Designing Multi-Protocol ECU Network

The model demonstrates how a Auto OEM can assemble the car platform network with 100's of ECU. This model has 16 CAN Nodes, CAN-to-CAN Bridge, Gateway, 6 Ethernet switches, multiple local networks of CAN and FlexRay (Abstracted as traffic models) connected to each Switch. The models represents over 50 Nodes. The model demonstrates the ease of construction using pre-built library blocks. When you run the simulation, you can see that the simulation is also extremely fast. Also, the model contains less than 100 blocks for such a large system. It extremely easy to assemble the entire network.

The purpose of the model is to:

1. Demonstrate the construction of large, multi-protocol networks
2. Demonstrate the types of analysis that can be performed
3. Show the pre-built libraries in VisualSim and their usage
4. Other functionality such as traffic generation, pre-built reports, ease of model construction, and visualize the system

The types of analysis are:

1. Determine the latency for each message
2. Capacity of a network to handle additional new messages
3. Create a network with multiple protocols to determine the traffic utilization across the networks
4. Design Gateways and Bridges based on traffic and types of traffic
5. Effective throughput and utilization of each network

Model Overview:

The multi-protocol model contains the following:

1. Ethernet switches connected in a star formation. There are 7 switches in the Hierarchical block connected to the top-level switch.
2. The two CAN networks are connected via a CAN-to-CAN bridge.
3. The Ethernet backbone switch and the Bridge are connected to the Gateway.
4. The Gateway block is a generic solution. Details or proprietary implementation can be connected to the top two ports. This bypasses the standard implementation. The Gateway can be designed using standard library blocks.
5. The protocols covered in this model are CAN, Bridge, Gateway, Ethernet, Ether Switch, TCP and UDP
6. Traffic for the CAN bus uses the standard CANdb tables. Now this can be easily customized for any proprietary implementation required. Just need to modify the columns and rows of the table.
7. CAN Bus can be modified easily to create CAN FD.
8. CAN Bus has both the elements of hardware timing and the protocol arbitration behavior accurately modeled.
9. The Bridge handles all the messages to be transferred to the networks connected to the Bridge.
10. The Gateway routes the messages between the CAN segments and the Ether Switch.
11. The Ethernet Segments uses UDP to handle both broadcast and multicast using the standard Multicast block.
12. The Routing Table handles the IP routing for the entire network.
13. The Ethernet Segments have traffic from other networks in some cases (CAN and FlexRay) and direct sensor connection for other sub-systems. The CAN and FlexRay networks connected to

the Ethernet are modeled as complex statistical traffic models. This is done to show how to abstract parts of the network that impact the Gateway but the design of the network is fixed.

14. The UDP traffic is handled using the Multicast block.
15. The TCP traffic has handshaking and is defined in the Msg_Proc block
16. The connection between the Ethernet switches and the Star_Node (top-level Ethernet Switch block) is handled by the Routing Table database block.

Traffic Types:

1. There is a unique traffic profile for each segment.
2. Each CAN bus has sensor and message rates and their relationship defined from the CANdb database. The Can Bus block has the CANdb details.
3. Each Can segment has a unique table.
4. All messages from all nodes on both CAN segments are sent to all the Nodes.
5. One message from the CAN Bus 1 is sent to the Ethernet network
6. Similarly, one Ethernet segment message is sent to CAN Bus 2.
7. All Ethernet segments send a TCP diagnostic message to the Switch 7.
8. All regular messages on the Ethernet are UDP and sent as multicast.
9. The multicast details are in the Init_memory block within the Ethernet Segment.
10. Each Switch uses one list from the Multicast table. The number matches the switch. So Switch1 uses Multicast1 and so on.

Plots and Reports:

1. CAN to Ethernet Latency Plot
2. Ethernet to CAN Latency Plot
3. TCP/IP Latency at Switch7
4. UDP/IP Latency at Switch1
5. The three Gannt charts are for the message periodic rate, the arrival at each CAN Node and the selected Message Filering at each node
6. Statistics shows the Routing Table, the statostics for each node in the Ethernet segment, the links to the Gateway and the CAN segments

Analysis:

1. At 0.4, notice that there is a spike in latency. This is because there is a sudden burst of diagnostic traffic with high priority and a large data size. This is causing all the other traffic to start queuing. It might be better to schedule the diagnostic messages at different times.
2. The link from the Star Node to the Switch are heaving utilized at 20% while the Switches to the Star Node are heavily under-utilized. This provides for greater expansion opportunity at the Core switch.
3. Ethernet to CAN is within a narrow range because it goes from the Switch to the Star_Node, Gateway to the CAN segment. The CAN segments are only 18% utilized with a lot of capacity.
4. Notice that the UDP messages are actually faster than the TCP messages. This is because of the handshaking request.