Adapting the Java project of Monitoring in Omnet++

Major Components of Monitoring :

CentralizedMonitoringClient :- This component collects the local monitoring values and send them to the Server . The procedure of collection and sending happens periodically.

CentralizedMonitoringServer :- This component comprises of the server . The server handles the receipt of the values sent by the clients. The server processes these data.

Major Implementations :

RealTransportUDP : This component actually deals with the implementation of UDP for the simulator. This class extends JSimpleModule, which results in it being a component in the omnet++. It provides interface like bind, connect, send, close to the java based components. Upon receiving the calls from java components, it creates a packet with arguments and the function identifiers and send to the C++ based components (UDPAppC).

UDPAppC: This is a C++ based component. This component is a wrapper that uses the UDPSocket class to create sockets and perform basic operations on them according to the function identifiers received in the messages from RealTransportUDP.

RealTimeScheduler: This is a Java based component. It provides interface(scheduleIn) to set the timers. The timers are implemented by using the scheduleAt function used in the omnet framework. The interface also stores the corresponding EventHandler which acts as a callback whenever the timeout is fired.

UDPJavaMessage : This is the class which contains the message structure that packet follows which is passed from the Java based component and the C++ based component. Similarly udppacket.msg is used by the C++ component to decipher the messages received.

Simonstratorhost.ned : This ned file basically keeps the information of all connections for the application. It shows the connections in the application,

transport and ip layer. Here we are currently focused on the udp transport layer functionalities.

Interaction between components: The components mostly interacting by fetching run time reference to the component and calling an interface exposed by it. But it was not possible to fetch a java based component in the run time from the c++ based component and vice versa. Therefore the interaction between these two components was done on the basis of exchange of messages.

CURRENT STATUS : The real time component of CentralizedMonitoringClient or Server was not possible to be fetched due to the declaration of one generic name in the ned file for both the application. There is a need of exchange of messages between these components. The overall implementation could not be made more generic so that other java based projects could be ported on to the omnet.