ENSC 427: COMMUNICATION NETWORKS
SPRING 2011

FINAL PROJECT
TRAFFIC ANALYSIS OF A SIMULATED BPL OVER PLC NETWORK

Web Address: http://www.sfu.ca/~gba2/

Gary Brykov
Yash Trivedi

301026383
301047924
gba2@sfu.ca
ymt1@sfu.ca

1 ABSTRACT

The use of Broadband over Power Lines (BPL) using Power Line Communication (PLC) can be used in medium/low-voltage supply networks for the delivery of various communication services, such as Internet access, voice over IP (VoIP), automatic meter reading (AMR), and home and building automation. BPL uses PLC by sending and receiving radio signals over power lines to provide access to the Internet. The BPL modems are capable of handling power line noise on a wide spectrum and are mandated to be designed to notch out frequencies on which radio interference with short-wave and ham radios occurs. Successful implementation of BPL enables distributing electricity via a digital network. This decentralized “smart grid” approach would overlay the ordinary electrical grid with an information and net metering system that could be used to manage and provision power distribution. For this project, we used OPNET to simulate a basic BPL over PLC network to investigate the burstiness (variation in data throughput) of three different types of traffic: constant-sized self-similar traffic, exponential-interarrival time traffic, and traffic following a Poisson-distribution pattern.

2 INTRODUCTION

The development of new devices which are able to communicate through PLC networks has changed its goal. Nowadays, these networks are used not only for power distribution, but to add new applications such as Internet access and IP (Internet protocol) based services. AMR and power transmission control using BPL over PLC can allow for closed-loop control using an observer-based controller design which is an ideal control method for the multiple-input multiple-output (MIMO) nature of the power distribution grid. This approach may result in lower electric power costs, less pollution and greater reliability, and home automation could be achieved by allowing home appliances to communicate with each other.

2.1 BPL Technology Overview

BPL technology provides an internet connection over electric distribution lines by installing power line adaptors at centralized locations to carry broadband Internet traffic over medium voltage power lines. These adaptors receive internet data and translate it using orthogonal frequency-division multiplexing (OFDM), which is the ideal modulation scheme for the noisy power line environment due to its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long wire and narrowband interference) without complex filters. The endpoint BPL modems separate the data from the electricity, sending the data to an Ethernet port. BPL can be used for providing up to 200Mbps (total, upstream plus downstream) by means of the power grid. The coverage is often the last half mile (typical distance from the medium-to-low voltage transformer to the customer premise meter) and as an in-home network.

Access BPL is a new technology to carry broadband Internet traffic over medium voltage (MV) power lines. Commonly there are three electric phases each carrying 1kV-30kV volts. One phase is usually enough to power the houses on a residential street, and two or three phases can be joined together to power the big electric motors in an industrial or commercial area. BPL modems are connected to the medium voltage power lines using inductive couplers which transfer the communications signal onto the power line without physical contact. A potential challenge is the delivery of the signal from the MV line to the low voltage (LV) line that enters the home or office, because the transformer that lowers the electric power from several thousand volts down to 110V/120V could act as a bottleneck to the broadband signal.

2.2 Traffic Models

Good traffic modeling is a basic requirement for accurate network planning. This report attempts to provide an overview of throughput variation for 3 network traffic models over the PLC link.

Analysis of the traffic provides information like the average load, the throughput, and numerous other details. Traffic models enable network designers to make assumptions about the networks being designed based on past experience and also enable prediction of performance for future requirements. Traffic models are used either as part of an analytical model or to drive a Discrete Event Simulation (DES), which is the case in this project.

The three traffic models used in the simulation are: (1) self-similar, (2) constant-sized exponential-interarrival, and (3) following a Poisson distribution.

Internet traffic can be modeled as a sequence of arrivals of discrete packets. A counting process $\{P(t)\}_{t=0..\infty}$ is a continuous-time, integer-valued process, where *P(t)* represents the number of packet arrivals in the time interval *(0,t]*. An interarrival time process is a non-negative random sequence $\{I_p\}$, where $I_p=T_p-T_{p-1}$ indicates the length of the interval separating packets *p-1* and *p*. Counting processes and interarrival time are related by

$$\{P(t)=p\}=\left\{T_p \le t < T_{p+1}\right\}=\left\{\sum_{k=1}^{p} I_k \le t < \sum_{k=1}^{p+1} I_k\right\}.$$

One important issue is the ability to describe traffic burstiness. A sequence of arrival times will be bursty if the *T_p* tend to form clusters and if the corresponding *$\{I_p\}$* sees a mix of relatively long and short interarrival times. Traffic experiencing high burstiness will vary fluctuate more in throughput than non-bursty traffic. The Hurst parameter *H* can be used as a measure of burstiness in case of self-similar traffic, and OPNET allows for configuration of this parameter.

2.2.1 Self-Similarity in Traffic

Self-similarity describes the phenomenon where a certain property of an object is preserved with respect to scaling in space or in time. The fact that much of network traffic is self-similar in nature is a reminder that new discoveries do not necessarily require new mathematical concepts, statistical techniques, or in this case, new networking technologies. Instead, the discovery often lies in applying a well-known mathematical concept such a self-similarity in a newer context such as networking. Network traffic that is bursty on many or all timescales can be described statistically using the notion of self-similarity, where the traffic structure remains unchanged at varying timescales.

The definition of self-similarity involves a packet sequence

$$X = X(i), i \geq 1.$$

Let

$$X^m(k) = \frac{1}{m}\sum_{i=(k-1)m+1}^{km} X(i), k = 1,2,3 \ldots$$

be the corresponding aggregated sequence with level of aggregation *m*, obtained by dividing the original series *X* into non-overlapping blocks of size *m* and averaging over each block. The index *k* labels the block. If *X* is the increment process of a self-similar process, then if for all integers *m*,

$$X = m^{1-H}X^m,$$

where *H* is the Hurst Parameter and the sequence is said to be exactly self-similar.

2.2.2 Exponential Interarrival ON-OFF Traffic

In the exponential interarrival on-off model, the packet stream from a single source is modeled as a sequence of alternating burst periods and silence periods. The duration of each burst is exponentially distributed with mean *1/a*. During such a period, packets are generated with constant interarrival time *T*. After that, an exponentially distributed silence period with mean value *1/b* follows. Three parameters that can be configured in OPNET for exponentially-arriving traffic are:

- Peak bit rate (*p*): peak bit rate of packets when the source is in the active state, or the maximum amount of network bandwidth needed by the source.
- Mean bit rate (*m*): the mean bit rate, or the average amount of network bandwidth requested by the source.
- Mean active time(T_{on}): the mean active/holding duration.

Burstiness (*β*), can be defined as the ratio of the peak packet rate and the average packet rate (*b = p/m)*. The corresponding values for a^{-1}, b^{-1}, *T* can therefore be calculated as

$$T=1/p,$$

$$a^{-1} = T_{on,}$$

and

$$b^{-1}=a^{-1}*(\beta -1).$$

2.2.3 The Poisson Distribution and Network Traffic

Assuming pure-chance packet arrivals and pure-chance terminations means that the number of packet arrivals in a given time has a Poisson distribution:

$$P(a) = \left(\frac{\mu^{a}}{a!}\right)e^{-\mu},$$

where *a* is the number of call arrivals and μ is the mean number of call arrivals in time *T*. For this reason, pure-chance traffic is also known as Poisson traffic. The number of call departures in a given time also has a Poisson distribution:

$$P(d) = \left(\frac{\lambda^{d}}{d!}\right)e^{-\lambda},$$

where *d* is the number of call departures and λ is the mean number of call departures in time *T*. The intervals, *T*, between call arrivals and departures are intervals between independent, identically distributed random events. It can be shown that these intervals have a negative exponential distribution:

$$P[T \geq t] = e^{-t/d},$$

where *d* is the mean holding time.

3 NETWORK MODELING

3.1 General Network Topology

In the BPL over PLC network, there is a central device which distributes the power to all endpoint units within homes. The power line link utilized by BPL will be modelled in OPNET at medium voltage (1000 – 35,000 V) over standard aluminum power lines. The internet signal is transmitted from the BiPAC 2300 BPL Access head-end unit over this transmission medium and received at the destination building by a BiPAC 2103 BPL Access end-point unit (modem), which is connected by Ethernet to the computer or device. The general network topology is illustrated in **Figure 1**.

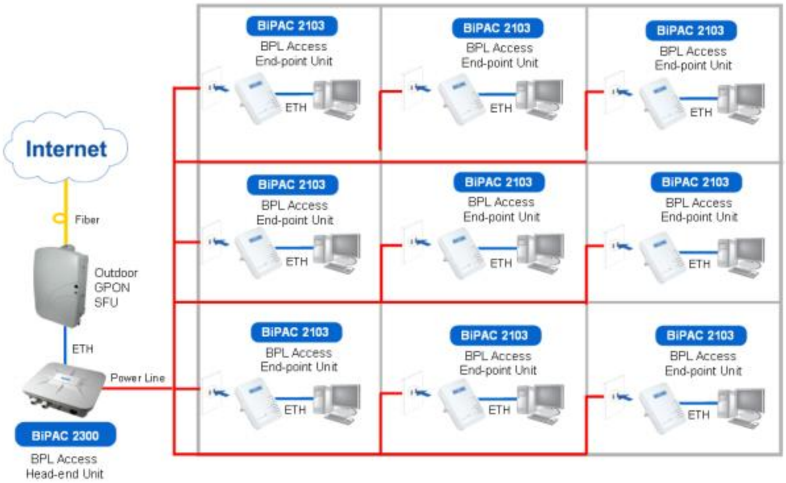


Figure 1: BPL Network Topology

For the purposes of this project, the simulated network model consists of the BiPAC 2300 Access Head-end unit, two BiPAC 2103 Access End-point units, and a workstation connected to each end-point unit. The head-end unit is connected to internet source traffic nodes, which generate one of the three traffic types discussed in the section above. The simplified model diagram is shown in **Figure 2**.

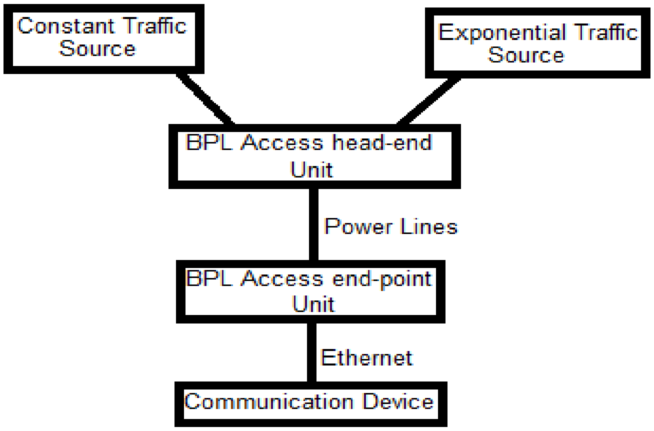


Figure 2: Simplified BPL Network Diagram

3.2 OPNET Modelling

OPNET is a discrete event simulator that allows simulating and analyzing protocols. In the case of this project, since the power-line physical layer is not natively supported by OPNET, the improvised solution is the modification of the centralized head-end node's processor model in order for it to act as a medium for communication between the other network nodes. The central process model encapsulates the physical layer emulator and emulates the duplex 100Mbit/second link connecting the head-end and end-point units as MV power line.

Each OPNET processor model is defined by a finite state machine which represents the module's logic and contains the module's behaviour described in OPNET Proto C code.

For this project, two scenarios were created and simulated. The first deals with self-similar and exponential traffic being generated and carried through PLC MV link to a head-end unit, where it is bridged to the end-point units, where it is demodulated and converted into Ethernet data and delivered to the PC workstation. The second scenario is identical except for the fact that exponentially-distributed traffic is replaced with Poisson-distributed traffic.

3.2.1 Node and Process Modelling of BPL End-point and Head-end Units

The BiPAC 2103 BPL Access End-point Unit was modeled in OPNET as a modem and demodulator utilizing OFDM to transmit signal over power line into Ethernet data. **Figure 3** shows the node model of the end-point unit.

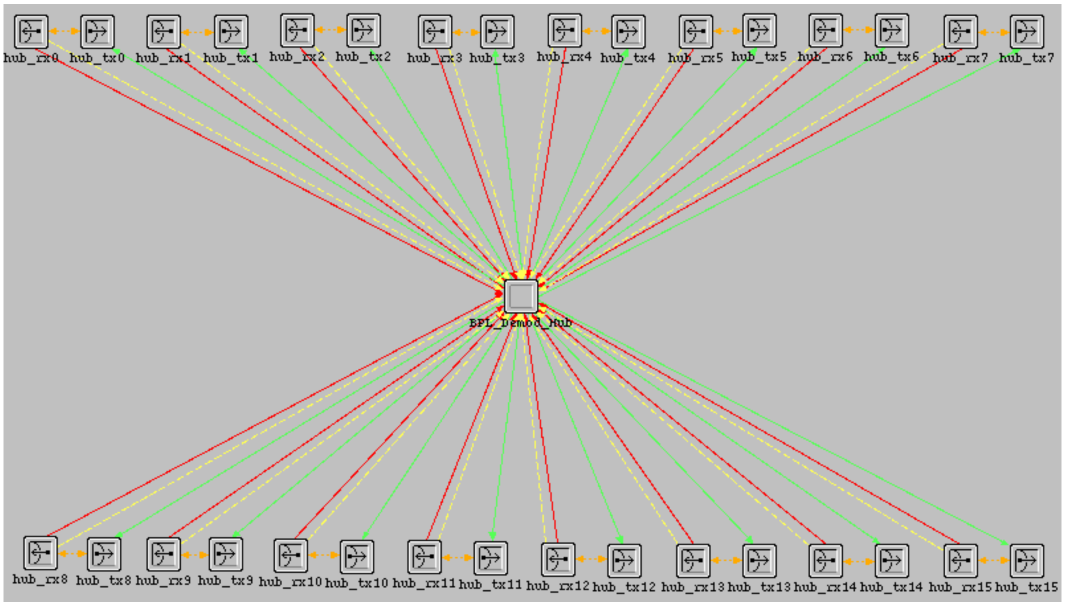


Figure 3: BPL End-point Unit Node Model

Figure 4 shows the associated process model of the BPL_Demod_Hub process.

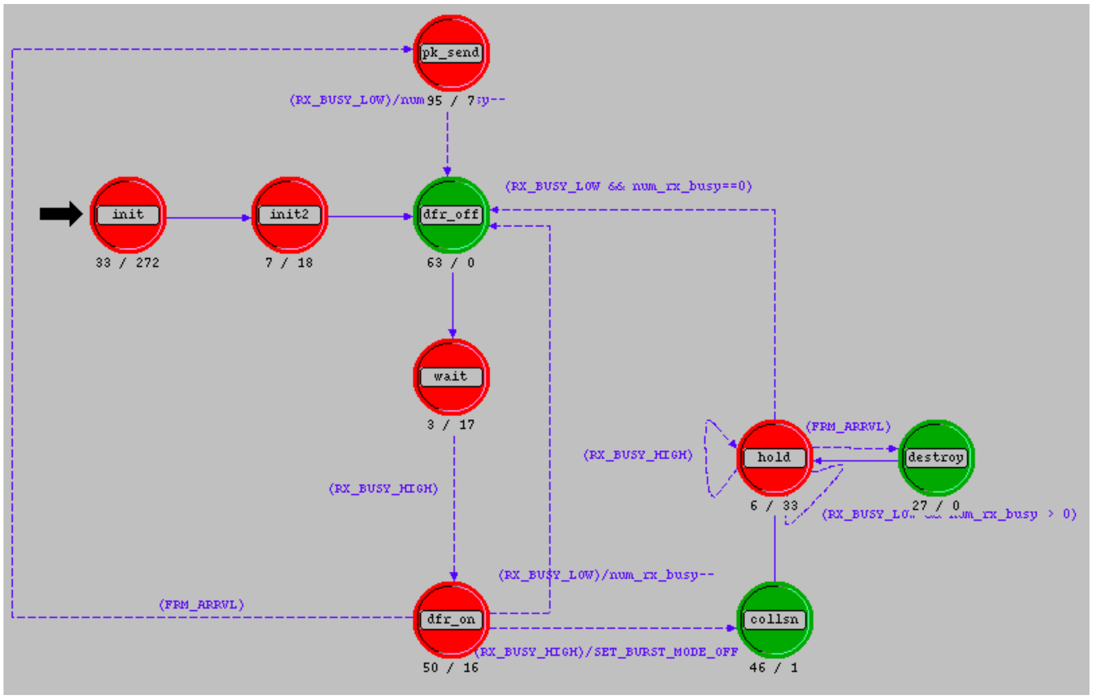
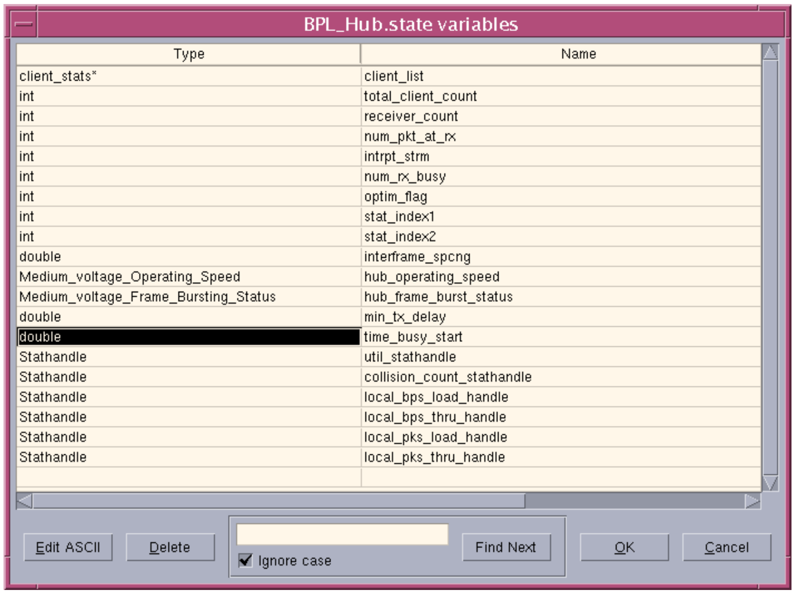

pk_send
(RX_BUSY_LOW)/num 95 / 7 sy--
init
33 / 272
init2
7 / 18
dfr_off
63 / 0
(RX_BUSY_LOW && num_rx_busy==0)
wait
3 / 17
(RX_BUSY_HIGH)
(FRM_ARRVL)
hold
6 / 33
destroy
27 / 0
(RX_BUSY_LO... num_rx_busy > 0)
(RX_BUSY_HIGH)
(RX_BUSY_LOW)/num_rx_busy--
(FRM_ARRVL)
dfr_on
50 / 16
(RX_BUSY_HIGH)/SET_BURST_MODE_OFF
collsn
46 / 1

Figure 4: BPL End-point Unit Process Model

The state variables for the above shown process model are displayed in **Figure 5**.



BPL_Hub.state variables

Type	Name
client_stats*	client_list
int	total_client_count
int	receiver_count
int	num_pkt_at_rx
int	intrpt_strm
int	num_rx_busy
int	optim_flag
int	stat_index1
int	stat_index2
double	interframe_spcng
Medium_voltage_Operating_Speed	hub_operating_speed
Medium_voltage_Frame_Bursting_Status	hub_frame_burst_status
double	min_tx_delay
double	time_busy_start
Stathandle	util_stathandle
Stathandle	collision_count_stathandle
Stathandle	local_bps_load_handle
Stathandle	local_bps_thru_handle
Stathandle	local_pks_load_handle
Stathandle	local_pks_thru_handle

Figure 5: BPL End-point Unit Process State Variables

The BiPAC 2300 BPL Access Head-end Unit was modeled in OPNET as a BPL bridge and a medium which encapsulates the physical layer emulator in order to allow approximate modelling of the PLC link. **Figure 6** shows the node model of the head-end unit.

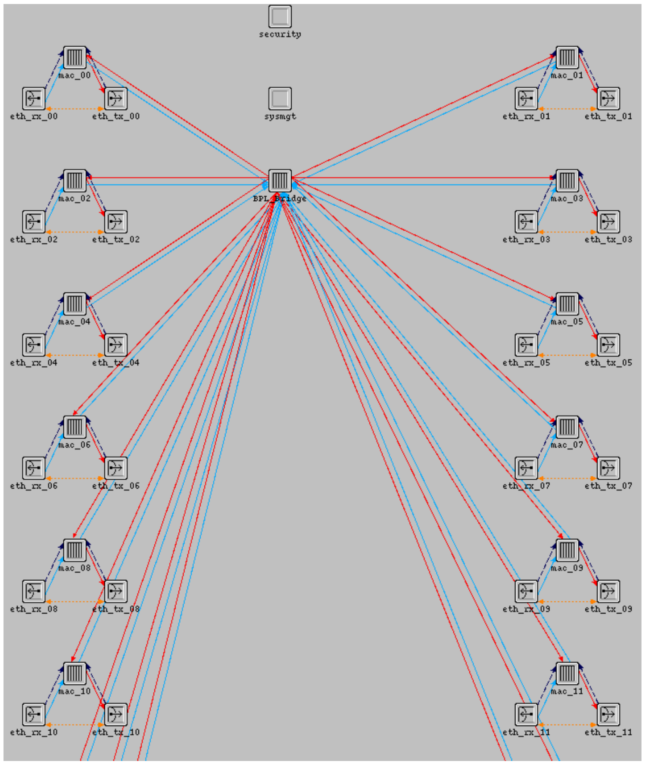


Figure 6: BPL Head-end Unit Node Model

Figure 7 shows the associated process model of the BPL_Bridge process, which ended up consisting of five states, with physical layer emulation being performed in the idle state's exit block.

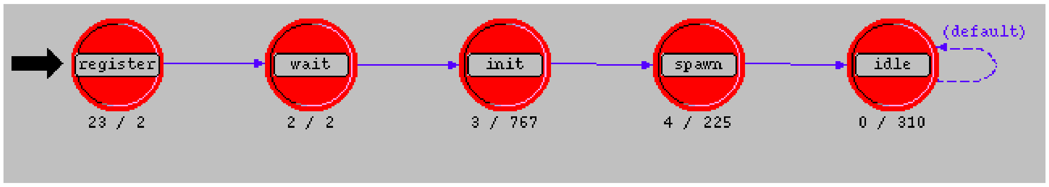
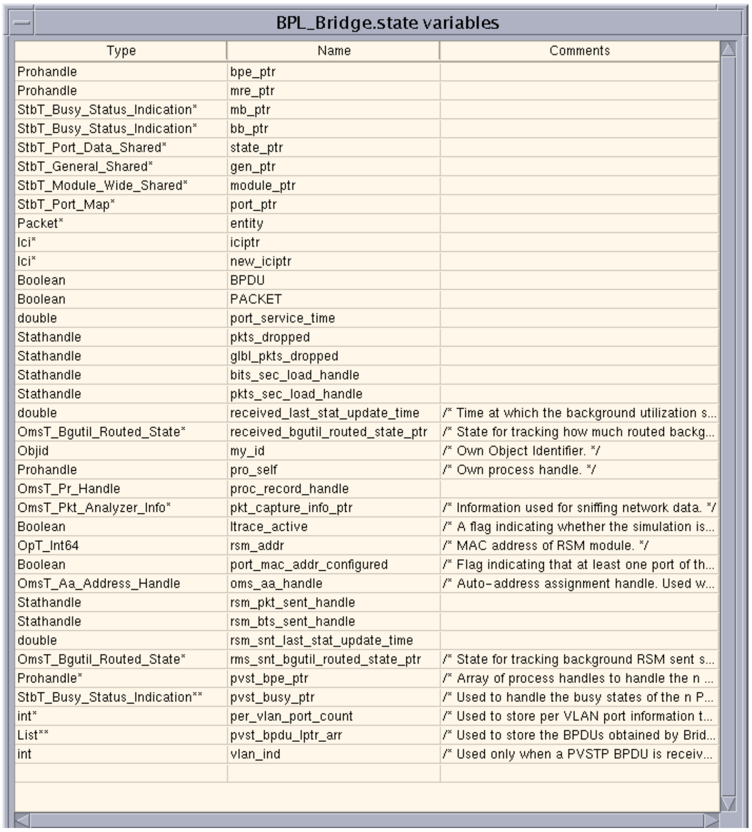


Figure 7: BPL Head-end Unit Process Model

The state variables for the above shown process model are displayed in **Figure 8**.



Type	Name	Comments
Prohandle	bpe_ptr	
Prohandle	mre_ptr	
StbT_Busy_Status_Indication*	mb_ptr	
StbT_Busy_Status_Indication*	bb_ptr	
StbT_Port_Data_Shared*	state_ptr	
StbT_General_Shared*	gen_ptr	
StbT_Module_Wide_Shared*	module_ptr	
StbT_Port_Map*	port_ptr	
Packet*	entity	
Ici*	iciptr	
Ici*	new_iciptr	
Boolean	BPDU	
Boolean	PACKET	
double	port_service_time	
Stathandle	pkts_dropped	
Stathandle	glbl_pkts_dropped	
Stathandle	bits_sec_load_handle	
Stathandle	pkts_sec_load_handle	
double	received_last_stat_update_time	/* Time at which the background utilization s...
OmsT_Bgutil_Routed_State*	received_bgutil_routed_state_ptr	/* State for tracking how much routed backg...
Objid	my_id	/* Own Object Identifier. */
Prohandle	pro_self	/* Own process handle. */
OmsT_Pr_Handle	proc_record_handle	
OmsT_Pkt_Analyzer_Info*	pkt_capture_info_ptr	/* Information used for sniffing network data. */
Boolean	ltrace_active	/* A flag indicating whether the simulation is...
OpT_Int64	rsm_addr	/* MAC address of RSM module. */
Boolean	port_mac_addr_configured	/* Flag indicating that at least one port of th...
OmsT_Aa_Address_Handle	oms_aa_handle	/* Auto-address assignment handle. Used w...
Stathandle	rsm_pkt_sent_handle	
Stathandle	rsm_bts_sent_handle	
double	rsm_snt_last_stat_update_time	
OmsT_Bgutil_Routed_State*	rms_snt_bgutil_routed_state_ptr	/* State for tracking background RSM sent s...
Prohandle*	pvst_bpe_ptr	/* Array of process handles to handle the n ...
StbT_Busy_Status_Indication**	pvst_busy_ptr	/* Used to handle the busy states of the n P...
int*	per_vlan_port_count	/* Used to store per VLAN port information t...
List**	pvst_bpdu_lptr_arr	/* Used to store the BPDUs obtained by Brid...
int	vlan_ind	/* Used only when a PVSTP BPDU is receiv...

Figure 8: BPL Head-end Unit Process State Variables

3.2.2 PLC Link Modelling

Because a power-line link is not natively supported in OPNET, two problems needed to be solved. First, the speed in access BPL is not equal to that of Ethernet, and the speed also fluctuates over different environments. Second, the distribution of the users in a power grid is much more random than that of the bus model generated by OPNET.

Bus topology versus Star topology

The feasibility of replacing a bus topology by a star topology is tested using a contributed model downloaded from the OPNET online model library [4]. The same parameters are set for these two topologies as shown in **Figures 9** and **10**. The distance distribution from users to the sensor is identical in both topologies.

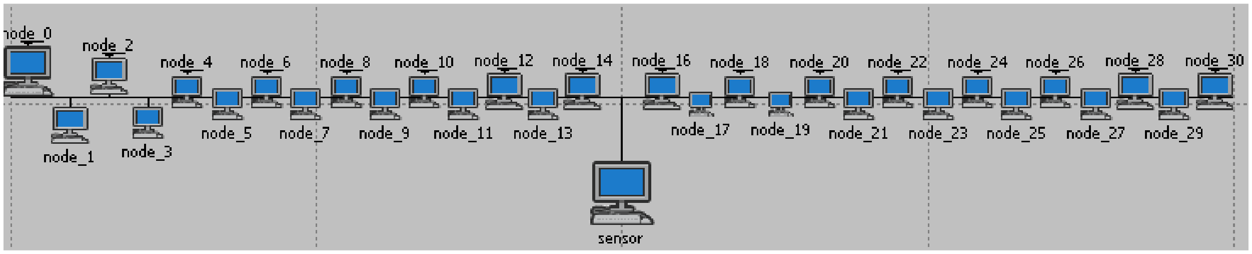


Figure 9: Bus Topology

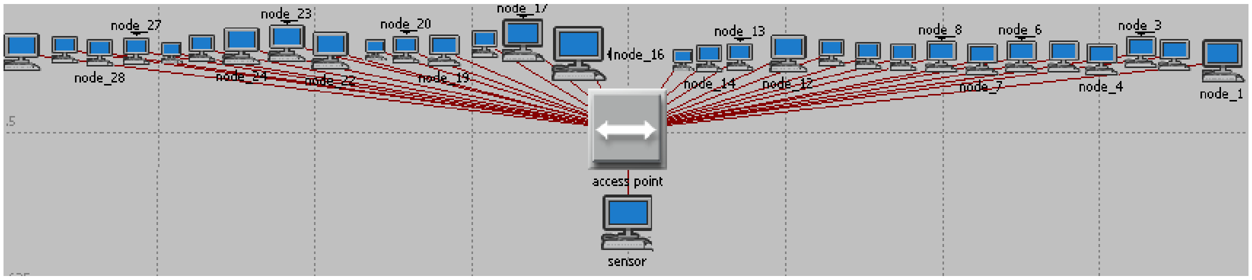


Figure 10: Star Topology

In order to eliminate the influence of the random number generator in the software, 100 different simulation seeds are defined. The end-to-end delay is chosen as the measurement criterion. The results are shown in **Figure 11**. From the results of the simulation, it can be found that for a large number of experiments with different random seeds, the delay difference between star topology and bus topology becomes smaller. This justifies that the bus topology can be replaced by a star topology in the lab environment.

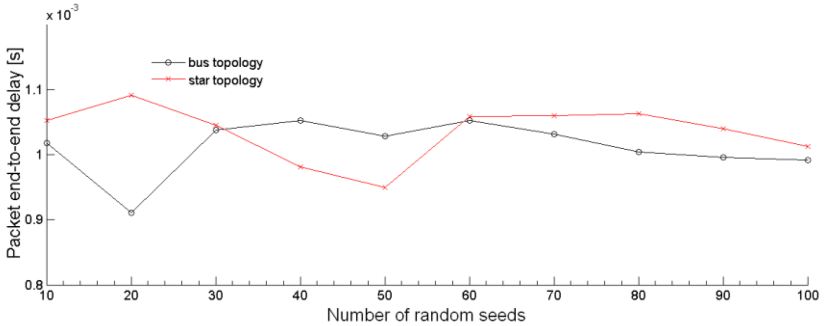
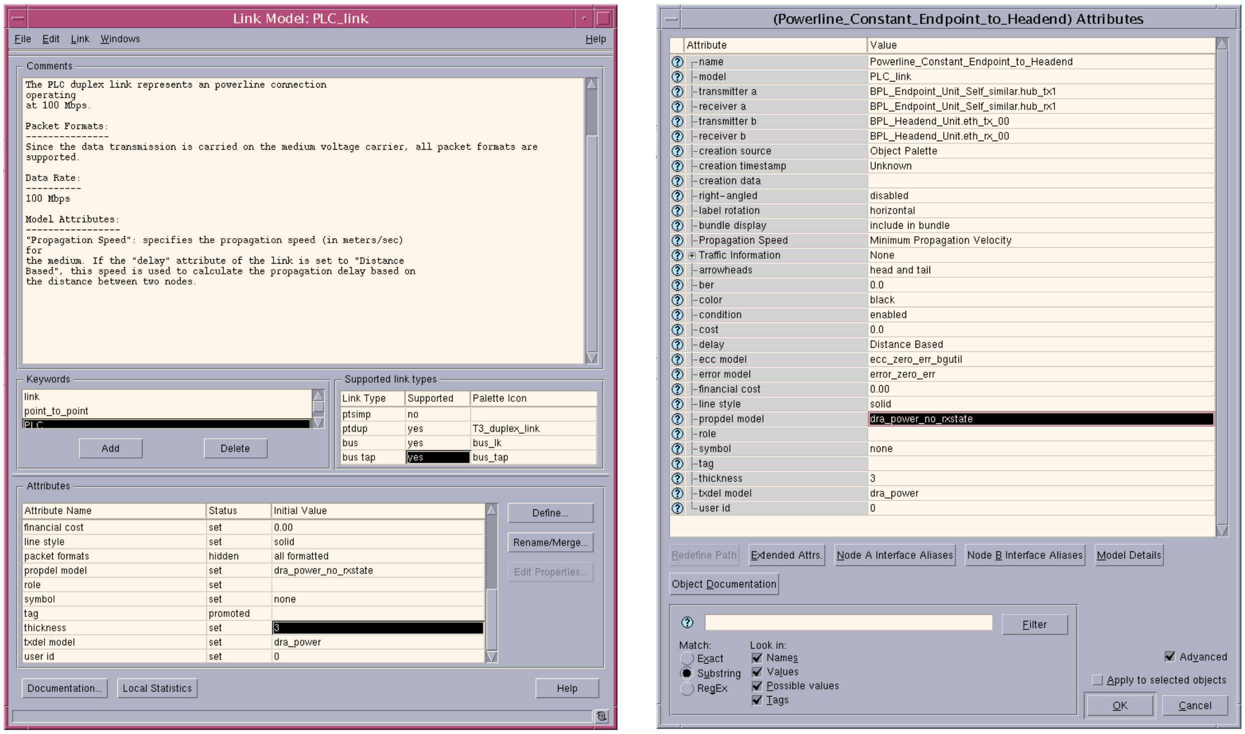


Figure 11: Comparison of average packet end-to-end delay

The PLC link model parameters and attributes are shown in **Figure 12**.



```
The PLC duplex link represents an powerline connection
operating
at 100 Mbps.

Packet Formats:
---------------
Since the data transmission is carried on the medium voltage carrier, all packet formats are
supported.

Data Rate:
----------
100 Mbps

Model Attributes:
-----------------
"Propagation Speed": specifies the propagation speed (in meters/sec)
for
the medium. If the "delay" attribute of the link is set to "Distance
Based", this speed is used to calculate the propagation delay based on
the distance between two nodes.
```

link
point_to_point
PLC

Link Type	Supported	Palette Icon
ptsimp	no	
ptdup	yes	T3_duplex_link
bus	yes	bus_lk
bus tap	yes	bus_tap

Attribute Name	Status	Initial Value
financial cost	set	0.00
line style	set	solid
packet formats	hidden	all formatted
propdel model	set	dra_power_no_rxstate
role	set	
symbol	set	none
tag	promoted	
thickness	set	3
txdel model	set	dra_power
user id	set	0

Attribute	Value
name	Powerline_Constant_Endpoint_to_Headend
model	PLC_link
transmitter a	BPL_Endpoint_Unit_Self_similar.hub_tx1
receiver a	BPL_Endpoint_Unit_Self_similar.hub_rx1
transmitter b	BPL_Headend_Unit.eth_tx_00
receiver b	BPL_Headend_Unit.eth_rx_00
creation source	Object Palette
creation timestamp	Unknown
creation data	
right-angled	disabled
label rotation	horizontal
bundle display	include in bundle
Propagation Speed	Minimum Propagation Velocity
Traffic Information	None
arrowheads	head and tail
ber	0.0
color	black
condition	enabled
cost	0.0
delay	Distance Based
ecc model	ecc_zero_err_bgutil
error model	error_zero_err
financial cost	0.00
line style	solid
propdel model	dra_power_no_rxstate
role	
symbol	none
tag	
thickness	3
txdel model	dra_power
user id	0

Match: ○ Exact ● Substring ○ RegEx

Look in: ☑ Names ☑ Values ☑ Possible values ☑ Tags

☑ Advanced

□ Apply to selected objects

Figure 12: PLC Link Model and Attributes

3.2.3 Network Construction of Scenario 1 – Self-similar vs. Exponential

The OPNET network model for Scenario 1 is shown below in **Figure 13**. The scenario consists of a star topology with one BPL head-end unit, two BPL end-point units, two PC workstations and two traffic sources – one generating constant-sized packet traffic using a self-similar, or fractal, distribution pattern and one generating constant-sized packet traffic using an exponential packet interarrival rate. The self-similar traffic was configured with *Hurst parameter = 0.8* and a *fractal onset time scale of 0.001*.

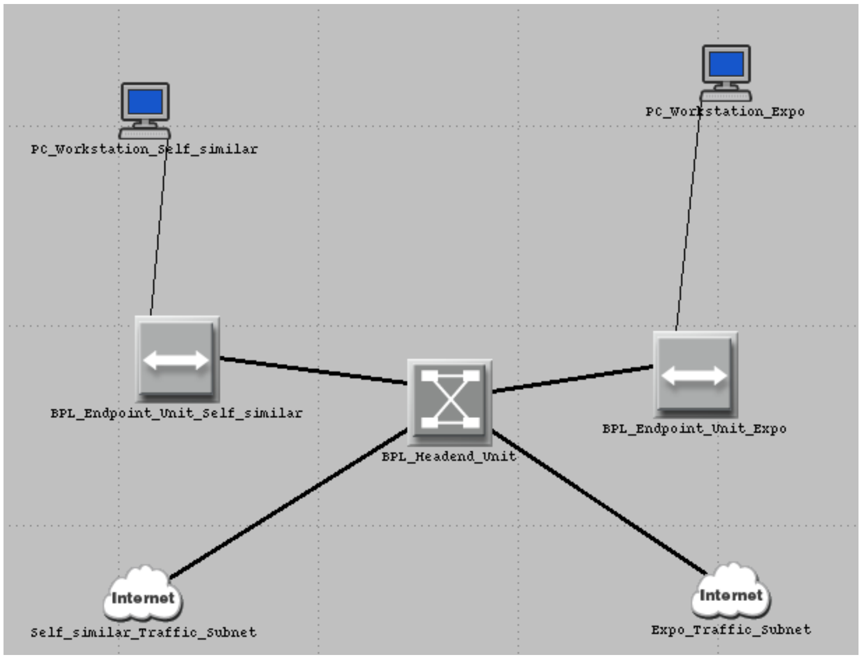


Figure 13: Scenario 1 Network Schematic

As evident from **Figure 13**, the head-end unit is connected to the end-point units and the traffic sources via PLC link (thick line), which has been approximately modelled as a 100Mbit/second data link with high susceptibility to background noise. The configured attributes for the PLC link are shown later on in this report. The PC workstations, each receiving their own type of traffic, are connected to the end-point units with 100BaseT duplex Ethernet link. **Figures 14** and **15** illustrates the node and process model for the self-similar traffic generator subnet, as well as configuration parameters and process-model state variables, respectively.

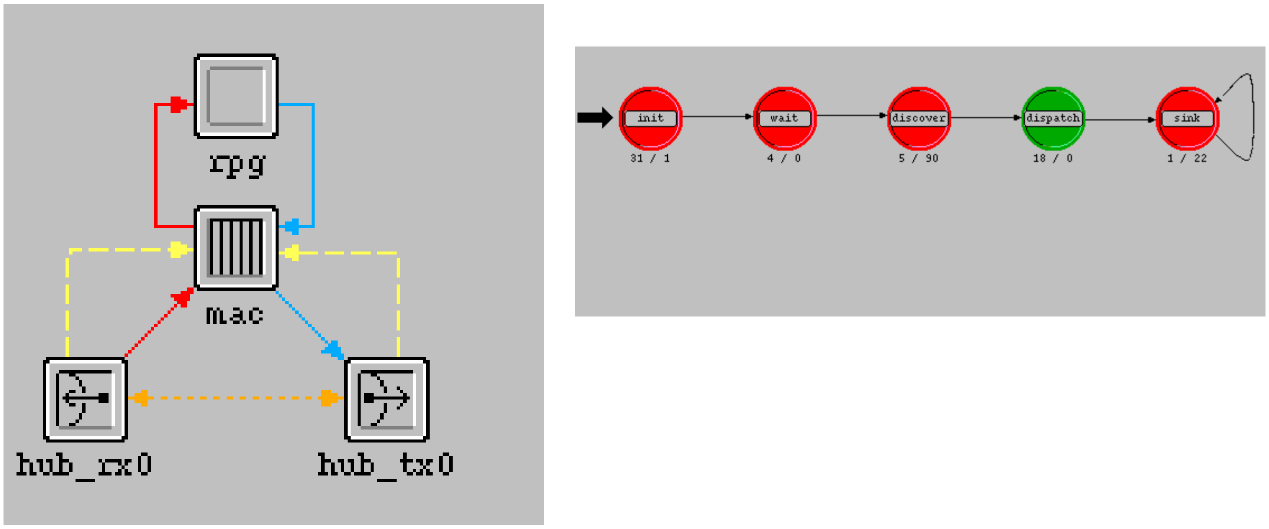
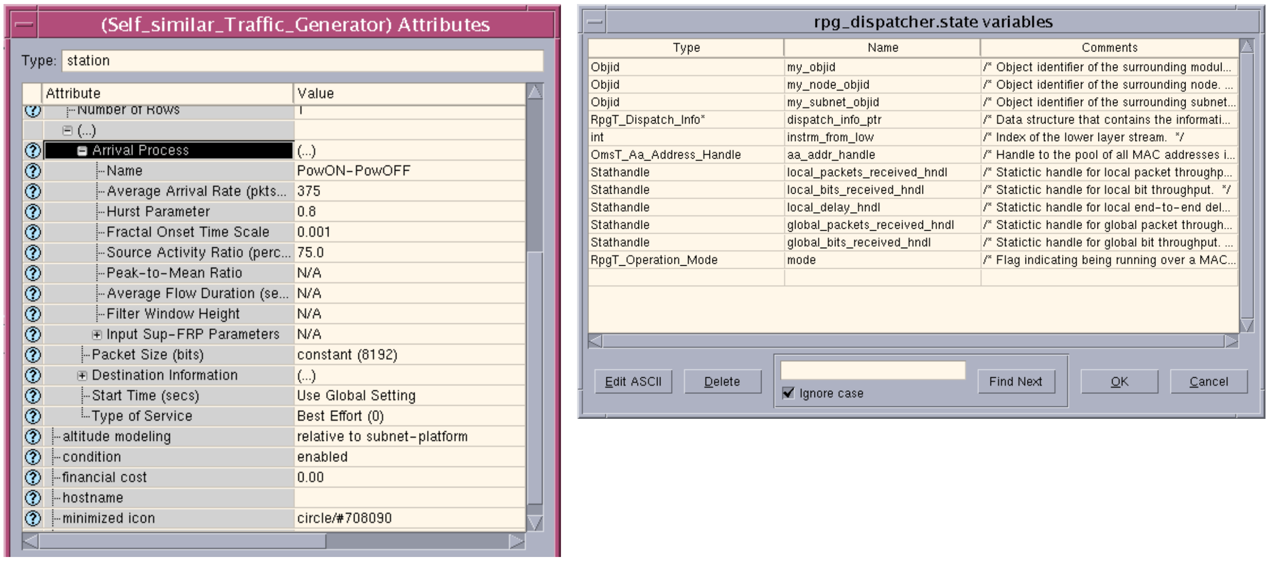


Figure 14: Self-similar Traffic Generator Node and Process Model



(Self_similar_Traffic_Generator) Attributes

Type: station

Attribute	Value
Number of Rows	1
⊟ (...)	
⊟ Arrival Process	(...)
Name	PowON-PowOFF
Average Arrival Rate (pkts...	375
Hurst Parameter	0.8
Fractal Onset Time Scale	0.001
Source Activity Ratio (perc...	75.0
Peak-to-Mean Ratio	N/A
Average Flow Duration (se...	N/A
Filter Window Height	N/A
⊞ Input Sup-FRP Parameters	N/A
Packet Size (bits)	constant (8192)
⊞ Destination Information	(...)
Start Time (secs)	Use Global Setting
Type of Service	Best Effort (0)
altitude modeling	relative to subnet-platform
condition	enabled
financial cost	0.00
hostname	
minimized icon	circle/#708090

rpg_dispatcher.state variables

Type	Name	Comments
Objid	my_objid	/* Object identifier of the surrounding modul...
Objid	my_node_objid	/* Object identifier of the surrounding node. ...
Objid	my_subnet_objid	/* Object identifier of the surrounding subnet...
RpgT_Dispatch_Info*	dispatch_info_ptr	/* Data structure that contains the informati...
int	instrm_from_low	/* Index of the lower layer stream. */
OmsT_Aa_Address_Handle	aa_addr_handle	/* Handle to the pool of all MAC addresses i...
Stathandle	local_packets_received_hndl	/* Statictic handle for local packet throughp...
Stathandle	local_bits_received_hndl	/* Statictic handle for local bit throughput. */
Stathandle	local_delay_hndl	/* Statictic handle for local end-to-end del...
Stathandle	global_packets_received_hndl	/* Statictic handle for global packet through...
Stathandle	global_bits_received_hndl	/* Statictic handle for global bit throughput. ...
RpgT_Operation_Mode	mode	/* Flag indicating being running over a MAC...

Figure 15: Self-similar Traffic Generator Attributes and State Variables

As seen in the figure, the Hurst parameter for the self-similar traffic generator was set at 0.8. This was done to ensure that the increments of packet traffic were positively correlated.

The enter executives of the initialization state in the self-similar packet generator is shown in the code snippet below.

```
/* Initialize the state variables used by this model.                 */
rpg_dispatcher_sv_init ();

/* Register RPG as a higher layer protocol over IP layer and retrieve  */
/* an auto-assigned protocol id.                                       */
higher_layer_proto_id = IpC_Protocol_Unspec;
Ip_Higher_Layer_Protocol_Register ("rpg", &higher_layer_proto_id);

/* Register this process in the network wide process registery so that */
/* lower layer can detect our existence.                               */
rpg_dispatcher_register_self ();

/* Read RPG related simulation attribute values into global variables  */
/* unless another rpg_dispatcher process already did this task.        */
if (rpg_sim_attrs_read == OPC_FALSE)
    {
    /*  We are the first. Read the attributes if they exist (otherwise */
    /* the default value is 100.0 secs).                               */
    if (op_ima_sim_attr_exists ("RPG Start Time") == OPC_TRUE)
        op_ima_sim_attr_get (OPC_IMA_DOUBLE, "RPG Start Time", &rpg_start_time);

    /* Switch the flag.                                                */
    rpg_sim_attrs_read = OPC_TRUE;
    }

/* Schedule a self interrupt to wait for lower layer process to        */
/* initialize and register itself in the model-wide process registry.  */
/* This is necessary since global RPG start time may have been set as  */
/* low as zero seconds, which is acceptable when operating over MAC    */
/* layer.                                                              */
op_intrpt_schedule_self (op_sim_time (), RPGC_INIT);
```

This code was commented in order to be explanatory.

Figures 16 and **17** illustrates the node and process model for the exponential traffic generator, and the configuration parameters and process-model state variables, respectively.

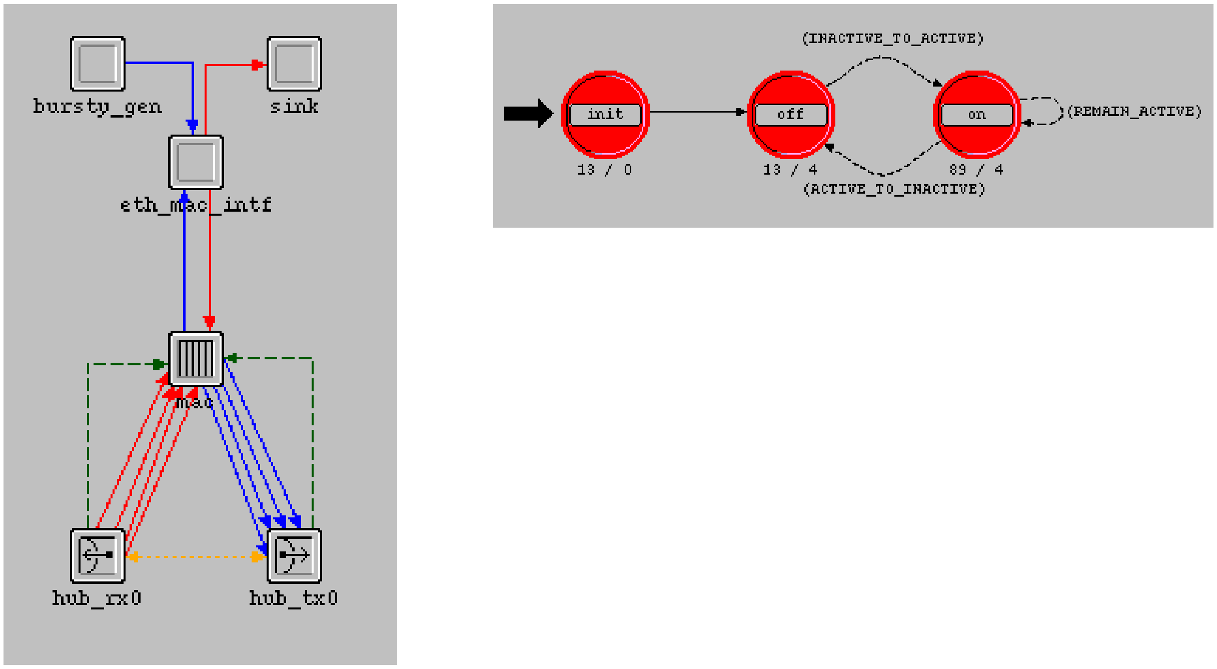
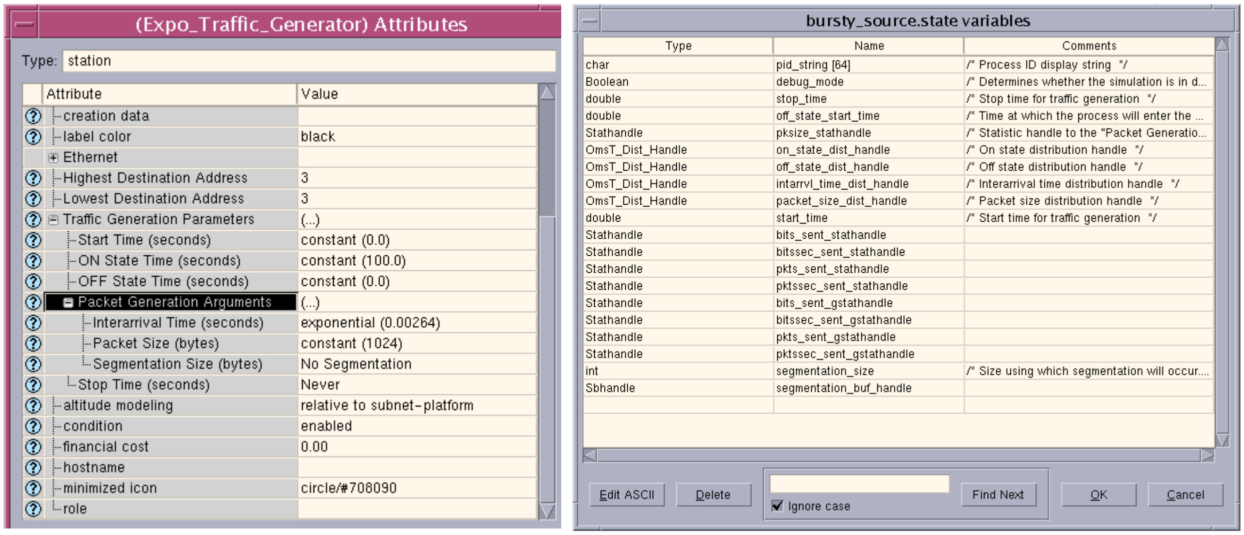


Figure 16: Exponential Traffic Generator Node and Process Model


(Expo_Traffic_Generator) Attributes

Type: station

Attribute	Value
creation data	
label color	black
Ethernet	
Highest Destination Address	3
Lowest Destination Address	3
Traffic Generation Parameters	(...)
Start Time (seconds)	constant (0.0)
ON State Time (seconds)	constant (100.0)
OFF State Time (seconds)	constant (0.0)
Packet Generation Arguments	(...)
Interarrival Time (seconds)	exponential (0.00264)
Packet Size (bytes)	constant (1024)
Segmentation Size (bytes)	No Segmentation
Stop Time (seconds)	Never
altitude modeling	relative to subnet-platform
condition	enabled
financial cost	0.00
hostname	
minimized icon	circle/#708090
role	

bursty_source.state variables

Type	Name	Comments
char	pid_string [64]	/* Process ID display string */
Boolean	debug_mode	/* Determines whether the simulation is in d...
double	stop_time	/* Stop time for traffic generation */
double	off_state_start_time	/* Time at which the process will enter the ...
Stathandle	pksize_stathandle	/* Statistic handle to the "Packet Generatio...
OmsT_Dist_Handle	on_state_dist_handle	/* On state distribution handle */
OmsT_Dist_Handle	off_state_dist_handle	/* Off state distribution handle */
OmsT_Dist_Handle	intarrvl_time_dist_handle	/* Interarrival time distribution handle */
OmsT_Dist_Handle	packet_size_dist_handle	/* Packet size distribution handle */
double	start_time	/* Start time for traffic generation */
Stathandle	bits_sent_stathandle	
Stathandle	bitssec_sent_stathandle	
Stathandle	pkts_sent_stathandle	
Stathandle	pktssec_sent_stathandle	
Stathandle	bits_sent_gstathandle	
Stathandle	bitssec_sent_gstathandle	
Stathandle	pkts_sent_gstathandle	
Stathandle	pktssec_sent_gstathandle	
int	segmentation_size	/* Size using which segmentation will occur....
Sbhandle	segmentation_buf_handle	

Figure 17: Exponential Traffic Generator Attributes and State Variables

3.2.4 Network Construction of Scenario 2 – Self-similar vs. Poisson

The OPNET network model for Scenario 2 is shown below in **Figure 18**. This scenario consists of a star topology with one BPL head-end unit, two BPL end-point units, two PC workstations and two traffic sources – one generating constant-sized packet traffic using a self-similar, or fractal, distribution pattern and one generating packet traffic using a Poisson distribution packet size and interarrival time. The self-similar traffic was configured with *Hurst parameter = 0.8* and a *fractal onset time scale of 0.001* and the Poisson traffic with a *mean of 5*.

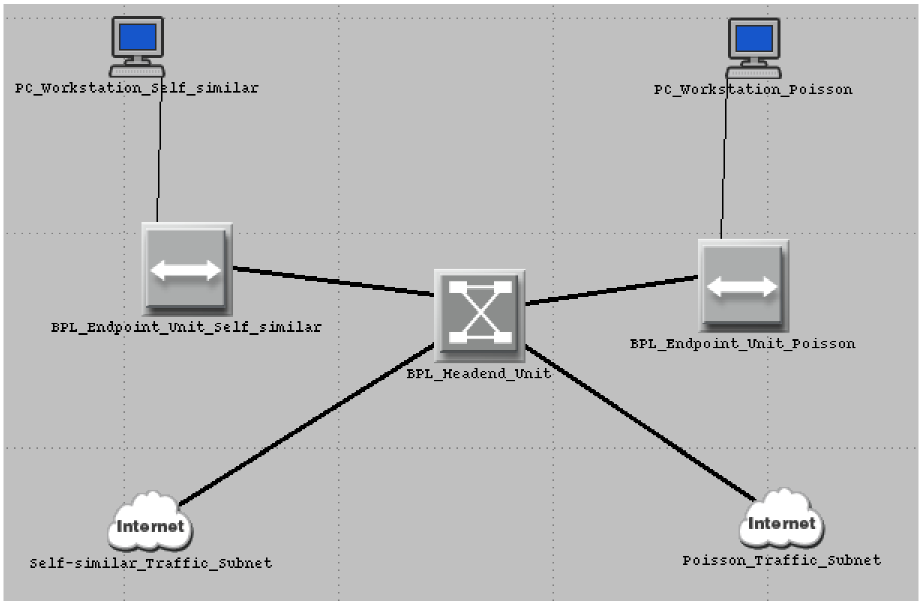
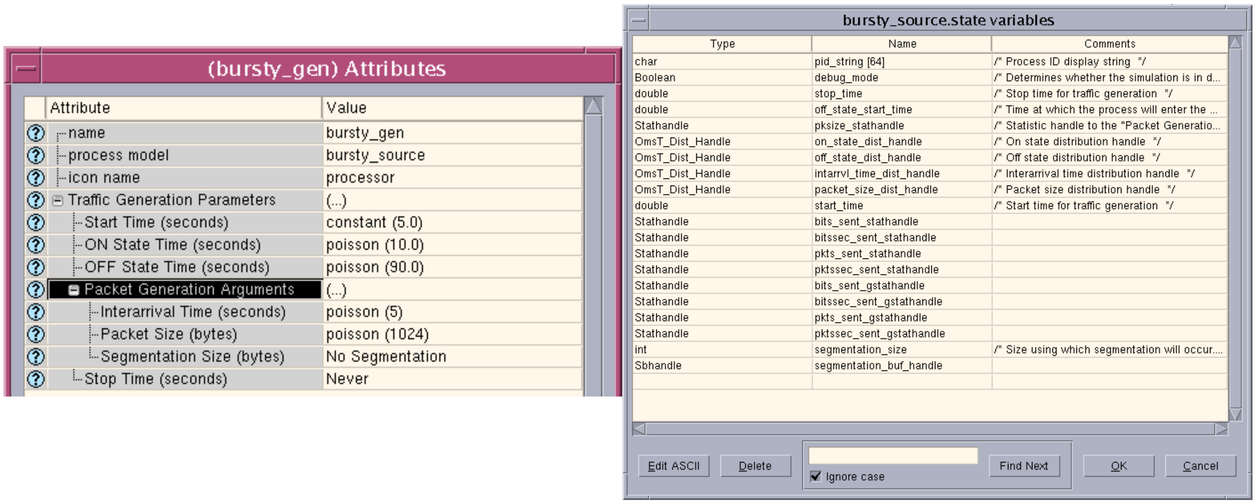

PC_Workstation_Self_similar
PC_Workstation_Poisson
BPL_Endpoint_Unit_Self_similar
BPL_Headend_Unit
BPL_Endpoint_Unit_Poisson
Internet
Self-similar_Traffic_Subnet
Internet
Poisson_Traffic_Subnet

Figure 18: Scenario 2 Network Schematic

As in Scenario 1, the head-end unit is connected to the end-point units and the traffic sources via PLC link (thick line), which has been approximately modelled as a 100Mbit/second data link with high susceptibility to background noise. The configured attributes for the PLC link are shown later on in this report. The PC workstations, each receiving their own type of traffic, are connected to the end-point units with 100BaseT duplex Ethernet link. The node and process model for the self-similar traffic generator subnet, as well as configuration parameters and process-model state variables are identical to those as in Scenario 1 and are shown in **Figures 14** and 1**5**.

Figure 19 illustrates the configuration parameters and process-model state variables for the Poisson-distribution traffic generator subnet.



(bursty_gen) Attributes

Attribute	Value
name	bursty_gen
process model	bursty_source
icon name	processor
⊟ Traffic Generation Parameters	(...)
Start Time (seconds)	constant (5.0)
ON State Time (seconds)	poisson (10.0)
OFF State Time (seconds)	poisson (90.0)
⊟ Packet Generation Arguments	(...)
Interarrival Time (seconds)	poisson (5)
Packet Size (bytes)	poisson (1024)
Segmentation Size (bytes)	No Segmentation
Stop Time (seconds)	Never

bursty_source.state variables

Type	Name	Comments
char	pid_string [64]	/* Process ID display string */
Boolean	debug_mode	/* Determines whether the simulation is in d...
double	stop_time	/* Stop time for traffic generation */
double	off_state_start_time	/* Time at which the process will enter the ...
Stathandle	pksize_stathandle	/* Statistic handle to the "Packet Generatio...
OmsT_Dist_Handle	on_state_dist_handle	/* On state distribution handle */
OmsT_Dist_Handle	off_state_dist_handle	/* Off state distribution handle */
OmsT_Dist_Handle	intarrvl_time_dist_handle	/* Interarrival time distribution handle */
OmsT_Dist_Handle	packet_size_dist_handle	/* Packet size distribution handle */
double	start_time	/* Start time for traffic generation */
Stathandle	bits_sent_stathandle	
Stathandle	bitssec_sent_stathandle	
Stathandle	pkts_sent_stathandle	
Stathandle	pktssec_sent_stathandle	
Stathandle	bits_sent_gstathandle	
Stathandle	bitssec_sent_gstathandle	
Stathandle	pkts_sent_gstathandle	
Stathandle	pktssec_sent_gstathandle	
int	segmentation_size	/* Size using which segmentation will occur....
Sbhandle	segmentation_buf_handle	

Figure 19: Poisson Traffic Generator Attributes and State Variables

Once the two scenarios were created, individual DES results were selected in order to enable collection of data throughput and queuing delay for the PLC links connecting the network nodes. These results are discussed in the next section of the report.

4 DISCUSSION AND CONCLUSIONS

The modelled scenarios were simulated in OPNET for a simulation duration of 25 minutes. The results are displayed below.

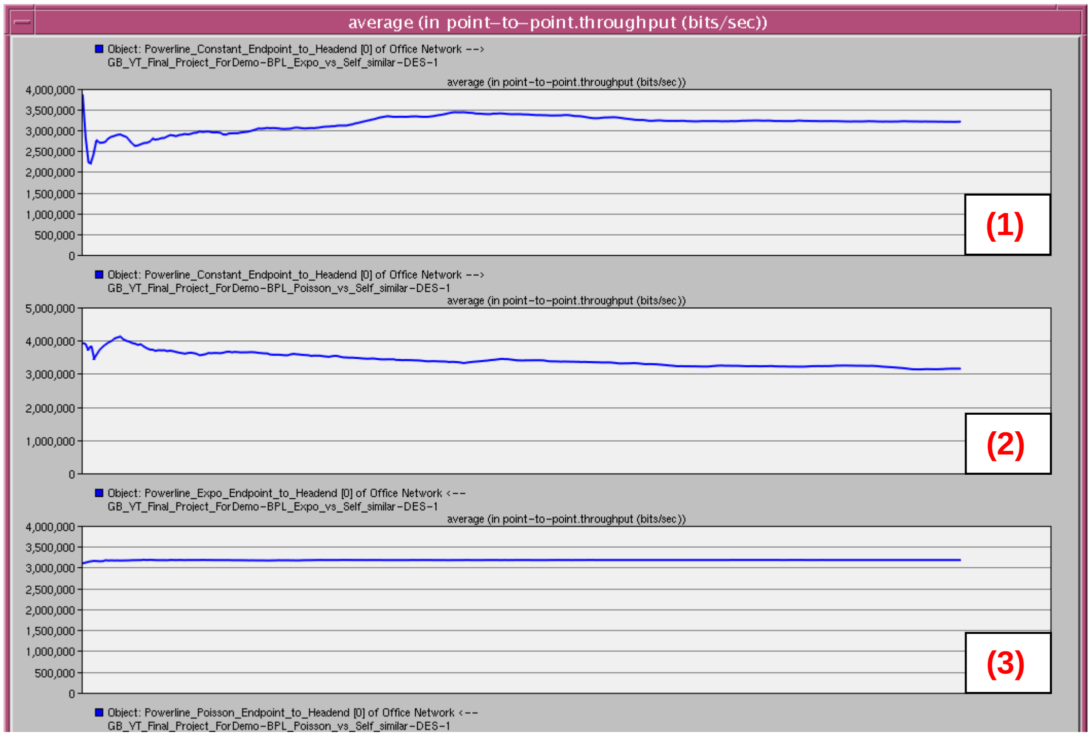


Figure 20: Simulation Results – Throughput

Figure 20 shows the data throughput simulation results for both scenarios. The graphs display the average data throughput in bits/second throughout the course of 25 minutes of simulated time. The graph labelled **(1)** shows the bandwidth in the PLC link connecting the BPL head-end unit to a BPL end-point unit. This link is transmitting self-similar traffic. This graph shows significant variation in throughput (burstiness) present in this traffic model.

The graph labelled **(2)** also shows the bandwidth in the PLC link connecting the BPL head-end unit to a BPL end-point unit, however this link is transmitting Poisson traffic. This graph shows moderate variation in throughput (burstiness) present in this traffic model.

The graph labelled **(3)** also shows the bandwidth in the PLC link connecting the BPL head-end unit to a BPL end-point unit, but this link is transmitting exponential traffic. This graph shows virtually no variation in throughput (burstiness) present in this traffic model.

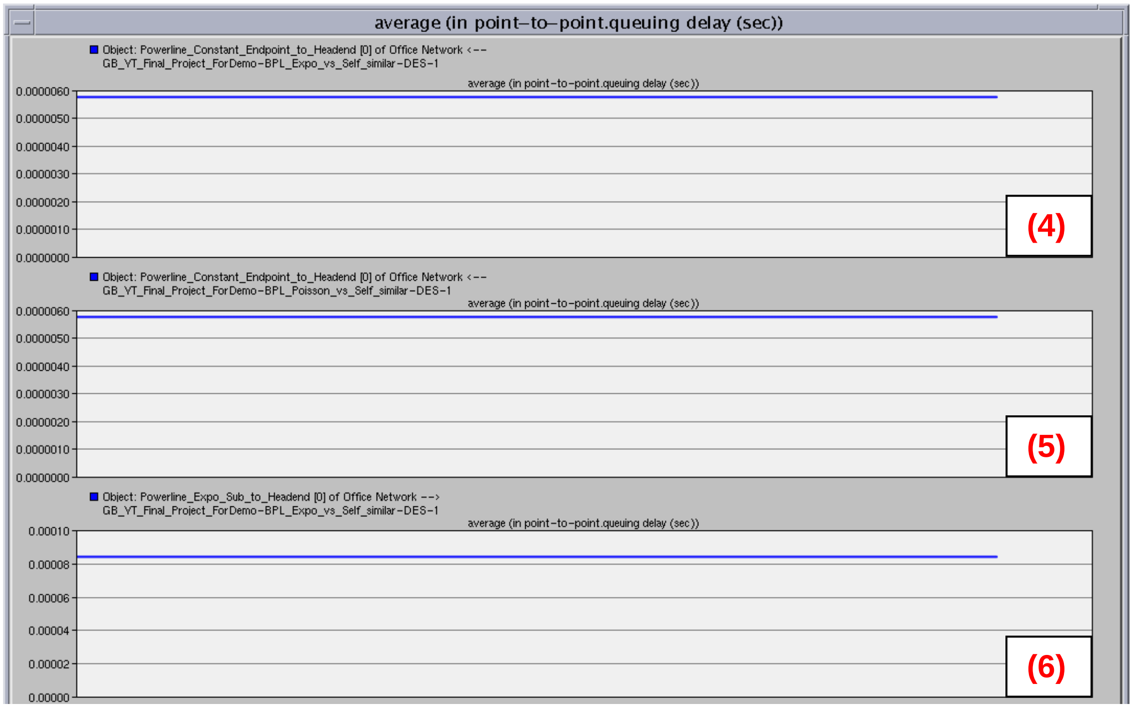


Figure 21: Simulation Results – Point-to-point Delay

Figure 21 shows point-to-point queuing delay simulation results for both scenarios. The graphs display the average point-to-point queuing delay throughput in seconds throughout the course of 25 minutes of simulated time. The graph labelled **(4)** shows the queuing delay for packets travelling from the BPL head-end unit to a BPL end-point unit.

This link is transmitting self-similar traffic and experiences an average point-to-point delay of *0.0000058 seconds.*

The graph labelled **(5)** shows the queuing delay for packets travelling from the BPL head-end unit to a BPL end-point unit, but this link is transmitting Poisson traffic and also experiences an average point-to-point delay of *0.0000058 seconds.*

The graph labelled **(6)** shows the queuing delay for packets travelling from the BPL head-end unit to a BPL end-point unit. This link is transmitting exponential traffic and experiences an average point-to-point delay of *0.000082 seconds.*

From the results obtained from simulation, it is apparent that traffic modelled following an exponential interarrival time experiences the least amount of burstiness in the PLC medium. Self-similar traffic and Poisson traffic undergo much more burstiness through the PLC link, with Poisson traffic showing slightly less burstiness than self-similar traffic.

Although exponential traffic shows the most promise regarding burstiness in the PLC link, the point-to-point delay is over an order of magnitude greater than the self-similar or Poisson modelled traffic. Whether this difference in delays is significant in a larger PLC network remains to be seen and could be investigated in future work using more robust BPL over PLC network simulations.

5 REFERENCES

[1]Ahmed, A., Asadullah, S., Palet, J., Popoviciu, C., Savola, P. “ISP IPv6 Deployment Scenarios in Broadband Access Networks.” Internet: www.ietf.org/rfc/rfc4779.txt, January 2007 [February 19, 2011].

[2]Balls, C., Battaglini, A., Haas, A., Lilliestam, J. “The SuperSmart Grid.” Internet: http://www.supersmartgrid.net/wp-content/uploads/2008/06/battaglini-lilliestam-2008-supersmart-grid-tallberg1.pdf, June 18, 2008 [February 19, 2011].

[3]Du Bois, D. “Broadband Over Powerlines (BPL) in a Nutshell.” Internet: http://energypriorities.com/entries/2004/12/broadband_over_1.php, December 09, 2004 [February 19, 2011].

[4]Iglesias, G., J. “Modelling and analyzing different MAC layers over PLC.” Internet: http://www.salle.url.edu/~zaballos/opnet/PLC.pdf, March 2008 [March 9, 2011].

[5]Motorola, Inc. “Motorola Canopy Broadband over Powerline-BPL MU (Multi-Unit) Solutions.” Internet: http://www.netlinkweb.com/solutions/motorola/BPL-applications/BPL-MU.asp, March 2010 [February 19, 2011].

[6]Motorola, Inc. “Practical, Proven Broadband over Power Line: The Powerline LV Solution.” Internet: http://www.motorola.com/governmentandenterprise/contentdir/he_il/files/productinformation/mibas%20brochure%20bpl%20powerline_lv_whitepaper.pdf, February 2006 [February 19, 2011].

APPENDIX A: OPNET 14.0 CONFIGURATION FILE

```
#number of logical channels
log_channels: 1
#tells what type of protocol exists in each logical channel
log_channel_0 : BPL_PLC
#log_channel_1 : BPL_alt
#log_channel_2 : BPL_alt
#tells the number of slave nodes
slave_nodes : 2
#initial repeater level
repeat_downlink : 1
repeat_uplink : 1
#number of retries
number_retries : 2
#timeslot for start of collecting statistics
start_timeslot : 0
```