Introduction to Intelligent Vehicles [3. Timing Analysis II]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

Fall 2019

Announcement

- Homework 1 due in two weeks

Revisit CAN Timing Analysis

- $Q_i = B_i + \sum_{(\text{for all } j,\, P_j<P_i)} \left\lceil \frac{Q_i + \tau}{T_j} \right\rceil C_j$
- $R_i = Q_i + C_i$

Message	Priority	Trans. Time	Period
μ_0	$P_0 = 0$	$C_0 = 4$	$T_0 = 10$
μ_1	$P_1 = 1$	$C_1 = 4$	$T_1 = 13$
μ_2	$P_2 = 2$	$C_2 = 4$	$T_2 = 13$

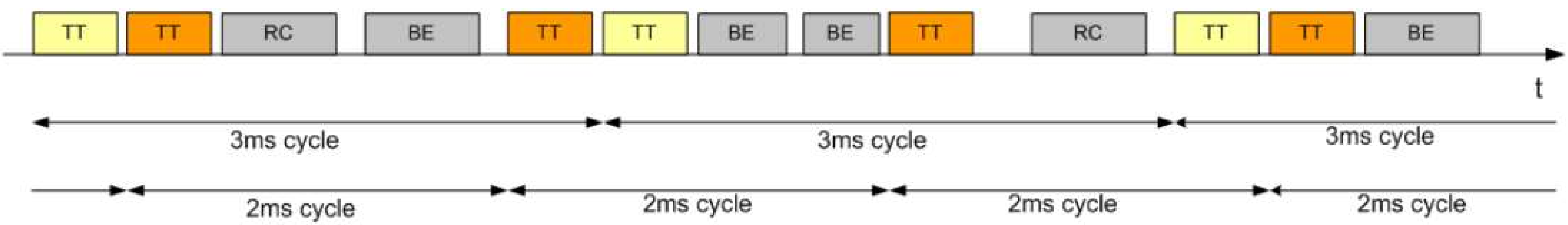
- If the B_i is replaced by maximum C_i with lower priority
  - The analysis will become schedulable

Outline

- **Introduction to Other In-Vehicle Networks**
- Timing Analysis of Time Division Multiple Access (TDMA) Based Protocols
- Real-Time Calculus (RTC)

TTEthernet

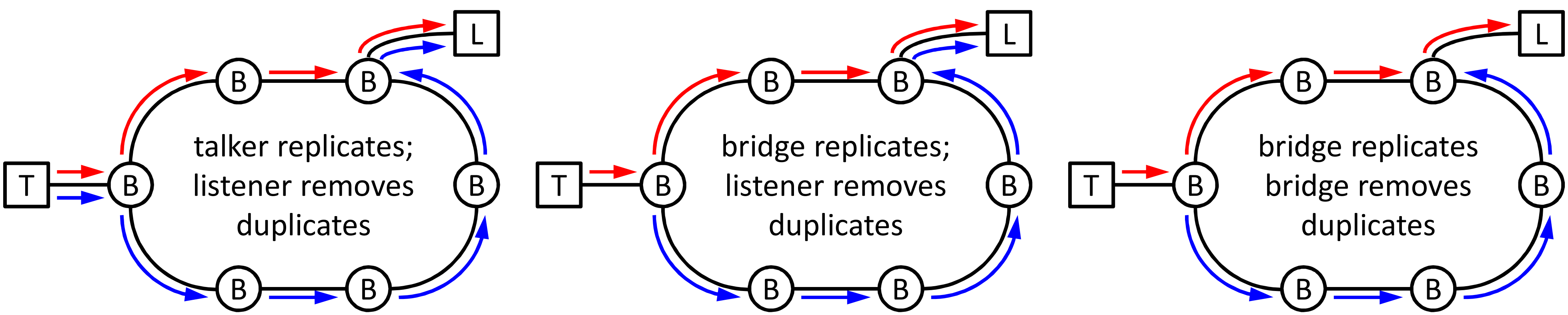
- Why TT, not pure Ethernet?
- Features
  - Quality of Service (QoS) and preemption
  - Time synchronization
- Traffic types
  - Time-Triggered (TT) traffic (highest priority)
    - Sent over the network at predefined (scheduled) time
  - Rate-Constrained (RC) traffic
    - Sent over the network with predefined bandwidth
  - Best-Effort (BE) traffic (lowest priority)
    - Conventional Ethernet



Time-Sensitive Networking (TSN)

❑ Features

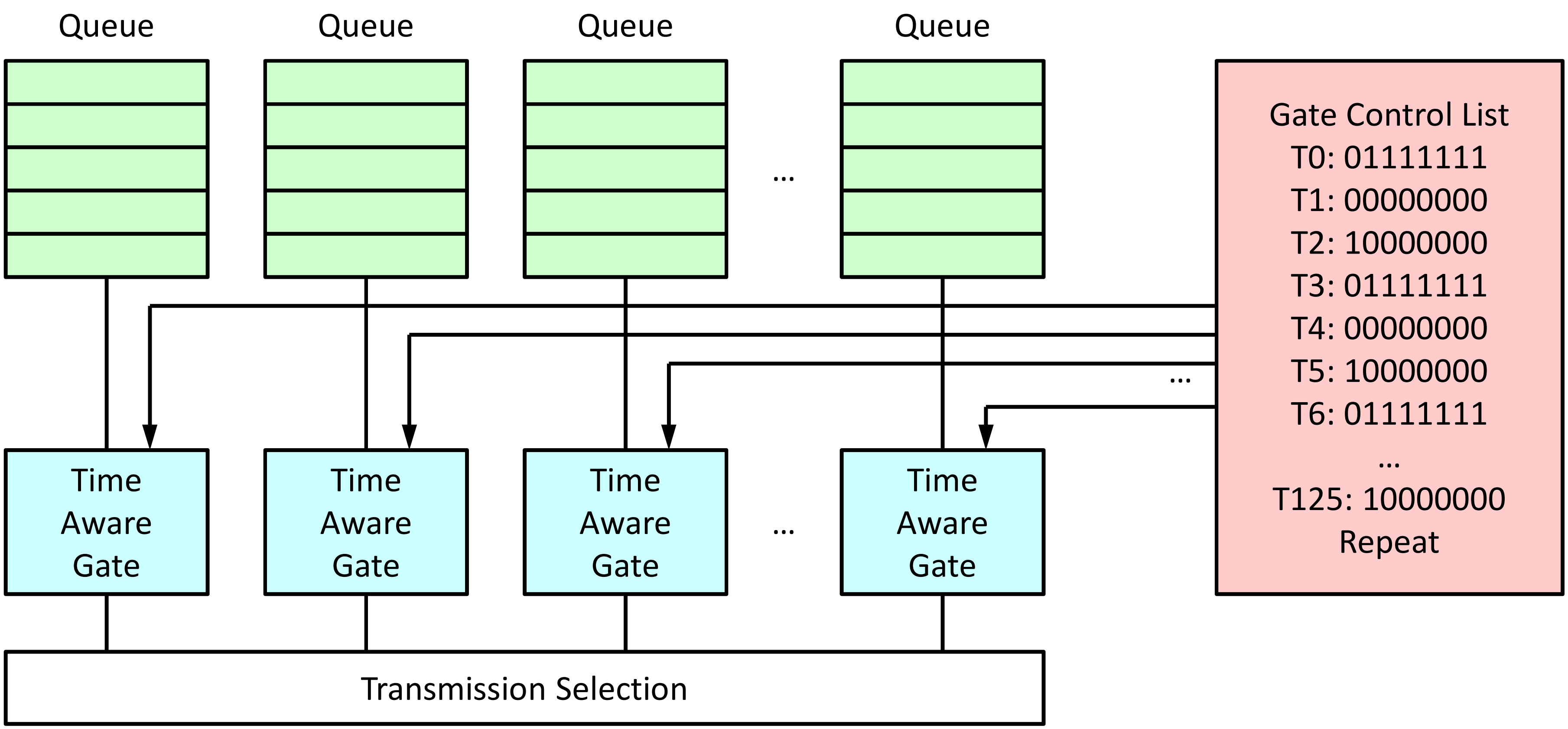
- Another name: Audio Video Bridging (AVB)
- Quality of Service and preemption
  - Achieve timing guarantees for high-priority messages
- Frame replication and elimination
- Time synchronization
- Time aware shaper



https://standards.ieee.org/events/automotive/2015/03_IEEE_TSN_Standards_Overview_and_Update_v4.pdf

Time-Sensitive Networking (TSN)

❑ Time aware shaper



http://www.ieee802.org/1/files/public/docs2012/bv-boiger-time-aware-shaper-0712-v01.pdf

Other Protocols with TDMA Concepts

- FlexRay
  - https://en.wikipedia.org/wiki/FlexRay
- Time-Triggered Protocol
  - https://en.wikipedia.org/wiki/Time-Triggered_Protocol

Outline

- Introduction to Potential In-Vehicle Networks
- **Timing Analysis of Time Division Multiple Access (TDMA) Based Protocols**
- Real-Time Calculus (RTC)

Abstraction

- [Wikipedia]
  - In software engineering and computer science, abstraction is
    - The **process** of removing physical, spatial, or temporal details or attributes in the study of objects or systems in order to more closely attend to other details of interest
      - It is also very similar in nature to the process of generalization
    - The **objects** which are created by keeping common features or attributes to various concrete objects or systems of study
      - i.e., the result of the process
  - John V. Guttag
    - "The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context"
- Example
  - Timing analysis of Controller Area Network (CAN)

Problem Formulation

- There is a set of time slots scheduled to serve a message in a TDMA-based protocol
  - The network schedule and the message arrivals are defined by "patterns"
- What is the worst-case response time of the message?

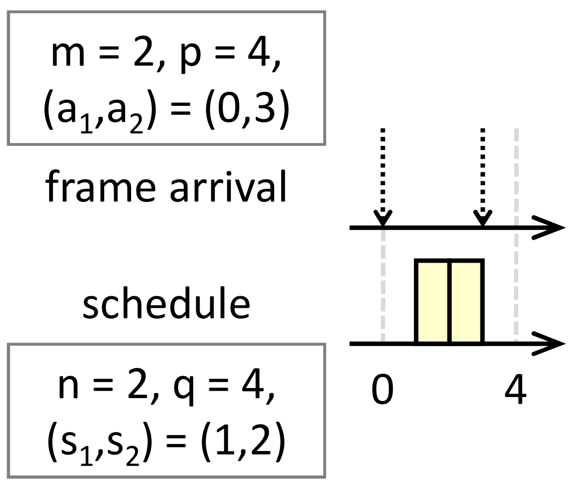
- Assumptions
  - Each time slot has the same length
  - Each time slot serves exactly one instance/frame
  - An instance/frame is transmitted only if the whole time slot is available
    - No transmission if the instance/frame arrives in the middle of the time slot

Message Definitions

- Synchronous message
  - The network knows the time that each frame of the message is sent
  - Example 1: Buses arrive at 7am, 8am, 9am, …
  - Example 2: Abstraction of TSN traffic in TSN
  - Example 3: Abstraction of Time-Triggered traffic in TTEthernet
- Asynchronous message
  - The network does not know the time that each frame of the message is sent but knows the period (or pattern) of the message
  - Example 1: Buses arrive every hour
  - Example 2: Abstraction of AVB traffic in TSN
  - Example 3: Abstraction of Rate-Constrained traffic in TTEthernet

Pattern Definition

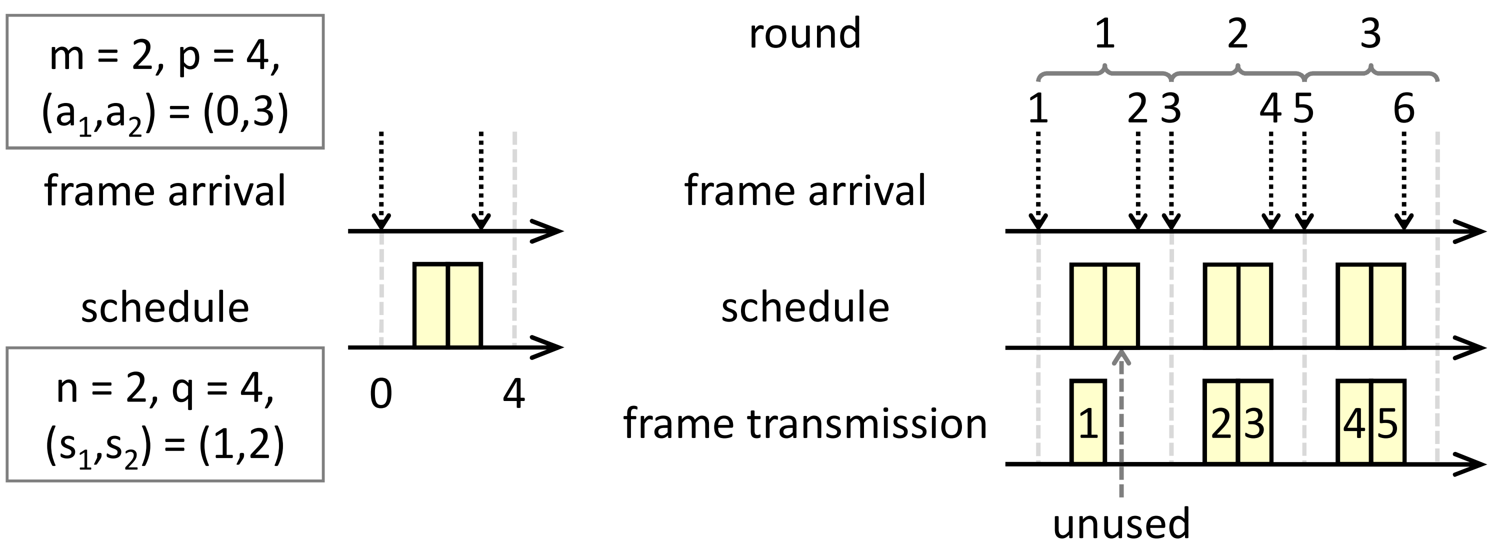
- Frame arrival pattern $(m, p, a_1, a_2, a_3, \ldots, a_m)$
  - Arriving times of frames: $a_1,a_2,a_3,\ldots,a_m$
  - The pattern repeats with a period p
- Schedule pattern $(n, q, s_1, s_2, s_3, \ldots, s_n)$
  - Starting times of time slots: $s_1,s_2,s_3,\ldots,s_n$
  - The pattern repeats with a period q



- If m/p > n/q (demand > supply), then it is not schedulable

Synchronous Message

- Theorem: we only need to consider **two rounds** to compute the worst-case response time
  - Length of a round = least common multiple of p and q

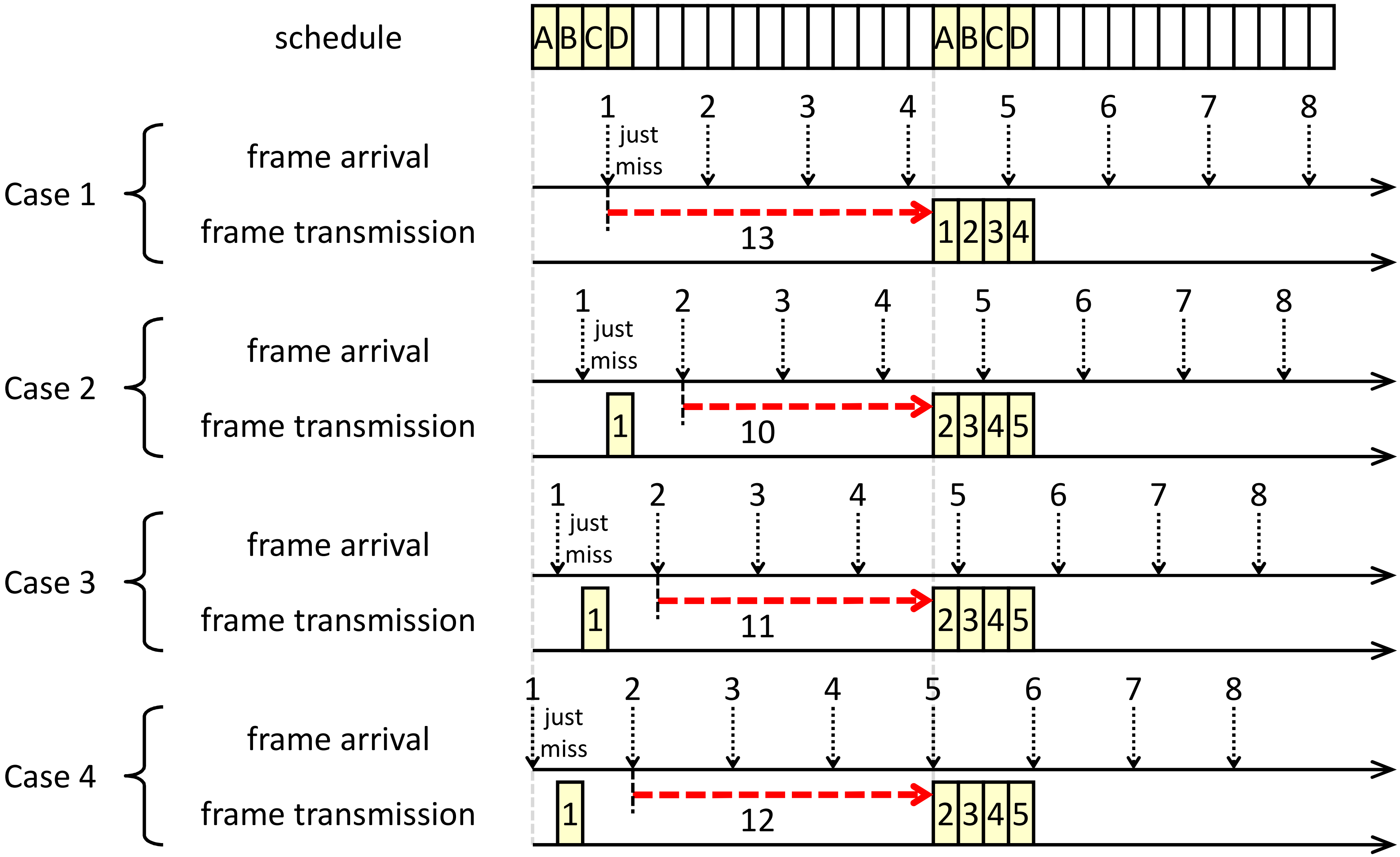


- For your reference
  - Why two rounds?
    - The numbers of unscheduled frames after first and second rounds are the same

Practice

- Analyze the following patterns
  - Assume they are synchronous
- Frame arrival pattern: (m = 4, p = 10, **a** = 0, 3, 5, 6)
- Schedule pattern: (n = 2, q = 5, **s** = 1, 2)

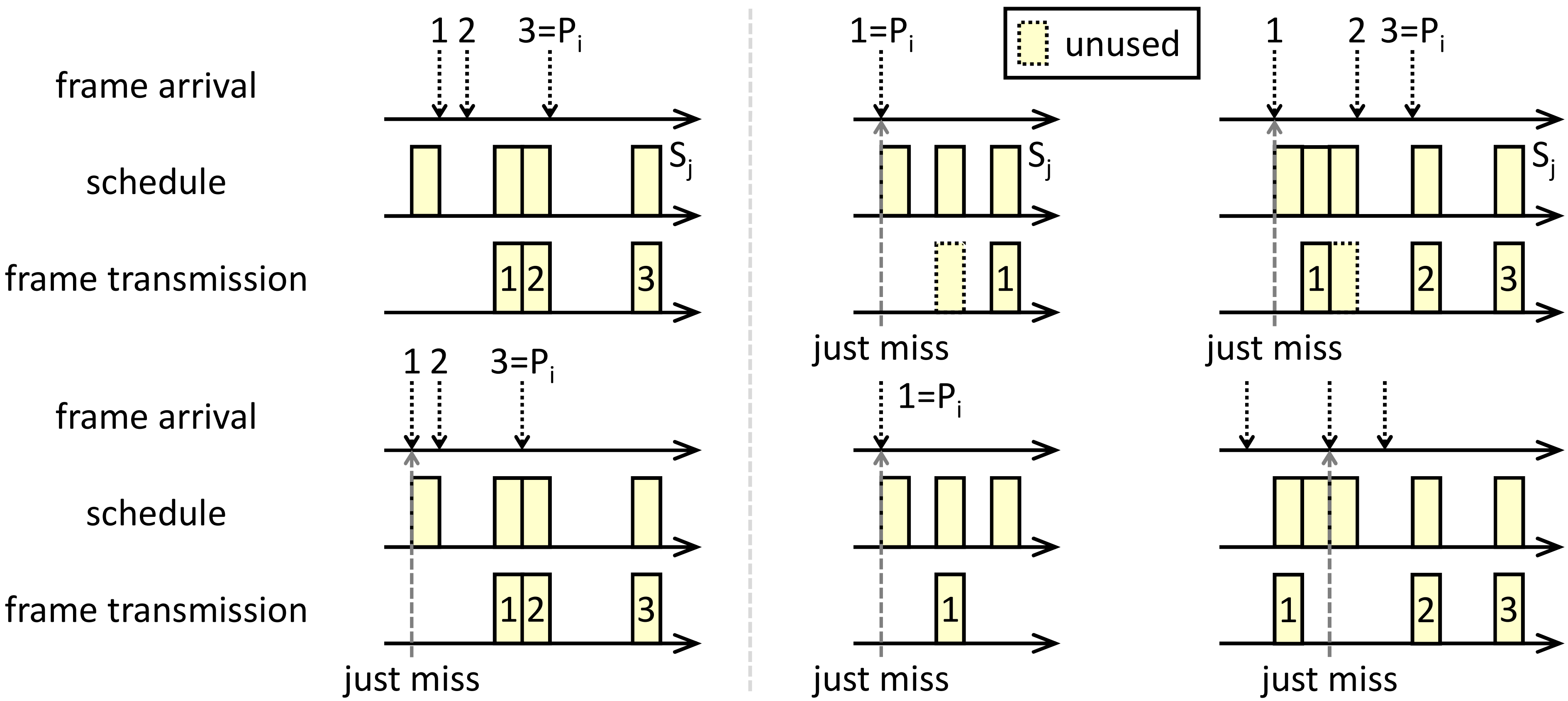
Asynchronous Message: Example



And more cases …

Asynchronous Message: Theorem

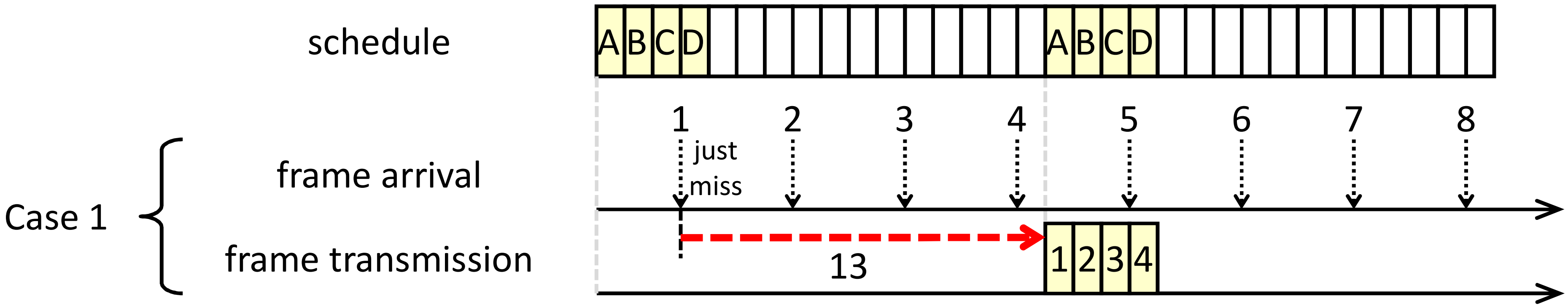
- Theorem: if the worst case happens when frame P_i is assigned to time slot S_j, then
  - P_i or one frame before P_i must **just miss** an assigned time slot, and
  - There must be no unused time slot between the ending time of the just-missed time slot and the starting time of S_j



Asynchronous Message: Duplication

- Frame arrival pattern $(m, p, a_1, a_2, a_3, \ldots, a_m)$
  - Arriving times of frames: $a_1,a_2,a_3,\ldots,a_m$
  - The pattern repeats with a period p
- Schedule pattern $(n, q, s_1, s_2, s_3, \ldots, s_n)$
  - Starting times of time slots: $s_1,s_2,s_3,\ldots,s_n$
  - The pattern repeats with a period q
- Assumption (or duplication until)
  - p = q = least common multiple of p and q
- Duplication again
  - $a_{m+1} = a_1+p$, $a_{m+2} = a_2+p$, …, $a_{2m} = a_m+p$
  - $s_{n+1} = s_1+q$, $s_{n+2} = s_2+q$, …, $s_{2n} = s_n+q$
  - **Fix m, n, p, q this time**

Asynchronous Message: Example



- Frame arrival pattern
  - (m = 4, p = 16, **a** = 3, 7, 11, 15, 19, 23, 27, 31)
- Schedule pattern
  - (n = 4, q = 16, **s** = 0, 1, 2, 3, 16, 17, 18, 19)

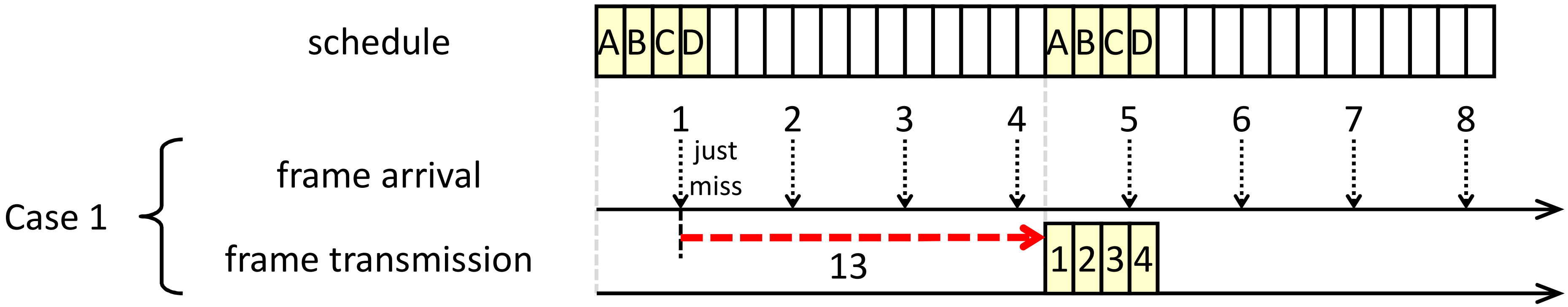
Asynchronous Message: Equation

- We only need to consider a finite number of different alignments of the frame arrival and the schedule
  - They are the cases that a frame just misses a time slot
  - The worst-case response time is

$$1 + \max_{1 \le k \le m} (\max_{1 \le j \le n} (s_{j+k} - s_j) - \min_{1 \le i \le m} (a_{i+k-1} - a_i))$$

    - "1" is the transmission time
    - "$\max_{1 \le k \le m}$ (…)" is the waiting time
  - What is the meaning of the equation?
    - Assume the i-th frame just misses the j-th time slot
    - Calculate the response time of the k-th frame after the i-th frame
      - k = 1 for the i-th frame itself
  - What is the meaning of the equation, again?
    - The densest part of the frame arrival pattern is served by the least dense part of the schedule pattern

Asynchronous Message: Example



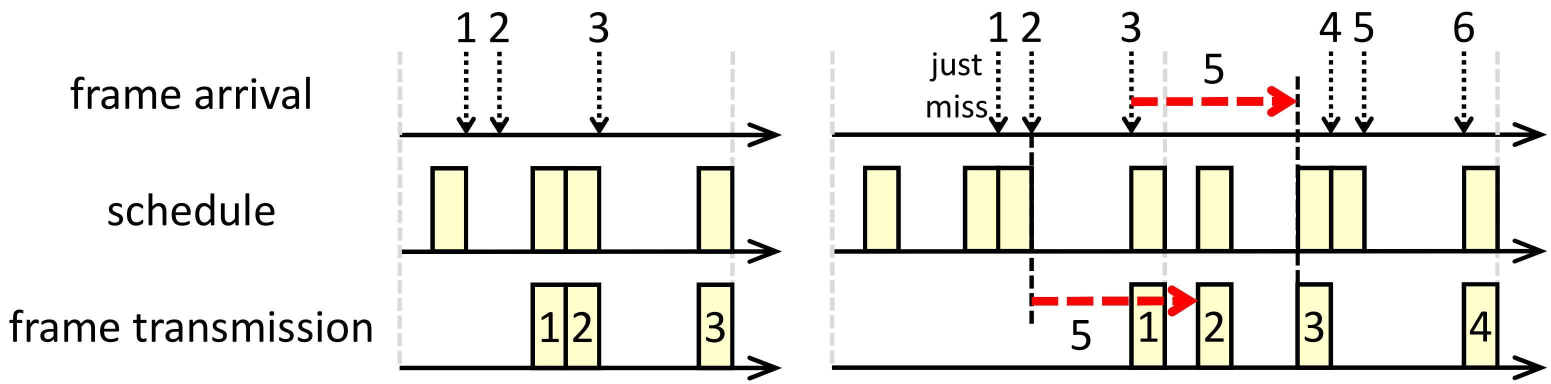
$$1 + \max_{1\le k\le m} \left(\max_{1\le j\le n} (s_{j+k} - s_j) - \min_{1\le i\le m} (a_{i+k-1} - a_i) \right)$$

- **Frame arrival pattern:** (m = 4, p = 16, **a** = 3, 7, 11, 15, 19, 23, 27, 31)
- **Schedule pattern:** (n = 4, q = 16, **s** = 0,1,2,3,16,17,18,19)

k	$\max_{1\le j\le n} (s_{j+k} - s_j)$	=	$\min_{1\le i\le m} (a_{i+k-1} - a_i)$	=	(…) in $\max_{1\le k\le m}$ (…)
1	$\max_{1\le j\le 4} (s_{j+1} - s_j)$	(j=4) → 13	$\min_{1\le i\le 4} (a_i - a_i)$	(i=1,2,3,4) → 0	13
2	$\max_{1\le j\le 4} (s_{j+2} - s_j)$	(j=3,4) → 14	$\min_{1\le i\le 4} (a_{i+1} - a_i)$	(i=1,2,3,4) → 4	10
3	$\max_{1\le j\le 4} (s_{j+3} - s_j)$	(j=2,3,4) → 15	$\min_{1\le i\le 4} (a_{i+2} - a_i)$	(i=1,2,3,4) → 8	7
4	$\max_{1\le j\le 4} (s_{j+4} - s_j)$	(j=1,2,3,4) → 16	$\min_{1\le i\le 4} (a_{i+3} - a_i)$	(i=1,2,3,4) → 12	4

note: this is waiting time

Asynchronous Message: Example



- Frame arrival pattern: (m = 3, p = 10, **a** = 2, 3, 6, 12, 13, 16)
- Schedule pattern: (n = 4, q = 10, **s** = 1, 4, 5, 9, 11, 14, 15, 19)

k	$\max_{1 \le j \le n} (s_{j+k} - s_j)$	=	$\min_{1 \le i \le m} (a_{i+k-1} - a_i)$	=	(…) in $\max_{1 \le k \le m}$ (…)
1	$\max_{1 \le j \le 4} (s_{j+1} - s_j)$	(j=3) → 4	$\min_{1 \le i \le 3} (a_i - a_i)$	(i=1,2,3) → 0	4
2	$\max_{1 \le j \le 4} (s_{j+2} - s_j)$	(j=3) → 6	$\min_{1 \le i \le 3} (a_{i+1} - a_i)$	(i=1) → 1	5
3	$\max_{1 \le j \le 4} (s_{j+3} - s_j)$	(j=3) → 9	$\min_{1 \le i \le 3} (a_{i+2} - a_i)$	(i=1) → 4	5

note: this is waiting time

Practice

- Analyze the following patterns
  - Assume they are asynchronous
- Frame arrival pattern: ($m = 4$, $p = 10$, $\mathbf{a} = 0, 3, 5, 6$)
- Schedule pattern: ($n = 2$, $q = 5$, $\mathbf{s} = 1, 2$)

Discussion

- They imply the optimal scheduling for a message
  - As early as possible for a synchronous message
  - As evenly as possible for an asynchronous message
  - How to resolve conflicts between multiple messages?
- Gaps to a practical protocol
  - Each time slot has the same length
  - Each time slot serves exactly one frame
  - Frame arrival and network schedule are described by patterns
  - Multiple switches?

Outline

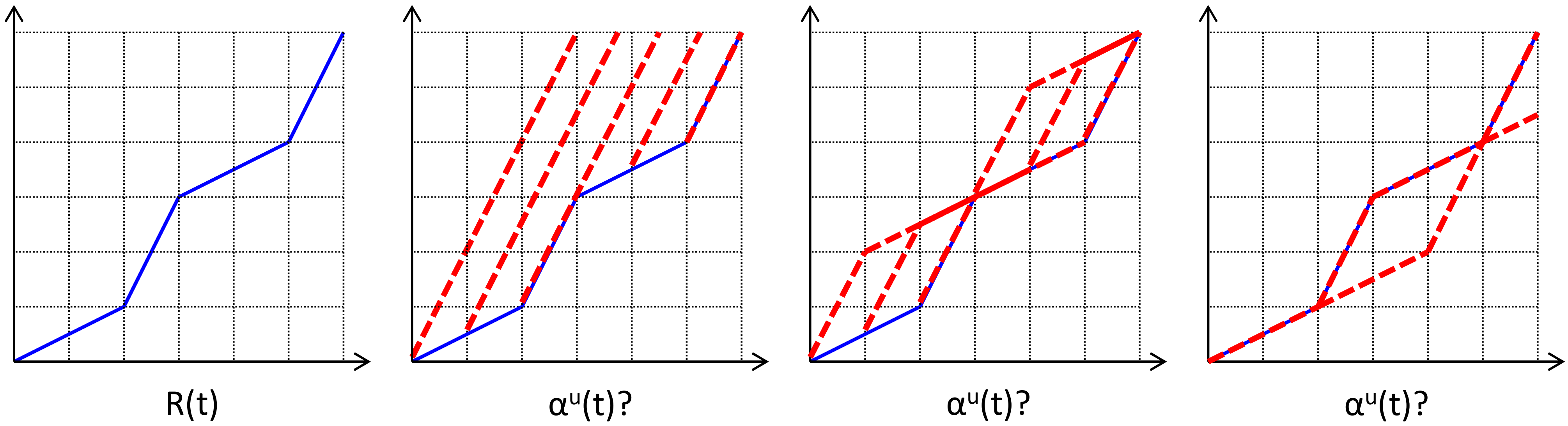
- Introduction to Potential In-Vehicle Networks
- Timing Analysis of Time Division Multiple Access (TDMA) Based Protocols
- **Real-Time Calculus (RTC)**
  - The midterm will not include RTC

Min-Plus Algebra

- Minimum
  - $(f \oplus g)(t) = \min (f (t), g(t))$
- Convolution
  - $(f \otimes g)(t) = \inf_{0 \le s \le t} (f (s) + g(t - s))$
  - Example: f(x) = x and g(x) = 2x
  - Example: f(x) = x if x ≤ 1; f(x) = 3x of x > 1; g(x) = 2x
- Deconvolution
  - $(f \oslash g)(t) = \sup_{u \ge 0} (f (t + u) - g(u))$
  - Example: f(x) = x and g(x) = 2x
  - Example: f(x) = x if x ≤ 1; f(x) = 3x of x > 1; g(x) = 2x

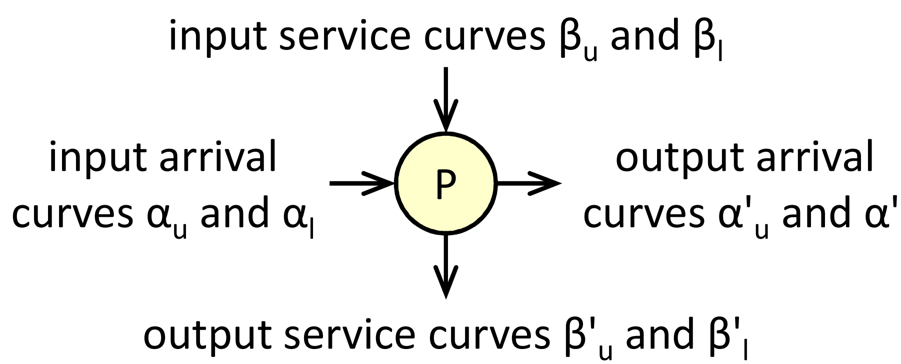
Input Arrival and Service Curves

- Input/output cumulative function, R(t) / R'(t)
  - R(t): the amount of load that arrives in time interval [0,t)
  - R'(t): the amount of load that leaves in time interval [0,t)
- Input upper/lower arrival curves, $\alpha_u(t)$ / $\alpha_l(t)$
  - $\alpha_l(t) \leq R(s + t) - R(s) \leq \alpha_u(t)$
- Input upper/lower service curves, $\beta_u(t)$ / $\beta_l(t)$
  - $(R \otimes \beta_l)(t) \leq R'(t) \leq (R \otimes \beta_u)(t)$

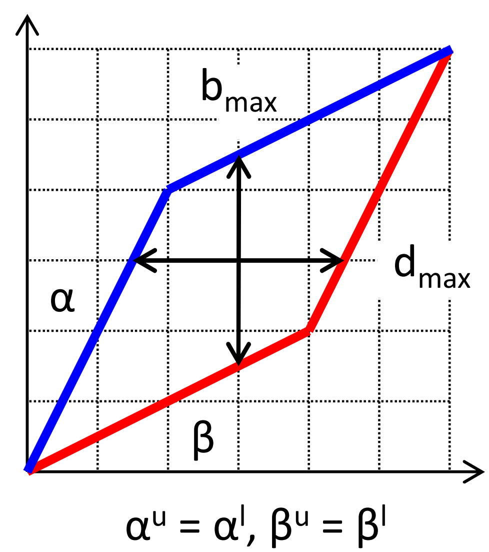


Output Arrival and Service Curves

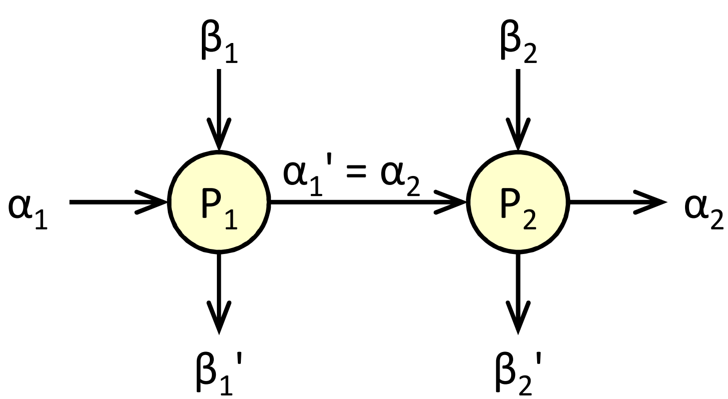
- Given a process with $\alpha_u(t)$, $\alpha_l(t)$, $\beta_u(t)$, $\beta_l(t)$
- Output upper/lower arrival curves, $\alpha'_u(t)$ / $\alpha'_l(t)$
  - $\alpha'_u = ((\alpha_u \otimes \beta_u) \oslash \beta_l) \oplus \beta_u$
  - $\alpha'_l = ((\alpha_l \oslash \beta_u) \otimes \beta_l) \oplus \beta_l$



- Maximal backlog
  - $b_{max} = \sup_{t \geq 0} (\alpha_u(t) - \beta_l(t))$
- Maximal delay
  - $d_{max} = \sup_{t \geq 0} [\inf_{s \geq 0,\, \alpha_u(t) \leq \beta_l(t+s)} (s)]$



- System analysis



Q&A