

**UNIVERSITY OF SOUTH WALES**  
**Assessment Cover Sheet and Feedback Form**  
**2017/18**

<b>Module Code:</b> NG4S800	<b>Module Title:</b> Satellite Networking	<b>Lecturer:</b> Prof I Otung
<b>Assignment No: 1</b>	<b>No. of pages in total including this page: 15</b>	<b>Maximum Word Count:</b>

**Assignment Title:** Selected Fundamental Theories in Satellite Networking

**Tasks:**  
Evaluation of various satellite network topologies.

Design of a telephone exchange with analysis of impact of traffic level and grade of service.

Design and analysis of digital hierarchies and TCP throughput on satellite links.

**Section A: Record of Submission**

**Record of Submission and Plagiarism Declaration**

I declare that this assignment is my own work and that the sources of information and material I have used (including the internet) have been fully identified and properly acknowledged as required in the referencing guidelines provided.

**Student Number: 14031329**

You are required to acknowledge that you have read the above statement by writing your student number(s) above.

(If this is a group assignment, please provide the student numbers of **ALL** group members)

**Details of Submission**

Note that all work handed in after the submission date and within 5 working days will be capped at 40%. No marks will be awarded if the assignment is submitted after the late submission date unless mitigating circumstances are applied for and accepted.

- IT IS YOUR RESPONSIBILITY TO KEEP A RECORD OF ALL WORK SUBMITTED.
- An electronic copy of your work should be submitted via Blackboard.
- Work should also be submitted to the member of academic staff responsible for setting your work.
- Work not submitted to the lecturer responsible may, **exceptionally**, be submitted (on the submission date) to the reception of the Faculty of Advanced Technology, which is on the 2nd floor of G block (Room G221) where a receipt will be issued.

**Mitigating Circumstances:** if there are any exceptional circumstances which may have affected your ability to undertake or submit this assignment, make sure you contact the Faculty Advice Shop on 01443 482540 (G221).

## Section B: Marking and Assessment

This assignment will be marked out of **100%**  
This assignment contributes to 40% of the total module marks.

This assignment is bonded / non- bonded. Details :

It is estimated that you should spend approximately (at least)

20 hours

**Date Set:** 29/11/2017

**Submission Date:** 08/01/2018

**Feedback Date:**

## Learning Outcomes

**This assignment addresses the following learning outcome(s) of the module:**

All learning outcomes

<b>Deliverable</b>	<b>Mark (Awarded)</b>	<b>Mark (Available)</b>
1.		<b>36</b>
2.		<b>34</b>
3.		<b>30</b>

## Section C: Marker's Feedback

**Lecturer's Comments:**

**Areas to concentrate on next time:**

Report structure

Research

Content

Team work

Referencing

Presentation

**Lecturer's signature:**

**Date:**

**Mark awarded:**

**All marks are subject to confirmation by the Board of Examiners**

## Task 1

Satellite networks are composed of three main elements: a user terminal (formally denominated as user segment or UES - User Earth Station), such as handheld terminals, portable radios and VSATS, the satellite itself (space segment), which may act either as a signal repeater/hub or as a node<sup>2</sup> that processes and redirects incoming packets and a base station, also known as ground segment or GES - Gateway Earth Station, made of large dishes that amplify the user terminals' signal from an outbound uplink and consequently inbound downlink.

Given the fact that satellite networks only support two transmission methods (point to point and broadcast)<sup>3</sup> between UES - satellite, satellite - satellite (ISL - Inter-Satellite Link) and satellite - GES, their only possible configurations must therefore be SHSS - Single Hop-Single Satellite, DHSS - Double Hop-Single Satellite, MHMS - Multiple Hop-Multiple Satellite and SHMS - Single Hop-Multiple Satellite. For each configuration, a different set of trade-offs must be considered, depending on the intended application.

In remote areas where there is lack of ground communication with the main network, for instance, in a small island or in a rural area, an SHSS configuration would be adopted. In this star topology, all user terminals connect to each other via a single satellite, however, these must use a large antenna in order to transmit to the destination terminal. As a consequence, the power requirements for such devices become much higher than the average UES. This is because the satellite is designed to consume as little power as possible, since all of the energy comes from its solar panels, therefore, all users must deal with the power "deficiency" problem by utilising large antennas. Besides, this topology is designed so that the cost is shifted towards the customer rather than the satellite itself or the GES, which means in the SHSS configuration the space segment is relatively low cost, whereas user segment is of higher cost (compared to DHSS).

Despite these issues, the advantage comes from the fact that each packet transmission is done in a single hop, directly from the source terminal > satellite > destination terminal. This means the delay is only one RTT (Round Trip Time). Moreover, implementation should be relatively simple, since there is no interconnection between satellites, no ISL and no handover. The satellite merely acts as a "mirror" that retransmits the signal to its destination. Also, this topology should be immune to ground segment downtimes or security breaches, since it is composed of only user terminals and a single satellite. In the end, it all comes down to the quality of the link (which unfortunately varies during the day) and the size of the user terminals' antennas.

On the other hand, when the cost and size of the terminals become too "impractical", a DHSS configuration should be considered instead. An example of this is standard TV broadcast such as DVB-S or in the case where the destination terminal is too far away for any quality link to take place<sup>4</sup>. This means that DHSS covers a much wider area than the previous configuration. Note that the implementation complexity and cost of the satellite in terms of power consumption have not changed as compared to SHSS. In this topology, all traffic is first redirected to a GES, where it is amplified and retransmitted back through the same satellite and consequently to the destination UES. This means the RTT has doubled, or, in other words, the delay has increased twice as high compared to the SHSS topology. In this case, latency has been sacrificed for a very beneficial trade-off. Now, users can use a very small antenna to connect with other terminals for both RX and TX communications. All of the TX power is now being amplified by the GES. One unfortunate trade-off resultant from the introduction of a single earth segment is the link vulnerability. Since all terminals are connected via a single base station in this topology, if this link goes down, then all of the network is also considered to be down.

In the case where the user needs to transmit data over much larger distances, say, from one country to another, a MHMS topology should be used. With this design, all terminals are connected to a complex and transparent satellite-GES network, where an indeterminate amount of hops are taken from source to destination. This interconnection is what is called a transit network. The first disadvantage is the delay. Round trip times are too difficult to predict as a result of the complex nature of the network. Secondly,

the earth-segment cost is much higher than the DHSS topology, due to the fact that there is more than one GES being used for the transmission. Moreover, the cost of space-segment is equally higher for the same reason. The advantages, however, come in the form of much wider coverage than the DHSS design, the space segment complexity is also low (since there is no ISL) and the size of the user terminals do not have to deal with any extra TX power. As can be seen, this topology's disadvantages outweigh the advantages (not quantitatively), which leads to the conclusion that this network is best fit for transient communications from one large gateway to another, or in other words, from one country/private organisation to another.

Lastly, in critical situations where there are no UES or GES nearby, for instance in the middle of the pacific ocean, an SHMS topology should definitely be adopted. This configuration implements ISL communications between many satellites with global coverage. This means the complexity is much higher due to handover and extra processing required. Also, because there is more than one satellite in the topology, the delay will always be more than one RTT. As a consequence from all this, the cost of space-segment becomes exponentially higher. To make it worse, the UES antennas must be capable of handling transmission in order to compensate for the inevitable power loss. This only leaves the advantages of having global coverage, no earth-segment vulnerabilities and low cost.

## Task 2

(i) From the Erlang B formula,  $P_B \equiv E_{1,N}(\rho) = \frac{10^{\left(N \log \rho - \sum_{r=1}^N \log_{10} r\right)}}{1 + \sum_{n=1}^N 10^{\left(n \log \rho - \sum_{r=1}^n \log_{10} r\right)}}$

the following MATLAB code was derived:

```

1 function PB = ErlangB(N, rho)
2     r = (1:N)';
3     x = 10.^(N * log10(rho) - sum(log10(r))); % First, calculate numerator separately %
4
5     y = 1; % And the denominator as well %
6     for n = 1:N, r = (1:n)';
7         y = y + 10.^(n * log10(rho) - sum(log10(r)));
8     end
9
10    PB = x / y; % Finally, divide the two to get the intended result %
11 end % -- END FUNCTION - ErlangB() --

```

Code 1: MATLAB function for calculating the call blocking probability on a telephone exchange system - ErlangB.m

For this exercise, a function named *plot\_N\_vs\_PB()* was created to draw a set of different sub-plots. Each sub-plot is associated with an Erlang value and a trunk vector, that ranges from 1 to a maximum (arbitrary) value. The function also returns the desired *N* value, as well as the vector *P<sub>B</sub>* with all of the blocking probabilities. Below is the main code to calculate the blocking probabilities of a telephone exchange system with 120 Erlangs and a goal of 10% grade of service:

```

1 erlangs = 120; % How many calls/hour on the telephone system %
2 grade = 10.0; % The grade of service desired for this system %
3
4 % Calculate and plot the blocking probability for every N and rho values %
5 [N, PB] = plot_N_vs_PB(...
6     1, ... % Create figure 1 %
7     250, ... % N will be plotted from 1 to 250 on the x-axis %
8     erlangs, ... % The desired telephone exchange load (Erlangs) %
9     40, ... % The erlang increment used to plot different loads %
10    grade ... % The desired grade of service / blocking probability %
11 );
12
13 % Show results %
14 fprintf('** Answer for task 2.i: N = %d, Rho = %d, PB = %.4f %*\n', N, erlangs, PB(N));

```

Code 2: task2.i.m

And the implementation for *plot\_N\_vs\_PB()*:

```

1 function [N_output, PB_output] = plot_N_vs_PB(figure_index, N_max, desired_rho, increment_rho_by, desired_grade)
2
3 fprintf('** Calculating N value for Erlangs = %d and Grade of Service = %.2f %% ... \n', desired_rho, desired_grade);
4
5 fig = figure(figure_index);
6 clf(fig);
7
8 PB = zeros(1, N_max);
9 N_vector = 1:N_max;
10
11 plot_index = 1;
12 legend_x_offset = 0;
13
14 if figure_index == 2
15     max_rho = desired_rho;
16     legend_x_offset = 8;
17 else
18     max_rho = desired_rho * 2 - increment_rho_by;
19 end
20
21 for rho = increment_rho_by:increment_rho_by:max_rho
22     % Calculate the Blocking Probability from values N = 1 up to N_max
23     % for the current Rho value
24     for N = N_vector
25         PB(N) = ErlangB(N, rho) * 100;
26     end
27
28     % Plot PB (Block Probability) in percentages (0 -> 100%)
29     % in a semi-logarithmic scale
30     graph_handle = semilogy(1:N_max, PB, 'LineWidth', 2);
31
32     % Set the title for all subplots and the labels for the xy axis (only once)
33     if plot_index == 1
34         title('N (# of trunks) vs PB (Blocking Probability (%))');
35         xlabel('N (# of trunks)');
36         ylabel('PB (Blocking Probability (%))');
37
38         grid on;
39         hold on;
40
41         xlim([1 N_max]);
42         ylim([10^-2 100]);
43     end
44
45     % Place the rho value on the current subplot at position y = 10^-1
46     N = find(PB <= 10^-1);
47     line_legend_text = text(N(1) + 4 + legend_x_offset, 10^-1, ['\rho = ' num2str(rho)], 'Rotation', -80 - legend_x_offset / 1.2 + plot_index * 1.5, 'FontSize', 12);
48
49     % Process the rho value in question in a special way
50     if rho == desired_rho
51         line_legend_text.FontSize = 14;
52         line_legend_text.FontWeight = 'bold';
53         set(graph_handle, 'LineWidth', 4, 'color', [1 0.85 0]);
54
55         % Look for a percentage value that fits the
56         % desired grade of service
57         N_output = 0;
58         PB_output = 0;
59         N_ = find(PB <= desired_grade);
60
61         if ~isempty(N_)
62             N = N_(1);
63             % Set the output values of this function
64             N_output = N;
65             PB_output = PB;
66
67             % Draw an arrow pointing to the result (no. of trunks)
68             text(N - 2.5 - legend_x_offset / 1.85, PB(N) * 1.5, '\downarrow', 'FontSize', 25, 'FontWeight', 'bold', 'Color', [1 0 0]);
69             text(N - 5, PB(N) * 2.25, ['N = ' num2str(N)], 'FontWeight', 'bold', 'FontSize', 12, 'Color', [0 0 1]);
70
71             % A reference line as well
72             ref_y = PB(N) * ones(N_max);
73             plot(1:N_max, ref_y, 'color', [0 0 1]);
74             text(N_max + 2, PB(N), ['PB = ' num2str(round(PB(N), 2)) ' %'], 'FontWeight', 'bold');
75
76             % Finally, draw an interception point
77             plot(N, PB(N), 'rx', 'MarkerSize', 8, 'LineWidth', 1.5, 'Color', [0 0 0]);
78         end
79     end
80     plot_index = plot_index + 1;
81 end
82
83 set(gca, 'XTick', sort([1, get(gca, 'XTick')])); % Add XTick at x=1 %
84 set(gca, 'YTickLabel', cellstr(['10^{-2}'; '10^{-1}'; '1'; '10'; '100']));
85 refreshdata;
86
87 end % -- END FUNCTION - plot_N_vs_PB() --

```

Code 3: plot\_N\_vs\_PB.m

This produces the graphical output:

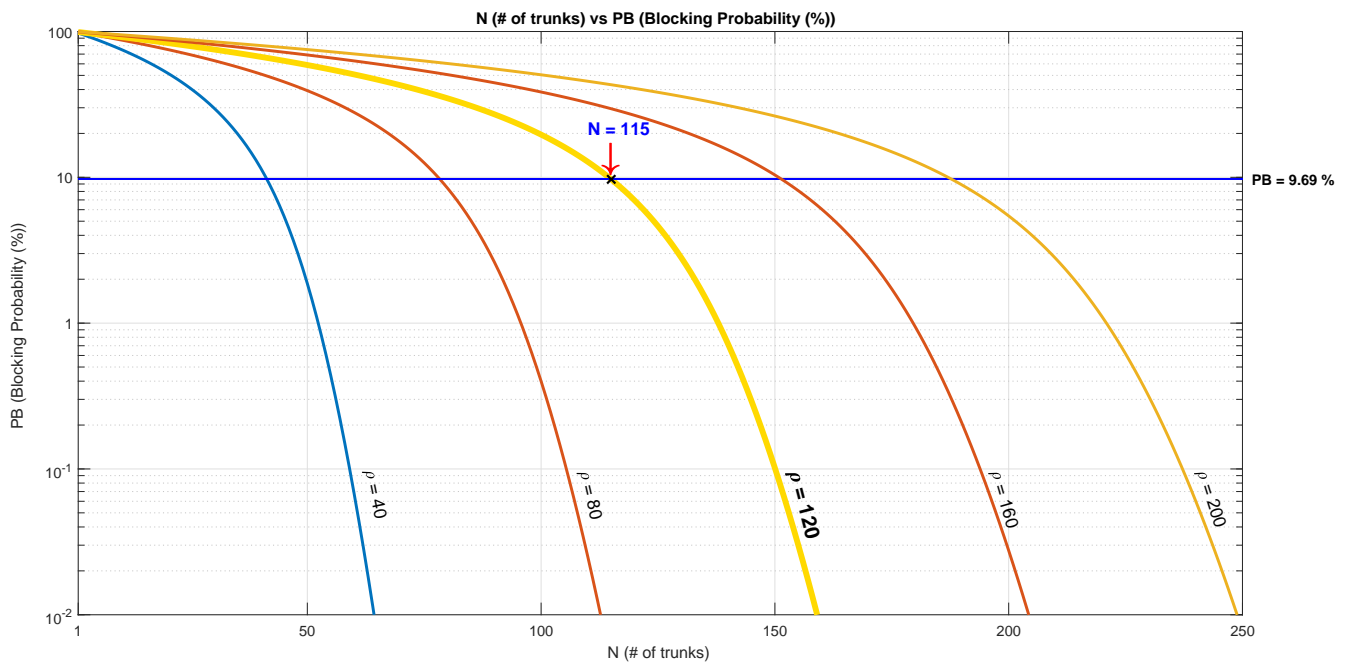


Figure 1: Finding # of trunks ( $N$ ) for telephone exchange with conditions: Erlangs ( $\rho$ ) = 120 and Grade of service (PB) = 10%

Concluding, the answer for task 2.i is:  $N = 115$

(ii) a) Changing the *erlangs* parameter in the MATLAB code to 600 shows the result:

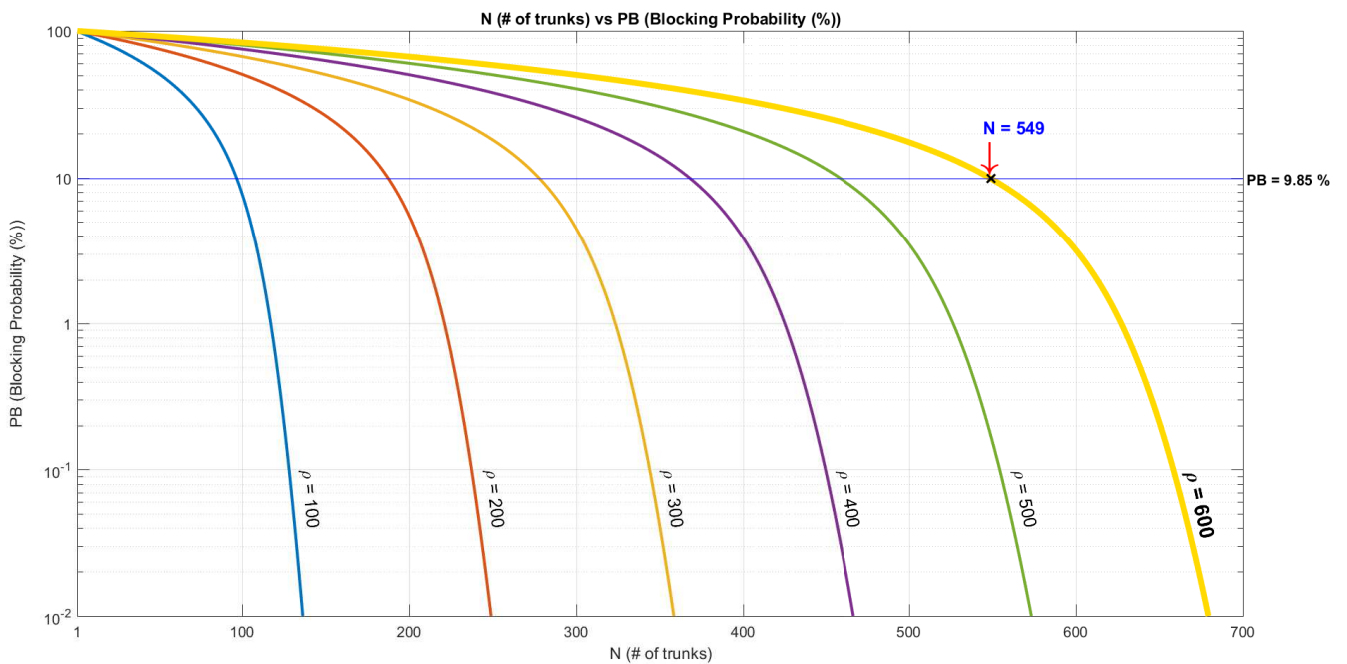


Figure 2: Finding # of trunks ( $N$ ) for telephone exchange with conditions: Erlangs ( $\rho$ ) = 600 and Grade of service (PB) = 10%

→ Answer for task 2.ii.a:  $N = 549$

(ii) b) Finally, after resetting the *erlangs* parameter to 120 and changing the *desired\_grade\_of\_service* value to 0.1% the result becomes:

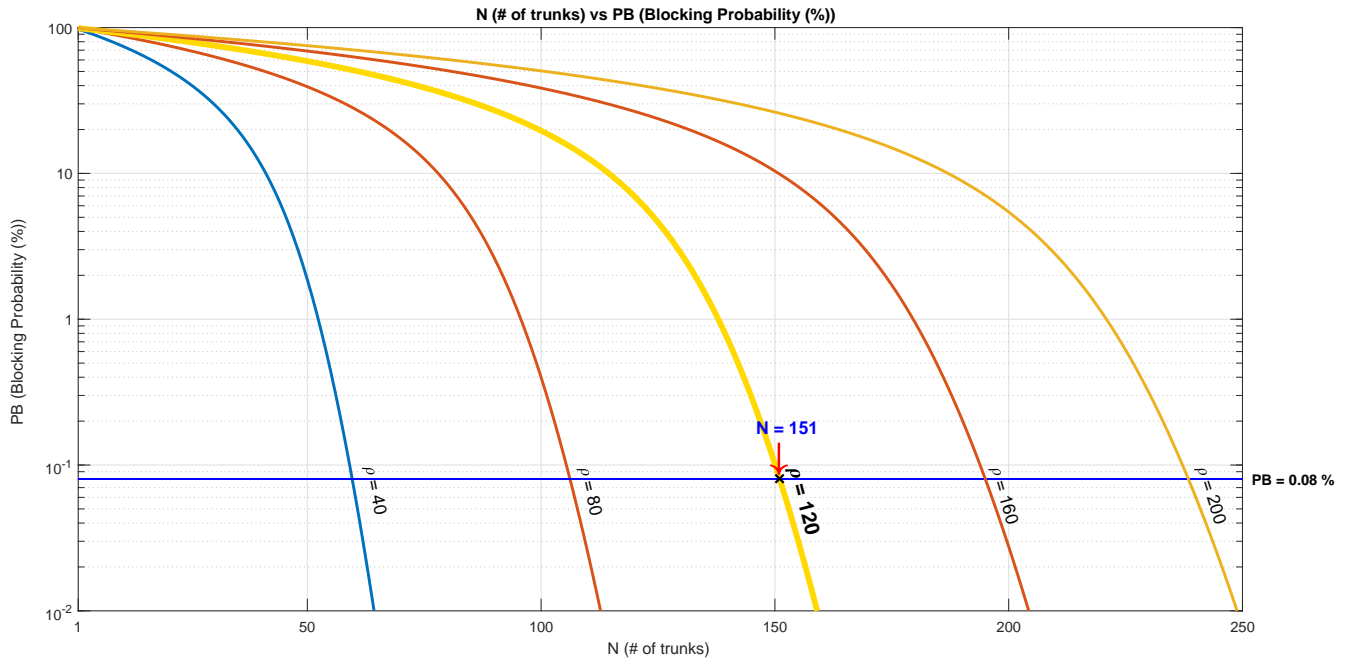


Figure 3: Finding # of trunks (N) for telephone exchange with conditions: Erlangs ( $\rho$ ) = 120 and Grade of service (PB) = 0.1%

→ Answer for task 2.ii.b: **N = 151**

(iii) In task 2.ii.a the *erlang* argument was increased by a factor of 5, which equals 600 calls per hour. The observed result for  $N$  with this load was 549, therefore, the increase ratio is  $\frac{N_2}{N_1} = \frac{549}{115} = 4.8$ . In other words, for a telephone exchange to keep the blocking probability at 10% with 600 calls per hour, the number of trunks need to be increased 4.8 times.

If, however, the grade of service is improved by a factor of 10 (as was done in task.ii.b), the number of trunks required to withstand the same load of 120 calls per hour is  $N = 151$ , which is an increase of only  $\frac{N_3}{N_1} = \frac{151}{115} = 1.3$ . All in all, the results indicate that the load in the telephone exchange produces a much larger impact on the value  $N$  than the grade of service.

Despite this conclusion, the decision on whether one parameter is more beneficial/profitable to the telephone company than the other depends on the cost of each of them. Clearly, increasing the number of trunks reduces the blocking probability dramatically, however, adding more trunks might be more costly than to change the grade of service to a higher but acceptable value (higher  $P_B \equiv$  more blocking  $\equiv$  less trunks  $\equiv$  lower cost). This is a trade-off that must be carefully considered when designing a telephone exchange system.

### Task 3

a) Based on the ITU-T specification, an STM-1 signal may be constructed from an E3 signal by first adding a POH (Path Overhead) field next to the STM-1 payload in the 10th frame column, which places the C3 signal into a virtual container in order to identify and manage the transmission from source to destination using the byte Path Trace and BIP-8 code to check for TX/RX errors. In simpler terms, the POH converts the C3 signal into a VC-3. This step is called **mapping**.

Afterwards, a pointer is inserted in the STM-1 header between the RSOH (Regenerator Section Overhead) and MSOH (Multiplex Section Overhead) fields on the 4th header row so that the C3 data can be accessed within the payload field using the correct offset. The output of this stage produces the AU-3 (Administrative Unit) signal. Also, this step is called **pointer processing** or **alignment**.

Next, three AU-3 signals from three different sources are multiplexed with byte-interleaving into one single output signal named AUG (Administrative Unit Group).

Finally, the STM-1 frame is completed after adding the SOH (Section Overhead) fields, composed of RSOH and MSOH fields. These are intended to be used in between regenerators and multiplexing points (respectively). The RSOH contains information about frame alignment, a channel for inter-regenerator communication (through byte E1) and error checking, such as RS trace message and BIP-8 code. Similarly, MSOH contains bits for error checking with BIP-24 and another E2 field for communication between two multiplexing points.

Following these steps should result in the diagram:

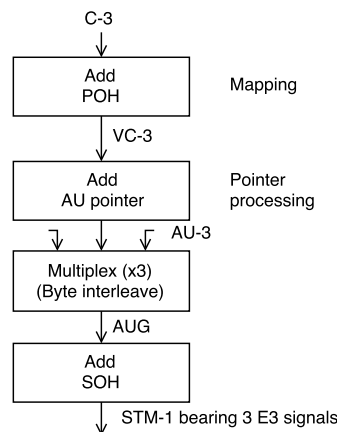


Figure 4: Building an STM-1 frame from three E3 signals

The efficiency is:

→ If the STM-1 signal contains 3 E3 signals;

→ And each E3 signal contains  $4 \times E2 = 4^2 \times E1 = 4^2 \times 30 \times E0 = 480$  voice channels;

→ Each of which has a bit rate of 64 kb/s, thus:

$$Bitrate_{Payload} = 3 \times E3 = 3 \times 480 \times 64 \text{ kb/s} = 92160 \text{ kb/s}$$

→ Therefore, the efficiency must be equal to:

$$\text{Efficiency}_{\text{STM-1}} = \frac{Bitrate_{Payload}}{Bitrate_{\text{STM-1}}} \times 100\% = \frac{92160 \text{ kb/s}}{155520 \text{ kb/s}} \times 100\% = \mathbf{59.26\%}$$



→ As opposed to the E3 efficiency:

$$\text{Efficiency}_{E3} = \frac{480 \text{ voice channels} \times 64 \text{ kb/s}}{34368 \text{ kb/s}} \times 100\% = \mathbf{89.39\%}$$

As can be seen, the STM-1 efficiency is quite low when compared to the PDH E3. This is because a single E3 is not capable of filling up the entire payload space on the STM-1 frame, therefore, the STM-1 transmission channel will carry a lot of overhead data just for three multiplexed E3 signals.

**b)** SDH allows the transmission of a PDH signal within its payload by allocating extra space in this same field and by allowing the PDH data to vary its offset position with respect to the payload's starting point on column 10 of the STM-1 frame structure. (Note that when it is said "allocating extra space", it is meant "a space that is larger than a single PDH frame". The STM-1 payload field is fixed at  $261 \text{ columns} \times 9 \text{ rows} = 2349$  bytes of size and does not grow in respect to the PDH signal being conveyed. If one STM-1 frame cannot fit enough PDH signals, then an STM-N of  $N > 1$  should be used instead.)

This way, the bit rate variance property that characterises the nature of the PDH signal is conserved and the SDH signal, being a synchronous transmission, will still be able to carry a plesiosynchronous signal. This is also why there is a pointer offset in the STM-1 header that varies its value depending on the payload's bit rate. It is used to provide access to the virtual container at the right offset within the payload field.

**c)** TCP was designed to handle transmission of IP packets between two hosts through the coaxial or fiber medium. This protocol handles congestion in these physical mediums quite well as opposed to the wireless medium, where it performs considerably worse. The reason is the fact that in these first two mediums the relative distances between two hosts is (most of the time) quite short. In satellite networks however, the distances are considerably larger, with LEO orbit altitudes reaching up to 5000 km, MEO 5000 - 20000 km and HEO above 20000 km.<sup>5</sup> The **absolute minimum** time it takes for radio waves to travel to the LEO orbit is  $\frac{5000 \text{ km}}{c} = \frac{5 \cdot 10^6}{3 \cdot 10^8} = 1.7 \cdot 10^{-2} \text{ s} = 17 \text{ ms}$ , whereas in the HEO orbit the travel time is  $\frac{20000 \text{ km}}{c} = \frac{20 \cdot 10^6}{3 \cdot 10^8} = 6.7 \cdot 10^{-2} \text{ s} = 67 \text{ ms}$ . This is only considering the speed of light with the best case scenario where there are no extra delays.

In other words, TCP works well on copper and fiber because the RTT (Round Trip Time) is in the order of milliseconds and timeout issues are much less of a problem, whereas in the wireless medium, such as the communication between a base station and a geostationary satellite, the RTT is considerably larger to the point where it takes more than one second for a packet to take a single hop (with all the delays being accounted for). The main issue is not necessarily that the packet or segment takes a very long time to reach its destination. (Since this issue is unavoidable, unless the satellite is moved to a lower orbit from HEO/GEO to MEO/LEO.) The true problem lies in the fact that TCP expects the segment to reach its destination within a negotiated amount of time. If the transmission takes too long, a timeout event will occur and the segment 'growth rate' (SSThresh) will begin to lower. If the timeout keeps occurring, the client might stop receiving any packet at all and it will stay in the initial mode Slow Start indefinitely until the RTT is lowered or the TCP header is optimized for the specific link. Another issue found in satellite networks is the inability to utilise the whole bandwidth provided by the link. Essentially, TCP defines a parameter in its segment header that limits the maximum data size that is received by the client, called the window parameter. This value depends on many things, mainly on what the client is capable or willing to receive.

Taking this into account, in order to calculate the link utilisation, a critical parameter must first be calculated: the BDP - Bandwidth-delay Product.

$$BDP = RTT \times Bandwidth$$

This represents the number of data bits that are “currently” travelling in a medium which have not yet been acknowledged. So, if the window size only allows  $2^{16} \text{ bits} - 1 = 64 \text{ kB}$  to be sent (this size is defined by the TCP protocol in the header structure) and the client is requesting a total amount of data of 128 kB, then the entire “transaction” should theoretically take  $\frac{128 \text{ kB}}{64 \text{ kB}} = 2 \text{ RTTs}$  to complete. This will negatively impact the congestion-control algorithm, especially if the RTT is larger than 500 ms. Moreover, the link utilisation will be (assuming  $WindowSize = 2^{16} - 1 = 65535 = 64 \text{ kB}$ ,  $RTT = 500 \text{ ms}$  and  $Bandwidth = 3 \text{ Mbit/s}$ ):

$$\text{Link Utilisation (\%)} = \frac{WindowSize}{BDP} \times 100\% = \frac{65535 \times 8}{0.5 \times 3 \cdot 10^6} \times 100\% = \mathbf{35\%}$$

Another way of getting the utilisation is by calculating the maximum ideal throughput:

$$\text{Throughput} = \frac{WindowSize}{RTT} = \frac{65535 \times 8}{0.5} = \mathbf{1048560 \text{ b/s}}$$

Which, when divided with the total bandwidth gives:

$$\text{Link utilisation (\%)} = \frac{\text{Throughput}}{\text{Bandwidth}} \times 100\% = \frac{1048560}{3 \cdot 10^6} \times 100\% = \mathbf{35\%}$$

So, the first solution to improve the congestion-control algorithm, link utilisation and throughput is to simply increase the window size. This can be done in the TCP header in the ‘TCP Options’ field by setting the first byte to a fixed value 3 (which indicates the operation ‘set window scale value’), the second byte to value 3 as well (the length of the entire option in bytes) and the third byte to an arbitrary value that ranges from 0 to 14. This third parameter controls the size of the window, scaling it with the bitwise left shift operation. Hence, the new window size is:

$$WindowSize = TCPHeader_{WindowField} \ll TCPHeader_{OptionsField}[2]$$

Where  $TCPHeader_{OptionsField}[2]$  is the 3rd byte with the scaling value that is in the range  $[0, 14]$ . This means that if  $TCPHeader_{WindowField}$  is at its maximum value ( $2^{16} \text{ bits} - 1 = 65535$ ) and the option field is set to 14, the maximum obtainable window size that can be used in any network is  $65535 \ll 14 = 1073725440 \text{ bytes}$ . Coming back to the previous example, a window of this size would allow the client to utilise 100% of its available bandwidth since link utilisation would be  $\frac{1073725440 \times 8}{0.5 \times 3 \cdot 10^6} > 100\%$  and the entire transmission would be concluded within 1 RTT (ideally, assuming that the transmission algorithm is already in the congestion avoidance mode and the segment size has reached the window size).

There are other solutions implemented for satellite networks to further improve the algorithm and the utilisation percentage<sup>6</sup>. Some of the techniques are:

- 1- Increase the initial SStHres value which controls the point at which the TCP protocol switches from Slow Start mode to Congestion Avoidance mode, for the same reasons presented above;
- 2- Also increase the minimum size ( $T_b$ ) of the granular traffic segment block that comprises an entire segment;
- 3- Use fast retransmission and fast recovery in order to allow the host to reach the Congestion Avoidance mode at a higher rate after a loss/timeout;
- 4- Compress the TCP header structure to reduce the entire segment size<sup>1</sup>;
- 5- Detect when a corruption loss occurs in a transmission;
- 6- Connect to multiple satellites instead of only one in order to “multiplex” the RX data stream<sup>1</sup>.

# References

- [1] Gerard Maral, Michel Bousquet, and Zhili Sun. *Satellite Communications Systems: Systems, Techniques and Technology*. Wiley-Blackwell, 2009, p. 355.
- [2] Zhili Sun. *Satellite Networking: Principles and Protocols*. Ed. by John Wiley & Sons. Wiley-Blackwell, 2005, p. 86.
- [3] Zhili Sun. *Satellite Networking: Principles and Protocols*. Ed. by John Wiley & Sons. Wiley-Blackwell, 2005, p. 28.
- [4] Zhili Sun. *Satellite Networking: Principles and Protocols*. Ed. by John Wiley & Sons. Wiley-Blackwell, 2005, p. 87.
- [5] Zhili Sun. *Satellite Networking: Principles and Protocols*. Ed. by John Wiley & Sons. Wiley-Blackwell, 2005, p. 30.
- [6] Zhili Sun. *Satellite Networking: Principles and Protocols*. Ed. by John Wiley & Sons. Wiley-Blackwell, 2005, p. 269.

# Appendices

# Appendix A

## MATLAB Code

### A.1 ErlangB.m

---

```
1 function PB = ErlangB(N, rho)
2     r = (1:N)';
3     x = 10.^(N * log10(rho) - sum(log10(r))); % First, calculate numerator separately %
4
5     y = 1; % And the denominator as well %
6     for n = 1:N, r = (1:n)';
7         y = y + 10.^(n * log10(rho) - sum(log10(r)));
8     end
9
10    PB = x / y; % Finally, divide the two to get the intended result %
11 end % -- END FUNCTION - ErlangB() --
```

---

Code A.1: ErlangB.m

### A.2 task2\_i.m

---

```
1 erlangs = 120; % How many calls/hour on the telephone system %
2 grade = 10.0; % The grade of service desired for this system %
3
4 % Calculate and plot the blocking probability for every N and rho values %
5 [N, PB] = plot_N_vs_PB(...
6     1,      ... % Create figure 1 %
7     250,    ... % N will be plotted from 1 to 250 on the x-axis %
8     erlangs, ... % The desired telephone exchange load (Erlangs) %
9     40,     ... % The erlang increment used to plot different loads %
10    grade    ... % The desired grade of service / blocking probability %
11 );
12
13 % Show results %
14 fprintf('** Answer for task 2.i: N = %d, Rho = %d, PB = %.4f %% **\n', N, erlangs, PB(N));
```

---

Code A.2: task2\_i.m

### A.3 task2\_ii\_a.m

---

```

1  erlangs = 600; % How many calls/hour on the telephone system %
2  grade   = 10.0; % The grade of service desired for this system %
3
4  % Calculate and plot the blocking probability for every N and rho values %
5  [N, PB] = plot_N_vs_PB(...
6      2,      ... % Create figure 2                                %
7      700,    ... % N will be plotted from 1 to 700 on the x-axis    %
8      erlangs, ... % The desired telephone exchange load (Erlangs)    %
9      100,    ... % The erlang increment used to plot different loads  %
10     grade   ... % The desired grade of service / blocking probability %
11 );
12
13 % Show results %
14 fprintf('** Answer for task 2.ii.a: N = %d, Rho = %d, PB = %.4f %% **\n', N, erlangs, PB(N));

```

---

Code A.3: task2\_ii\_a.m

### A.4 task2\_ii\_b.m

---

```

1  erlangs = 120; % How many calls/hour on the telephone system %
2  grade   = 0.1; % The grade of service desired for this system %
3
4  % Calculate and plot the blocking probability for every N and rho values %
5  [N, PB] = plot_N_vs_PB(...
6      3,      ... % Create figure 3                                %
7      250,    ... % N will be plotted from 1 to 250 on the x-axis    %
8      erlangs, ... % The desired telephone exchange load (Erlangs)    %
9      40,     ... % The erlang increment used to plot different loads  %
10     grade   ... % The desired grade of service / blocking probability %
11 );
12
13 % Show results %
14 fprintf('** Answer for task 2.ii.b: N = %d, Rho = %d, PB = %.4f %% **\n', N, erlangs, PB(N));

```

---

Code A.4: task2\_ii\_b.m

### A.5 plot\_N\_vs\_PB.m

---

```

1  function [N_output, PB_output] = plot_N_vs_PB(figure_index, N_max, desired_rho, increment_rho_by, desired_grade)
2
3  fprintf('** Calculating N value for Erlangs = %d and Grade of Service = %.2f %% ... \n', desired_rho, desired_grade);
4
5  fig = figure(figure_index);
6  clf(fig);
7
8  PB = zeros(1, N_max);
9  N_vector = 1:N_max;
10
11  plot_index = 1;
12  legend_x_offset = 0;
13
14  if figure_index == 2
15      max_rho = desired_rho;
16      legend_x_offset = 8;
17  else
18      max_rho = desired_rho * 2 - increment_rho_by;
19  end
20
21  for rho = increment_rho_by:increment_rho_by:max_rho
22      % Calculate the Blocking Probability from values N = 1 up to N_max
23      % for the current Rho value
24      for N = N_vector

```

---

```

25         PB(N) = ErlangB(N, rho) * 100;
26     end
27
28     % Plot PB (Block Probability) in percentages (0 -> 100%)
29     % in a semi-logarithmic scale
30     graph_handle = semilogy(1:N_max, PB, 'LineWidth', 2);
31
32     % Set the title for all subplots and the labels for the xy axis (only once)
33     if plot_index == 1
34         title('N (# of trunks) vs PB (Blocking Probability (%))');
35         xlabel('N (# of trunks)');
36         ylabel('PB (Blocking Probability (%))');
37
38         grid on;
39         hold on;
40
41         xlim([1 N_max]);
42         ylim([10^-2 100]);
43     end
44
45     % Place the rho value on the current subplot at position y = 10^-1
46     N = find(PB <= 10^-1);
47     line_legend_text = text(N(1) + 4 + legend_x_offset, 10^-1, ['\rho = ' num2str(rho)], 'Rotation', -80 - legend_x_offset / 1.2 +
        plot_index * 1.5, 'FontSize', 12);
48
49     % Process the rho value in question in a special way
50     if rho == desired_rho
51         line_legend_text.FontSize = 14;
52         line_legend_text.FontWeight = 'bold';
53         set(graph_handle, 'LineWidth', 4, 'color', [1 0.85 0]);
54
55         % Look for a percentage value that fits the
56         % desired grade of service
57         N_output = 0;
58         PB_output = 0;
59         N_ = find(PB <= desired_grade);
60
61         if ~isempty(N_)
62             N = N_(1);
63             % Set the output values of this function
64             N_output = N;
65             PB_output = PB;
66
67             % Draw an arrow pointing to the result (no. of trunks)
68             text(N - 2.5 - legend_x_offset / 1.85, PB(N) * 1.5, '\downarrow', 'FontSize', 25, 'FontWeight', 'bold', 'Color', [1 0 0]);
69             text(N - 5, PB(N) * 2.25, ['N = ' num2str(N)], 'FontWeight', 'bold', 'FontSize', 12, 'Color', [0 0 1]);
70
71             % A reference line as well
72             ref_y = PB(N) * ones(N_max);
73             plot(1:N_max, ref_y, 'color', [0 0 1]);
74             text(N_max + 2, PB(N), ['PB = ' num2str(round(PB(N), 2)) ' %'], 'FontWeight', 'bold');
75
76             % Finally, draw an interception point
77             plot(N, PB(N), 'rx', 'MarkerSize', 8, 'LineWidth', 1.5, 'Color', [0 0 0]);
78         end
79     end
80
81     plot_index = plot_index + 1;
82 end
83
84 set(gca, 'XTick', sort([1, get(gca, 'XTick')])); % Add XTick at x=1 %
85 set(gca, 'YTickLabel', cellstr(['10^{-2}'; '10^{-1}'; '1'; '10'; '100']));
86 refreshdata;
87
88 end % -- END FUNCTION - plot_N_vs_PB() --

```

Code A.5: plot\_N\_vs\_PB.m