BULK TRANSFER

Sara Shakil Qureshi

AGENDA

- Bulk Transfers - Introduction
- Maximum Data Packet Sizes
- Bus Bandwidth Allocation
- Bulk Transfer Data Sequences
- Error Recovery
- Stream Protocol
- Bulk Streams

Bulk Transfers

- Bulk transfers are useful for transferring data when time isn't critical. Bulk transfers are designed to support large communication at variable available bandwidths and at multiple times.
- If a bulk transfer has to be carried on, then it is sure that

1- Access to USB is on bandwidth-available basis.

2- Guaranteed delivery of data, but no guarantee of bandwidth or latency.

SAME IN BOTH VERSIONS

Stream Pipes

- Stream pipes are always uni-directional in their communication flow.
- If bidirectional data is needed, two bulk pipes must be used, each uni-directional. And follows FIFO.
- A stream pipe to a device is bound to a single device endpoint number in the appropriate direction.

SAME IN BOTH VERSIONS

Data Packet Sizes

- Endpoint is responsible for specifying the maximum data packet payload sizes, excluding PID and CRC bits.
- In USB 2.0, they are 8-, 16-, 32- or 64-bytes for full speed and 512 bytes for high speed.
- In USB 3.0, they are 1024 bytes as specified in the endpoint descriptor.

ENHANCED IN USB 3.0

Data Packet Sizes

- USB does not require that data payloads transmitted be exactly the maximum size. They can be equal or less than the maximum value.
- HOST has the responsibility to manage packet sizes.
- When a bulk IRP (IN/OUT Request Packet) involves more data than can fit into one maximum-sized data payload, all data payloads are of the maximum required size except the last one.

SAME IN BOTH VERSIONS

Data Packet Sizes

- A bulk transfer is complete when endpoint:

1- Has transferred exactly the amount of data expected.

2- Transfers a packet with a payload size less than maximum or transfers zero-length packet.

- Instead of zero-length packet, in USB 3.0, endpoint responds with a STALL handshake (Endpoint is halted or a control request is not supported).

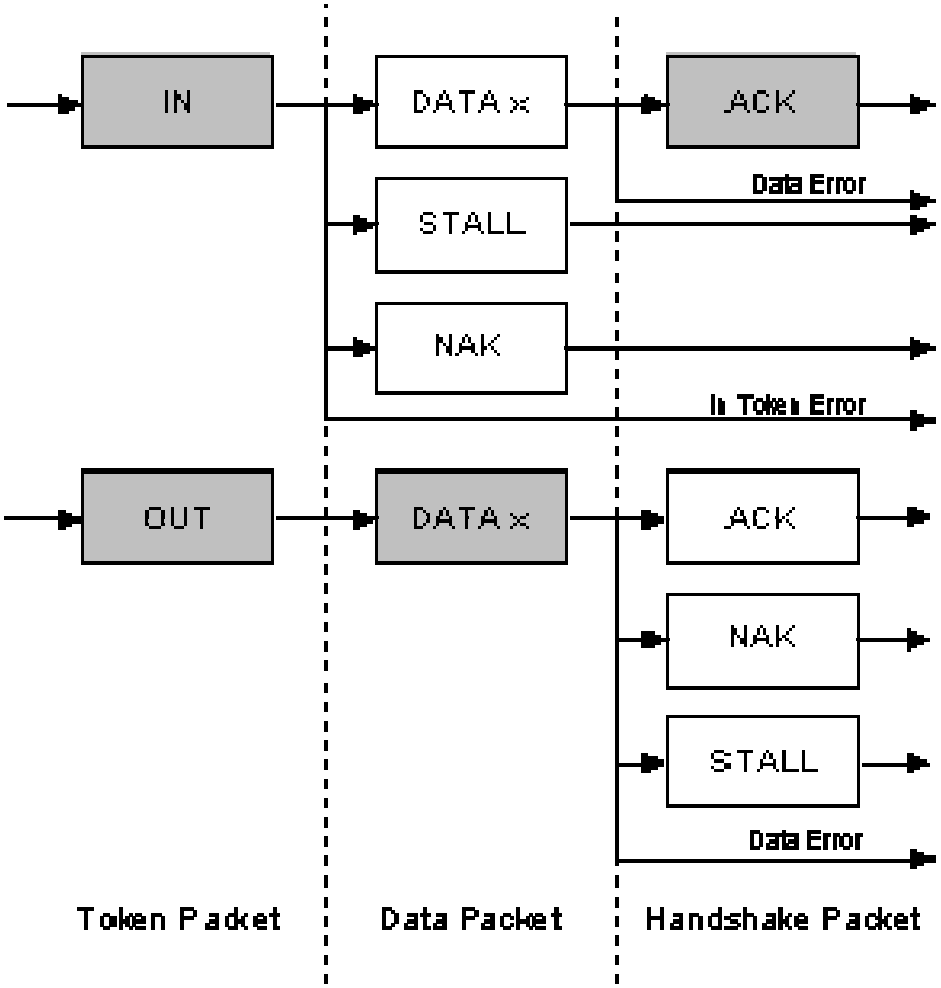
SMALL VARIATION IN USB 3.0

Data Packet Sizes

- If a data packet is received that is larger than the expected size, all pending IRPs for that endpoint will be aborted/retired.

SAME IN BOTH VERSIONS

IN/OUT Transactions



Bus Bandwidth Allocation

- Endpoint has nothing to do with allocation of bus bandwidth. Transactions occur on a bandwidth available basis.
- Of the bus bandwidth available, USB balances the access for all bulk pipes and IRPs through "Good Effort" delivery of data between client software and functions.

SAME IN BOTH VERSIONS

Priority of Bulk Transfer

- Control Transfers get around 10% of the total available bus bandwidth and periodic transfers (Interrupt and Isochronous) get maximum of 90%.
- Bandwidth left by the above transfers is allocated for the Bulk transfers. Thus bulk transfer has the lowest priority.

SAME IN BOTH VERSIONS

Bus Bandwidth Allocation

- If there are bulk transfers pending for multiple endpoints, they are then selected according to FAIR ACCESS POLICY that is Host Controller implementation-dependent.
- Bus time made available to a software client and its endpoint can vary depending upon insertion and removal of other devices and also as bulk transfers are requested for other devices too.

SAME IN BOTH VERSIONS

Data Sequences & Error Recovery

- Endpoint sets the data sequence as requested by host in the control transfer and then host also sets the same.
- Halt condition results in removal of all IRPs and is removed by software intervention through separate control pipe.
- There is error recovery and data integrity is more important than data rate.

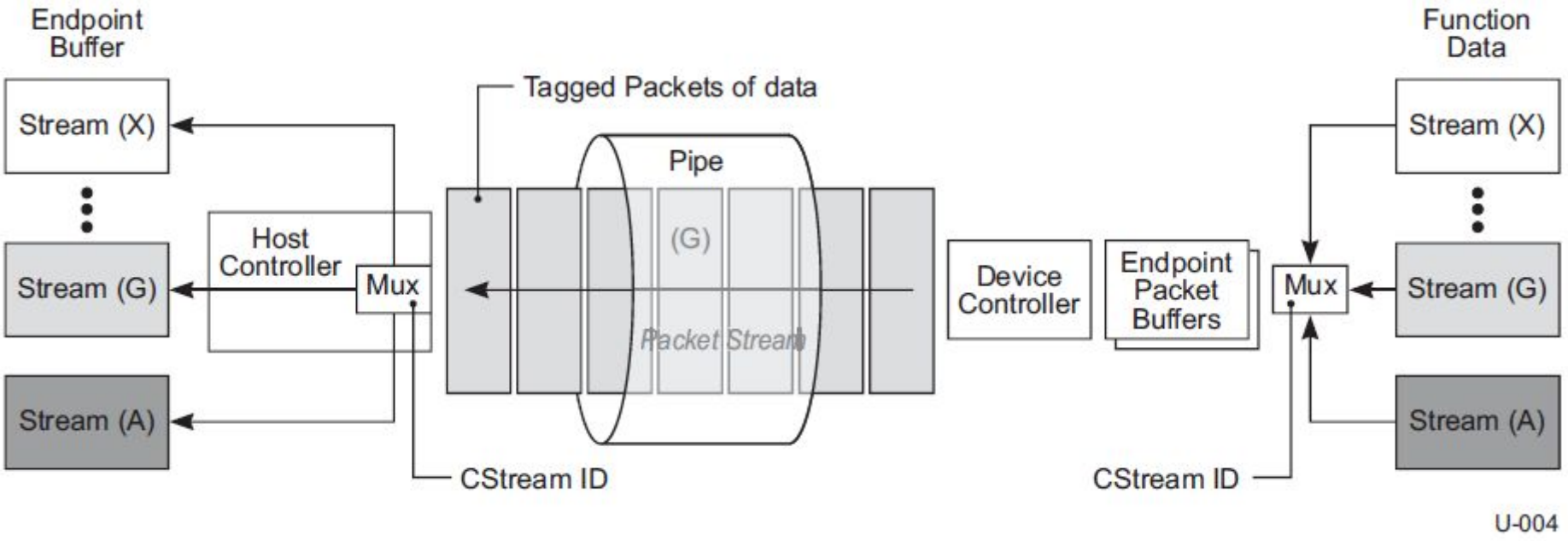
SAME IN BOTH VERSIONS

Bulk Streams

- Super speed streams used multi-stream model and utilize “stream” pipe communications mode.
- Stream Protocol is used and current stream IDs are used.
- Each endpoint has an associated CID.

NEW IN USB 3.0

Bulk Streams



NEW IN USB 3.0

Bulk Streams

- Each endpoint has maximum of 65533 buffers with buffers having 1:1 mapping with CID.
- Either host or device can initiate selection of CID but this can be rejected too.
- Termination of stream.
- Dynamic initialization of streams by host.
- Halt of pipe stops all stream activity.

NEW IN USB 3.0

WHY DO WE NEED MULTI-STREAM, IF WE HAVE BULK TRANSFER IN USB 3.0?

QUESTIONS