Laboratory

5

Link layer protocols

Objetive

In this lab session we are going to study and analyze the performance of link layer protocols over a point-to-point channel using the OPNET IT Guru tool. In particular, we will compare performance of stop&wait and pipeline protocols, analyzing the impact on performance of channel parameters like link delay, link data rate and link error rate (error-free and real channels)

Overview

OPNET IT Guru offers a virtual network framework that allows the evaluation of whatever kind of networks, by showing the behaviour of all the common devices you can find in one network (like hosts, hubs, switches, routers, firewalls, etc.) and also the performance of protocol stack and applications running in those devices.

In this session we need a model of a point-to-point link between two hosts in order to evaluate link layer protocol performance. For that purpose, we will use as the reference link layer protocol the one included in TCP protocol. We will change the TCP parameters to determine the window size, packet size, timers, etc. so we can tune the link layer protocol in each scenario with different point-to-point channel models.

Stop-and-Wait protocols have some desirable properties:

- Since the sender cannot transmit a new packet until the acknowledgement for the last one sent is received, the protocol is ACK-clocked. These protocols automatically adjust the transmission speed to both the speed of the network and the rate the receiver sends new acknowledgements.
- They respect an underlying conservation principle of computer networks: "Given a producer sending packets to a consumer, if the system is in equilibrium, that equilibrium will be preserved if a new packet is sent only when a previously sent packet is consumed."

However, these protocols become rather inefficient if the path connecting the sender and the receiver has a large bandwidth-delay product (that is, the maximum number of bits that can be seen in transit on the link). A possible solution to this problem would be to give the source a credit of packets that can be sent without having to wait for acknowledgements. This would allow the source to fill the pipe, then, the arrival of acknowledgements would maintain and adjust a continuous flow of data. Therefore, a Stop-and-Wait protocol would be a particular case of such protocols, where a credit of one packet is given to the source.

Procedure

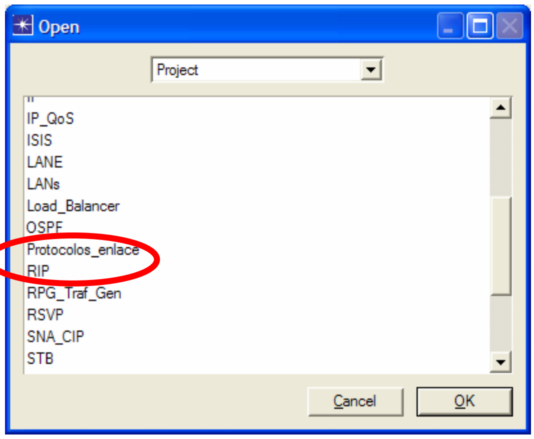
(1) Open the Project

1. Start OPNET IT Guru.

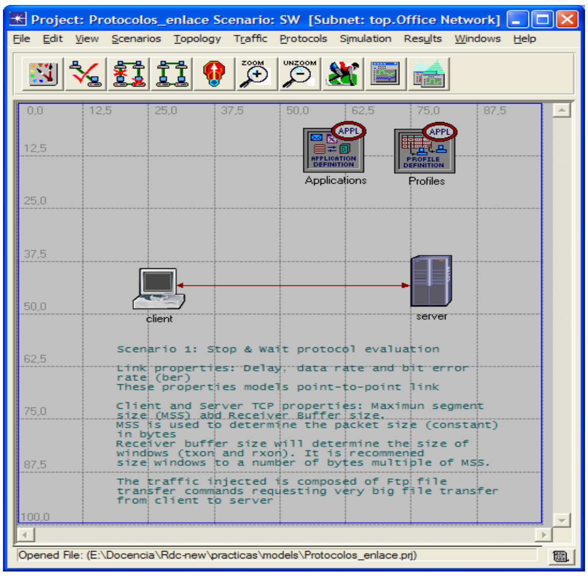
2. Copy the supplied Project files (these files are zipped in a file available in the course web) to your local model directory

3. Select **File→Open…** and be sure that "**Project**" is selected in drop-down menu.

4. Select "**Protocolos_enlace**" Project and click **OK**.



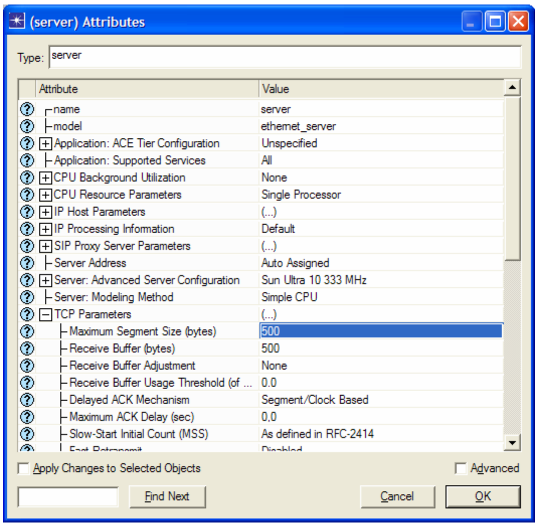
Then, you will have available the working scenario with two workstations (client and server) connected with a point-to-point link (modelled with **point_to_point_adv** model**)** working at 10 Mbps and with a 4 ms. link delay. Also, you can find two extra objects: **Applications** and **Profile.** With them we will define and FTP application as traffic pattern between client an server hosts. In simulation time, the client host will starts an FTP session with server host and after that the client request a file transfer (put command). The file size is big enough to keep the transfer until the simulation end.



(2) Performance impact of transmission window size.

With this scenario we are going to evaluate the performance of sliding-window protocols as a function of transmission window size. So, we want to know the link utilization when the link layer protocol uses a transmission window sizes of 1 (Stop&wait behaviour), 10 and 20 packets.

1. Right-Click on server workstation object and select "**Edit Attributes**" from the pop-up menu.

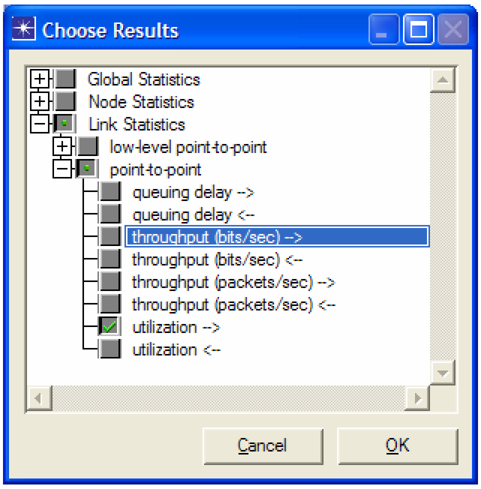


2. Among station properties we will work with TCP parameters, that determine the behaviour of the TCP running in server Workstation[1]. TCP property "**Receive Buffer (bytes)**" can be used to change the receiving window size (an as a consequence the transmission window size at the other end). The default value of this property is 500 bytes. As the TCP's Maximum Segment Size (MSS), in other words the maximum size of a packet) is also 500 bytes, we can receive only one segment after the previous one is acknowledged. So, we are talking about a Stop&Wait protocol. To increase the size of both windows we have to change the value of this property in multiple values of MSS (i.e.: 1000 bytes will correspond to a window size of 2 packets).

3. Right-Click on "**Receive Buffer (bytes)**" property and select "**Promote Attribute to Higher level**" in order to assign different values to this property for a simulation series. After that, the value of "**Receive Buffer (bytes)**" property will be "**promoted**". Click **OK**.

4. Right-Click on the scenario background and select "**Choose individual statistics**" to instruct simulator to gather the desired performance metrics. Check the "**Utilization →**" box that refers to the link utilization in client→server direction. Click **OK.**

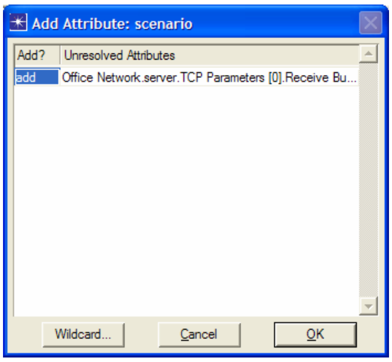
[1] Although we will study TCP protocol later in the course, now we are going to use only the parameters related to flow and error control..



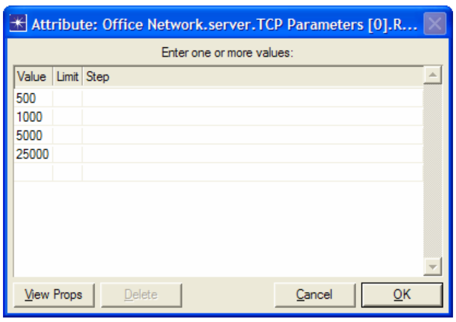
5. Click on **Configure/run simulation** button (toolbox menu) 

6. Be sure that simulation time is 50 seconds.

7. Select “**Object Attributes**” tab to instruct simulator the generation of several simulations based on “**Receive Buffer (bytes)**” property. Click on “**Add…**” button and select this property in the correspondent cell of “**Add?**” column. Click **OK.**



8. Next, we will specify the values this property is going to take in each simulation. Select “**Receive Buffer (bytes)**” property and click on “**Values…**” button. Introduce the following values: 500, 1000, 5000 and 25000. Click **OK.**



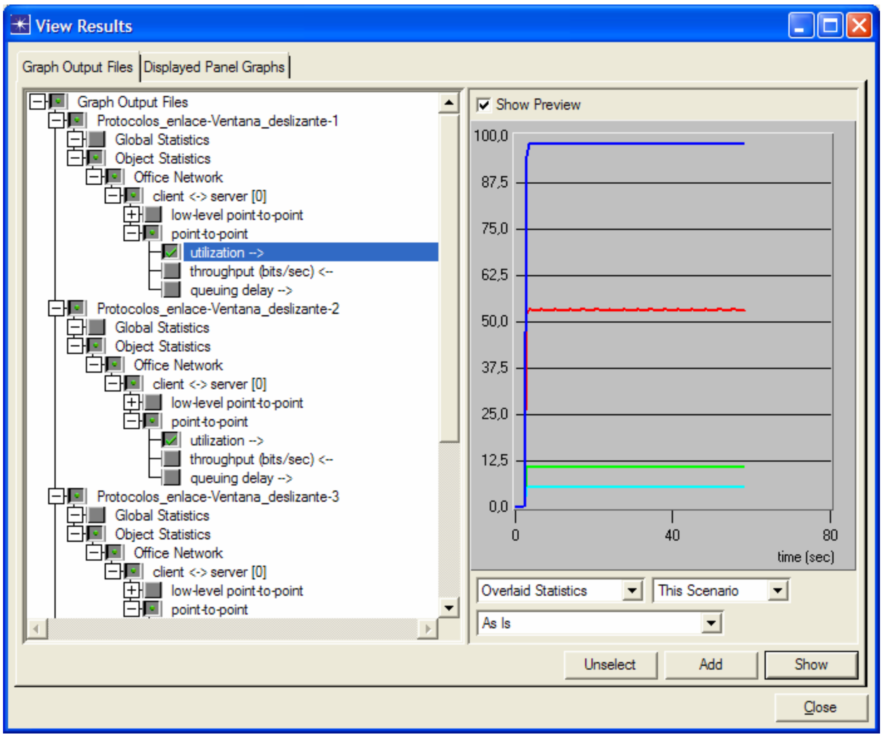
9. Start the simulation run by clicking on “**Run**” button.

Get simulation results.

Once simulation ends, select "**Results→View Results (Advanced)**" to launch a new stats window. Within the new window select "**Panels→Create vector Panel…**" menu entry to prepare the desired performance plots. In particular, we are interested in the impact of window size in the link utilization.

Mark "**Utilization →**" boxes in all simulations of the simulation series as we show in the figure below. In order to overlay the results of all simulations in one graphic window we should select "**Overlaid statistics**" in the drop-down menu placed bottom-right in the window. With respect to the way data is represented, we can employ different filters also available in another drop-down menu. We keep the default filter "**As Is**" (data no filtered).

The obtained result should be similar to the one shown in the figure below.



Click "**Show**" button to create the correspondent graphic window that will be associated to the current project. Right-Clicking on different areas of this window you will find popup menus that allow change text in caption labels, window title, export values to spreadsheet, curve style, colours, etc.

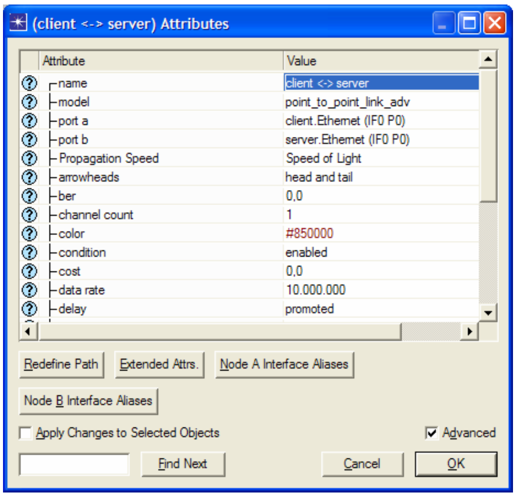
Explain why the link utilization increases when window size also increases. Calculate the smallest window size that will result in a maximum utilization (≈100%).

(3) Performance impact of link delay in an Stop&Wait protocol.

1. Duplicate current scenario (“**Scenarios→Duplicate scenario…**”). The new scenario’s name will be “**Ventana_Deslizante_retardo**”.

2. Suppose we are working with an Stop&Wait protocol (sliding-window with a window size of 1). To do that, we fix “**Receive Buffer (bytes)**” and “**Maximum Segment Size**” TCP properties in server and client hosts at the same value: 1000 bytes. Both windows will be space to keep one 1000 bytes segment.

3. Right-Click on the point-to-point channel object and select “**Edit attributes**” to change the values of “**data rate**” and “**delay**”[2] link properties. We will assign the value of 10 Mbps (10.000.000) to the former property. Then, we will right-click “**delay**” property and select “**Promote Attribute to Higher level**” so we could assign different values to this property in a simulation run series. Click **OK.**



4. Click on **Configure/run simulation** button (toolbox menu) 

[2] This property is and advanced property, so we have to check the “**Advanced**” box to access to it.

5. In “**Object Attributes**” tab add “**delay**” and assign it the following values (in seconds): 0,0003 (Typical LAN value), 0,001, 0,005 (typical WAN links), 0,03 y 0,1 (a similar value to satellite link delay).

6. Run simulation. Once simulation ends, build a graphic with the utilization curves for every delay values (in a similar way that in previous experiment). Explain the obtained results. Is there a lineal relationship between link utilization and link delay?.

(4) Stop&Wait performance in presence of transmission errors.

Until now, we have supposed that the link was error-free. But in real world this assumption is not realistic. So, we will study the performance of Stop&Wait protocols in links with an specific error rate. Before starting this study, we will configure the point-to-point link with an 10 Mbps bandwidth and an 0,001 seconds delay. The exchanged packets will be 1000 bytes in size.

1. Duplicate current scenario (“**Scenarios→Duplicate scenario…**”). The new scenario’s name will be “**Parada_y_Espera_error**”.

2. Adjust the timer (timeout value) to 0,1 seconds at TCP client host. To do that, you should assign this value to the three following TCP properties in TCP client host: “**Inital RTO (sec)**”, “**Minimum RTO (Sec)**” and “**Maximum RTO (Sec)**”. Associated with timers, also is required you set “**Timer granularity (sec)**” property to 0,001 seconds.

3. Right-Click on “**Ber**” (Bit error rate) property of the point-to-point link and select “**Promote Attribute to Higher level**” so we can assign different values to this property in simulation time. Click **OK.**

4. Be sure to select “**throughput (bits/sec)**” statistics in the point-to-point link (Link statistics) and TCP’s “**Retransmission count**” property (Node statistics).

5. Click on **Configure/run simulation** button (toolbox menu) 

6. In “**Object Attributes**” tab, configure the attribute you have promoted in 2 and assign it the following values: **5E-06**, **1E-06**, **5E-07** and **0,0**. Our purpose is to introduce errors in the link to study the behaviour of *Stop&Wait* protocol. As a reference we also simulate the error-free link.

7. Run simulation. Be sure that simulation time is 100 seconds.

8. Build one graphic with the resulting throughput of all performed simulations[3]. Apply "**time_average**" filter to all the curves. Explain obtained results. Build a second graphic where the number of retransmission can be shown (use here the "**sample_sum**" filter). From both graphics, explain the behaviour of Stop&Wait protocol under one link with errors.

(5) Sliding-window protocol performance in presence of errors.

We are going to do the same as in previous experiment (4) but now with an sliding-window protocol. So, we start with the same channel model than before (10 Mbps data rate and 1 ms delay), packet size of 1000 bytes and a window size that maximizes link utilization[4] (as it was done at the beginning). We will repeat the same steps than in previous experiment in order to measure the impact on performance (channel utilization) of channel errors with a sliding-window protocol.

1. Duplicate current scenario ("**Scenarios→Duplicate scenario…**"). The new scenario's name will be "**Ventana_deslizante_error**".

2. Follow the same steps that in previous experiment, repeating the same simulations and obtaining the same graphics. Analyze the results and compare them with the ones found with the Stop&Wait protocol.

[3] We are interested in the resulting throughput in client→server direction: "throughput (bits/sec) →"
[4] We have to calculate the smallest window size the maximizes the utilization in an error-free channel.

Questions

(1) When point-to-point channel introduces transmission errors in communications, the link layer protocols (Stop&Wait and sliding-window protocols) loose performance as a function of link error rate. Due to the possibility of packet loose in both data and acknowledge packets, both protocols employ timers for error control with a predefined timeout. Discover the optimum timeout value[5] for both link layer protocols taking into account the same link model and same packet and window sizes of previous experiments (5) and (6)[6]. Explain the obtained results.

(2) If we increase the link data rate up to 100 Mbps keeping the rest of parameters unchanged, which it will be the performance impact on both protocols if we are using an error-free link?. Ho can we improve the performance of the sliding-window protocol?

[5] Notice that you have to change three TCP properties related with timers (see experiment (1) step 5)
[6] Use only one link error rate. We suggest BER = 1E-06.

(3) Analyze and evaluate two variants of the sliding-window protocol: Goback-N and selective repeat. For that purpose, we will define the same link used in experiment (5) (10 Mbps, 1 ms. delay) with BER=1E-06. The size of transmission and receiving windows will be of 4 packets (4000 bytes) and timer's timeout will be set to 100 ms.

By default, TCP uses a Goback-N sliding-window protocol. In order to change this behaviour to the selective repeat variant we will enable "**Selective ACK (SACK)**" property in both the client and server TCPs. Comment the obtained results comparing both sliding-window approaches.