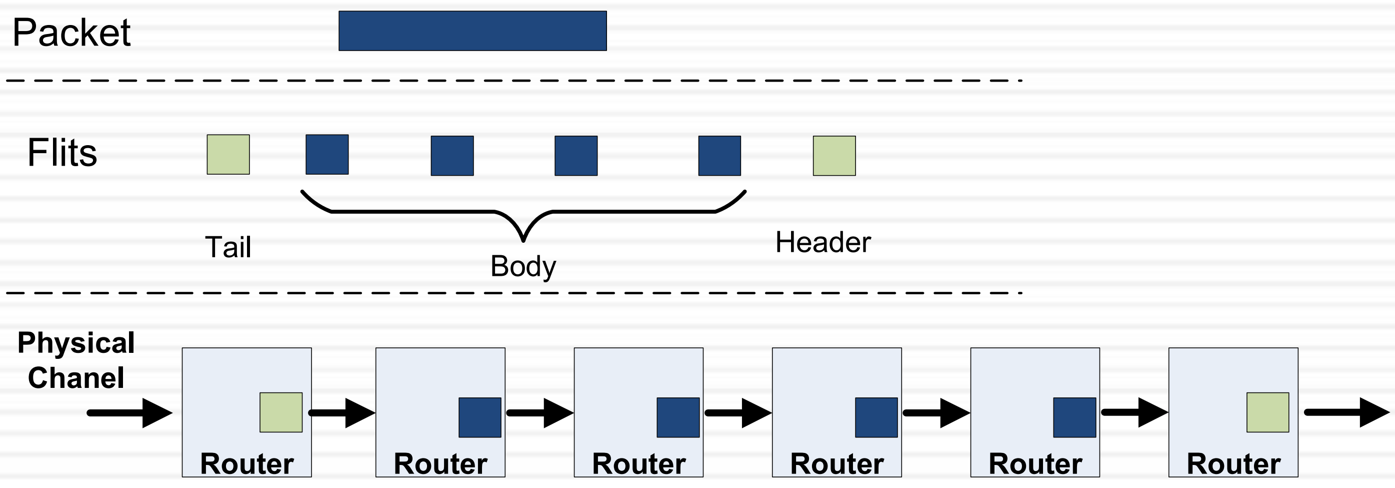
Outline

- Networks-on-Chip (NoC)
- Wormhole switching with priority sharing
- The problem of blocking
- Network latency analysis
- Summary

Introduction - NoCs

- ***NoC: packet-switched, shared, optimized for communications***
  - Resource efficiency
  - High scalability
  - IP reusability
  - High performance
- ***Real-Time Service:***
  - The correctness relies on not only the communication result but also the completion time bound (deadline).
  - For hard real-time service, it is necessary that all the packets must be delivered before their deadlines even under worst case scenario.

Wormhole switching



- Advantages (with Virtual Channels (VCs))
  - Small Buffer Size
  - High Average Throughput
  - Low Average Latency

Possible Solution

- ***Contradiction***
  - The network gives more efficiency and flexibility but introduces the unpredictable delay due to the contention. Real-time service, requires the timing to be predictable even under the worst case situation.
- ***Priority based Wormhole Switching***
  - Each flow has *distinct* priority and *exclusive* VC.
  - Deterministic routing from source to destination.
  - Support priority preemption.
  - Hard real-time communication can be supported.

Related Works

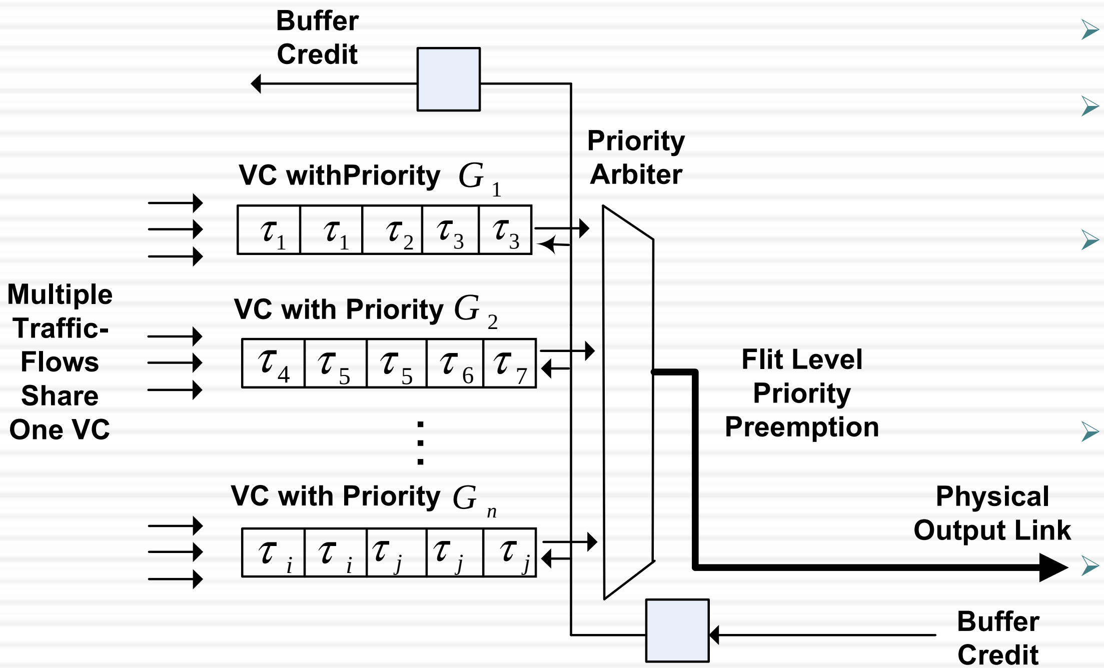
- Schedulablity Analysis
  - RM Utilization test [Mutka,96]
  - Lumped Link Scheme [Balakrishnan&Ozguner,98]
  - Blocking Dependency Graph [Kim&Lee,98]
  - Contention Tree [Lu&Jantsch,05]
  - Single Resource Scheduling [Shi&Burns,08]

Easy analysis but implementation cost is huge

Priority Share is a possible solution!

RTS York

Priority Share Router Structure

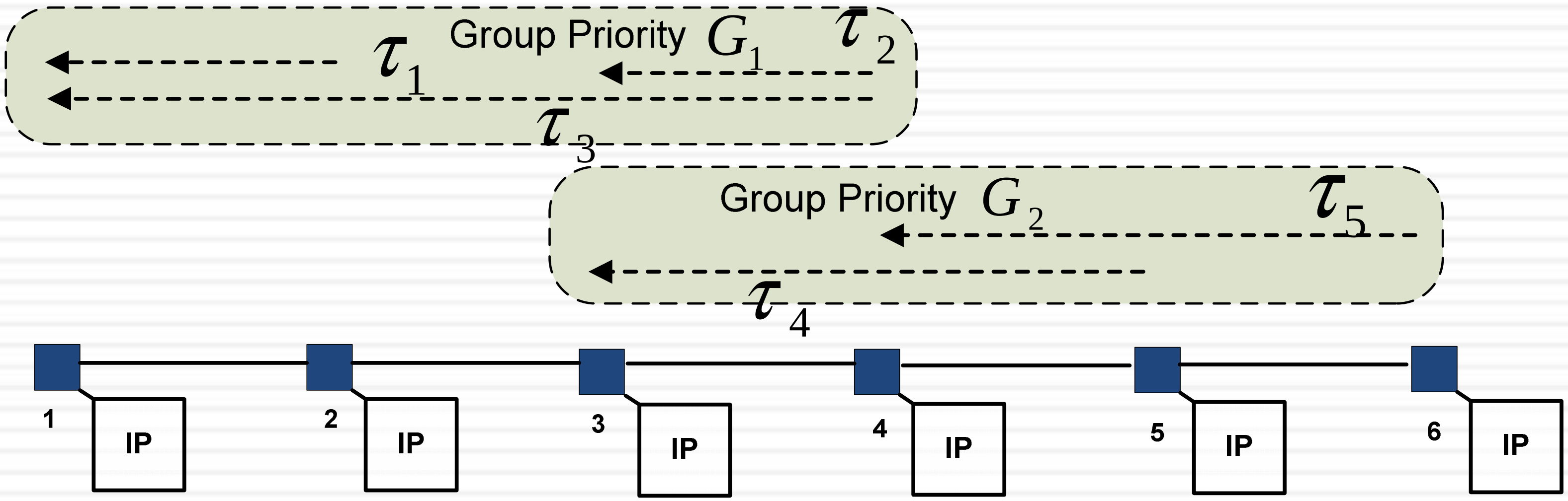


- There limited VCs at each port
- Each VC is assigned distinct global priority
- More than one flow can be allocated with same VC and have same priority level.
- Flow only requests the VC with same priority
- At any time, only the flit with highest priority and credit can access the output link
- Flit-level priority preemption between different VCs

Distinct v.s Share

- Distinct priority scheme: each flow has distinct priority level and exclusive virtual channel
  - The analysis only needs to consider the interference from higher priority flows
  - Implementation cost is huge
- Priority share scheme: more than one flow share a single virtual channel and assigned with same priority
  - Reduced cost
  - Introducing the blocking delay

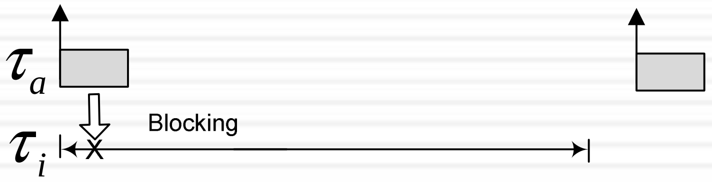
Blocking problem



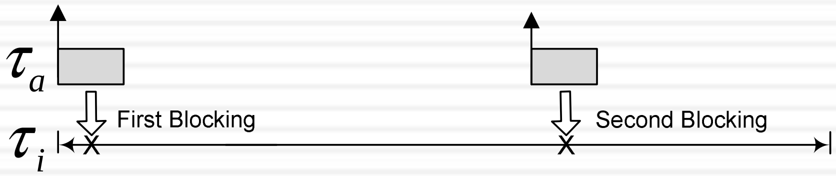
- The packet release from τ_4 could block τ_5.
- The packet release from τ_2 and τ_3 could pre-empt τ_5 and delay further.
- Activity from τ_1 also can introduce extra delay.

Three possible blockings

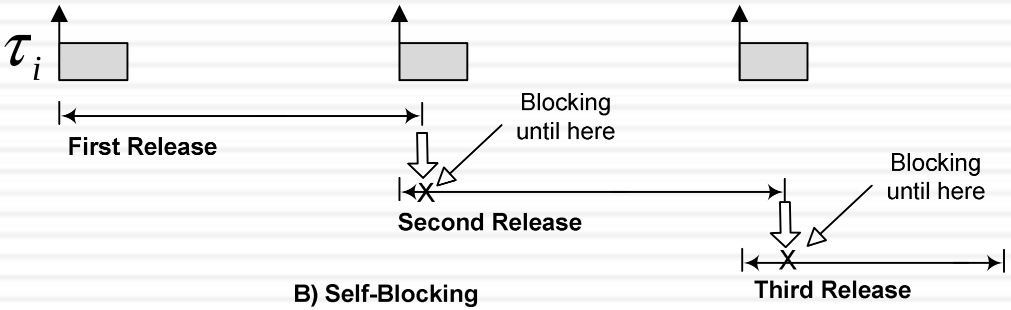
- Single blocking



- Multiple blocking



- Self-blocking



B) Self-Blocking

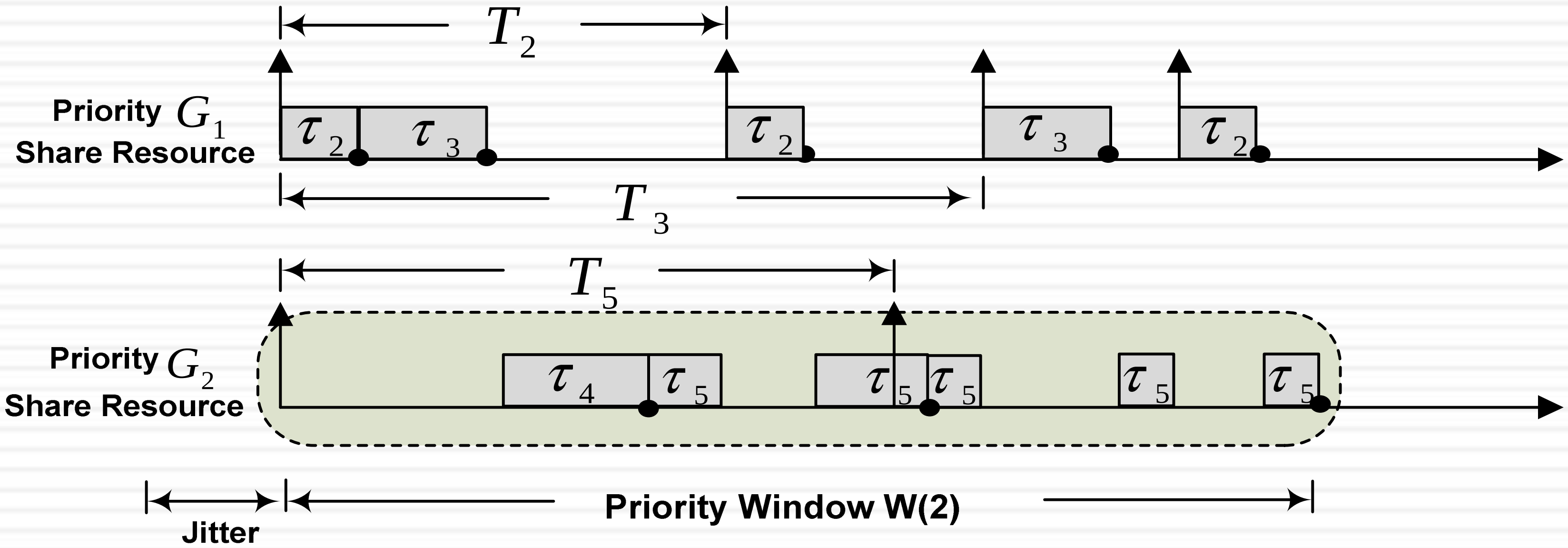
System Models

- Characterize traffic-flow

A traffic-flow is packet stream which traverses the same route from source to destination and requires the same grade of service.

- Attributes
  - G : Priority
  - C : Basic network latency
  - D : Deadline
  - T : Period for periodic flow or minimal interval for sporadic flow
  - J^R : Release jitter
- Interrelationship
  - Direct competing relationship
  - Indirect competing interference
- Set
  - S(i) is the set including all the flows sharing the priority G_i .

Priority window model



- Priority level G_i share resource: all the link resources required by the flows with the same priority $G_{i.}$
- Priority window W(i) a contiguous time interval during which is this priority level G_i keep the network busy and serves all the flows with priority higher or equal to G_i

Priority window

- The priority level G_i window W(i) upper bound

$$W(i) = E(i) + I(i)$$

- E(i): summation of service requirements generated by all the flows with the priority G_i

$$E(i) = \sum_{\forall \tau_m \in S(i)} \lceil \frac{W(i) + J_m^R}{T_m} \rceil \cdot C_m$$

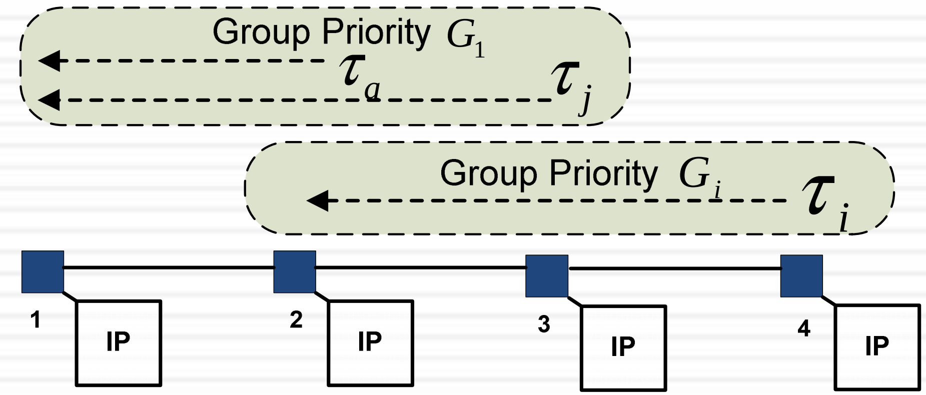
- I(i): summation of interference by all the higher priority flows

$$I(i) = \sum_{\tau_j \in hp(i)} \lceil \frac{W(i) + J_j^R + R_j - C_j}{T_j} \rceil C_j$$

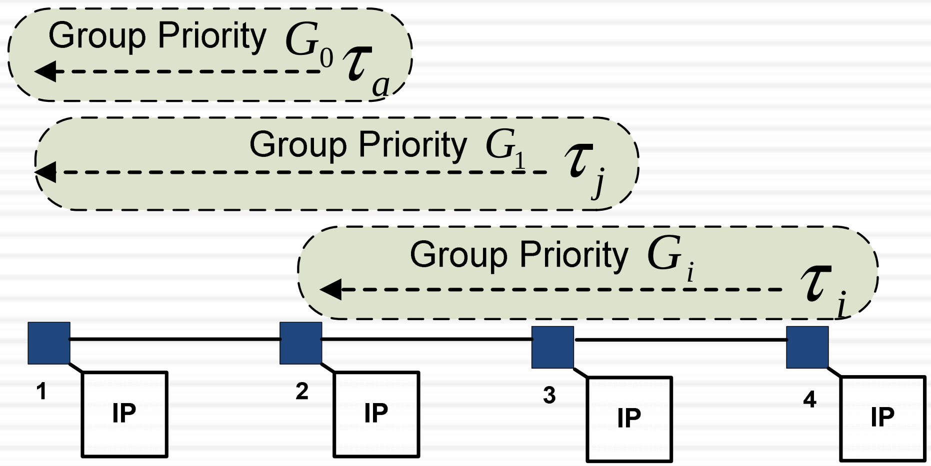
- Upper bound

$$W(i) = \sum_{\forall \tau_m \in S(i)} \lceil \frac{W(i) + J_m^R}{T_m} \rceil \cdot C_m + \sum_{\forall \tau_j \in hp(i)} \lceil \frac{W(i) + J_j^R + R_j - C_j}{T_j} \rceil C_j$$

Indirect Interference



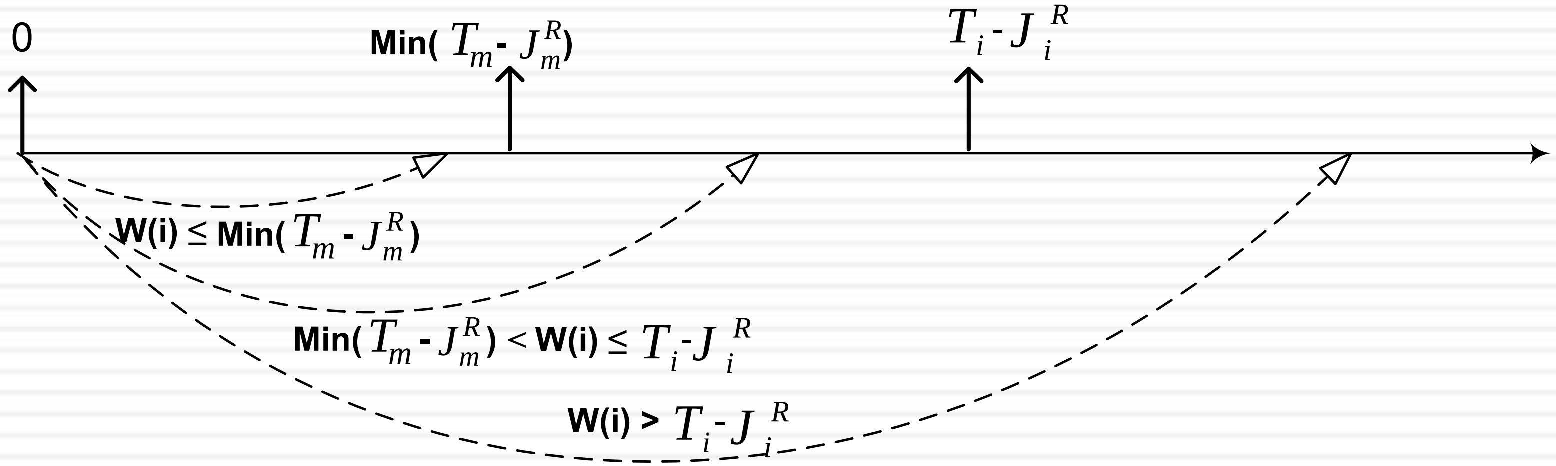
Case (A): Indirect interference from same system priority



Case (B): Indirect interference from higher system priority

Maximum network latency

- Three relations between priority window and period
- τ_m is any flow in set S(i) which includes all the flows sharing the same priority and τ_i is the observed flow



Maximum network latency (2)

- The network latency upper bound R_i is givne by: $R_i = W(i) + J_i^R$

when $W(i) \leq min(T_m - J_m^R)$

- The network latency upper bound R_i is given by $R_i = W(i) + J_i^R$

when $min(T_m - J_m^R) \leq W(i) \leq T_i - J_i^R$

Maximum network latency (3)

- The network latency upper bound R_i is given by:

$$R_i = \max_{q=1,\ldots,\lceil \frac{W(i)+J_i^R}{T_i} \rceil} (w_q(i) - (q-1)T_i + J_i^R)$$

where q is the index of packet instance, and $w_i(q)$ is given by:

$$w_q(i) = qC_i + \sum_{\forall \tau_m \in S(i), \tau_m \neq \tau_i} \lceil \frac{w_q(i) + J_m^R}{T_m} \rceil \cdot C_m + \sum_{\forall \tau_j \in hp(i)} \lceil \frac{w_q(i) + R_j - C_j + J_j^R}{T_j} \rceil C_j$$

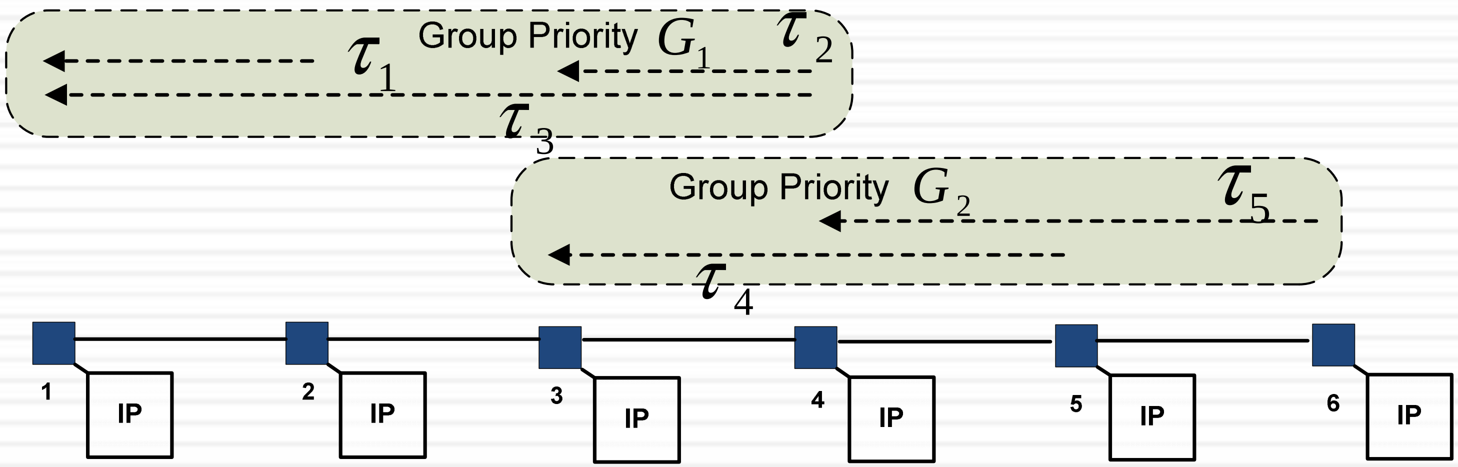
when $W(i) > T_i - J_i^R$.

Summary

- Priority share can effectively reduce priority and virtual channel overhead but also introduce the blocking.
- Three different blocking situations are addressed.
- The priority window approach offers a network latency upper bound analysis to handle all the possible situations.
- With this new schedulability analysis, a broad class of real-time communication can be explored and developed in a SoC/NoC platform.

Thanks and Question Time

A set of traffic-flows case



Priority ordering:

$$G_1 > G_2$$

Set:

$$S_1 = \{\tau_1, \tau_2, \tau_3\}$$
$$S_2 = \{\tau_4, \tau_5\}$$

Interrelationship:

$$S_1^D = \phi, S_1^I = \phi$$
$$S_2^D = \phi, S_2^I = \phi$$
$$S_3^D = \phi, S_3^I = \phi$$
$$S_4^D = \{\tau_2, \tau_3\}, S_4^I = \{\tau_1\}$$
$$S_5^D = \phi, S_3^I = \{\tau_1, \tau_2, \tau_3\}$$