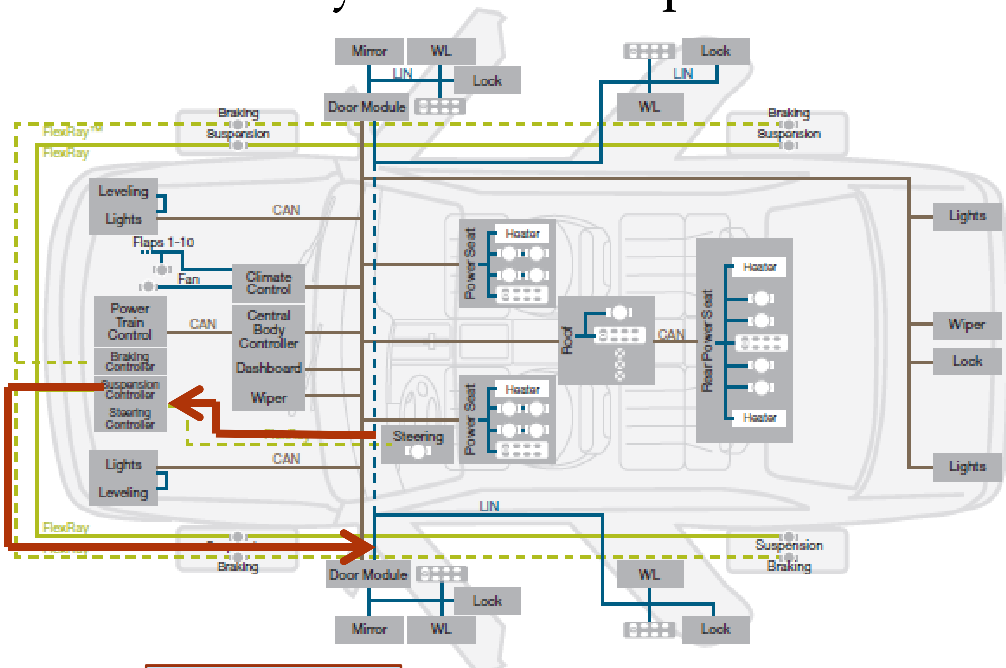
Response Time of Embedded Networks: FlexRay and Time Triggered Ethernet

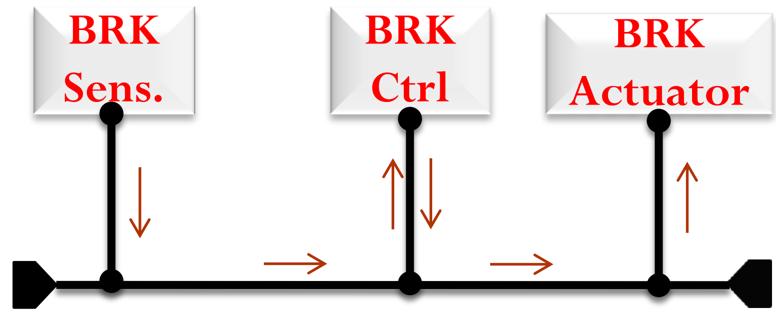
Presented by
Ratnesh Kumar
Fellow IEEE, Prof. ECE, Iowa State

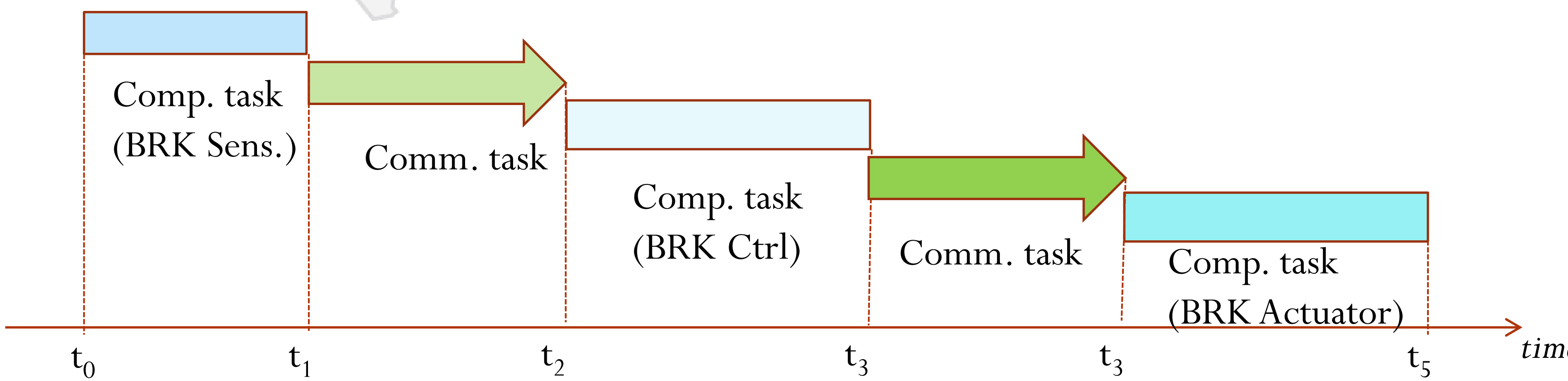
Work done in collaboration with
Lucien Ouedraogo
Research Fellow, ECE, Iowa State (Presently with MathWorks)

A Time-Critical Task in Vehicle Embedded Network

- A Brake-by-Wire example







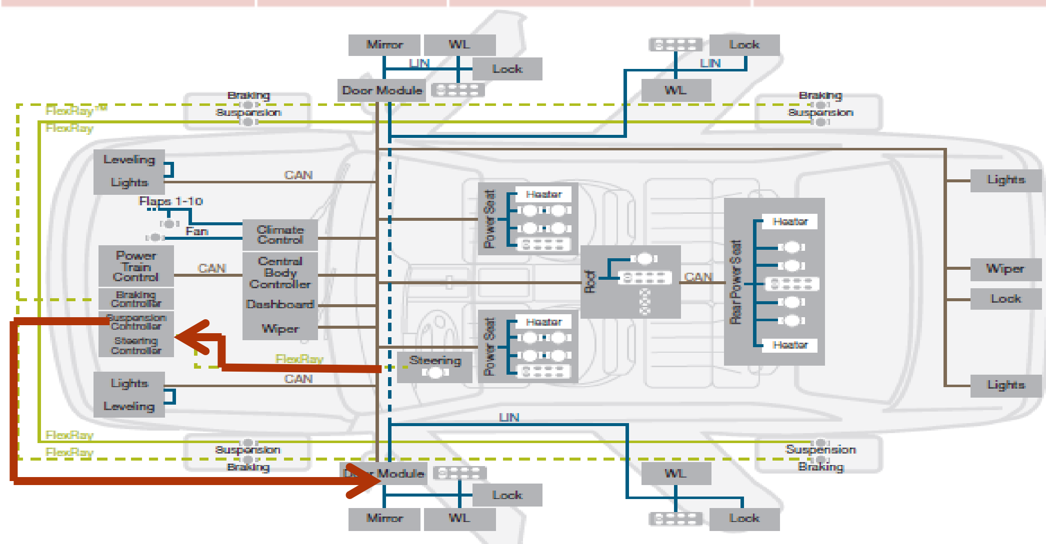
Emerging Networks: FlexRay/TTEthernet

- Properties:
  - Time-driven + event-driven traffic
  - Distributed clock synchronization
  - Fault-tolerance

- Response time important concern for time-critical apps: E.g. Brake-by-wire

- Our work: Computationally efficient method for computing worst-case response time and communication schedule

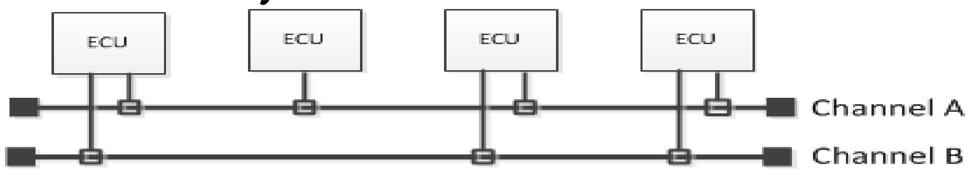
	CAN	FlexRay	TTE
Medium access	CSMA/CA	TDMA & FTDMA	TDMA & CSMA/CD
Bandwidth	1 Mbit/s	10 (x2) Mbit/s	100 Mbit/s
Frame size	14 – 8 bytes	264 – 254 bytes	1538 – 1500 bytes
Topology	Bus	Bus, Star	Star, Line
Composability	No	Yes	Yes
Incremental update?	No	Yes	No
Utilization in real setting	<40-50%	~30-50%, Max. 60-70%	~70%



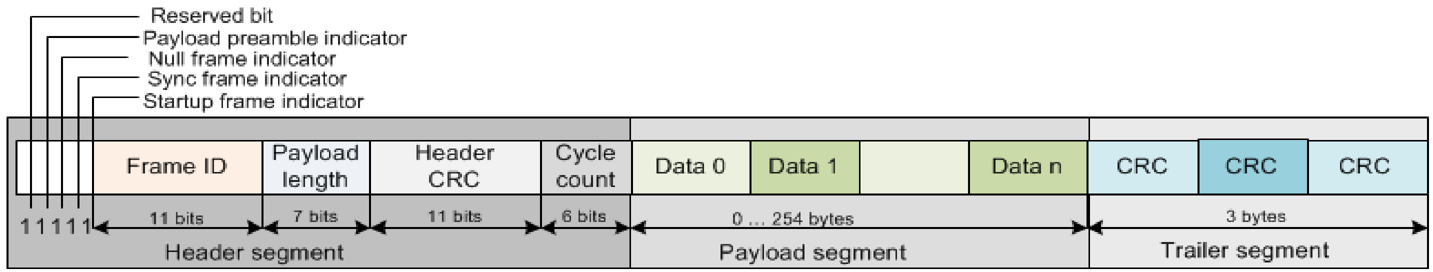
Overview of FlexRay

- Developed by an industry consortium, now ISO 10681 1&2: 2010

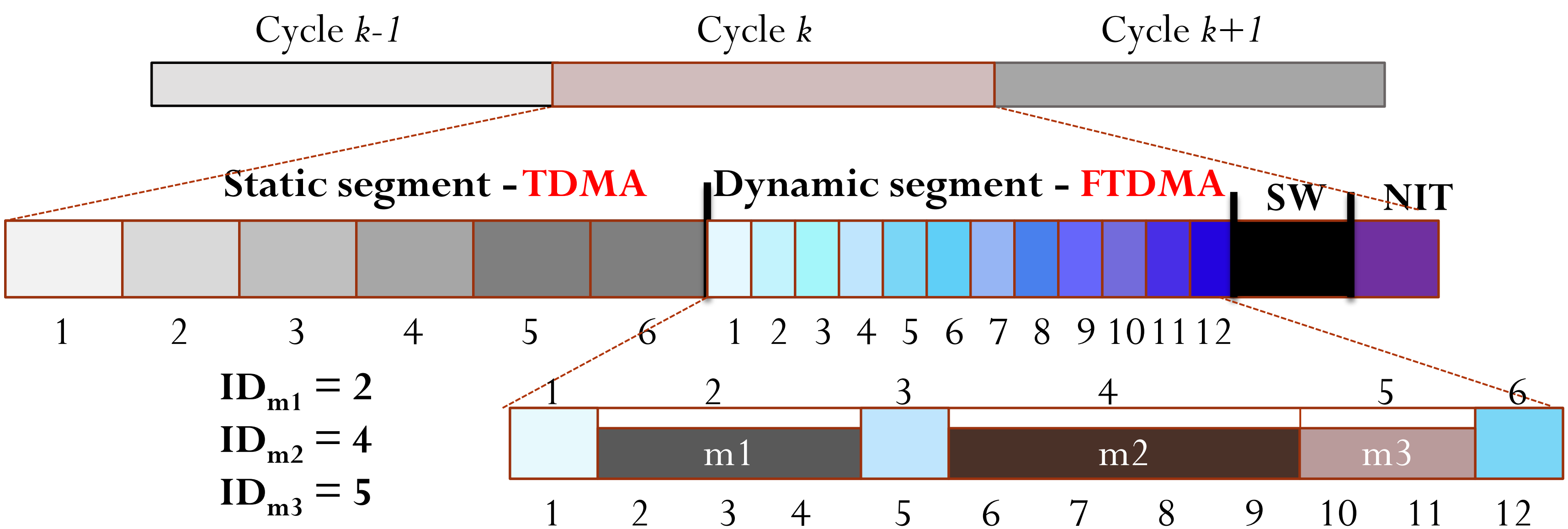
Dual channel bus



Frame format

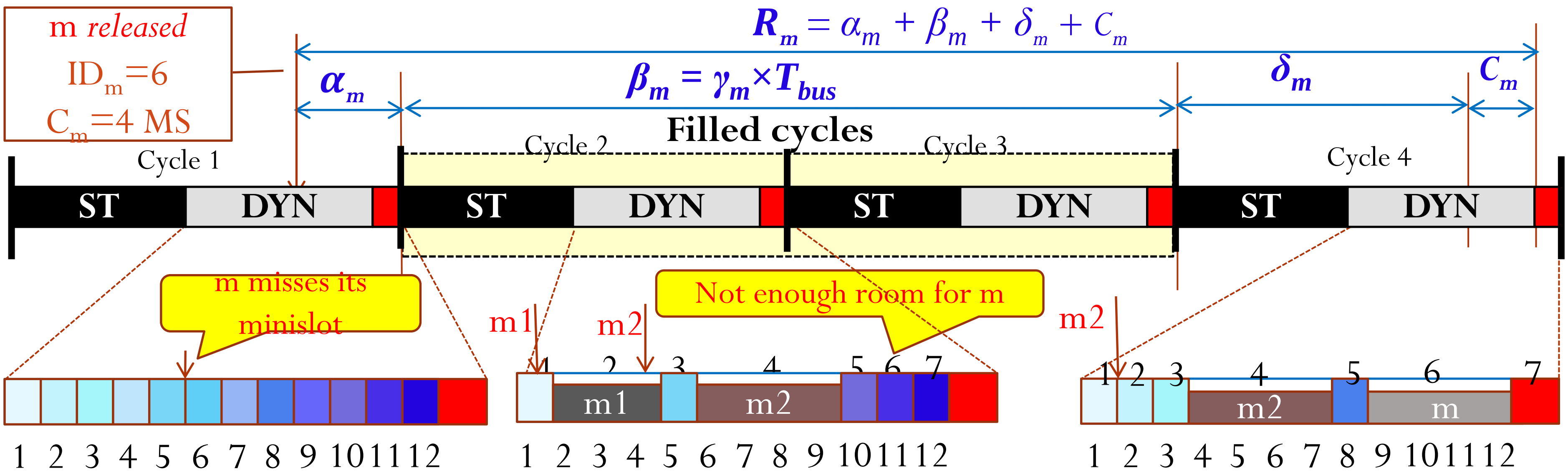


MAC



Response Time Computation Problem

- Within DYN segment, higher priority msga delay lower priority ones
- For response time: Find worst case arrival sequence of higher priority msgs



- We formulate computation of β_m and δ_m as ILP (α_m and C_m are constants)

Our approach – Variables and Objective Functions

- Parameters (constants):
  - $\boldsymbol{Dom}(m) = \{m_1, m_2, \ldots, m_{p-1}\}$: Priority ordered messages of higher priority than *m*;
  - Binary constant $\boldsymbol{f_{ik}} = \mathbf{1}$ iff $ID_i = ID_k$;
  - $T_i^{\boldsymbol{last}}$: Last minislot in which message *i* can be transmitted;
  - φ_i: extra minislots needed for transmission of message *i*;
- Variables:
  - Binary decision variable $\boldsymbol{x_i^j} = \mathbf{1}$ iff message *i* is transmitted in cycle *j;*
  - Binary (auxiliary) variable $\boldsymbol{u_i^j} = \mathbf{1}$ iff message *i* is not transmissible in cycle *j*.
- Objective functions:
  - For computing β_m, Maximize number of nontransisible cycles:

$$\boldsymbol{Maximize} \sum_{j=0}^{U} u_p^j$$

  - For computing δ_m, Maximize number of messages in the final cycle:

$$\boldsymbol{Maximize} \sum_{\boldsymbol{i=1}}^{\boldsymbol{p-1}} \boldsymbol{x_i^{\{\beta_m+1\}}} \times \boldsymbol{\varphi_i}$$

Our approach – Constraints

- Non transmissibility condition (filled cycle condition) that defines u_i^j:

$$[u_i^j = 1] \Leftrightarrow \left[\sum_{k<i} x_k^j \times \varphi_k + ID_i > T_i^{last}\right] \vee \left[\sum_{k<i} x_k^j \times f_{ik} > 0\right], \forall i, j$$

- A messages is transmitted only if it is transmissible:

$$x_i^j \leq 1 - u_i^j, \forall i, j$$

- Filled cycles for $m(= m_p)$ must be contiguous:

$$u_p^j \geq u_p^{\{j+1\}}, \forall j > 0$$

Our approach – Constraints

- Arrival of messages must satisfy their minimum interarrival times:

$$[u_i^j = x_i^j = 0] \wedge [x_i^k = 1] \Rightarrow \left[\Delta_i^{\{j,k\}} > \left(\sum_{l=j}^{j+k} x_i^l - 1 \right) T_i \right], \forall i, j, k$$

where:

$$\Delta_i^{\{j,k\}} := k \times T_{bus} + \left(\sum_{n<i} (x_n^k - x_n^j) \varphi_n \right) \times T_{MS}$$

is duration between j^{th} and $(j+k)^{th}$ cycles in which message i can be transmitted.

- Initial condition for cycle 0 (no transmission and transmissibility):

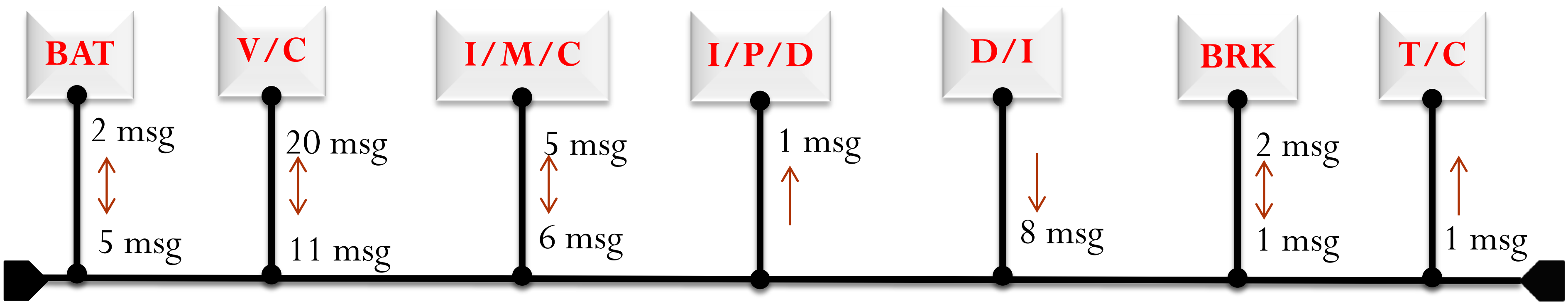
$$x_i^0 = u_i^0 = 0, \forall i$$

- (Additional condition for 2nd objective function) Initial $\boldsymbol{\beta_m}$ cycles are filled:

$$u_p^j = 1, \forall 0 < j \leq \beta_m$$

Validation over SAE Benchmark

- SAE benchmark: 7 ECUs exchanging 22 periodic & 31 aperiodic signals



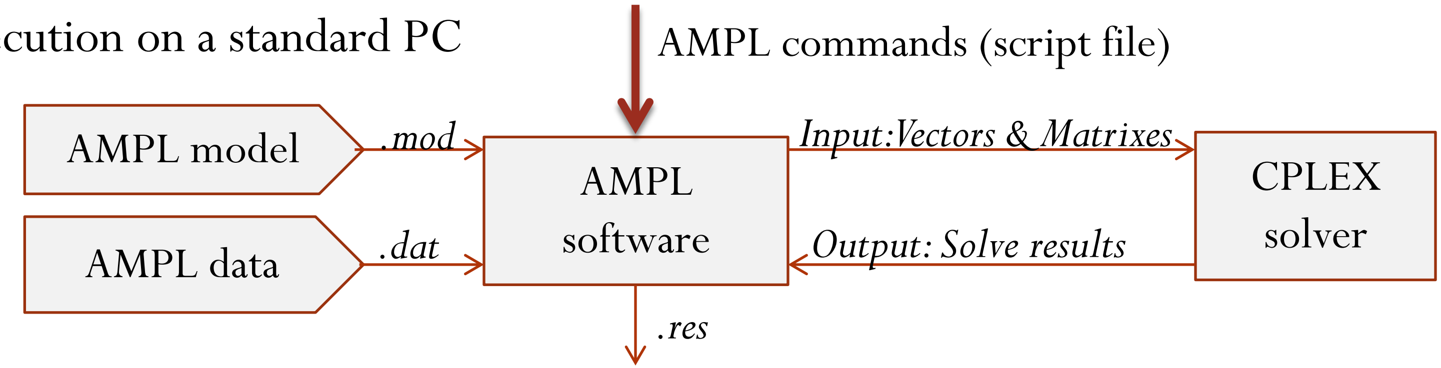
- We consider three FlexRay configurations:

Parameter	Length Bus	Length ST segment	Length Minislot	Number Minislots
Config. C1	170	60	2	50
Config. C2	120	40	2	40
Config. C3	150	30	2	60

- C1 & C2: some messages are unschedulable
- C3: all the messages are schedulable

Validation over SAE Benchmark

- Software framework for simulation:
  - Modeling language and tool: AMPL
  - ILP solver: CPLEX V12.2
  - Hardware: Execution on a standard PC

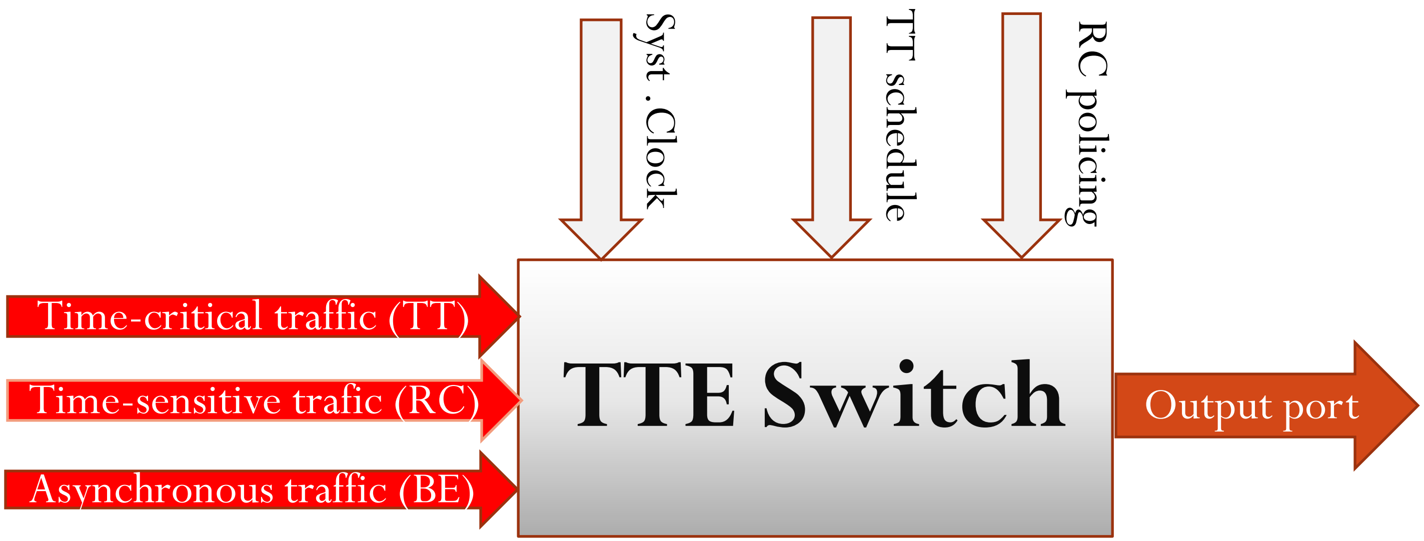


Msg No		22		23		24		25		26		27		28	
Config.		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
R_m	Ours	N.S	1104	1730	632	N.S	1342	N.S	754	N.S	N.S	N.S	1110	N.S	N.S
	Pop et al.	-	-	N.S.	632	-	1342	-	980	-	-	-	1112	-	-

- Our formulation computes within 6 minutes for all msgs; Pop et al. failed to compute within an hour for certain messages (shown as -).
- For msgs Pop et al. is able to compute, it overestimates by as much as 30%

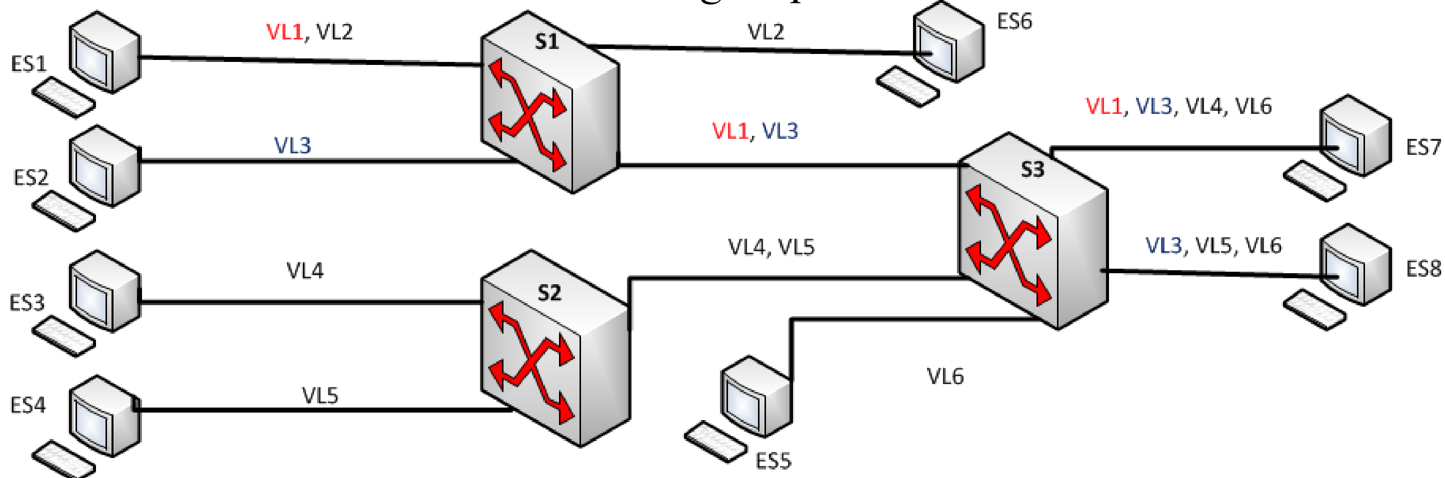
TTEthernet Protocol: Overview

- Defines Time-triggered (TT) service over Ethernet – SAE AS6802
- Allows mixed time critical communication on a single physical network
  - TT (Time-Triggered) frames: Period – deterministic latency
  - RC (Rate-constrained) frames: Minimum interarrival time – bounded latency
  - BE (Best-Effort) frames: Standard Ethernet frames – no transmission/delay guarantee
  - Priority: TT → RC → BE



AFDX protocol for RC/TT Frames

- Circuit Switching: Each frame sent from one node to a group of nodes over a virtual circuit (route) of links



- Parameters (constants):
  - $s(i)$: source node of frame *i*
  - $d(i)$: destination node of frame *i*
  - T_i: minimum interarrival time of frame *i*
  - N_i: Number of instances of frame *i* to be transmitted
  - C_i: transmission time of frame *i*
  - $X_{kk'}^{ij} = 1$ iff frames *i* and *j* share the link between nodes *k* and *k'*
  - $Y_{kp}^{ij} = 1$ iff frames *i* and *j* both transit through node *k* at its input port p
  - $Y_{kp}^{i} = 1$ iff frame *i* transits through node *k* at its input port *p*

TTEthernet – Variables/Objective Function

- Variables:
  - a_k^{in}: arrival time of *n*th instance of frame *i* at node *k*
- Objective function:

$$Maximize_n \, [a_{d(i)}^{in} - a_{s(i)}^{in}]$$

TTEthernet – Constraints for RC Frames

- Link is shared resource: Two frames arriving at the same switch port must be separated by at least the transmission time of the first arrived frame:

$$[a_k^{im} - a_k^{jn} \geq C_i Y_{kp}^{ij}] \vee [a_k^{jn} - a_k^{in} \geq C_j Y_{kp}^{ij}], \forall i,j,m,n,k,p$$

- Minimum interarrival time:

$$a_{s(i)}^{i(n+1)} - a_{s(i)}^{in} \geq T_i, \forall n,i$$

- First-Come First-Serve for frames:

$$[a_k^{im} - a_k^{jn}] \wedge [X_{kk'}^{ij} = 1] \Rightarrow [a_{k'}^{im} < a_{k'}^{jn}], \forall i,j,m,n,k,k'$$

- No idling of link when a frame is queued up for transmission:

$$[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \leq a_k^{im}) \vee (a_k^{lo} \geq a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} \leq a_{k'}^{im}] \Rightarrow [a_{k'}^{jn} = a_{k'}^{im} + C_j], \forall i,j,k,k',m,n$$

$$[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \leq a_k^{im}) \vee (a_k^{lo} \geq a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} > a_{k'}^{im}] \Rightarrow [a_{k'}^{in} = a_k^{in} + C_j], \forall i,j,k,k',m,n$$

TTEthernet – Constraints for TT Frames

- Additional parameter:
  - $P^{ij} = 1$ iff frame i higher priority than frame j
- (Modified constraint) Replace first-come-first-serve by highest-priority-first:
  - $[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \le a_k^{im}) \vee (a_k^{lo} \ge a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} \le a_{k'}^{im} - C_i] \wedge [P^{ij} > 0] \Rightarrow [a_{k'}^{im} = a_{k'}^{jn} + C_i], \forall i, j, k, k', m, n$
  - $[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \le a_k^{im}) \vee (a_k^{lo} \ge a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} \le a_{k'}^{im} - C_i] \wedge [P^{ij} \le 0] \Rightarrow [a_{k'}^{jn} = a_{k'}^{im} + C_j], \forall i, j, k, k', m, n$
  - $[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \le a_k^{im}) \vee (a_k^{lo} \ge a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} > a_{k'}^{im} - C_i] \wedge [a_k^{jn} < a_{k'}^{im}] \Rightarrow [a_{k'}^{jn} = a_{k'}^{im} + C_j], \forall i, j, k, k', m, n$
  - $[a_k^{im} < a_k^{jn}] \wedge [(a_k^{lo} \le a_k^{im}) \vee (a_k^{lo} \ge a_k^{jn})] \wedge [X_{kk'}^{ij} = 1] \wedge [a_k^{jn} > a_{k'}^{im}] \Rightarrow [a_{k'}^{jn} = a_k^{jn} + C_j], \forall i, j, k, k', m, n$

Research Contributions

- FlexRay:
  - A new ILP formulation for worst-case response time
  - Our formulation: is exact and also non-iterative (so computationally more efficient) compared to literature
  - SAE benchmark validations shows applicability to practical-sized problems
  - To appear in IEEE Transactions on Automation Sc. & Engr.
- TTEthernet:
  - A new MILP formulation for worst-case response time
  - Implementation and evaluation ongoing
- Future direction:
  - A MILP-based framework for system level end-to-end timing analysis
  - Integrate for system level assurance (eg, correctness of synchronous semantics under asynchronous execution)