

Demonstration of Wired Communication using TCP protocol in Omnet++

Title: To study wired communication and effect of various parameters on the communication.

Abstract:

Wired communication includes wired media for transfer of information. It uses mediums such as copper cables, fibre optic cables, etc which can be further distributed in various types.

There are several parameters that affect the transmission like bandwidth, delay, error which are studied in this project.

I. Requirements

Software:

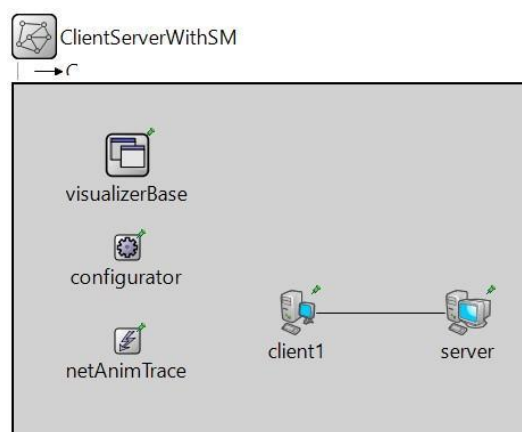
1. OMNET++

Pre-requisite knowledge:

1. C++
2. Wired communication mediums and transmitter, receivers.
3. Protocols basics

II. Network Description(.ned) file

1. Design



2. Source

```
import inet.common.misc.NetAnimTrace; import
inet.common.scenario.ScenarioManager;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator; import
inet.node.inet.StandardHost;
```

```

import inet.visualizer.integrated.IntegratedVisualizer; import
ned.DatarateChannel; network ClientServerWithSM
{
    @display("bgb=456,311"); types: channel
C extends DatarateChannel
{
    datarate = 10Mbps; delay =
0.1us;
    submodules:
    client1: StandardHost { parameters:
        @display("p=250,200;i=device/pc3");
    }
    server: StandardHost { parameters:
        @display("p=400,200;i=device/pc2");
    }
    configurator: Ipv4NetworkConfigurator { parameters:
        @display("p=100,143;is=s");
    }
    netAnimTrace: NetAnimTrace { @display("p=100,225;is=s");
    }
    visualizerBase: inet.visualizer.integrated.IntegratedVisualizer {
        @display("p=100,62");
    } connections:
    client1.ethg++ <--> C { disabled = true; } <--> server.ethg++; }

```

III. Terminologies in NED file

1. Channel: Channels encapsulate parameters and behaviour associated with connections.

Datarate Channel has following parameters:

- a. 'delay' is a double parameter which represents the propagation delay of the message
- b. 'disabled' is a boolean parameter that defaults to false; when set to true, the channel object will drop all messages
- c. 'datarate' is a double parameter that represents the data rate of the channel
- d. 'BER' and 'PER' stand for Bit Error Rate and Packet Error Rate, and allow basic error modelling

2. NetAnimTrace: Trace events denote what happens to a packet in the transmit queue of a node. NetAnimeTrace is used to show packet handling.

3. Configurator: This module assigns IPv4 addresses and sets up static routing for an IPv4 network. It assigns per-interface IP addresses, strives to take subnets into account, and can also optimize the generated routing tables by merging routing entries.

4. Client, Server: IPv4 host with SCTP, TCP, UDP layers and applications.
5. Visualizer: An integrated visualizer for visualizing packet drop when there is no connection from server to client.

IV. Configuration(.ini) file

[General]

network = ClientServerWithSM total-stack = 7MiB

**server.numPcapRecorders = 0

**server.pcapRecorder[0].pcapFile = "results/server.pcap"

**client1.numPcapRecorders = 0

**client1.pcapRecorder[0].pcapFile = "results/client1.pcap"

**scenarioManager.script = xmldoc("scenario.xml")

tcp apps

**numApps = 1

**client*.app[*].typename = "TcpSessionApp"

**client*.app[0].active = true

**client*.app[0].localPort = -1

**client*.app[0].connectAddress = "server"

**client*.app[0].connectPort = 1000

**client*.app[0].tOpen = 0.2s

**client*.app[0].tSend = 0.4s

**client*.app[0].sendBytes = 2000000B

**client*.app[0].sendScript = ""

**client*.app[0].tClose = 25s

**server*.app[*].typename = "TcpEchoApp"

**server*.app[0].localPort = 1000

**server*.app[0].echoFactor = 2.0

```

**.server*.app[0].echoDelay = 0s

# NIC configuration

**.ppp[*].ppp.queue.typename = "DropTailQueue" # in routers

**.ppp[*].ppp.queue.packetCapacity = 10 # in routers

*.configurator.config=xml("<config><interface hosts='*' address='192.168.1.x'
netmask='255.255.255.0'/></config>")

**.packetDropVisualizer.displayPacketDrops = true

```

V. Information about INI file:

Configuration and input data for the simulation are in a configuration file usually called omnetpp.ini

The configuration file contains settings that control how the simulation is executed, values for model parameters, etc. The configuration file can also prescribe several simulations runs; in the simplest case, they will be executed by the simulation program one after another.

Submodule type name is expected to be defined via typename pattern assignments. Typename pattern assignments look like pattern assignments for the submodule's parameters, only the parameter name is replaced by the typename keyword. Typename pattern assignments may also be written in the configuration file.

VI. Observations

1. Wired communication uses medium to send information.
2. The medium has several parameters like bandwidth, transmission delay, noise interference.
3. After packet is transferred there is ACK from the receiver.
4. Packet drop is visualized when there is no connectivity between client and server.

VII. Conclusion

1. Omnet++ can be used to visualize behaviour of wired network.
2. The parameters can be controlled to match real values of medium such as bandwidth in .ned file.
3. Error can be seen through the BER and per flags during simulation.

4. Packet drops can be visualized using visualizer.

References

1. <https://doc.omnetpp.org/omnetpp/manual/>
2. <https://inet.omnetpp.org/docs/users-guide/ch-network-nodes.html>
3. <https://inet.omnetpp.org/docs/developers-guide/ch-packets.html>