Considerations for Isochronous Data Services over the Proximity-1 Space Link

Presented by Jay L. Gao

SpaceOps
22 June 2006, Rome Italy

Background & Definition

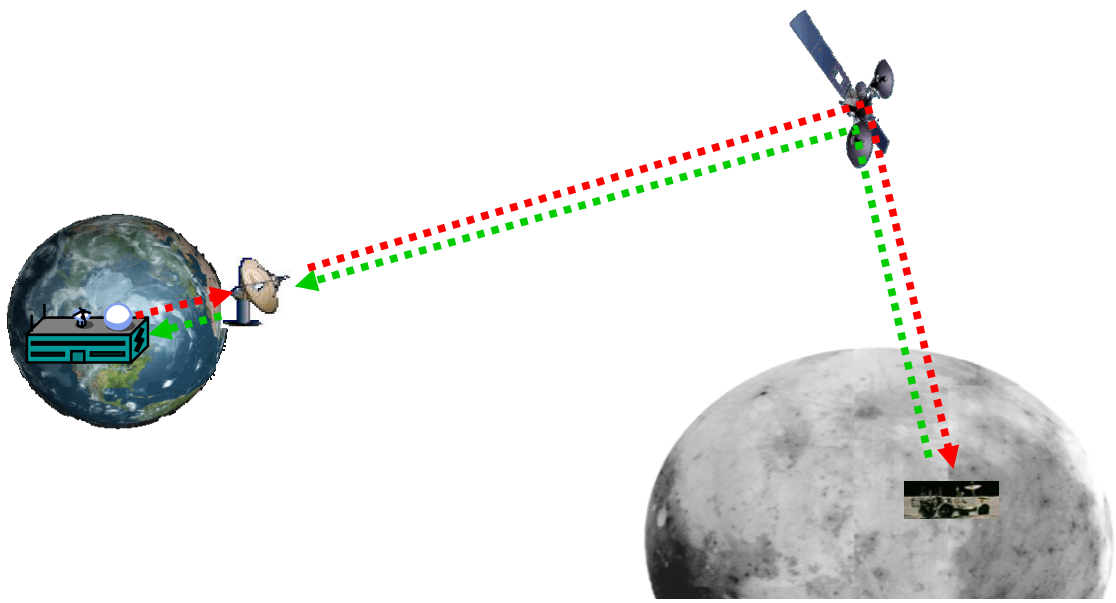
- Isochronous means “having equal time difference”
  - The term ‘Isochronous’ describes a system where the time difference between arrival and departure of data is constant
- However, one can also consider degrees of “isochronicity” by measuring the consistency in the latency of a system
  - Deterministic systems (such as TDMA) has higher degree of isochronous property because resource assignments are static
  - For system that dynamically shares resources, variations in latency can still be controlled
- Proximity-1 is not a deterministic protocol in the sense that it does not designate specific ‘time slots’ to the transmissions of data and control messages. However, is it still sufficiently isochronous for carrying real-time traffic?

Example of Isochronous Services

- Emerging lunar exploration missions requires isochronous services for the transport of real-time voice, command/control data, and video streams with very low delay jitter

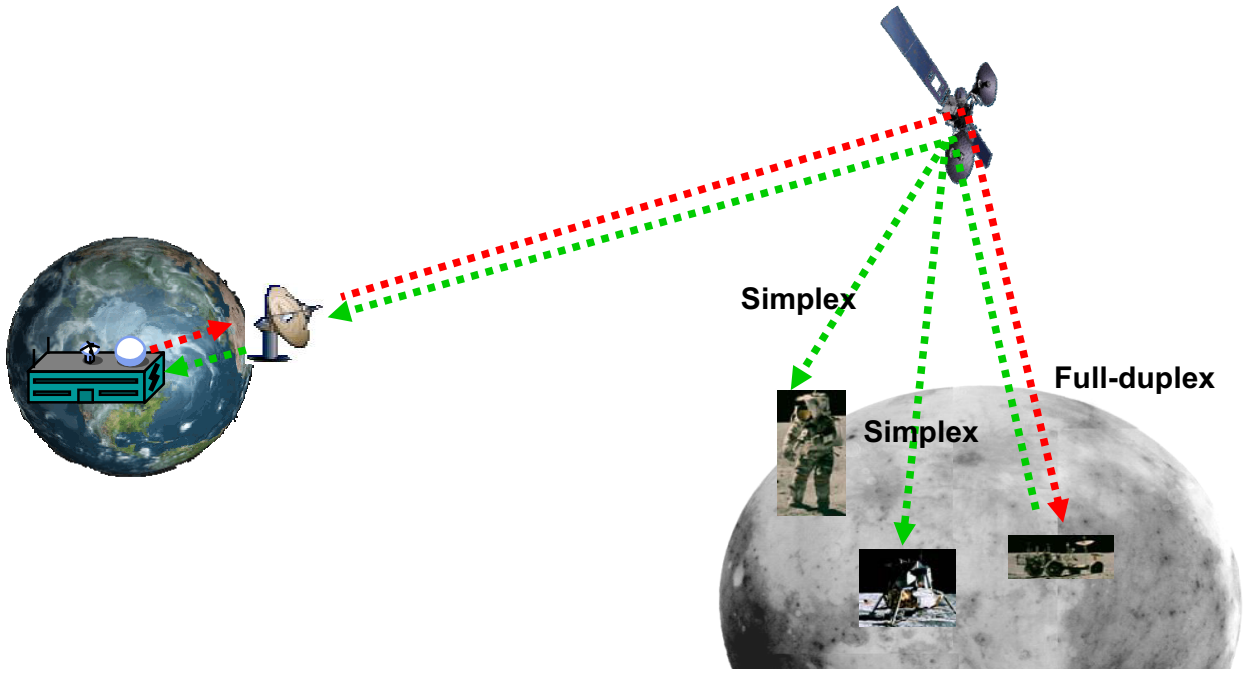
Robotic Tele-operation (1)

- Tele-operation from Earth
  - Forward links carry control data
  - assume re-transmission mechanism is not useful because of the real-time nature of control traffic
- Return traffic could be a mixture of reliable data type (e.g., science) and expedited data type (e.g., status and motion-imagery)



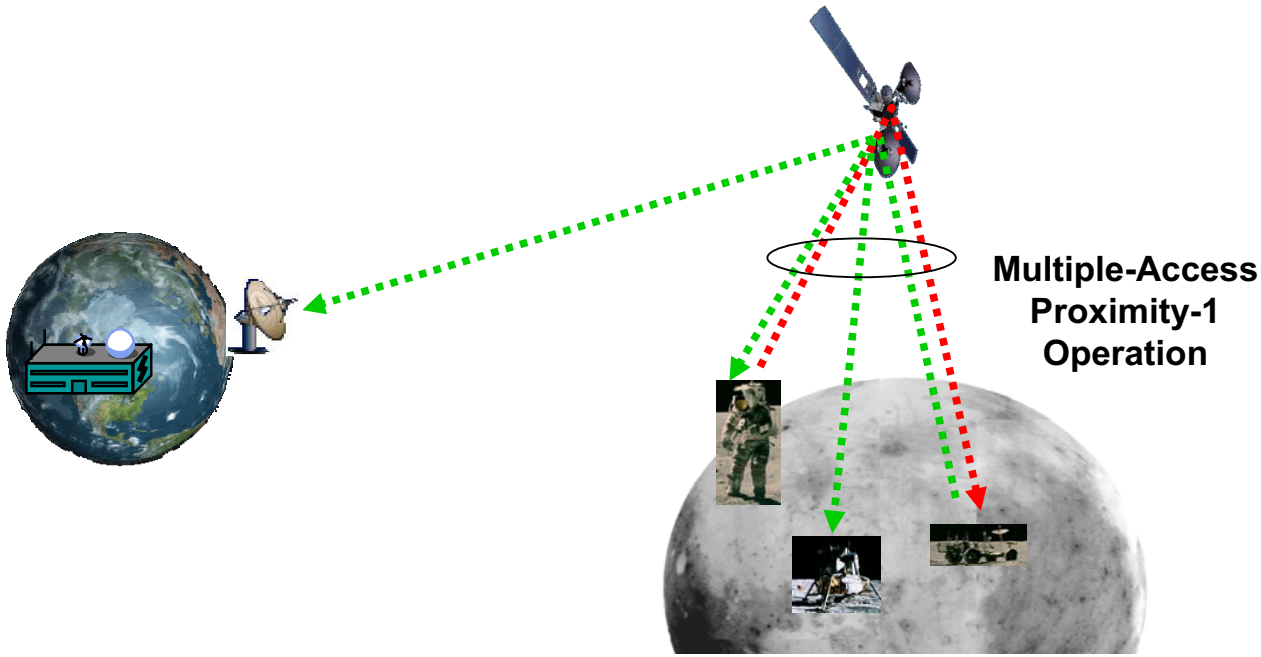
Robotic Tele-operation (2)

- Mission Ops tele-operating rover with EVA astronaut and LSAM monitoring progress and receiving copies of science/status telemetry
- One-to-many transmissions from orbiter to surface assets
  - Orbiter operates in full-duplex mode; EVA and LSAM in simplex mode (receive only)
  - Higher loading on forward link



Robotic Tele-operation (3)

- EVA astronaut tele-operating rover with Mission Ops and LSAM monitoring progress and receiving science/status telemetry
- Multiple-access Proximity-1 (one-to-many and many-to-one transmission for the orbiter)
  - Requires modification to current standard
  - Multiple access extension under investigation



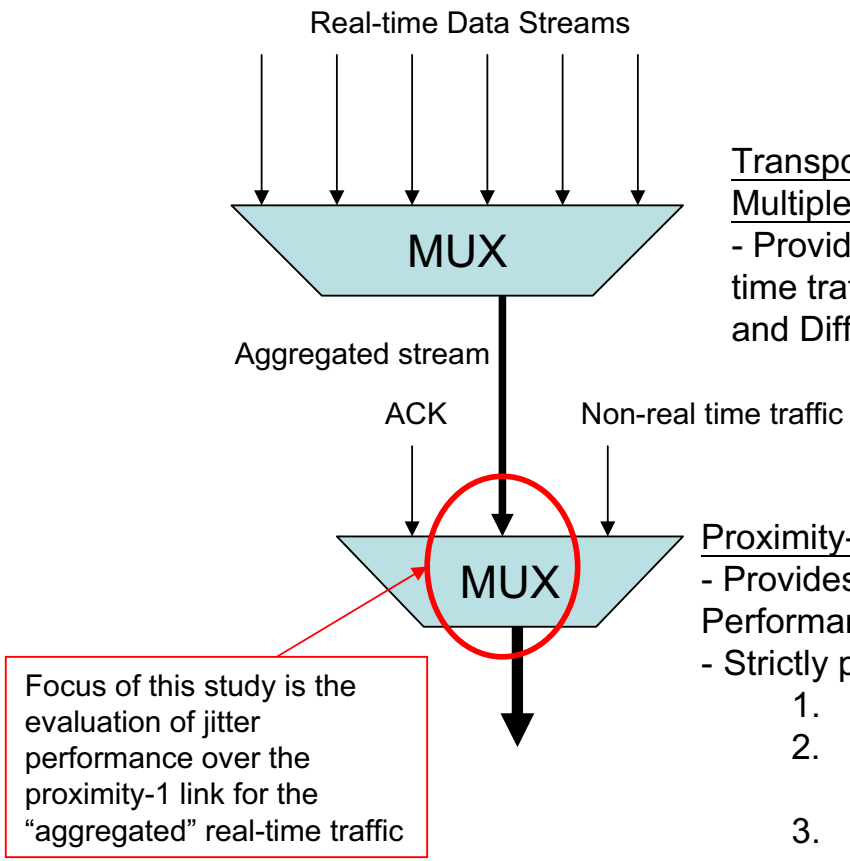
Metric for Isochronous Service

- Performance of isochronous service can be characterized by the *delay jitter* metric
  - Delay jitter metric is defined as the time interval between the arrival of a data unit to the link layer to the moment it gains access to the channel, i.e., began transmission over the underlying physical medium
  - We do not analyze queuing delay, only the head-of-the-queue delay arising from the Proximity-1's data link layer channel access mechanism (strict-priority handling of data and ACK frames)

- How does jitter hurt the end-to-end performance?
  - Real-time application uses codec with limits on process speed and tight jitter requirement to preserve timing. Jitter is mitigated by buffering at the receiving entity before decoding
  - The buffering at the decoder increases the end-to-end latency and degrades quality of voice and command-control loop applications

- What is tolerance for jitter?
  - For voice
    - Usually a 20 msec is suggested to get toll-quality voice
    - The quality of two-way conversation is affected by other delay introduced by codec, processing, FEC, propagation, and transmission over multiple links. Overall, ITU-T G.114's recommendation states that a150 msec is the maximum tolerable one-way end-to-end *delay* for voice conversation before subjective experience of users becomes awkward (i.e., talking over each other)
  - For video and command/control loop it is very application dependent

Metric for Isochronous Service (cont'd)



<u>Transport & Network Level Prioritization & Multiplexing</u>

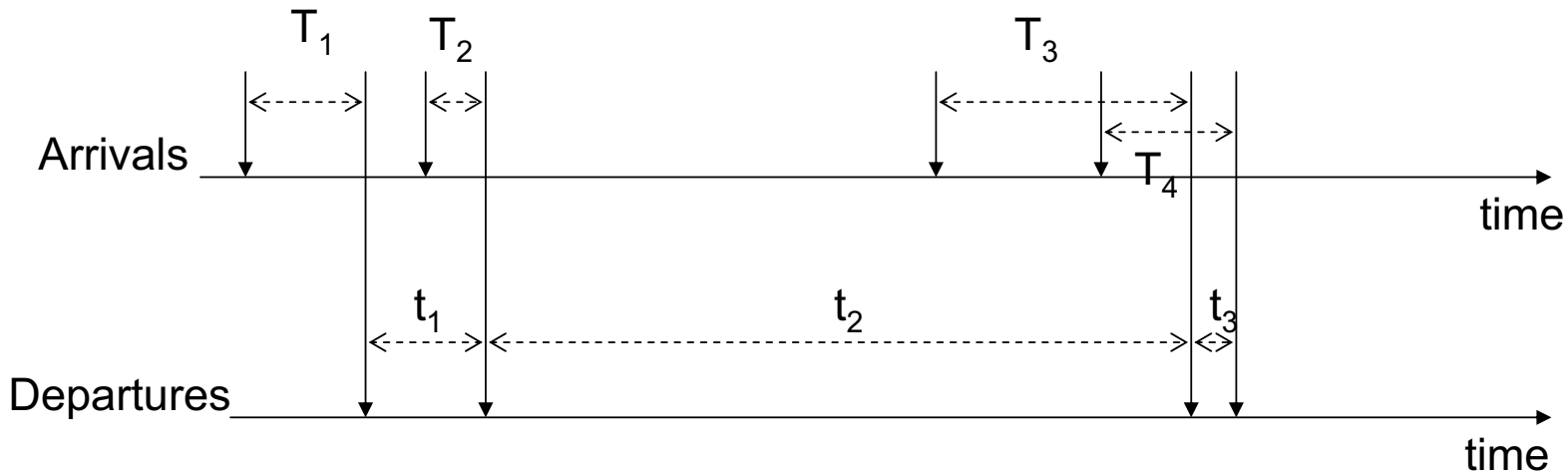
- Provides differentiated QoS for individual real-time traffic streams via mechanisms such as RTP and DiffServ

<u>Proximity-1 Prioritization & Multiplexing</u>

- Provides differentiated reliability and latency Performance
- Strictly priority mechanism deployed for:
    1. Protocol signaling & ACK
    2. Isochronous traffic (aggregated) – expedited frames
    3. Non-real time traffic – sequence-controlled frames

Jitter of Proximity-1

- Jitter here is defined as the variability of latency *through* proximity-1 layer (T), not the “inter-departure time” of frames (t).
- Real-time traffic is not necessarily constant rate

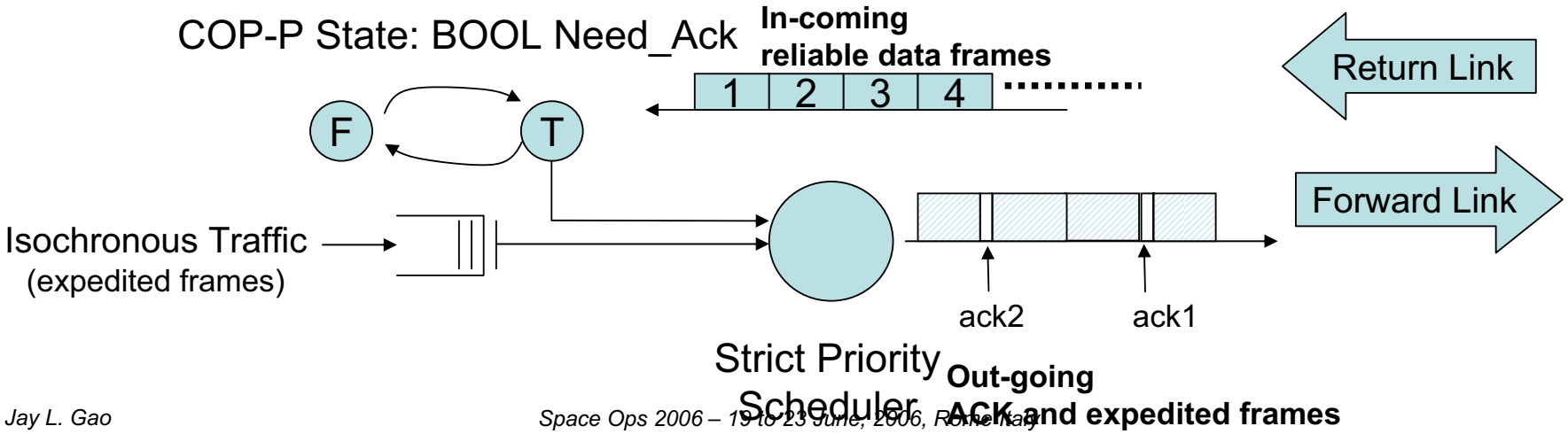


Frame Prioritization

- Currently Proximity-1 handles three types of frames:
  - Directives & ACK
  - Expedited Data
  - Reliable Data
- The protocol does not explicitly assign transmission time to each data type but arbitrates access to the channel based on a “strict-priority” mechanism
  - Highest priority goes to directives & ACK frames, then expedited data frames, then reliable (ARQ) data frames

Scenario Analyzed

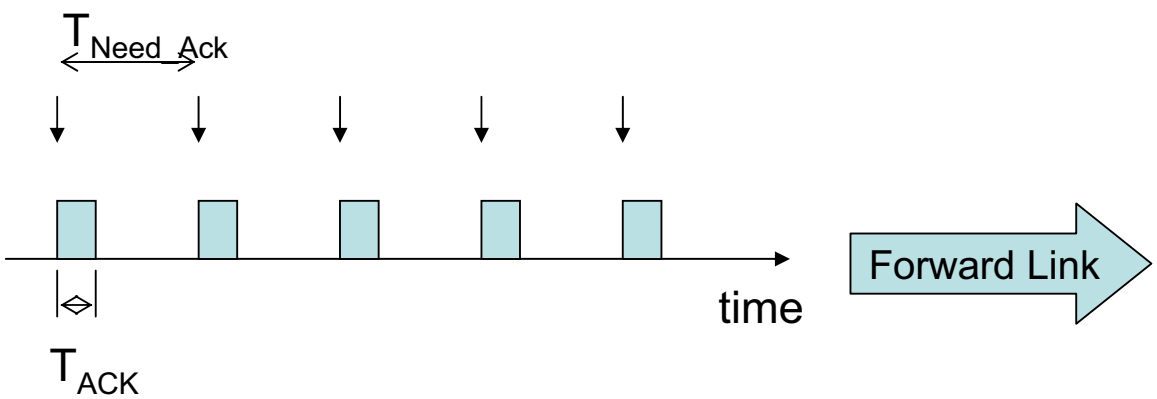
- The following scenario is used for analysis of the jitter performance of proximity-1:
  - On the forward link, isochronous traffic (voice, command loop, and video) is carried by expedited frames since re-transmissions does not make sense for real-time data
  - On the return link, Proximity-1 is supporting a reliable data transfer (using ARQ) and generating acknowledgement (ACK) traffic on the forward link as well
- Jitter is created by the contention for the channel between the expedited data frames and acknowledgement frames



Acknowledgement Traffic

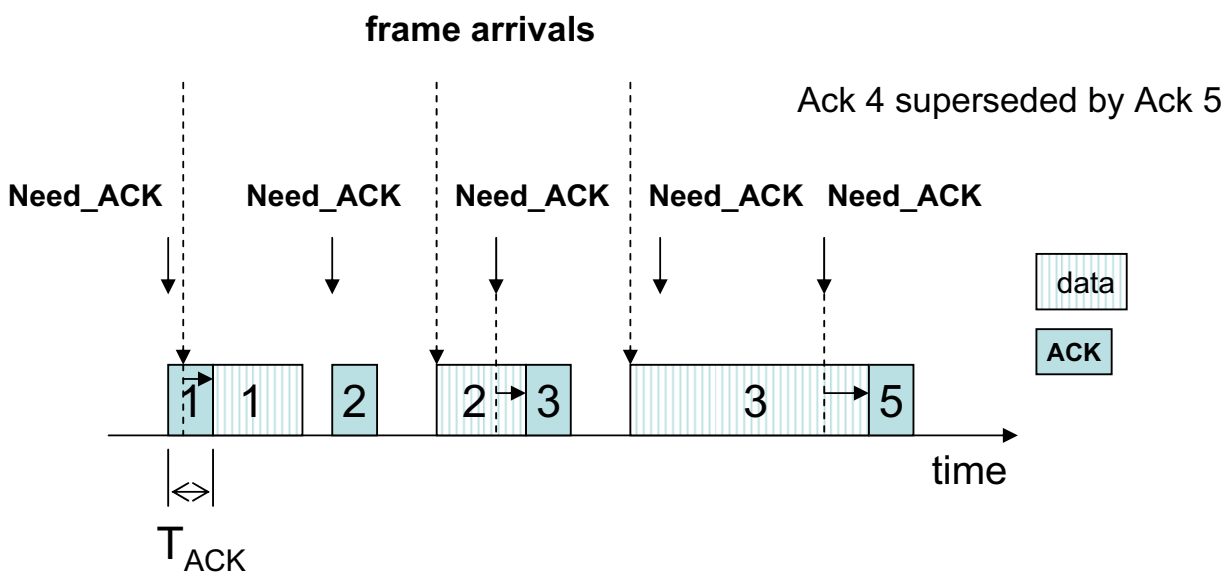
- Nominally, ACK transmissions are triggered on a periodic basis by arrival of data frames from the other peered Proximity-1 protocol entity. The periodicity is equal to the arrival rate of the reliable data frames.
- If there are no data traffic on the forward link, then ACK frame will be sent out at the same periodicity
- The residual capacity, normalized to the data rate, on the forward link is:

$$\rho_{data} < 1 - \rho_{ACK} = 1 - \frac{N_{plcw}}{N_{pltu_ret}} \frac{R_{return}}{R_{forward}}$$



Contention Between ACK and Data Frames

- However, when there is data traffic, there is contention.
- While ACK traffic has strict priority over data traffic, depending on the relative timing between the arrival time of the data traffic and the triggering of ACK transmission, either traffic could be pre-empted by the other and create *variable delay* through the system



Light Loading Conditions

- $\rho_{data} << 1 - \rho_{ack}$
  - Intuitively this means that the arrival of data frames are sufficiently separated in time that their contention process with the ACK frames do not couple. i.e., the jitter experienced by each data frame are essentially independent

- The maximum jitter experienced by an arriving data frame is the transmission time of the ACK frame
  - When a data frame arrives at the beginning of an ACK frame transmission

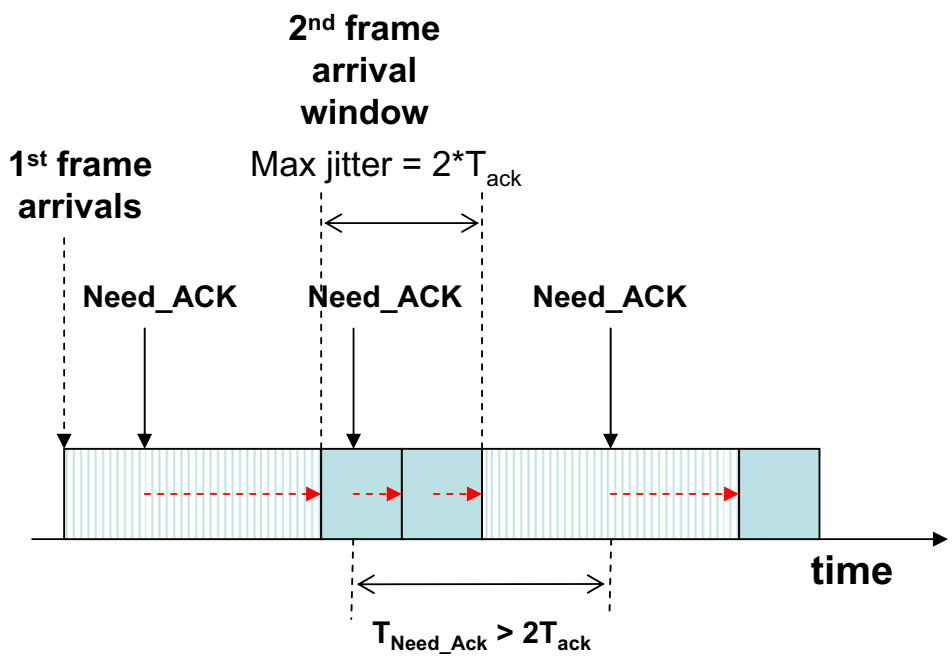
Light Loading Conditions (cont'd)

- ARQ Performance on the return link:
  - Sub-case 1: Expedited frame size is "small"
    - Transmission time of data frame < periodicity of the ACK frames
    - ACK frames experience some jitter but no gap; minimal impact on ARQ process
  - Sub-case 2: Expedited frame size is "large."
    - Transmission time of data frame > n * periodicity of ACK frames
    - Gaps in ACK appears, causing retransmissions of data on the return link
    - ARQ throughput degrades
- Fix: *use larger window size to defer retransmissions and maintain throughput until the end of the ACK gap*
- *No need to change current Proximity-1 Data Link Specification*

$$\eta_{ARQ} = 1 - \lambda_{data_frame} \cdot \max\left(\left\lfloor \frac{N_{pltu_for} \cdot R_{return}}{N_{pltu_ret} \cdot R_{forward}} \right\rfloor, \left\lceil \frac{T_{round-trip} \cdot R_{return}}{N_{pltu_ret}} \right\rceil - 1 \right) \cdot \frac{N_{pltu_ret}}{R_{return}}$$

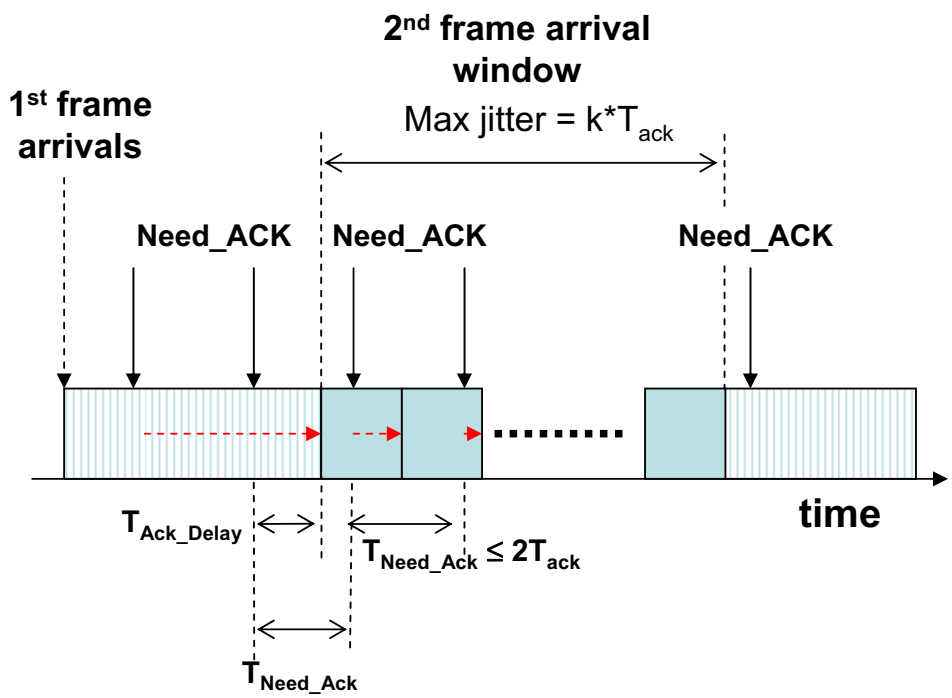
High Loading Condition

- High loading means the total traffic (data + ACK) is significant compare to the capacity of the forward link
- Sub-case 1: $\rho_{ack} < 1/2$
  - The acknowledgement traffic takes less than half of the forward link capacity
  - An arriving data unit maybe blocked by at most two consecutive ACK frames (caused by first ACK frame being deferred sufficient that an ARQ state change occurs during its transmission)
  - Maximum jitter is 2 * transmission time of ACK frame on the forward link



High Loading Condition (cont'd)

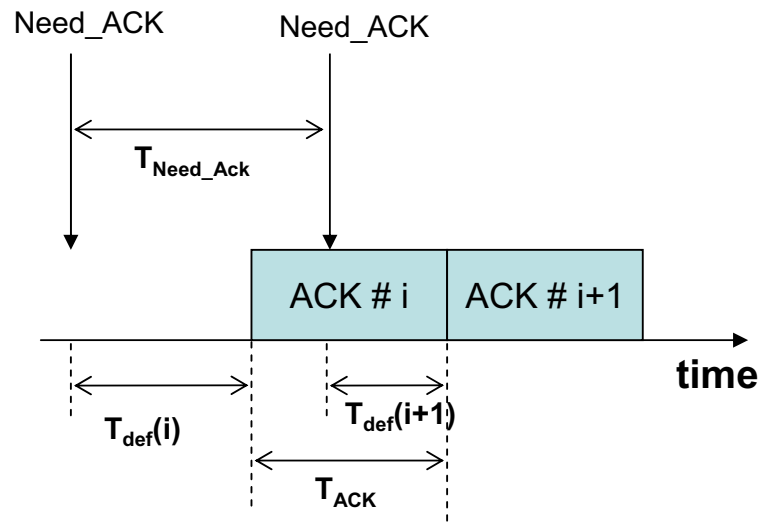
- Sub-case 2: $1/2 \leq \rho_{ack} < 1$
  - The acknowledgement traffic takes more than half of the forward link capacity
  - An arriving data frame maybe blocked by a burst of k consecutive ACK frames
  - Maximum jitter is k * transmission time of ACK frame on the forward link



High Loading Condition (cont'd)

- The maximum jitter is basically the maximum number of ACKs in a burst
- Close-form solution for the maximum jitter can be found
- A recursive relationship exists for the T_{def} parameter, which is the time an ACK frame is deferred during a burst of ACK frames

$$T_{def}(i+1) = T_{ACK} - T_{Need_Ack} + T_{def}(i)$$



- Solving the iterative relationship, one can compute the maximum jitter to be,

$$D_{\max_jitter} = T_{ACK} \max_{i>0} \left\{ T_{def}(i+1) > 0, T_{def}(1) = T_{Need_ACK} \right\} = T_{ACK} \left\lceil \frac{T_{Need_ACK}}{T_{Need_ACK} - T_{ACK}} \right\rceil = T_{ACK} \left\lceil \frac{1}{1-\rho_{ACK}} \right\rceil$$

Contention with Sequence-Controlled Frames

- If sequence-controlled frames are also present on the forward link, the maximum jitter will increase by the transmission time of one data frame
- Due to the strict-priority of the expedited frame over sequence-controlled frame, this increase in jitter will remain constant regardless of the loading of the sequence-controlled traffic
- The increase, however, could be substantial since a data frame could be up to 100 times larger than an acknowledgement frame, so it is recommended that user properly configure the packet/frame size based the possible performance impact.

Recommendations

- Jitter performance is derived for various loading conditions:
  - Under light loading condition, the maximum jitter is bounded by the transmission time of one ACK frame
  - Under high loading condition the maximum jitter will increase proportionally with: 1/(1 – normalized loading of ACK traffic)
  - The presence of sequence-controlled frame increase jitter by transmission time of one data frame on the forward link
- Recommendations:
  - Under light loading conditions, no change is needed for the protocol
    - Transmission time of an ACK frame at 512kbps is about 0.2 msec, jitter due to channel contention is minimal
    - Typical VoIP stream requires 8 to 64kbps throughput (depending on the codec and quality needed), which should be easily supported by future radios with high forward link capacity
    - Impact on ARQ can be minimized by using larger window size

Recommendations (cont'd)

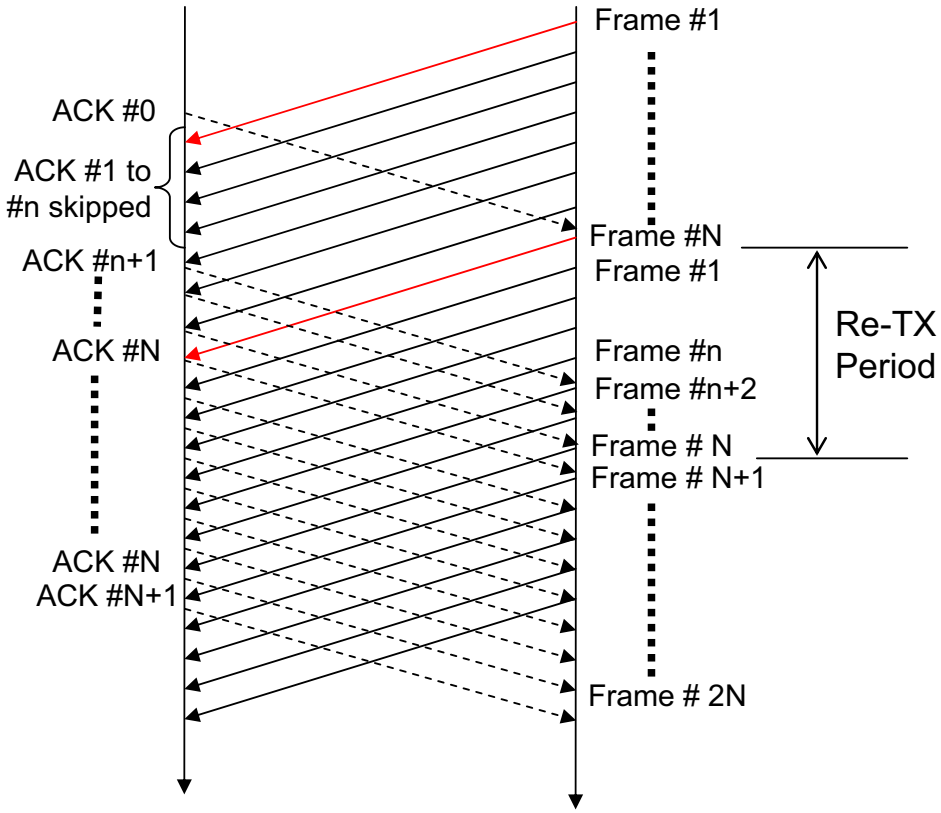
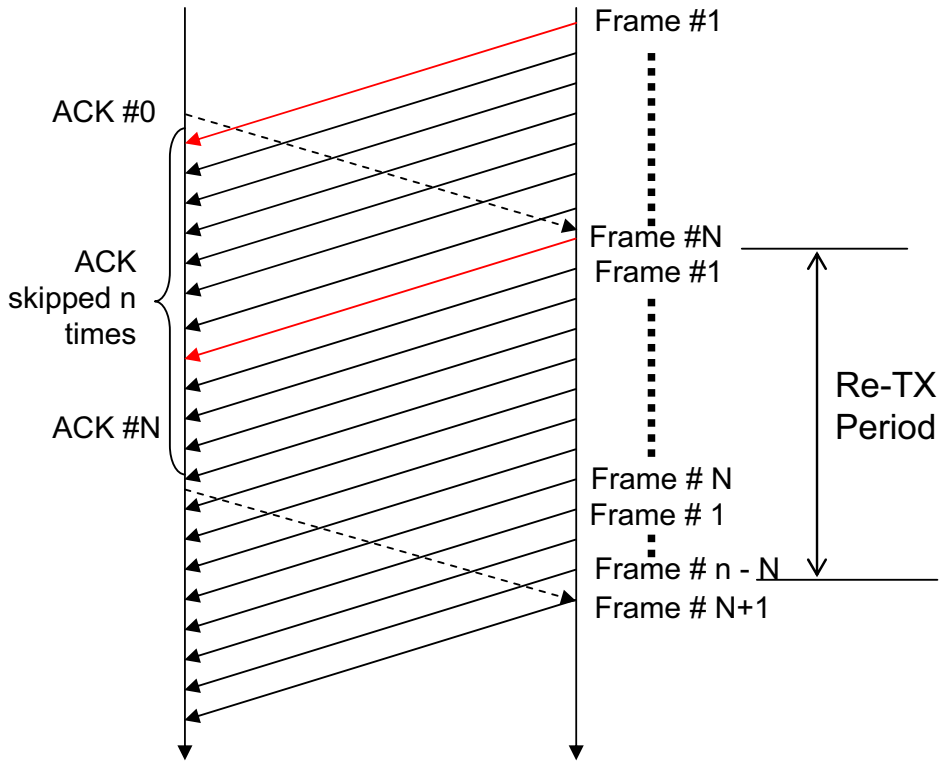
- Recommendations:
  - In the presence of sequence-controlled frames on the forward link, the increase of jitter can be mitigated by user configurations that limit the data frame size
  - Under higher loading, changes to the protocol could be considered
    - With highly asymmetric forward and return link data rates (> 20dB higher return link capacity) and high isochronous traffic load, long ACK frame bursts on the order of 100+ could create significant jitter (>20msec), which is significant part of the end-to-end latency budget
    - Fix #1: Piggy-back ACK information in the Isochronous frames via the OCF option to reduce the number of ACK frames
      - May not be feasible because the frame rate of the isochronous traffic may not matching frame rate for the ACK traffic
      - Requires additional algorithm to determine whether to create separate ACK frame or piggy-back (could be left as an implementer's choice)
      - May change code-block length
    - Fix #2: Implement weighted-fair or other algorithms that dynamically controls channel access based on latency/priority
      - Modification to current standard's strict-priority scheme
      - This is a preferred choice given that it is more general and can provide differentiated latency performance over multiple isochronous streams on the forward link

Other Considerations

- This analysis focused on jitter arising from data and ACK frame contention and did not address jitter arising from contention among multiple streams of data frames on the forward link
  - It is anticipated that using a dynamic algorithm to control channel access can provide a comprehensive solution for minimizing jitter across a variety of traffic sources and jitter requirements
- This analysis does not address the performance of systems that queues multiple ACK frames
  - The Proximity-1 specifications requires that only the latest ARQ state be sent in an ACK message when the channel becomes available, so the system should only keep one ACK frame in the buffer and always update the content of this ACK message, prior to transmission, to reflect the latest changes in ARQ state
  - If an implementation used a FIFO queue instead of updating a buffered ACK frames, then jitter will be higher.
- This analysis did not address queuing delay since it is tightly coupled to flow control issue that depends on higher layer protocols. Most system does not maintain long queues and will drop real-time traffic during link outage or “congestion periods” because the data quickly lose its timeliness to the application.
- This analysis did not assume any particular coding or physical layer characteristics, except that a full-duplex operation is supported at constant data rate. If the Proximity-1 data link layer is applied on top of alternative coding or PHY, the result of this analysis still applies

Backup Charts

ARQ with an ACK gap (n<N)



ARQ with an ACK gap (n>N-1)