Small cycle impact in pulsed queues

Yizhou Li (Huawei)

Guanhua Zhuang (Huawei)

Li Dong (Shenyang Institute of Automation)

Wenbin Dai (Shanghai Jiao Tong University)

Sept, 2021

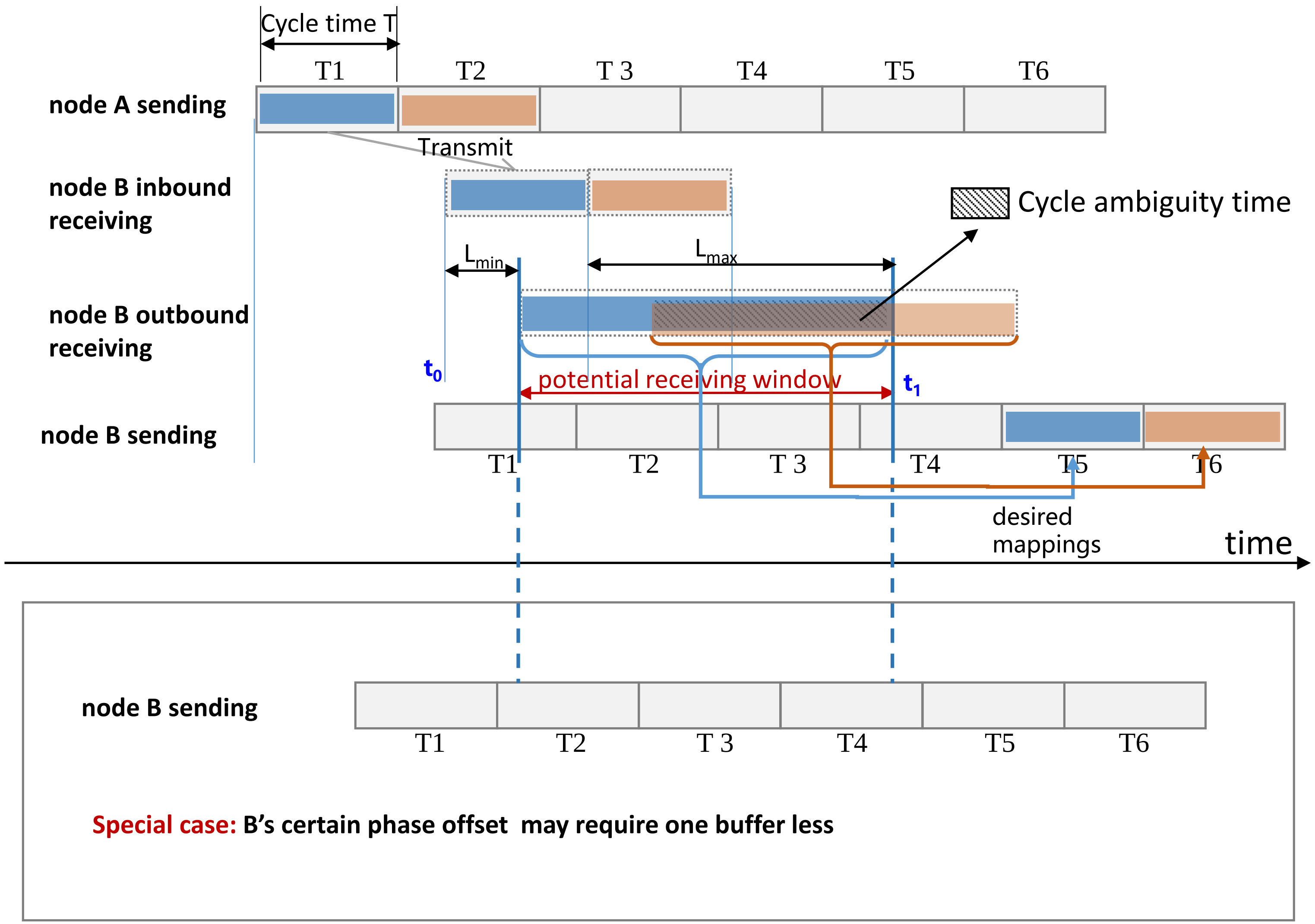
Introduction

- Pulsed Queues was presented in Nendica as a continuation of early discussions towards the PAR

(https://www.ieee802.org/1/files/public/docs2021/new-finn-pulsed-queuing-0821-v03.pdf)

- Suggested to use syntonization (frequency synchronization) instead of synchronization for CQF and its variance (CQF+).
- A measurement of phase offset is required to correctly match the cycles of two neighbor nodes.
- Goal of the slides
  - Potential impact in syntonized CQF/pulsed queues when the cycle time is small

Large receiving window time at the transmitter port may cause cycle identification ambiguity



- Assumptions:
  - per-hop latency variance between： Lmin & Lmax
  - Scheduled traffic takes full cycle time T
- Max receiving window time at the transmitter:

  $(T + L_{max}) – (0 + L_{min})$

  $= T + (L_{max} - L_{min})$
- # of receiving buffers to accommodate the max receiving window time:

  $\text{floor}((T + (L_{max} - L_{min}))/ T) + 2$

  $= \text{floor}((L_{max} - L_{min}) / T) +3$
- # of sending buffer = 1
- Total # of buffers = $\text{floor}((L_{max} - L_{min}) / T) +4$
- Special case: depending on B's phase offset relative to A, one buffer less may still work
- Example in left case:
  - $(L_{max} - L_{min})$ is 1.6T
  - Total # of buffers: 5 (or 4 in special case)
  - If $(L_{max} - L_{min})$ is < T, total # of buffers = 4 (or 3 in special case)

When such ambiguity has significant negative impact

- Cycle time T is small, and
- Processing latency variance (L_{max}- L_{min}) at node is large so that it is comparable to T

- Recall:
  - When (L_{max}- L_{min}) << T, to eliminate the cycle ambiguity：
    - Make potential receiving window always fall in a single cycle:
      - increase the dead time (not allowing sending traffic) in a cycle to absorb the processing latency variance
  - Not full utilization of 100% of cycle time T currently due to dead time
    - Dead time contributor 1: Guard band at the beginning of T for the interruption from lower priority traffic
    - Dead time contributor 2: Dead time at the end of T to absorb the latency variance and cycle shifting between two neighbor nodes
  - When T is small, the dead time can eat T up. This is not desired.

Cycle time T can be small in the order of 10x us

- Rough factors to determine cycle time T
  - allow at least one 1500B/max size packet to be sent within T
  - preferably multiple packets can be sent within T
  - T < e2e bounded latency requirement / # of hops (roughly, not the exact number)
- Existing CQF usage scenario: T is no less than 100x us
  - Sufficiently good for streaming traffic as e2e bounded latency requirement is in the magnitude of few ms
- Some observations:
  - With increasing of link speed, the same amount of data can be transmitted within a smaller cycle time

Cycle Time (µs)	**Buffer Size per Cycle (Byte)**		
	Link bandwidth		
	100Mbps	1Gbps	10Gbps
1	12.5	125	1250
1.2	15	150	1500
2	25	250	2500
4	50	500	5000
10	125	1250	12500
12	150	1500	15000
120	1500	15000	150000

Cycle time decreasing:
100x us -> 10x us -> few us

  - Smaller cycle makes CQF+ applicable to more strict e2e bounded latency requirement usage scenarios.
    - Application period requirement in industry automation [60802-d1-2]: 100 µs to 2 ms (isochronous), 500 µs to 1 ms (Cyclic-synchronous), 2 to 20 ms (Cyclic-Asynchronous), 100 ms to 1s as latency (Alarms and Events), 50 ms to 1 s (network control traffic), latency < 2ms (video), latency < 100ms (Audio/Voice)
- **Cycle time in the order of magnitude of ~10x us would be desired**

Processing latency variance at node is relatively large

- Store and forward time variance per packet

Bit_Rate	**Store and forward latency (us)**		
	Min frame size (64B)	Max frame size (1518B)	Latency variance
100Mbps	64 *8 / 100Mbps ≈ 5 us	1518 *8 / 100Mbps ≈ 121 us	116 us
1Gbps	0.5	12.1	11.6
10Gbps	0.05	1.21	1.16

- Switch Fabric Latency in incast case
  - For n port switch, latency variance = (n-1) * frame_size / bit_rate

Bit_Rate	**Incast Latency variance (us)**		
	16-port switch	24-port switch	32-port switch
100Mbps	116 us *16 ≈1856 us	2784 us	3712 us
1Gbps	185.6 us	278.4 us	371.2 us
10Gbps	18.6 us	27.8 us	37.1 us

- **Processing latency variance is not negligible (or even larger) when cycle time T is ~10x us**

Some Thoughts

- Cycle identification ambiguity is an issue when cycle T has to be small
- Consider the explicit cycle labeling to remove the ambiguity, especially when # of buffer >3
- Require the measurement of latency variance to estimate the potential receiving window time for a single cycle
    - # of buffers required
        - May need a guess at the very beginning for dry run
        - Adjust based on measurement to proper value
        - Monitor to check the violation of the assumed latency variance
    - Determine the cycle/buffer mapping relationship between neighbor nodes