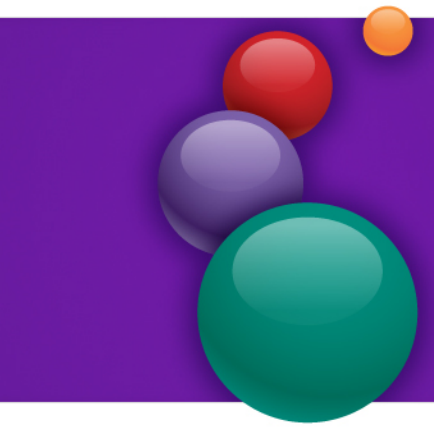
Gigabit Debug over IP Sockets: A Technical Overview



Jason Peck
Texas Instruments, Inc.
MIPI Debug WG Vice Chair

[21 September 2016]

Legal Disclaimer

The material contained herein is not a license, either expressly or impliedly, to any IPR owned or controlled by any of the authors or developers of this material or MIPI®. The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and MIPI hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL.

All materials contained herein are protected by copyright laws, and may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of MIPI Alliance. MIPI, MIPI Alliance and the dotted rainbow arch and all related trademarks, tradenames, and other intellectual property are the exclusive property of MIPI Alliance and cannot be used without its express prior written permission.

IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR MIPI BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

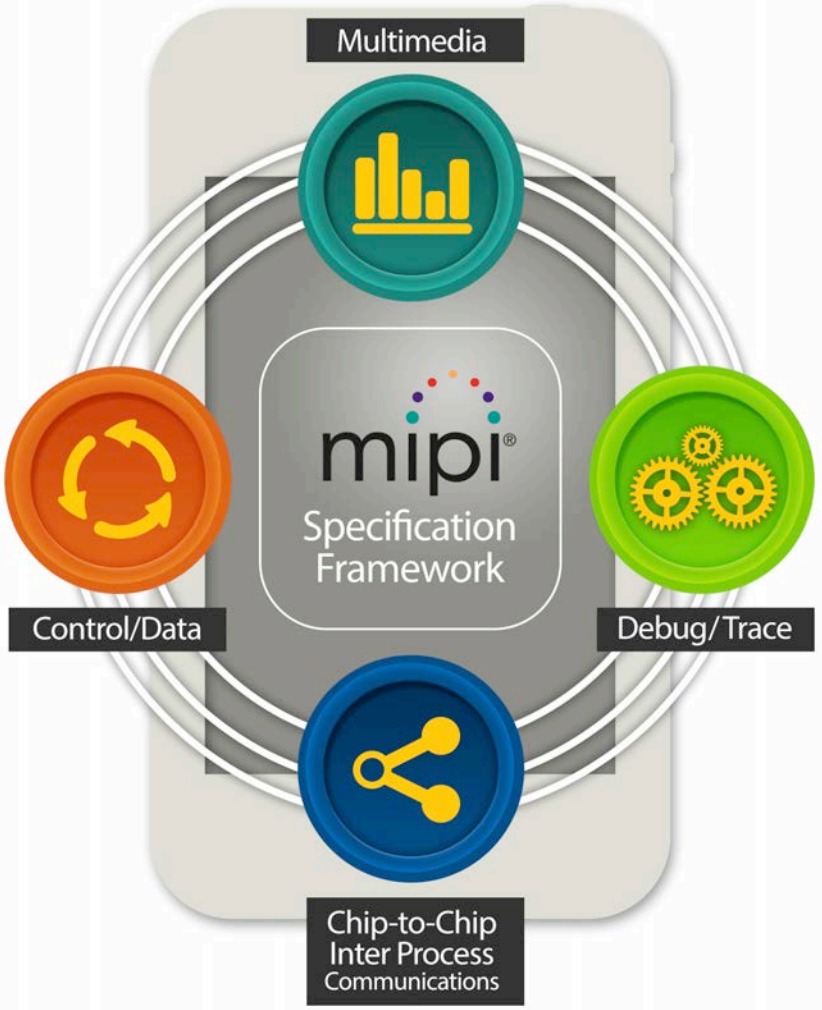
Agenda

- MIPI Overview
- MIPI Debug WG
- Gigabit Debug Framework Overview
- Gigabit Debug over IP Sockets
- Conclusion
- Q & A

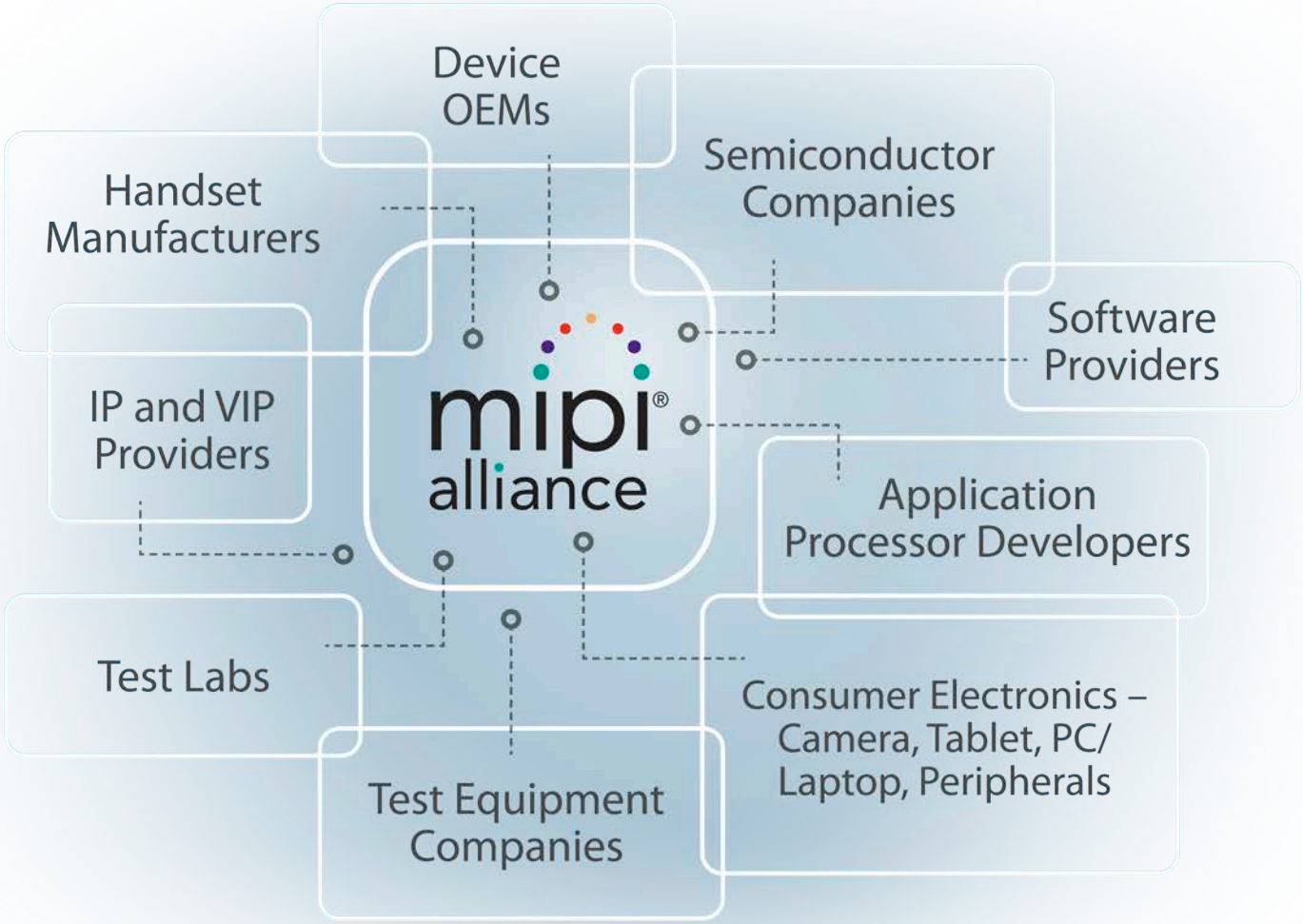
About MIPI Alliance

We are a global, collaborative organization comprised of over 280 member companies spanning the mobile and mobile-influenced ecosystems.

MIPI Alliance is leading innovation in mobile interface technology.



MIPI Alliance Member Ecosystem



Active Technical Working Groups

Camera	Debug	Display	Low Latency Interface
Low Speed Multipoint Link	PHY (C/D/M)	Reduced Input Output	RF Front End
Sensor / I3C℠	Software	Test	UniPro℠

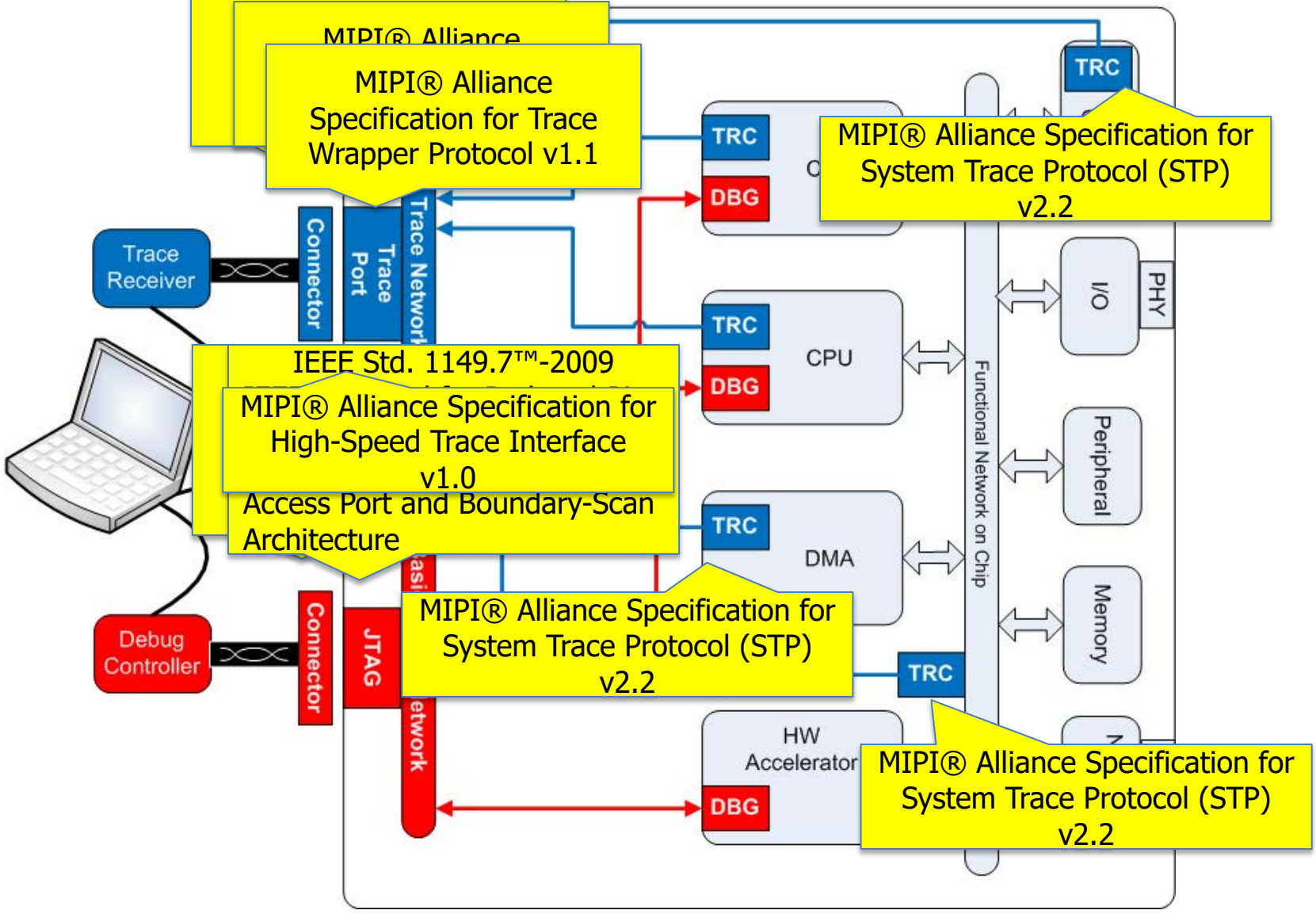
Workgroup Interaction

- WG consists of ~8 actively participating companies (SOC, IP, and Tools Vendors)
  - New faces are welcome, please join!
- Teleconferences every other Tuesday at 1600 or 1700 UTC
  - Summer/daylight time switch
  - 90-120 minutes depending on the agenda
- Three F2F sessions with all MIPI WGs
  - Generally meet for 4 days

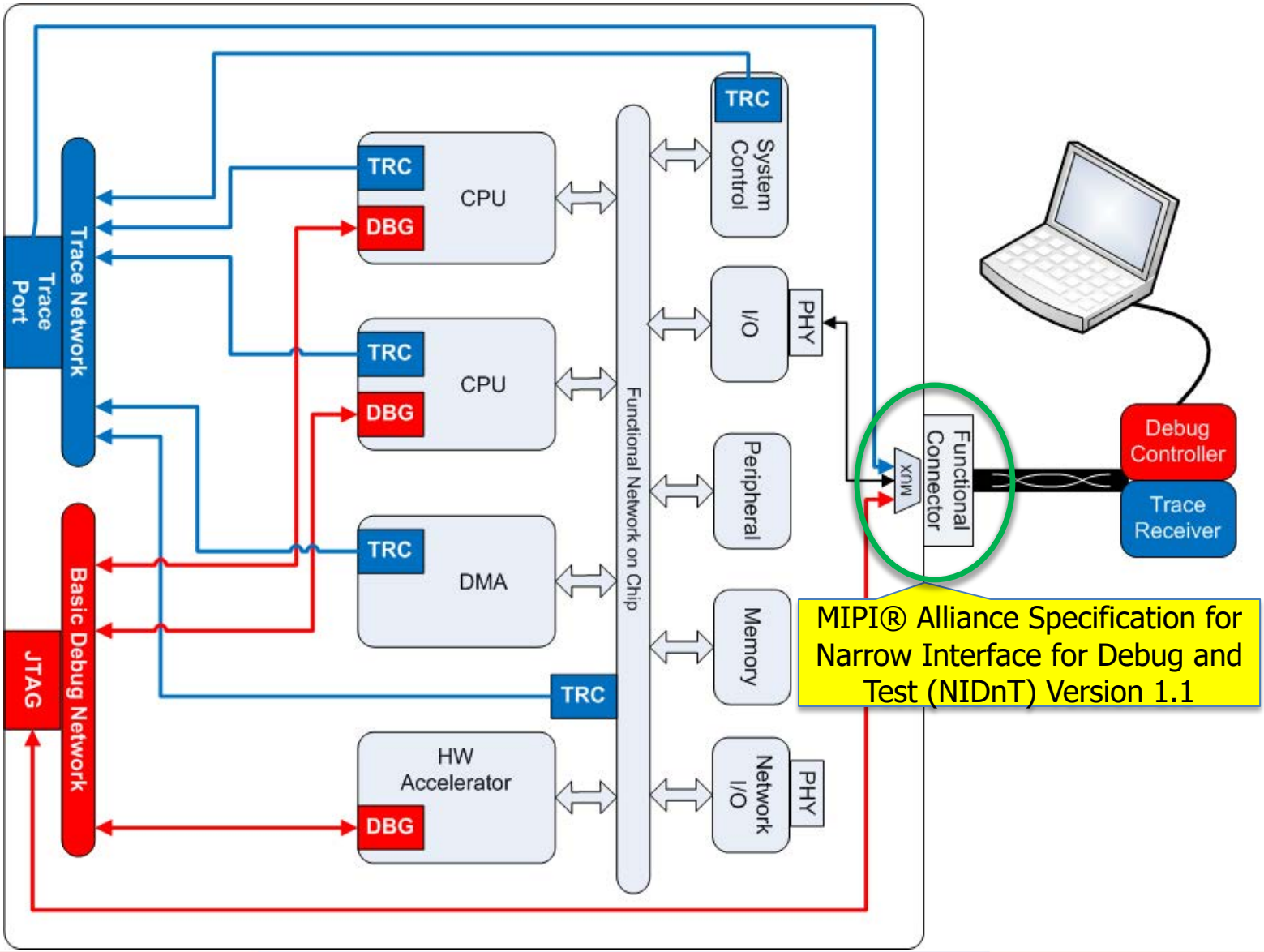
History of the MIPI Debug WG

- Formed as a MIPI IG and became a WG in 2004
- Always focused beyond mobile
  - Debug is a fairly universal problem space
- Initial focus on consolidating debug interfaces
  - Min Pin/1149.7
    - Reduced pin JTAG was spun out of MIPI to IEEE in 2006
  - Trace interface
  - Connectors
- Moved deeper into the platform (SoC and system HW)
  - Low level trace transport and merging protocols
- Current focus
  - Debug in form factor and remote debug
    - Debug reuse of functional interfaces
    - Debug over functional interfaces and networks

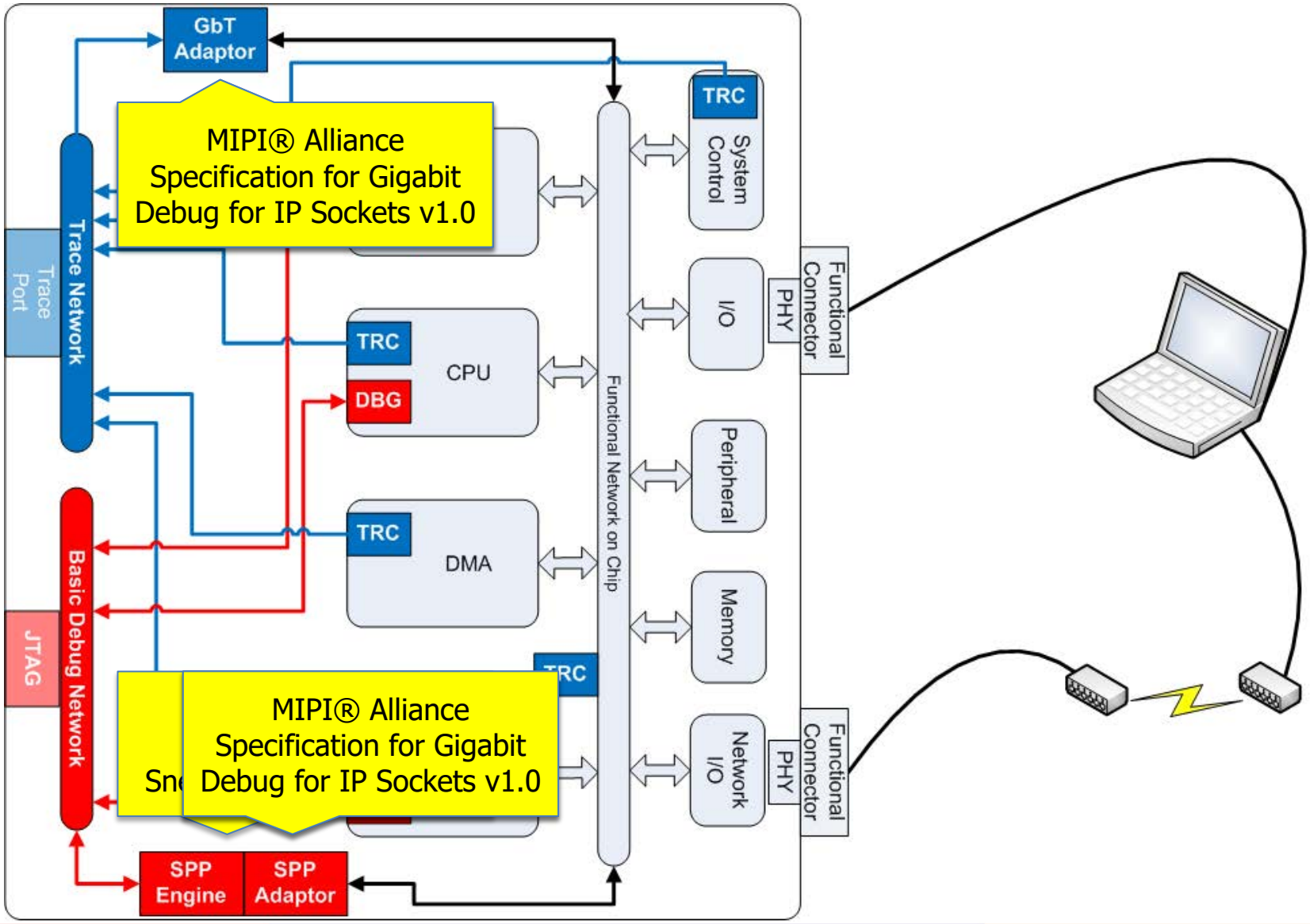
MIPI Debug Specifications and the SoC



MIPI Debug Specifications and the SOC : NIDnT



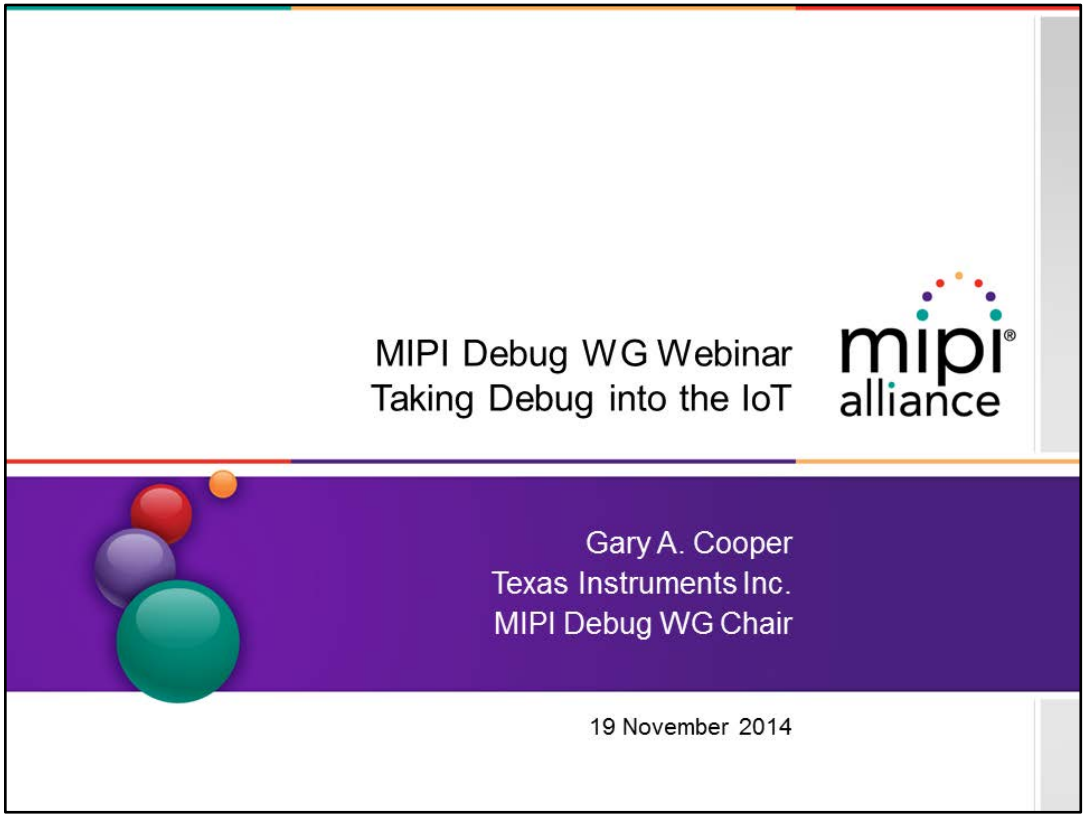
MIPI Debug Specifications and the SOC : GbD



Gigabit Debug Framework Overview

Gigabit Debug Fundamentals

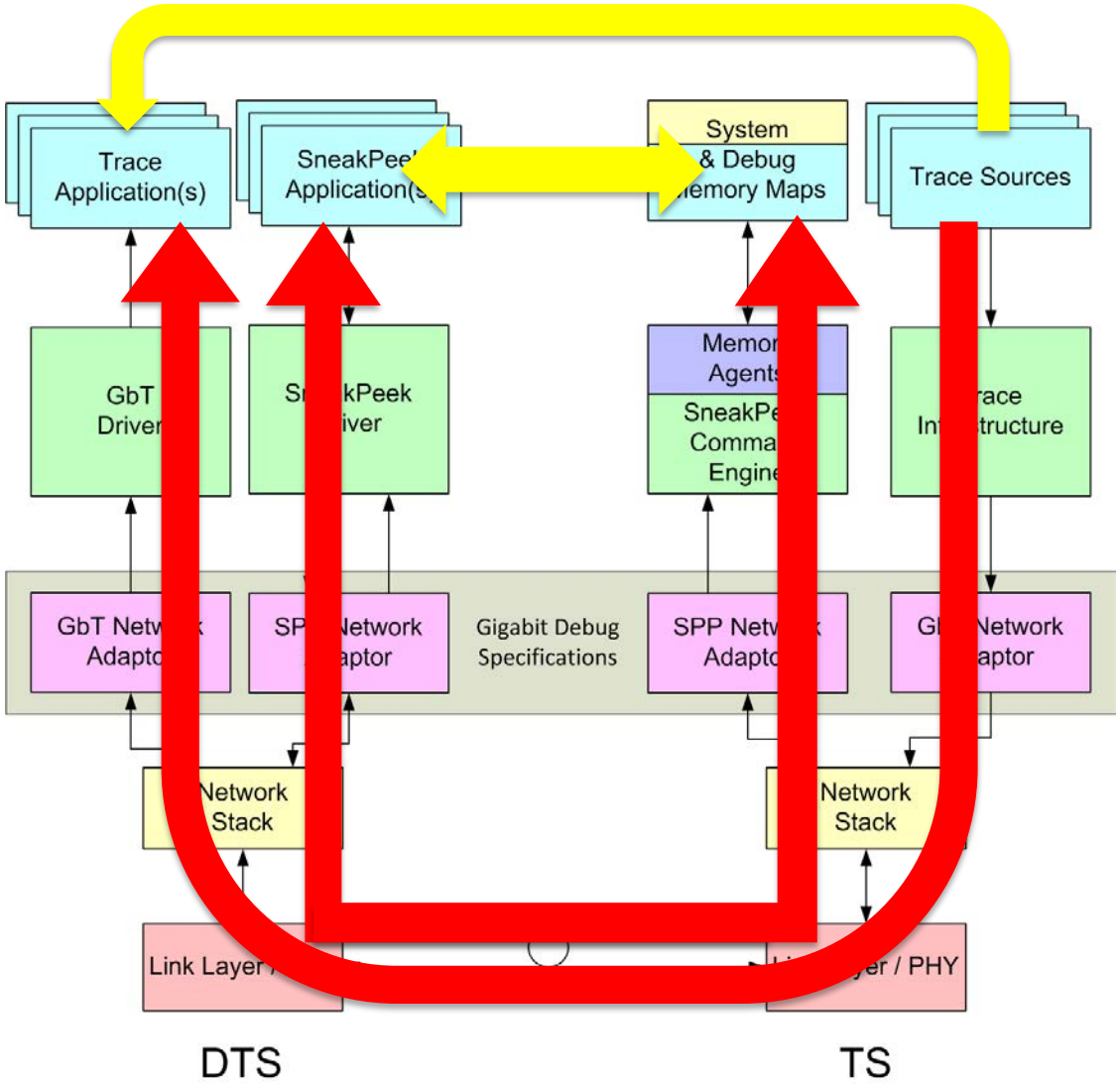
Replay: LINK
Presentation Material: LINK



Terminology

- **Debug Test System (DTS)** : The system that hosts the Debug Application, usually viewed or controlled by a human.
- **Target System (TS)** : The system being designed and debugged
- **Gigabit Trace (GbT)**: Defines a scheme to support the interleaving and transport of trace data
- **SneakPeek Protocol (SPP)**: Facilitates Debug Applications within the DTS to debug operations on the TS by providing a mechanism to perform address-mapped read and write transactions
- **Gigabit Debug (GbD)**: Transporting debug (SPP) and trace (GbT) protocols as normal network clients over a high-speed network
- **Network Adaptor**: An entity that facilitates a Debug Application communicating over a particular network, managing a link, controlling data transfers, handling network errors, etc.

Gigabit Debug Layers and Data Flow



Gigabit Debug over IP Sockets

What and Why?

- What is it?
  - A "How To" for sending SneakPeek and Gigabit Trace debug traffic over the transport protocols TCP and UDP
- Why IP Sockets?
  - Pervasiveness
    - IEEE 802.3 (Wired Ethernet) and 802.11 (Wireless LAN) are widely available across DTS and TS
  - Ease of adoption
    - TCP and UDP are well known and broadly supported transport protocols

Transport Protocols : TCP

- **TCP:**
  - The send and receive unit is called a stream (sequence of bytes).
- **Main advantage:**
  - TCP guarantees that data makes it to the destination, that it gets there in the proper order and without duplication.
- **Main disadvantages:**
  - TCP has a lot of features, some of which may not be needed—wasting bandwidth and time.
  - Generally relies on a high-level OS to perform reassembly, acking, flow control, etc.

Transport Protocols : UDP

- **UDP:**
  - The send and receive unit is called a datagram (includes data, and a header with data length, source port, destination port, and a checksum).
- **Main advantages:**
  - Very light-weight. Can be implemented in hardware or require a minimal amount of software overhead.
- **Main disadvantages:**
  - No guarantees : Datagrams may be lost, duplicated, or arrive out of order.

The Network Adaptor and UDP

- Problem :
  - *Network Adaptor Design Assumptions*
    - ✓ *Independence from Network Topology and other Network Clients*
    - X *In-order Network Communication*
      - UDP: Datagrams can come in out-of-order or arrive multiple times
    - X *Lossless and Error-Free Network*
      - UDP: Datagrams can be lost
    - ✓ *Independent Full-Duplex Communication (SPP)*
    - X *Link Flow Control*
      - UDP: No flow-control

The Network Adaptor and UDP

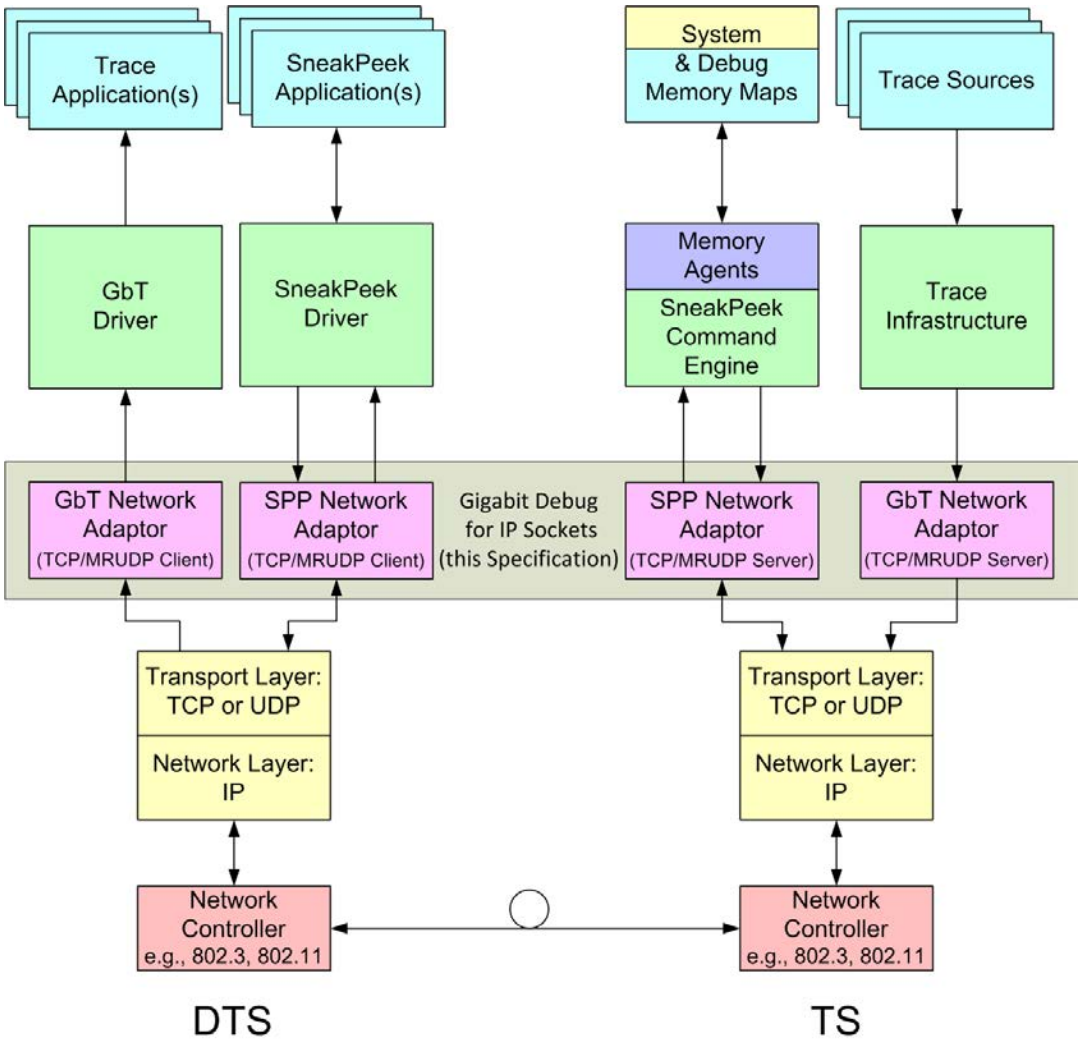
- Solution : MIPI Reliable UDP (MRUDP)
  - Define a reliability layer that resides above UDP and creates a reliable link where data is delivered to and from the debug application layer in order and without data loss or repetition.
  - MRUDP Manager : The collection of elements that implement MRUDP

Other Challenges

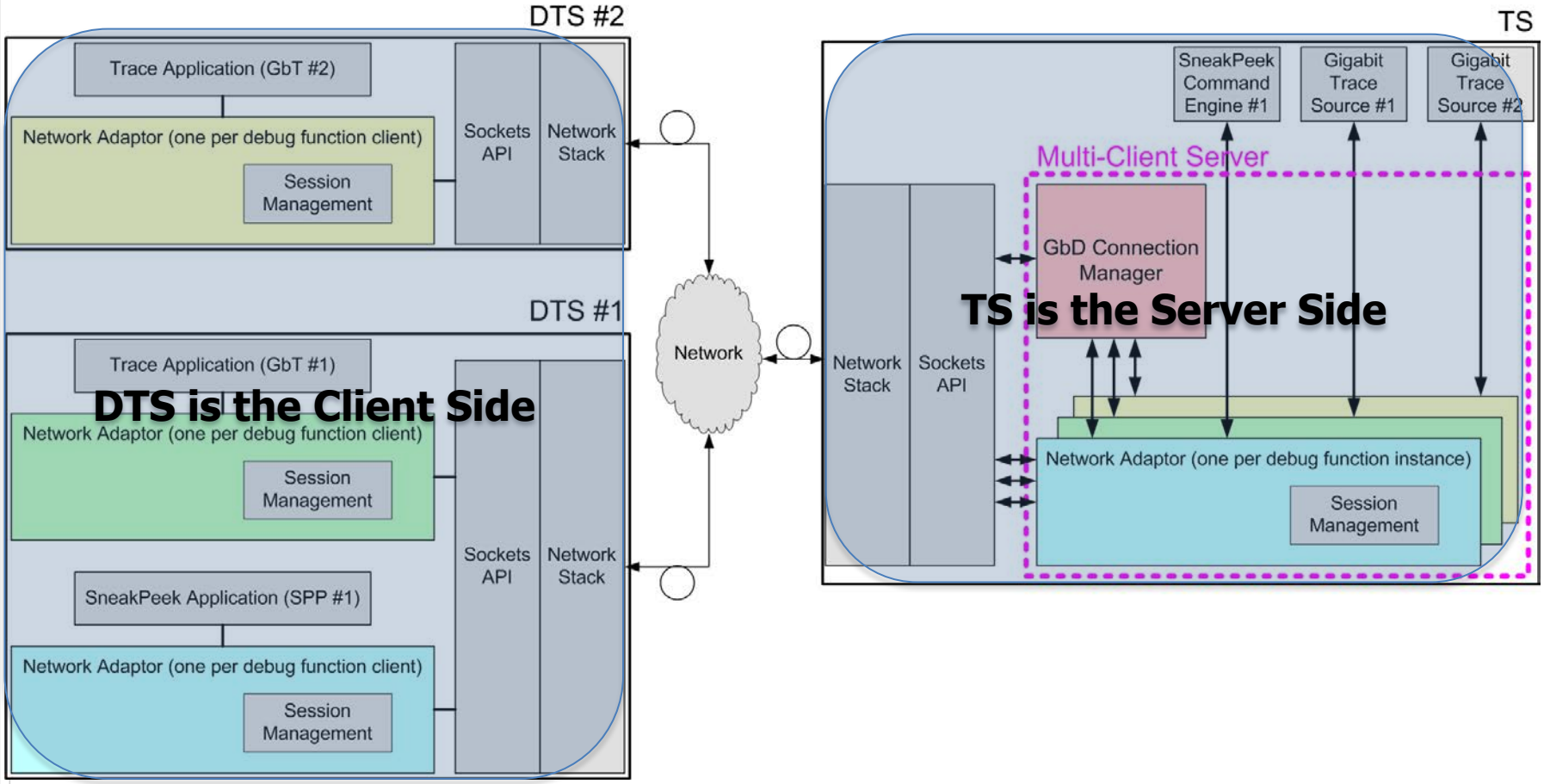
- *TS-side support for multiple and simultaneous DTS connections (e.g. GbT client and SPP client)*
  - Connection manager defined that establishes the initial connection and assigns to a specific Network Adaptor
- *TS-side support for systems with > 1 debug functions*
  - First four bytes (GbD Header) transmitted by the DTS after a connection is established is used to define the debug function and instance needs of the DTS
- *Discovery of Servers that support Gigabit Debug over IPS*
  - IANA registered User Port and Service Name

Gigabit Debug over IPS Specification

- DTS and TS side behaviors
- Debug Protocols (GbT, SPP)
- Transport Protocols (TCP, MRUDP)

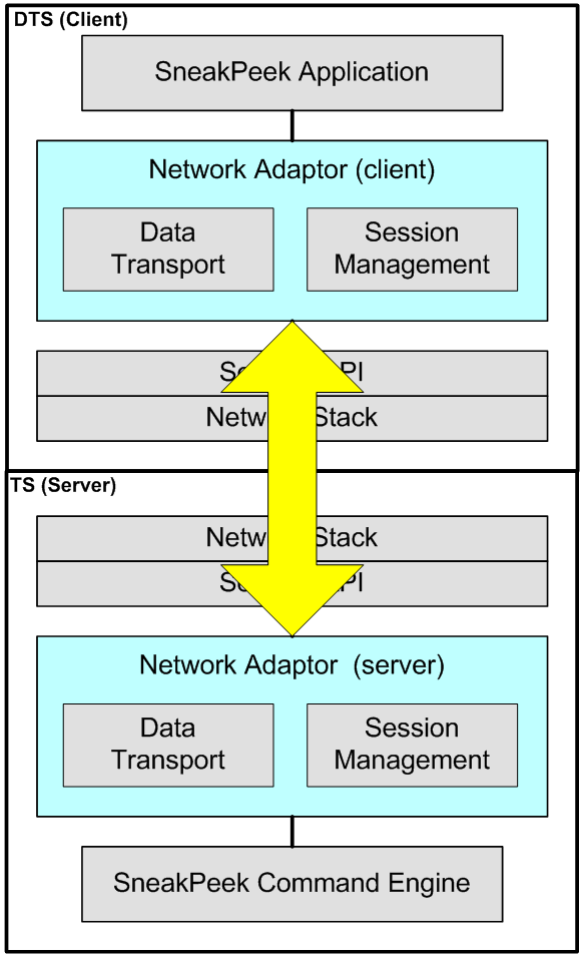


Client / Server Model



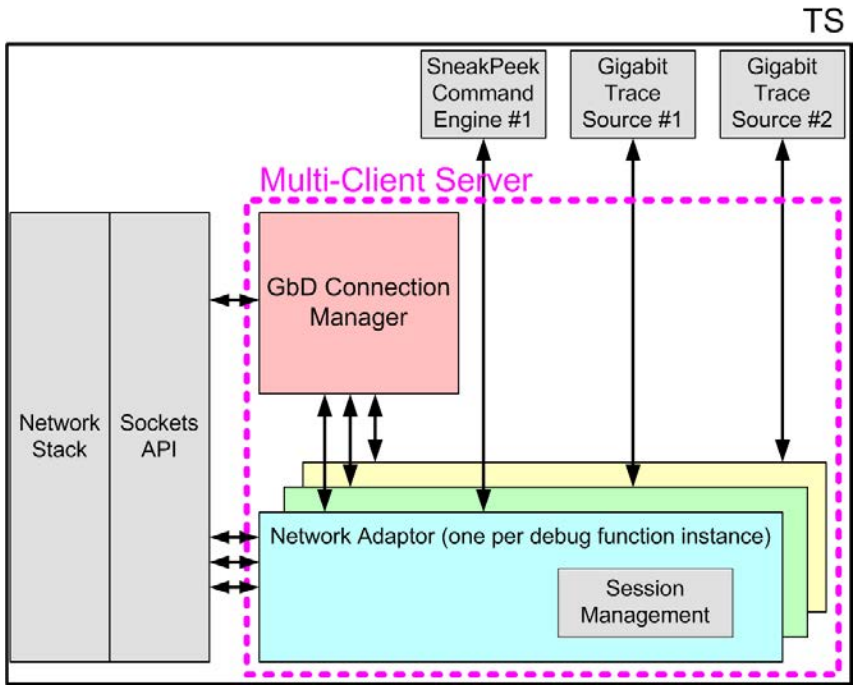
Components : Network Adaptor

- Network Adaptor
  - One instance of a Network Adaptor per debug application/function
  - **Data Transport**: Facilitates the transport of data between a debug application (DTS) and a debug function (TS) using TCP or MRUDP
  - **Session Management**: provides control functions associated with system setup, initialization, and operation



Components : GbD Connection Manager

- Accepting a Client's connection request
- Assigning a Network Adaptor to a connection
  - A TS can have as many Network Adaptors as it has debug functions
  - The maximum number of simultaneous Client Connections that a TS can have is limited by the number of Network Adaptors.



Lifecycle of a GbD Debug Session

- Discovery
- Establishing a Connection
- Transferring Data
- Terminating a Connection

Gigabit Debug Session: The period from initializing to de-initializing the network path

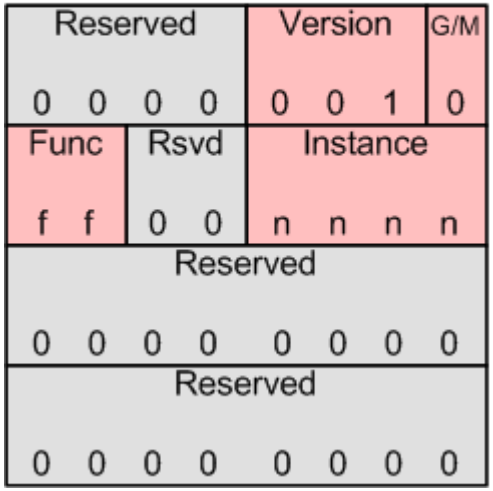
Lifecycle of a Debug Session: Discovery

- IANA Registered Resources (TCP, UDP)
  - User Port : 7606
  - Service Name : mipi-debug
- Server-side
  - Binds to User Port 7606
- Client-side
  - Determines IP address of Server (e.g. Fixed, DNS lookup, or DNS-SD)
  - Connects to Server located at IP address and port 7606

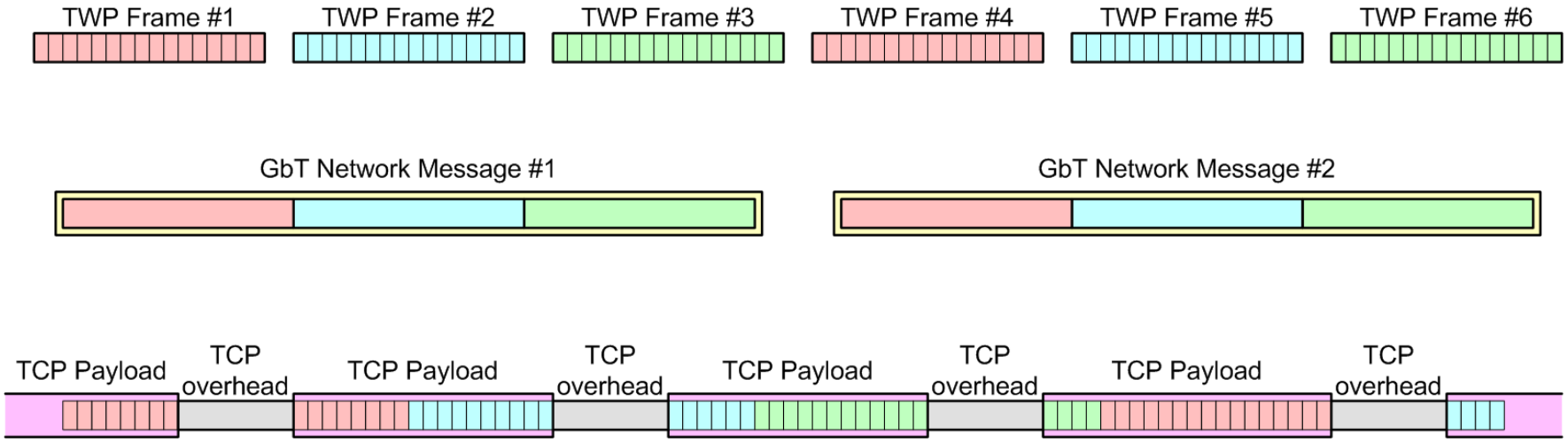
Lifecycle of a Debug Session: Establishing a Connection

- GbD Header:
  - The first four bytes transmitted by the Client to the Server during a Session conveys the desired debug function and instance
  - The first four bytes transmitted by the Server to the Client during a Session conveys the success or failure of the Client's request

GbD Header
(TCP and MRUDP Open)

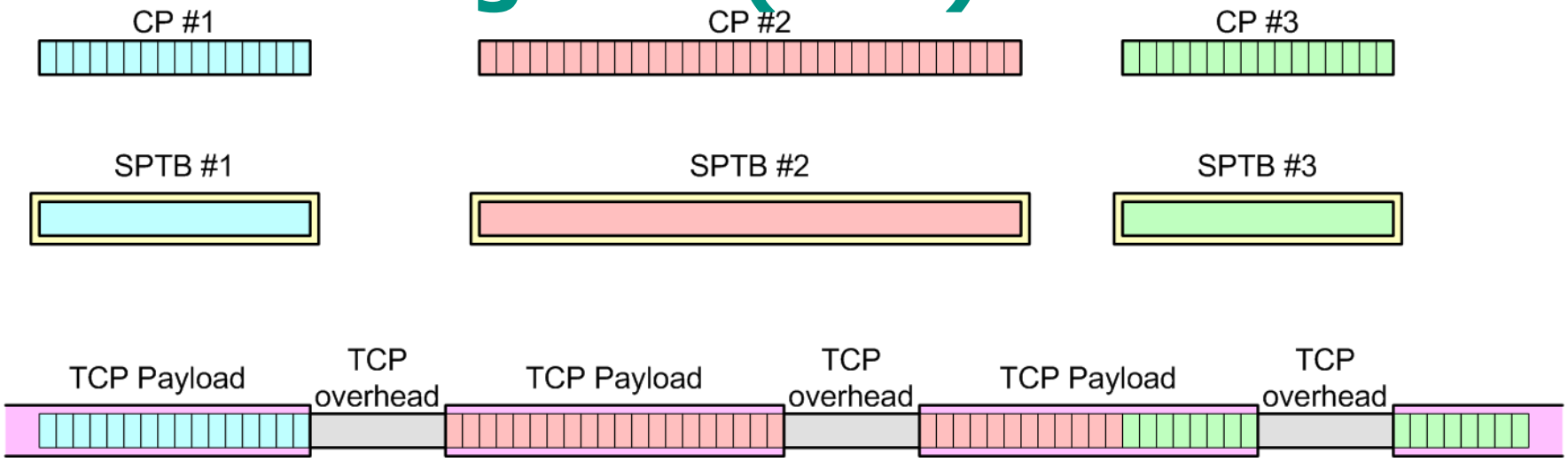


Lifecycle of a Debug Session: Transferring GbT (TCP)



- TWP Frames (16B) are packed into GbT Trace Network Messages
- GbT Network Messages have a size that is a multiple of 16B (i.e. contain an integer number of TWP Frames) and this size can not change during the lifetime of the session
- TCP transfers complete GbT Network Messages

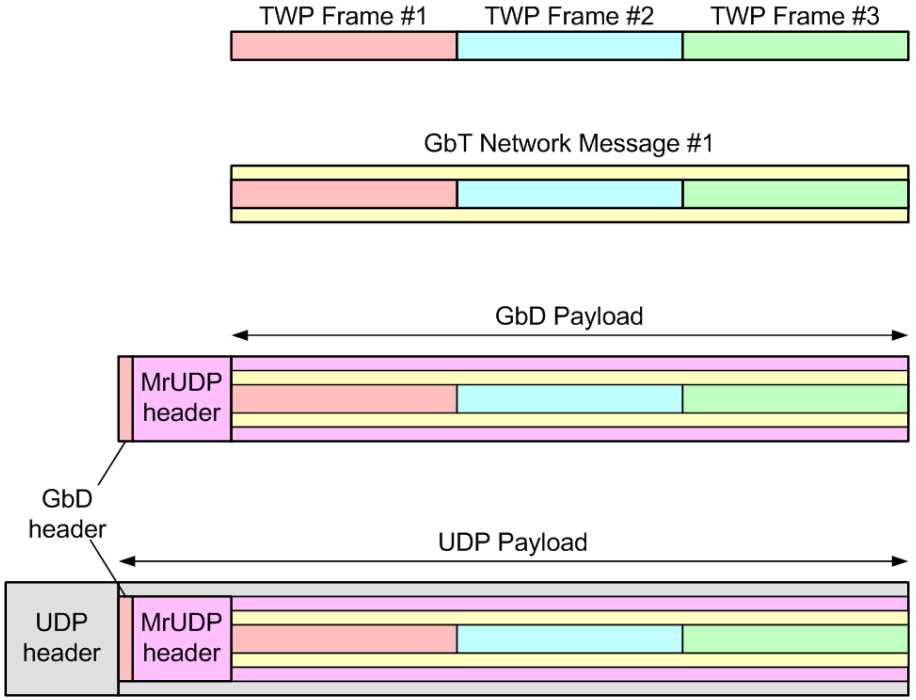
Lifecycle of a Debug Session: Transferring SPP (TCP)



- (DTS→TS) A Command SPTB contains exactly one Command Packet
- (TS → DTS) A Response SPTB contains exactly one Response Packet
- TCP transfers complete SPTBs

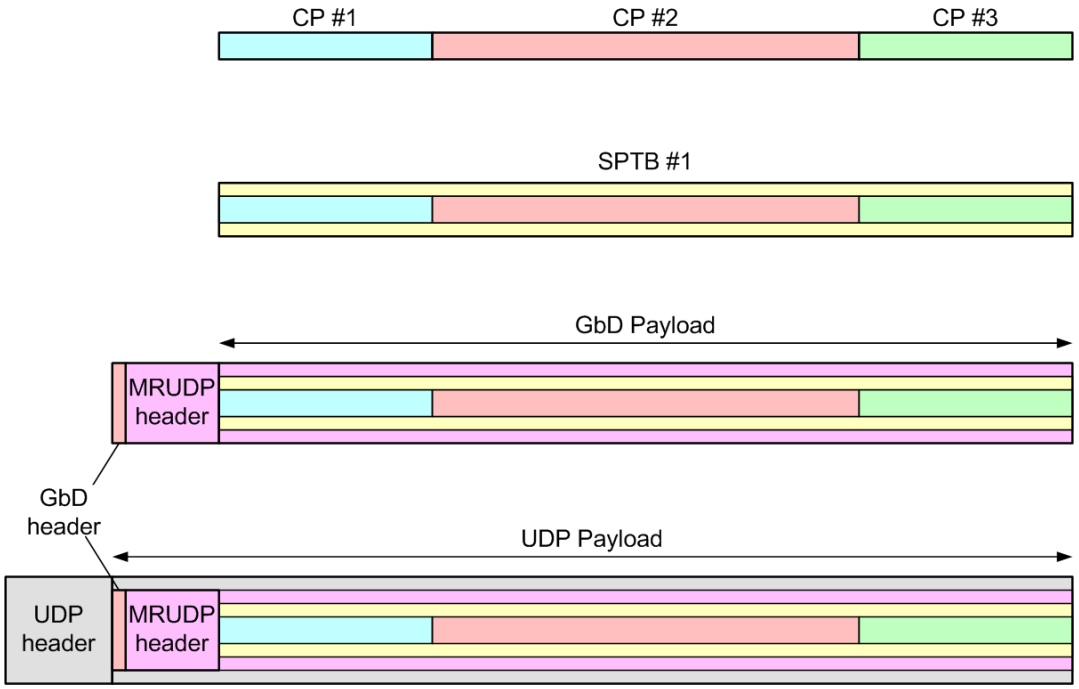
*SPTB = SneakPeek Transport Block

Lifecycle of a Debug Session: Transferring GbT (MRUDP)



- TWP Frames (16B) are packed into GbT Trace Network Messages
- GbT Network Messages contain an integer number of TWP Frames
- MRUDP transfers complete GbT Network Messages over UDP as part of its GbD Payload

Lifecycle of a Debug Session: Transferring SPP (MRUDP)

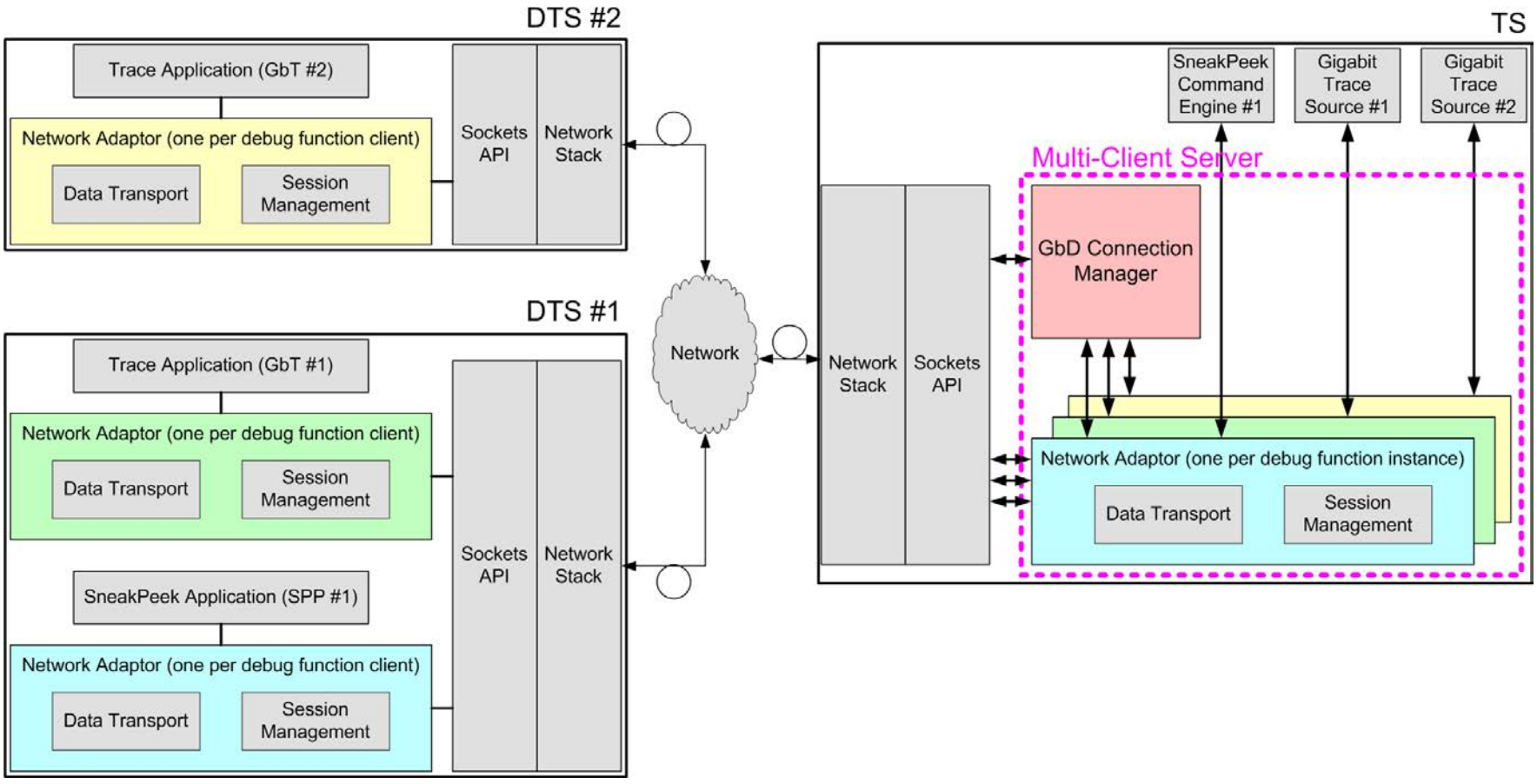


- (DTS→TS) A Command SPTB contains an integer number of Command Packets
- (TS → DTS) A Response SPTB contains an integer number of Response Packets
- MRUDP transfers complete SPTBs over UDP as part of its GbD Payload

Terminating a Connection

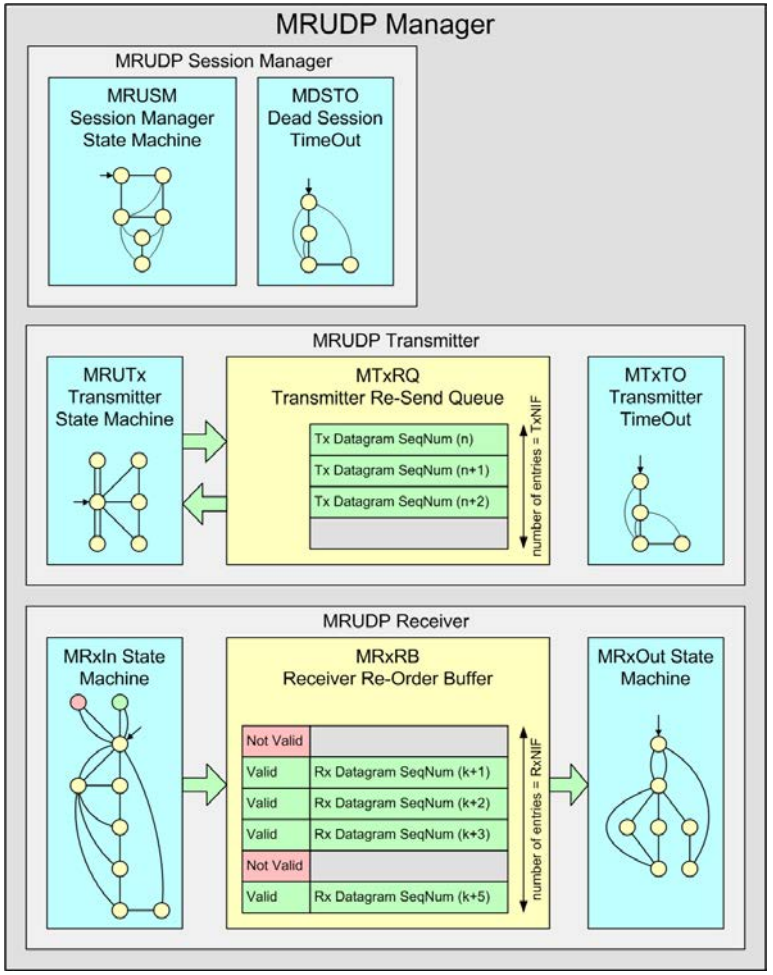
- Termination of a Connection is handled by Session Management
  - Via Debug Management Function
  - MRUDP : In-band requests and time-outs can facilitate connection termination

Client / Server Recap



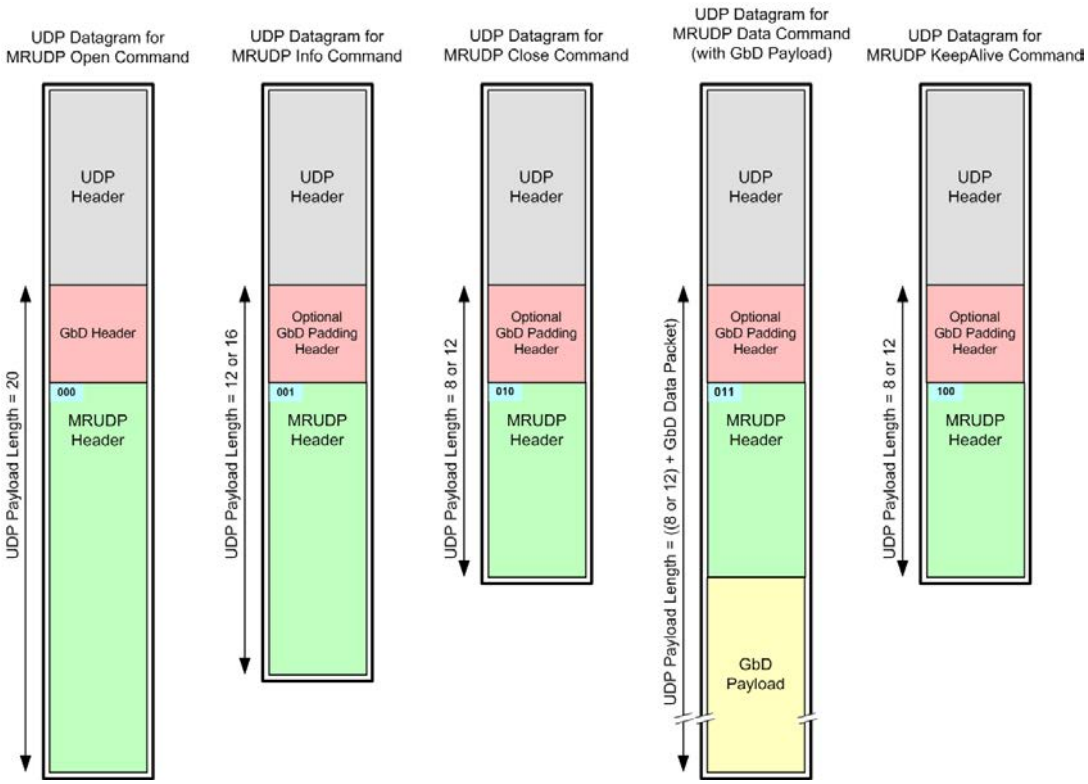
MRUDP Manager Overview

- MRUDP Managers on the TS and DTS communicate with each other over UDP
- **MRUDP Session Manager**: Supports the Network Adaptor's Session Management function
- **MRUDP Transmitter**: Implements a reliable send mechanism
- **MRUDP Receiver**: Implements a reliable receive mechanism



MRUDP Datagrams

- **Open** – Notification to the remote MRUDP Manager that the MRUDP Session is beginning. Operational parameters (size of buffers, Number of In Flight settings for Rx and TX) of the local MRUDP Manager are conveyed.
- **Info** – The local MRUDP Manager provides its operational parameters to the remote MRUDP Manager
- **Close** – Notification to the remote MRDUP Manager that the MRUDP Session is ending
- **Data** – Conveys application payload data
- **KeepAlive** – Informs the remote MRUDP Manager that the local Manager is still functioning

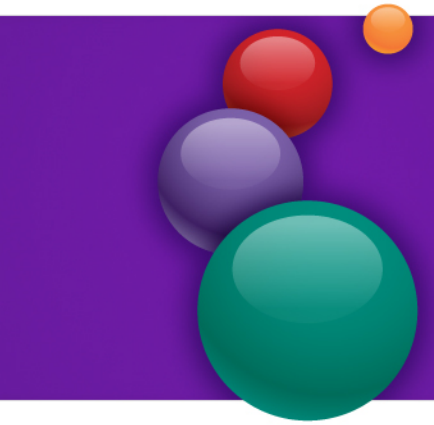


Conclusion

- More Information:
  - MIPI Debug WG Public Page: LINK
  - MIPI Architecture Overview for Debug: LINK
  - MIPI Debug on Wikipedia: LINK
  - *Gigabit Debug: Taking Debug into the IoT*: LINK
- MIPI Gigabit Debug Specifications : LINK
  - Framework
    - Trace Wrapper Protocol (TWP) v1.1
    - SneakPeek Protocol (SPP) v1.0
  - Implementations
    - Gigabit Debug for USB v1.0
    - Gigabit Debug for IPS v1.0
- Join the MIPI Debug WG!

Questions?

Thank You!



Backup Slides

Overview of Current Documents [1/3]

- MIPI® Alliance Recommendations for Debug and Trace Connectors Version 1.10.00
  - Basic Debug Connector (JTAG, cJTAG, SWD) connector
  - High-performance Trace Connector with Basic Debug
  - MIPI60 Trace Connector and MIPI Debug Connectors deployed in many EVMs and development systems. Supported by many debug tools vendors
  - Status : Stable
- MIPI® Alliance Specification for Parallel Trace Interface (PTI℠) v2.0
  - Transport trace information over single-ended parallel lines
  - Interface timing specification
  - Training patterns
  - Supports multi-drop trace
  - A large number of SoC vendors and tools vendors support MIPI compatible PTIs
  - Status : Stable
- MIPI® Alliance Specification for High-speed Trace Interface (HTI℠) v1.0
  - Transport trace information over high-speed serial lines
  - Aurora 8B/10B protocol
  - Status : Stable
- MIPI® Alliance Specification for Narrow Interface for Debug and Test (NIDnT℠) Version 1.1
  - Maps Basic Debug and Trace signals to functional interfaces
  - Interface switching protocol
  - Interfaces covered : microSD, USB 2.0, USB Type-C, HDMI, and Display Port
  - Adopted by several SoC vendors and is supported by multiple tools vendors
  - Status : Updates Under Development

Overview of Current Documents [2/3]

- MIPI® Alliance Specification for System Trace Protocol (STP) v2.2
  - Supports HW and SW messaging in a simple nibble stream
  - Primitives for time-stamping, message framing, data integrity, source identification, stream synchronization
  - Over 60 companies from across the industry have licensed MIPI-compatible IP from ARM. Multiple debug tools vendors support STP decode. Linaro effort underway to support STP compatible IP in Linux.
  - Status : Stable

- MIPI® Alliance Specification for Trace Wrapper Protocol (TWP℠) v1.1
  - Part of the MIPI Gigabit Debug Framework
  - Merging multiple byte-streams into a single trace stream
  - Padding and stream synchronization
  - Over 60 companies from across the industry have licensed MIPI-compatible IP from ARM. Multiple debug tools vendors support TWP decode.
  - Status : Stable

- MIPI® Alliance Specification for SneakPeek Protocol (SPP℠) v1.0
  - Part of the MIPI Gigabit Debug Framework
  - Communication protocol to be used between a device under debug and a system containing debug tooling
  - Status : Stable

Overview of Current Documents [3/3]

- MIPI® Alliance Specification for Gigabit Debug for USB (GbD for USB) v1.0
  - An implementation of the MIPI Gigabit Debug Framework
  - Defines the data transport models for each of SPP and TWP on a USB network
  - Status : v1.1 Update Under Development
- MIPI® Alliance Specification for Gigabit Debug for Internet Protocol Sockets (GbD for IPS℠) v1.0
  - An implementation of the MIPI Gigabit Debug Framework
  - Defines the data transport models for each of SPP and TWP on a network using TCP or UDP via IP Sockets
  - Status : Stable