University of South Wales
Prifysgol De Cymru

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

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

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

Deliverable	**Mark** *(Awarded)*	**Mark** *(Available)*
1.		**36**
2.		**34**
3.		**30**

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 connected with the following three elements: a user terminal, formally named 'user segment' or User Earth Station (UES), such as handheld terminals, portable radios and VSATS; the satellite itself, which makes up the space segment and may act either as a signal repeater or as a node[2] that processes and routes incoming frames; and a base station, also known as 'ground segment' or Gateway Earth Station (GES), made of large satellite dishes that amplify the UES signal from an outbound uplink and consequently inbound downlink.
Considering the fact that satellite networks support only two transmission methods: point to point and broadcast[3] , between UES $\iff$ satellite, satellite $\iff$ satellite (also known as an Inter-Satellite Link [ISL]) and satellite $\iff$ GES, the only possible network configurations are Single Hop-Single Satellite (SHSS), Double Hop-Single Satellite (DHSS), Multiple Hop-Multiple Satellite (MHMS) and Single Hop-Multiple Satellite (SHMS). For each topology, a different set of trade-offs must be considered based on the desired 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 mesh topology, all user terminals connect to each other via a single satellite. These must use a large high gain antenna in order to transmit to the destination terminal, hence, the power requirements for such devices are much higher than the average UES.
This condition is explained by the fact that satellites are designed to consume as little power as possible due to its limited power source (i.e. solar panels). Besides, the SHSS topology is purposefully built with the cost shifted towards the customer rather than the satellite itself or the GES (not present in this case). This means the space and earth segments are relative low cost, whereas user segment is of higher cost. Despite this, the benefit comes in the form of short transmission delays, simplicity and security. Specifically, in this topology a frame is sent and received in a single hop directly from source UES > satellite > destination UES within one Round Trip Time (RTT). In addition, implementation should be relatively simple as there is no interconnection between satellites (ISL). The satellite will merely act as a 'mirror' in order to retransmit the signal to its destination. Also, it should be immune to ground segment downtimes or security breaches since it is composed of user terminals and a single satellite. In the end, it all comes down to the quality of the link (which may unfortunately vary during the day) and the size of the user terminals' antennas.

On the other hand, when the cost and size of the UES become too impractical, a DHSS configuration should instead be considered. An example use-case scenario for this topology is standard TV broadcast (DVB-S) or in the situation where the destination terminal is too far to reach and the signal needs to be boosted by a GES.[4] Considering this, DHSS covers a much wider area than the previous topology, i.e. multiple cities or an entire small country.
In this configuration, all traffic is first redirected to a GES, where it is amplified and retransmitted back through the same satellite (or a different one[2]) and consequently to the destination UES. Unfortunately, the RTT has doubled in respect to the SHSS topology; however, this comes with a very beneficial trade-off: users can now use a very small antenna to connect with other terminals for both RX and TX communications since all of the TX power is amplified by the GES. Also, the implementation complexity and cost of the satellite (in terms of design, construction and power consumption) should be the same compared to the SHSS, which could be an advantage to the telecoms company.
One unfortunate consequence of having a single GES is link vulnerability. Since all terminals are connected through a single base station, in the unlucky event of the link going down, the network will completely lose the ability to reach external networks and all of the low gain UES will not be able to communicate with each other. Moreover, if a suspecting user gains physical access to the single GES it is possible for him to compromise the entire DHSS network.

In a different setting 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 due to 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. Also, the cost of the space-segment is equally higher for the same reason. The advantages, however, come in the form of much wider coverage than the DHSS, the space segment complexity is relatively low (since there is no ISL) and the user terminals do not have to compensate for any extra TX power; hence, small antennas are enough to reach the main network. 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 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 and handover between multiple satellites providing global coverage, meaning the space-segment complexity is much higher than the previous configurations. Moreover, because there is more than one satellite in this topology, the delay will always be more than one RTT. Therefore, for all the listed reasons, the cost of the space-segment will be exponentially higher. In addition to this, the users must handle the transmission with large antennas in order to compensate for the inevitable power loss. This only leaves the advantages of having global coverage and no earth-segment vulnerabilities/cost.

Task 2

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

the following MATLAB code was derived:

```
function PB = ErlangB(N, rho)
    r = (1:N)';
    x = 10.^(N * log10(rho) - sum(log10(r))); % First, calculate numerator separately %

    y = 1; % And the denominator as well %
    for n = 1:N, r = (1:n)';
        y = y + 10.^(n * log10(rho) - sum(log10(r)));
    end

    PB = x / y;  % Finally, divide the two to get the intended result %
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_B 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:

```
erlangs = 120;  % How many calls/hour on the telephone system  %
grade   = 10.0; % The grade of service desired for this system %

% Calculate and plot the blocking probability for every N and rho values %
[N, PB] = plot_N_vs_PB(...
    1,       ... % Create figure 1                                   %
    250,     ... % N will be plotted from 1 to 250 on the x-axis       %
    erlangs, ... % The desired telephone exchange load (Erlangs)       %
    40,      ... % The erlang increment used to plot different loads   %
    grade    ... % The desired grade of service / blocking probability %
);

% Show results %
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 $plot\_N\_vs\_PB()$:

```
function [N_output, PB_output] = plot_N_vs_PB(figure_index, N_max, desired_rho, increment_rho_by, desired_grade)

fprintf('** Calculating N value for Erlangs = %d and Grade of Service = %.2f %% ...\n', desired_rho, desired_grade);

fig = figure(figure_index);
clf(fig);

PB = zeros(1, N_max);
N_vector = 1:N_max;

plot_index = 1;
legend_x_offset = 0;

if figure_index == 2
   max_rho = desired_rho;
   legend_x_offset = 8;
else
   max_rho = desired_rho * 2 - increment_rho_by;
end

for rho = increment_rho_by:increment_rho_by:max_rho
    % Calculate the Blocking Probability from values N = 1 up to N_max
    % for the current Rho value
    for N = N_vector
        PB(N) = ErlangB(N, rho) * 100;
    end

    % Plot PB (Block Probability) in percentages (0 -> 100%)
    % in a semi-logarithmic scale
    graph_handle = semilogy(1:N_max, PB, 'LineWidth', 2);

    % Set the title for all subplots and the labels for the xy axis (only once)
    if plot_index == 1
        title('N (# of trunks) vs Grade of Service (PB [%])');
        xlabel('N (# of trunks)');
        ylabel('Grade of Service (PB [%])');

        grid on;
        hold on;

        xlim([1 N_max]);
        ylim([10^-2 100]);
    end

    % Place the rho value on the current subplot at position y = 10^-1
    N = find(PB <= 10^-1);
    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);

    % Process the rho value in question in a special way
    if rho == desired_rho
        line_legend_text.FontSize = 14;
        line_legend_text.FontWeight = 'bold';
        set(graph_handle, 'LineWidth', 4, 'color', [1 0.85 0]);

        % Look for a percentage value that fits the
        % desired grade of service
        N_output = 0;
        PB_output = 0;
        N_ = find(PB <= desired_grade);

        if ~isempty(N_)
            N = N_(1);
            % Set the output values of this function
            N_output  = N;
            PB_output = PB;

            % Draw an arrow pointing to the result (no. of trunks)
            text(N - 2.5 - legend_x_offset / 1.85, PB(N) * 1.5,'\downarrow', 'FontSize', 25, 'FontWeight', 'bold', 'Color', [1 0 0]);
            text(N - 5, PB(N) * 2.25, ['N = ' num2str(N)], 'FontWeight', 'bold', 'FontSize', 12, 'Color', [0 0 1]);

            % A reference line as well
            ref_y = PB(N) * ones(N_max);
            plot(1:N_max, ref_y, 'color', [0 0 1]);
            text(N_max + 2, PB(N), ['PB = ' num2str(round(PB(N), 2)) ' %'], 'FontWeight', 'bold');

            % Finally, draw an interception point
            plot(N, PB(N), 'rx', 'MarkerSize', 8, 'LineWidth', 1.5, 'Color', [0 0 0]);
        end
    end

    plot_index = plot_index + 1;
end

set(gca, 'XTick', sort([1, get(gca, 'XTick')])); % Add XTick at x=1 %
set(gca, 'YTickLabel', cellstr(['10^{-2}'; '10^{-1}'; '      1'; '     10'; '    100']));
refreshdata;

end % -- END FUNCTION - plot_N_vs_PB() --
```

Code 3: plot_N_vs_PB.m

This produces the graphical output:

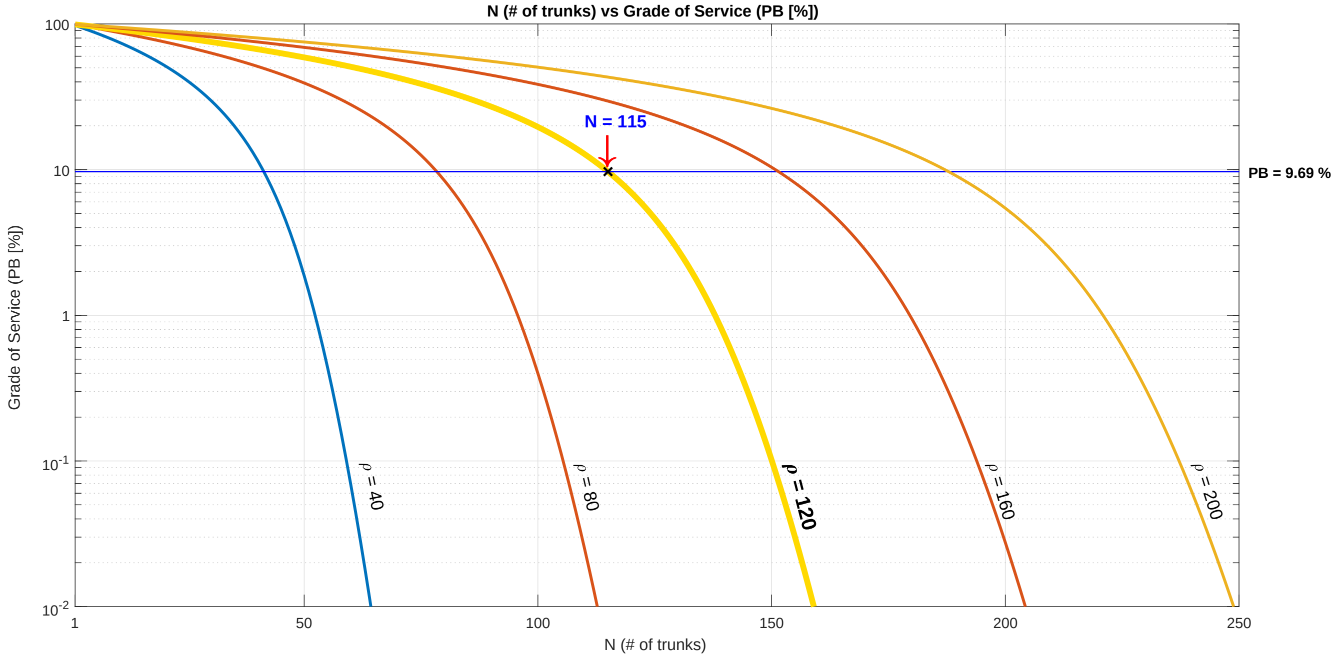

Figure 1: Finding # of trunks (N) for telephone exchange with conditions: Erlangs (ρ) = 120 and Grade of Service (PB) = 10%

Concluding, the answer for task 2.i is: $\underline{\mathbf{N = 115}}$

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

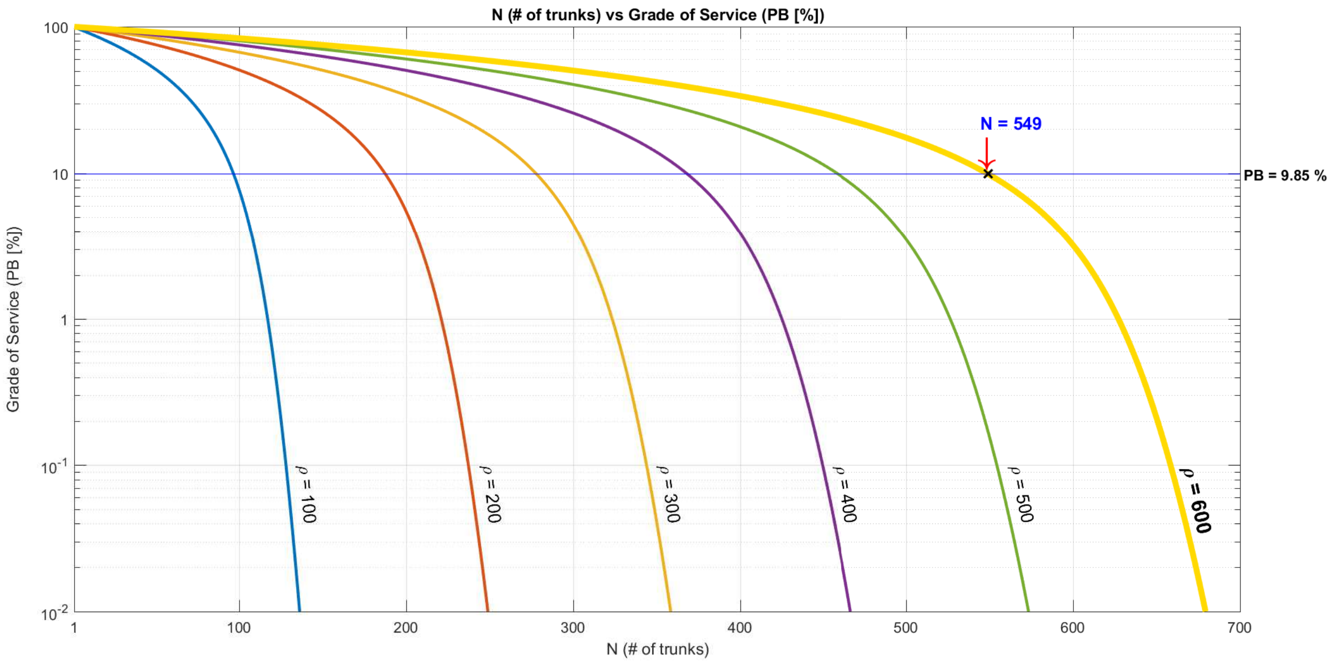

Figure 2: Finding # of trunks (N) for telephone exchange with conditions: Erlangs (ρ) = 600 and Grade of Service (PB) = 10%

$\rightarrow$ **Answer for task 2.ii.a: $\underline{\mathbf{N = 549}}$**

(ii) b) Finally, after resetting the *erlangs* parameter to 120 and changing the *grade* value to 0.1% the result becomes:

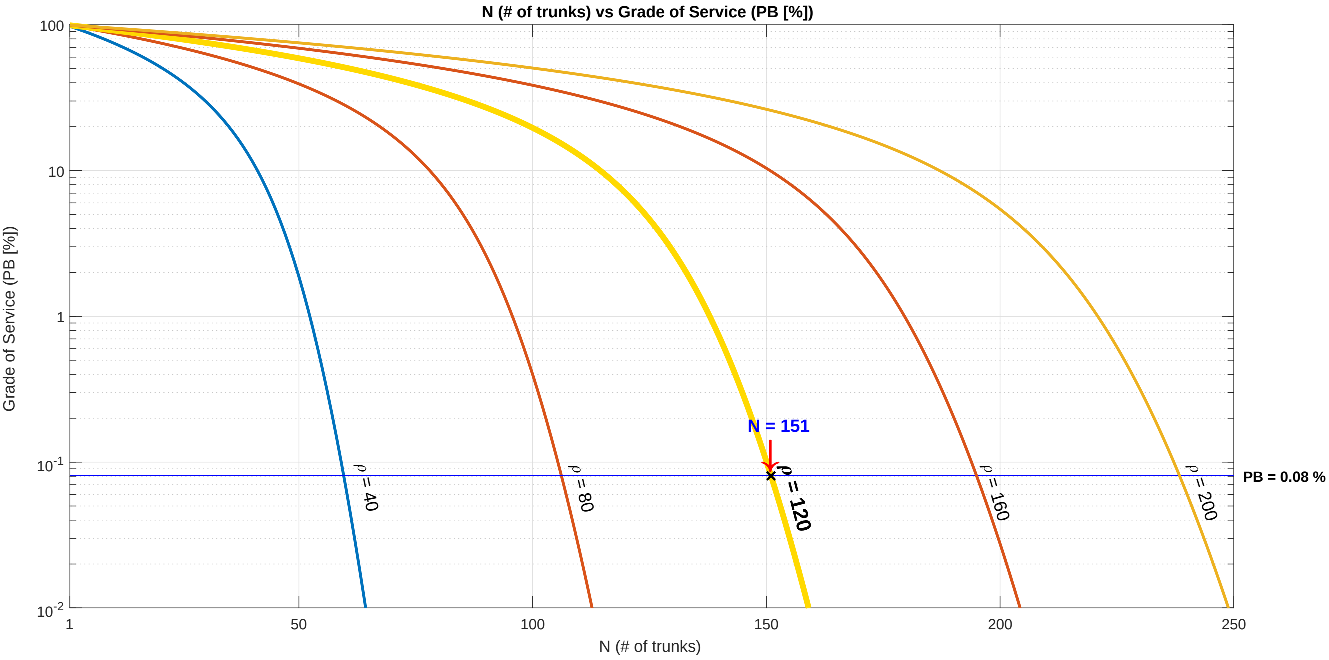

Figure 3: Finding # of trunks (N) for telephone exchange with conditions: Erlangs (ρ) = 120 and Grade of Service (PB) = 0.1%

→ **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 was $\frac{N_2}{N_1} = \frac{549}{115} = \mathbf{4.8}$. 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 $\frac{N_3}{N_1} = \frac{151}{115} = \mathbf{1.3}$. 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 the RSOH and MSOH fields. These are meant to be used in-between regenerators and multiplexing points. The RSOH contains information about frame alignment, as well as 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:

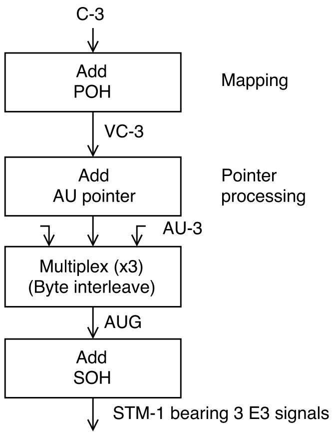

Figure 4: Building an STM-1 frame from three E3 signals

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:

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

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 columns × 9 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 specially designed to handle transmission of IP packets between two hosts ideally 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.[5] The **absolute minimum** time it takes for radio waves to travel to the LEO orbit is $\frac{5000 \text{ km}}{c} = \frac{5 \cdot 10^6}{3 \cdot 10^8} = 1.7 \cdot 10^{-2}$ s = 17 ms, and in the HEO orbit the travel time is $\frac{20000 \text{ km}}{c} = \frac{20 \cdot 10^6}{3 \cdot 10^8} = 6.7 \cdot 10^{-2}$ s = 67 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 $2^{16 \text{ bits}} - 1 = 64$ 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 $\frac{128 \text{ kB}}{64 \text{ kB}} = 2$ 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 $WindowSize = 2^{16} - 1 = 65535 = 64$ kB, $RTT = 500$ ms and $Bandwidth = 3$ 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:

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

Which, when divided with the total bandwidth gives:

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

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:

$$WindowSize = TCPHeader_{WindowField} << TCPHeader_{OptionsField[2]}$$

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 $65535 << 14 = 1073725440$ 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.[6] 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;[1]
- 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.[1]

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

```
function PB = ErlangB(N, rho)
    r = (1:N)';
    x = 10.^(N * log10(rho) - sum(log10(r))); % First, calculate numerator separately %

    y = 1; % And the denominator as well %
    for n = 1:N, r = (1:n)';
        y = y + 10.^(n * log10(rho) - sum(log10(r)));
    end

    PB = x / y;  % Finally, divide the two to get the intended result %
end  % -- END FUNCTION - ErlangB() --
```

Code A.1: ErlangB.m

A.2 task2_i.m

```
erlangs = 120;  % How many calls/hour on the telephone system  %
grade   = 10.0; % The grade of service desired for this system %

% Calculate and plot the blocking probability for every N and rho values %
[N, PB] = plot_N_vs_PB(...
    1,       ... % Create figure 1                                %
    250,     ... % N will be plotted from 1 to 250 on the x-axis       %
    erlangs, ... % The desired telephone exchange load (Erlangs)       %
    40,      ... % The erlang increment used to plot different loads   %
    grade    ... % The desired grade of service / blocking probability %
);

% Show results %
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

```
erlangs = 600;  % How many calls/hour on the telephone system  %
grade   = 10.0; % The grade of service desired for this system %

% Calculate and plot the blocking probability for every N and rho values %
[N, PB] = plot_N_vs_PB(...
    2,       ... % Create figure 2                                %
    700,     ... % N will be plotted from 1 to 700 on the x-axis       %
    erlangs, ... % The desired telephone exchange load (Erlangs)      %
    100,     ... % The erlang increment used to plot different loads  %
    grade    ... % The desired grade of service / blocking probability %
);

% Show results %
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

```
erlangs = 120; % How many calls/hour on the telephone system  %
grade   = 0.1; % The grade of service desired for this system %

% Calculate and plot the blocking probability for every N and rho values %
[N, PB] = plot_N_vs_PB(...
    3,       ... % Create figure 3                                %
    250,     ... % N will be plotted from 1 to 250 on the x-axis       %
    erlangs, ... % The desired telephone exchange load (Erlangs)      %
    40,      ... % The erlang increment used to plot different loads  %
    grade    ... % The desired grade of service / blocking probability %
);

% Show results %
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

```
function [N_output, PB_output] = plot_N_vs_PB(figure_index, N_max, desired_rho, increment_rho_by, desired_grade)

fprintf('** Calculating N value for Erlangs = %d and Grade of Service = %.2f %% ...\n', desired_rho, desired_grade);

fig = figure(figure_index);
clf(fig);

PB = zeros(1, N_max);
N_vector = 1:N_max;

plot_index = 1;
legend_x_offset = 0;

if figure_index == 2
   max_rho = desired_rho;
   legend_x_offset = 8;
else
   max_rho = desired_rho * 2 - increment_rho_by;
end

for rho = increment_rho_by:increment_rho_by:max_rho
    % Calculate the Blocking Probability from values N = 1 up to N_max
    % for the current Rho value
    for N = N_vector
```

```
            PB(N) = ErlangB(N, rho) * 100;
        end

        % Plot PB (Block Probability) in percentages (0 -> 100%)
        % in a semi-logarithmic scale
        graph_handle = semilogy(1:N_max, PB, 'LineWidth', 2);

        % Set the title for all subplots and the labels for the xy axis (only once)
        if plot_index == 1
            title('N (# of trunks) vs Grade of Service (PB [%])');
            xlabel('N (# of trunks)');
            ylabel('Grade of Service (PB [%])');

            grid on;
            hold on;

            xlim([1 N_max]);
            ylim([10^-2 100]);
        end

        % Place the rho value on the current subplot at position y = 10^-1
        N = find(PB <= 10^-1);
        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);

        % Process the rho value in question in a special way
        if rho == desired_rho
            line_legend_text.FontSize = 14;
            line_legend_text.FontWeight = 'bold';
            set(graph_handle, 'LineWidth', 4, 'color', [1 0.85 0]);

            % Look for a percentage value that fits the
            % desired grade of service
            N_output = 0;
            PB_output = 0;
            N_ = find(PB <= desired_grade);

            if ~isempty(N_)
                N = N_(1);
                % Set the output values of this function
                N_output  = N;
                PB_output = PB;

                % Draw an arrow pointing to the result (no. of trunks)
                text(N - 2.5 - legend_x_offset / 1.85, PB(N) * 1.5,'\downarrow', 'FontSize', 25, 'FontWeight', 'bold', 'Color', [1 0 0]);
                text(N - 5, PB(N) * 2.25, ['N = ' num2str(N)], 'FontWeight', 'bold', 'FontSize', 12, 'Color', [0 0 1]);

                % A reference line as well
                ref_y = PB(N) * ones(N_max);
                plot(1:N_max, ref_y, 'color', [0 0 1]);
                text(N_max + 2, PB(N), ['PB = ' num2str(round(PB(N), 2)) ' %'], 'FontWeight', 'bold');

                % Finally, draw an interception point
                plot(N, PB(N), 'rx', 'MarkerSize', 8, 'LineWidth', 1.5, 'Color', [0 0 0]);
            end
        end

        plot_index = plot_index + 1;
    end

    set(gca, 'XTick', sort([1, get(gca, 'XTick')])); % Add XTick at x=1 %
    set(gca, 'YTickLabel', cellstr(['10^{-2}'; '10^{-1}'; '      1'; '     10'; '    100']));
    refreshdata;

    end % -- END FUNCTION - plot_N_vs_PB() --
```

Code A.5: plot_N_vs_PB.m