ECEU-628: Computer Communications Networks
Sample Final Examination with Solutions

Prof. A. Bruce McDonald†
mcdonald@ece.neu.edu
†**Department of Electrical and Computer Engineering**
Northeastern University

April 5, 2004

Question One: Short-Answer with Solutions

(A) Given a Go-Back-N ARQ scheme with a window size of 7 what are the necessary and sufficient set of sequence numbers required for correct protocol operation?

Solution

The solution is to have a sequence space of $N + 1$*, hence, the "necessary and sufficient set of sequence numbers" are:* $\{0, 1, 2, 3, 4, 5, 6, 7\}$.

(B) Explain the limitation in packet length for standard Kermit: what is the maximum packet length and why is it limited?

Solution

In standard Kermit (without extensions) the length of a Kermit packet is specified is one character field: LEN. Printable ASCII ranges from decimal 32 *to decimal* 126*. Kermit must ensure that LEN is printable ASCII. Hence, at the sender the actual length is modified to ensure it is printable: LEN = actual-packet-length + 32; at the receiver the reverse is done: actual-packet-length = LEN - 32. Thus, the maximum length is limited to* $126 - 32 = 94$ *bytes.*

(C) Using Shannon's Capacity Theorem determine the minimum signal-to-noise ratio (in dB) that can be tolerated in order to reliably transmit a digital bit stream at a rate of 1.544 Mbps over a 96 kHz band-limited channel?

Solution

According to Shannon the maximum theoretic information carrying capacity of a bandwidth limited noisy channel is given by: $C = W \log_2(\frac{S}{N} + 1)$*; hence* $\frac{C}{W} = \log_2(\frac{S}{N} + 1)$*. You are given* $C = 1.544Mbps$ *and* $W = 96kHz$*. The problem is to find SNR in dB which is given by:* $SNR(dB) = 10 \log_{10}(\frac{S}{N})$ *Substituting what is known and using the following approximations—*$2^{16.08} \approx 65586$ *and* $\log_{10}(65535) \approx 4.75$*: we get* $SNR \geq 47.5$ *dB (The exact answer is* $SNR \geq 48.4$ *dB.*

(D) *Given an IP address: 150.192.128.27 with a netmask 255.255.224.0 answer the following: What class network does this address come from? Identify the class 'X' network ID and all feasible subnetworks (e.g. 150.192.128.0 is one of them) associated with the network and the given mask?*

Solution

Using the class-based address structure observer that the first octet (150) is 10110110 represented in binary. Hence, it must be a class-B network, namely network 150.192.0.0. The subnet mask is 255.255.224.0 indicates that the network has been subdivided into subnets using a 3-bit subnet mask. The feasible subnets are easy to find—from the binary numbers, e.g. $00100000, 01000000, \ldots = 32, 64, \ldots$. Clearly they advance by 32, hence, the set of feasible subnets are: $\{150.192.32.0, 150.192.64.0, 150.192.32.0, 150.192.96.0, 150.192.128.0, 150.192.160.0, 150.192.192.0\}$

(E) *What is the difference between switching and multiplexing?*

Solution

Both switching and multiplexing are fundamental network functions or building blocks—switching is the mechanism by which an input signal arriving on a given input line, time-slot, frequency or wavelength is received, processed and transmitted on a given—possibly different output line, time-slot, frequency or wavelength. Switching supports the routing service... Multiplexing is the mechanism by which multiple input signals are combined into a single output signal. The purpose is to reduce the complexity of the network and cost of infrastructure and control.

Question One: Short-Answer—Solutions with Detailed Discussion

(A) Given a Go-Back-N ARQ scheme with a window size of 7 what are the necessary and sufficient set of sequence numbers required for correct protocol operation?

Solution

The problem occurs at the 'wrap-around' point—if the number distinct sequence numbers is insufficient the result is **ambiguity***: a state that cannot exist in a correct algorithm. Specifically, if there are only* 7 *sequence numbers and the sender transmits seven frames and then receives ACK-0 it can either be interpreted as "all seven frame were received correctly" or "the first frame (and possibly others) were received incorrectly, hence all seven must be re-transmitted".* **The solution is to have a sequence space of** $N + 1$**, hence, the "necessary and sufficient set of sequence numbers" are:** $\{0, 1, 2, 3, 4, 5, 6, 7\}$**.** *The sender increments the sequence number as follows: if* $N(S)$ *is the sequence number last sent, the next number to be sent is* $N(S) + 1 \ \% \ (N + 1)$*. Where* % *is the modulus operator.*

(B) Explain the limitation in packet length for standard Kermit: what is the maximum packet length and why is it limited?

Solution

The fundamental design criteria to achieve inter-operability and portability in file transfers using Kermit was based on ensuring that only printable ASCII characters are transmitted. This ensured (with high probability) that existing proprietary communications software and hardware would not 'react' to the incoming serial data without modification. Thus, the popularity of Kermit and its successors prior to widespread Internet access. Using standard Kermit (without extensions) the length of a Kermit packet is specified is one character field: LEN. Printable ASCII ranges from decimal 32 *to decimal* 126 *(*127 *is the DEL character); non of the 8-bit ASCII characters are permissible. The designers of Kermit had the choice to either have a minimum packet length of* 32*, which could have been achieved by padding short packets, thus they could have had a larger maximum size. However, since one half of the packets in a standard (stop-and-wait) Kermit session are ACK packets that are very short it was felt that a better, and less processing intensive solution was using the* **characterization** *technique: At the sender the actual length is modified to ensure it is printable:* $LEN = actual - packet - length + 32$*; at the receiver the reverse is done:* $actual - packet - length = LEN - 32$*. Thus, the maximum length is limited to* $126 - 32 = 94$ *bytes.*

(C) Using Shannon's Capacity Theorem determine the minimum signal-to-noise ratio (in dB) that can be tolerated in order to reliably transmit a digital bit stream at a rate of 1.544 Mbps over a 96 kHz band-limited channel?

Solution

According to Shannon the maximum theoretic information carrying capacity of a bandwidth limited noisy channel is given by: $C = W\log_2(\frac{S}{N}+1)$*; hence* $\frac{C}{W} = \log_2(\frac{S}{N}+1)$*. Where the signal to noise ratio is a ratio without units (not in dB) You are given* $C = 1.544Mbps$ *and* $W = 96kHz$*. The problem is to find* SNR *in dB which is given by:* $SNR(dB) = 10\log_{10}(\frac{S}{N})$ *Note that we use* $10\log_{10}$ *to covert to dB for voltages. Substituting what is known and using the following approximations—*$2^{16.08} \approx 65586$ *and* $\log_{10}(65535) \approx 4.75$*: we get* $SNR \geq 47.5$ *dB (The exact answer is* $SNR \geq 48.4$ *dB).*

$$\begin{aligned}
1.544 * 10^6 &= 96 * 10^3 \log_2(\frac{S}{N}+1) \\
\frac{1.544 * 10^6}{96 * 10^3} &= \log_2(\frac{S}{N}+1) \\
16.08 &= \log_2(\frac{S}{N}+1) \\
2^{16.08} &= \frac{S}{N}+1 \\
\frac{S}{N} &= 2^{16.08}-1 \\
SNR(db) &= 10\log_{10}(2^{16.08}-1) \\
SNR(db) &\approx 10\log_{10}(65536-1) \\
SNR(db) &\approx 10\log_{10}(65535) \\
SNR(db) &\approx 10(4.75) \\
SNR(db) &\geq 47.5
\end{aligned} \tag{1}$$

(D) Given an IP address: 150.192.128.27 with a netmask 255.255.224.0 answer the following: What class network does this address come from? Identify the class 'X' network ID and all feasible subnetworks (e.g. 150.192.128.0 is one of them) associated with the network and the given mask?

Solution

*Using the class-based address structure observer that the first octet (*150*) is* 10110110 *represented in binary. Hence, it must be a class-B network based on the rule that class A networks begin with "1", class B networks begin with "10" and class C networks begin with "110'. The class B network ID is: 150.192.0.0. The subnet mask is 255.255.224.0 indicates that the third octet has been provided with a modified subnet structure. Namely, the network 150.192.0.0 has been subdivided into multiple smaller subnets. The mask shows how this is done. Observe the third octet of the mask. Without subnets the class B mask is 255.255.0.0, here we find 224 in the third octet. In binary this is 11100000, hence, the firsts three bits have been allocated for the subnet number: netid/subnetid/hostid. With three bits there are 8 combinations, however '111' is reserved as the broadcast address—i.e. broadcast on all the subnets—if broadcast is supported then the host address portion must also be set to all one's as well. The subnet '000' means 'this subnet only', hence, by setting all the host bits to one is enables a broadcast within the source subnet only. As such there are* $8-2=6$ *permitted subnets. Each subnet can support* $2^{13}-2 = 8190 hosts$*. Thus, a total of* $6 * 8190 = 49140$ *hosts can be supported on the class B network 150.192.0.0. A flat class B network supports* $2^{16}-2 = 65534$ *hosts, hence, with a three-bit subnet mask* 16394 *host addresses are lost. The feasible subnets are easy to find—from the binary numbers, e.g.* $00100000, 01000000, \ldots = 32, 64, \ldots$*. Clearly they advance by 32, hence, the set of feasible subnets are:* $\{150.192.32.0, 150.192.64.0, 150.192.32.0, 150.192.96.0, 150.192.128.0, 150.192.160.0, 150.192.192.0\}$

(E) What is the difference between switching and multiplexing?

Solution

Both switching and multiplexing are fundamental network functions or building blocks—**switching is the mechanism by which an input signal arriving on a given input line, time-slot, frequency or wavelength is received, processed and transmitted on a given—possibly different output line, time-slot, frequency or wavelength. Multiplexing is the mechanism by which multiple input signals are combined into a single output signal.** *The purpose is to reduce the complexity of the network and cost of infrastructure and control. Multiplexing may be deterministic or statistical; packet switching is associated with statistical multiplexing wherein the aggregate output rate my be significantly less than the sum of the inputs. This is due to the bursty nature of data traffic—other multiplexing techniques include FDM (frequency modulation), TDM (time-slots), WDM (wavelength for photonics), DWDM (dense wavelength). Multiple access techniques for LANs are actually a form of distributed multiplexing. The stochastic nature of the inputs and the distributed control lead to flexibility and large performance variations. TDMA and CDMA are controlled multiplexing techniques used in wireless environments. TDMA is a TDM scheme in which time slots are allocation on a demand basis and released when no longer needed. CDMA uses spread-spectrum technology to provide* channels *that follow different frequency hopping sequences. Noise and interference can be reduced. It is common in present-day wireless systems to use hybrid systems wherein each CDMA channel is divided in time using TDMA. Thus, a used is allocated a spreading code and time slot.*

Switching can involve different transfer mechanisms, hardware, multiplexing techniques and can occur at different 'levels' of a protocol stack. The two basic switching methods are circuit and packet switching. Packet switching is further divided into datagram and virtual-circuit methods. Switching systems involve hardware and software. The earliest switches were electromechanical devices that set up an end-to-end electrical path for the information signal. Today there are a number of switching technologies used to fit the application. The most simple is the software switch that reads the input and processes the information contained in the packet (for packet switching) or stored in the switches configuration tables for circuit switching to select to correct output to copy the data/signal element to. Faster techniques are based on hardware: (1) The shared memory switch reads incoming data from each input port and copies it into a common shared memory space. DMA (direct-memory-access) can be used to avoid extra copies and used in conjunction with a global input list and a per-port output list. With careful timing and memory access control data is copied only once. The cost is complex control a real-time software, hence, the benefit of the simple architecture (cost) may be lost. (2) The common high speed bus copies data from the input ports to the output port buffer along a common bus that runs at the aggregate speed of the input ports. The architecture is not scalable, but works well for high speed switching of small fixed-length packets or slots. (3) The switching fabric provides the most flexible and extensible solution to both circuit and packet switching problems. The 'cross-bar' is the simplest version, wherein, all input lines connect directly to all output lines. The shortcoming of the approach is the number of cross-points (relays) and the control to manage them. Multistage architectures can be designed with far fewer relays and greatly reduced internal connection*, thus, allowing many input to output transfers to proceed simultaneously.*

Today, high speed switches can move a packet or time slot in multiple dimensions: space, time and wavelength—consider a T-1 input, the T-1 line is actually one of hundreds of T-1s multiplexed onto an OC-48 SONET link (2.48 Gbps); moreover the same input line is using DWDM (Dense-Wave-Division-Multiplexing) to transport 50 OC-48s at different wavelengths on the same fiber (124 Gbps). A core switch can switch (1) in **space***—move the desire time-slot from one physical input port to a different physical output port, (2) in* **time***—if the same time-slot is not available on the outgoing 'T-1' a TSI (time-slot-interchanger) can switching it to a free outgoing time-slot, and (3) in* **wavelength or frequency***—in the case of WDM or DWDM we consider wavelength switching as moving an input arriving at a giving wavelength (color) to be switched to a different output wavelength (color) if it is necessary to support to connection or packet transfer. Switching is done to 'find the best fit', i.e. the one that will involve the fewest changes in the switch control and the least latency. Sometimes re-arrangement of existing control structures are required to support new connections or packet flows. A common technique in switching systems is to use 'self routing' tags. The switching control is established during the first time traversing the switch; subsequently a control word is stored and then appended to the data element as it moves through the switch and provides specific switching direction at each stage and time interval.*

Question Two: Idle-RQ Data-Link Communication Efficiency

Give an approximate analysis of the *effective-transmission-rate* for an Idle-RQ DLC ARQ (Stop-and-Wait) algorithm [1] . Be sure to provide clear solutions for each part: (A) - (E). Assume that data flow is in one direction only—from Node-A to Node-B. In the analysis do not put a fixed restriction on the number of errored frames within an arbitrary time interval, rather, account for the probability of an errored frame based upon the given bit-error-rate (BER). Furthermore, account for the following system parameters for which details will follow:

- The number of overhead bits associated with the DLC of each data frame in the forward direction.
- The length of the acknowledgment frames, both positive and negative, in the reverse direction.
- The processing times at the sender and the receiver.
- The propagation delay between the sender and the receiver.

Define the effective transmission rate in a similar manner to efficiency as follows:

$$R_e = \frac{Total\ number\ of\ received\ bits\ accepted\ by\ DLC}{Elapsed\ time\ to\ transfer\ those\ bits\ and\ accept\ them.}$$

Based on this definition it should be apparent that the link efficiency can be computed as follows, where R is the link's transmission rate in bits-per-second:

$$\eta = \frac{R_e}{R}$$

(A) Compute the probability distribution for the number of transmissions required to have a given frame to be accepted by the receiver's DLC. Assume that **P** is the probability of an error in any given frame. Furthermore, assume that frame errors are independent over all frame transmissions.

- Let $p = bit\ error\ rate$ (BER), the probability of a bit being in error for the link.
- Assume that bit errors are independent.
- Assume that the frame-length is fixed and equal to K bits.

The independence of frame errors implies that the total number of transmissions required for a given frame to be correctly received must be geometrically distributed. Compute the probability that it takes j transmissions to correctly receive the frame: Pr{j transmissions are required} = ?

Solution

Since the outcome of each frame transmission is independent with a fixed probability of error—and only one success is required, the number of transmissions is geometrically distributed:

$$Pr\{j\ transmissions\ are\ required\} = (1-P)P^{j-1}$$

All K bits of a frame must be received correctly for the frame transmission to be a success. Since bit errors are assumed independent P is itself determined with respect to p using a geometric distribution:

$$Pr\{j\ transmissions\ are\ required\} = (1-p)^k(1-(1-p)^k)^{j-1}$$

[1] Be sure to read all the question *very* carefully, including what *isn't said directly* and consider all the assumptions.

(B) One of the limitations of the present computation is that it is based on the assumption that all bit errors occur independently; however, the occurrence of single bit errors spread out randomly over all transmitted bits is unlikely. Typically bit errors arrive in a large burst affecting many closely spaced bits. What is the effect of ignoring the burst-effect and assuming independent bit errors on the result computed in (A) ?

Solution

The random independent occurrence of bit errors, as given in the assumptions leads to a simple mathematical solution to the problem. However, given that frame lengths are fixed and equal to k bits, the physical result is that every frame has an equal chance of being in error. Hence, the same number of bit errors will be spread over a larger number of frames. Specifically, since a single bit error leads to a bad frame, the assumption leads to a larger number of frame errors for the same BER. The end result is that the effective transmission rate ,R_e, will tend to be overly conservative, i.e. less that what would actually be observed given a more realistic model of burst errors, which would affect fewer frames, but cause more bit errors within those frames.

Assumptions for remaining parts of problem: Make the following assumptions in solving the remaining questions in this problem:

- Assume that the error detecting algorithm is ideal in that it always catches frames that are in error, and never detects an error in an error-free frame.
- Assume that acknowledgment frames (both positive and negative) are very small relative to the data frames. Hence, given reasonable line error rates the probability of receiving an ACK or NAK in error is negligible.
- Assume that frames have a fixed length of K bits inclusive of overhead bits for DLC framing.
- Let n_h represent a fixed number of DLC overhead bits in each K bit frame.
- Let n_a represent a fixed number of bits in both ACK and NAK frames, where, $n_a \ll K$.
- Assume there is a fixed processing delay equal to t_{ta} that effectively acts as a "line-turn-around" time in the stop-and-wait ARQ. The same processing delay is assumed for processing all frames/frame types.
- Let t_p represent the propagation delay of the link.
- Let R be the transmission rate of the link in bits-per-second.
- The system has an ideal timeout interval, namely, the sender will timeout and retransmit is the time is *would have taken* to receive and ACK or a NAK frame.

(C) Compute how long it takes to send a single frame and receive the acknowledgment (ACK or NAK) from the receiving side. This is assuming a single transmission. Illustrate all the delay components using the standard A-to-B frame transmission model used in analyzing ARQ algorithms:

Solution

Let S be the fixed round trip time for any frame transmission whether or not an error is detected. Assume that the timeout at the sender is also equal to S:

$$S = \frac{K}{R} + t_p + t_{ta} + \frac{n_a}{R} + t_p + t_{ta} = \frac{K + n_a}{R} + 2(t_p + t_{ta})$$

(D) Compute the average number of transmissions that are required before a frame is received correctly and accepted by the receiving DLC:

Solution

This is simply the expected number of trials required before a "success" is achieve given the geometric distribution, i.e. the expectation of the discrete geometric distribution.

$$E[T] = \frac{1}{1 - P}$$

(E) Utilizing all the information you have computed or was given to you—**show that** the effective transmission rate, R_e, of the data-link is given by the following expression:

$$R_e = \frac{(1 - P)(K - n_h)R}{K + n_a + 2(t_p + t_{ta})R}$$

Solution

All that remains to solve the problem is to divide the total number of (new) data-bits transmitted (data frame minus overhead) by the total expected time that it takes to receive and accept those bits, which is equal to the product of the expected number of transmissions, $E[T]$ and the total round-trip time S and use a little algebraic manipulation to obtain the desired form:

$$\begin{aligned} R_e &= \frac{K - n_h}{\frac{1}{1-P}\left(2(t_p + t_{ta}) + \frac{K+n_a}{R}\right)} \\ &= \frac{(1 - P)(K - n_h)R}{K + n_a + 2(t_p + t_{ta})R} \end{aligned}$$

Problem Two Solution: Detailed Explanations

In this section I have presented a more in-depth analysis of each part of the second problem to ensure that all students can see the precise concepts methods used to solve each part of the problem. The problem itself is quite simple once you are comfortable with the concepts! The question is what might be asked on the exam itself? Try to guess two or three possibilities and solve the problem(s) you anticipate. This is the optimal strategy for exam preparation once concepts are understood.

Part (A)

Since the probability of frame errors are independent for every transmission, and a frame only needs to be received correctly one time in j transmissions, there must be $j-1$ times the frame is received with at least one bit-error. Thus, the is simply a matter of direct application of the geometric distribution, or applying the logic just stated:

$$Pr\{j \text{ transmissions are required}\} = (1-P)P^{j-1}$$

Although it was not required within the context of this problem, the solution could easily be stated in terms of the BER by the following relationship: $P = 1-(1-p)^k$. This expression is the result of the independence of bit errors and the fixed frame length, k. Clearly, all k bits must be received correctly for the frame to be received correctly, hence, the probability of a frame error is this quantity subtracted from 1. The fully expanded result is then give by the following:

$$Pr\{j \text{ transmissions are required}\} = (1-p)^k(1-(1-p)^k)^{j-1}$$

Part (B)

The random independent occurrence of bit errors, as given in the assumptions leads to a simply mathematical solution to the problem. However, given that frame lengths are fixed and equal to k bits, the physical result of this assumption is that every frame has an equal chance of being in error. Hence, the same number of bit errors will be spread over a larger number of frames. Specifically, since a single bit error leads to a bad frame, the assumption leads to a larger number of frame errors for the same BER. The end result is that the effective transmission rate ,R_e, will tend to be overly conservative, i.e. less that what would actually be observed given a more realistic model of burst errors, which would affect fewer frames, but cause more bit errors within those frames.

Part (C)

The solution to this problem is to simply sum up all the components of delay in the forward and the reverse path using all the given information. This is the same as the entity S that we have used in previous examples and homework problems. The only difference is that we are not ignoring the transmission times of the ACK/NAK frames, and we are assuming that the "processing" delay at each end is large enough relative to the other delay components that it cannot be ignored. The solution is give be the following expression (Define the total round-trip delay to be the fixed value S):

$$S = \frac{K}{R} + t_p + t_{ta} + \frac{n_a}{R} + t_p + t_{ta} = \frac{K+n_a}{R} + 2(t_p + t_{ta})$$

Part (D)

This is simply the expected number of trials required before a "success" is achieve given the geometric distribution, i.e. the expectation of the discrete geometric distribution. If you did not recall this value, it is simple to compute using the definition of expectation and knowledge of the probability distribution from Question (A). Let T be the number of transmissions required to achieve a success. The solution is given by the following expression:

$$E[T] = \frac{1}{1-P}$$

This solution could be expanded in terms of p, but this was not necessary. To compute *the expected value of T proceed as follows:*

$$\begin{aligned}
E[T] &= \sum_{n=0}^{\infty} n(1-P)P^{n-1} \\
&= \sum_{n=1}^{\infty} n(1-P)P^{n-1} \\
&= (1-P)\sum_{n=1}^{\infty} nP^{n-1} \\
&= (1-P)\sum_{n=1}^{\infty} \frac{d}{dP}(P)^n \\
&= (1-P)\frac{d}{dP}\left(\sum_{n=1}^{\infty}(P)^n\right) \\
&= (1-P)\frac{d}{dP}\left(\sum_{n=0}^{\infty}(P)^n - 1\right) \\
&= (1-P)\frac{d}{dP}\left(\frac{1}{1-P} - \frac{1-P}{1-P}\right) \\
&= (1-P)\frac{d}{dP}\left(\frac{P}{1-P}\right) \\
&= \frac{(1-P)}{(1-P)^2} \\
E[T] &= = \frac{1}{(1-P)}
\end{aligned}$$

Explanation: *First apply the definition of expectation to the geometric distribution and perform some simple algebraic manipulation. The strategy in problems like these is transform a problem that you do not know how to solve into a problem or problems that you do know how to solve. A common obstacle involves variables that cannot be separated or index terms that appear as coefficients. A common trick to eliminate this problem is to look for anti-derivatives. In this case observe that $nP^{n-1} = \frac{d}{dP}(P)^n$. Thus, the index term of the summation can be eliminated as a coefficient by introducing the derivative of a power series. The equation is clearly an example of term-by-term differentiation of a power series, which, by the Differentiability Theorem is equal to the derivative of the power series; hence the summation and derivative can be interchanged. For the power series to converge the lower limit on the summation is changed to 0, consequently, to balance the equation $1 = \frac{1-P}{1-P}$ is subtracted from the result. Noting that for $0 < a < 1, \sum_{n=0}^{\infty}(P)^n = \frac{1}{1-a}$ and taking the derivative of the reduced expression leads to the final result as shown.*

Part (E)

All that remains to solve the problem of effective transmission rate is to divide the total number of data-bits transmitted (data frame minus overhead) by the total expected time that it takes to receive and accept those bits, which is equal to the product of the expected number of transmissions, $E[T]$ *and the total round-trip time* S*:*

$$
\begin{aligned}
R_e &= \frac{K - n_h}{\frac{1}{1-P}\left(2(t_p + t_{ta}) + \frac{K+n_a}{R}\right)} \\
&= \frac{(1-P)(K-n_h)R}{K + n_a + 2(t_p + t_{ta})R}
\end{aligned}
$$

Discussion

The following presents some discussion of the relevance of this problem and other things you should think about when studying it: We covered this problem fairly extensively during the two review sessions that I held. The main points for this type of problem are too (1) read the question very carefully and take note of details, for example, the fact that the K *bit frames include the* n_h *bits of overhead and that overhead is not counted as data; (2) take note of all the assumptions; (3) develop and understand the strategy for solving the problem; and, (4) consider the problem in its context: what are the effects of changing different parameters? Is there an* optimal value *for frame length* K*? Can you reason how changing the frame length from small to large would effect the performance—how and why? Using the parameters that are given what scenarios would provide efficient performance? How would things change for a different ARQ scheme? You may have notice some probability in this problem—You ought to be able to identify the that given independent experiments with two mutually exclusive outcomes, success or failure, the number of trials (transmissions) required before a success is* geometrically distributed*. Furthermore, the expected value of the* geometric distribution *is one of those important and common things that should be committed to memory. However, for the final exam the only thing I would assume is that you could express the probability in Part-A based on the independence assumption and knowledge that only one correct transmission is required.*

Some Final Notes on Probability

Probability has shown up on a few occasions during the semester—in class, in the readings and in some of the homework assignments. To this end I ask that you ensure you are familiar with the following distributions:

1. Geometric (discrete)
2. Poisson (discrete)
3. Binomial (discrete)
4. Exponential (continuous)
5. Uniform (continuous)

You are permitted to bring/use a cheat-sheet *if it helps, but no probability text books. You must know the pdf or pmf and cdf for each of the distributions and you must either know or be able calculate the expected value of each. More importantly, you need to be able to apply them to a problem, for example* this problem. *Also, recall from class the emphasis on the Poisson distribution for modeling packet arrivals, specifically it was used in the analysis of the Aloha MAC algorithm. How did I* justify *the assumption of Poisson arrivals, even though I indicated that the individual sources were* not Poisson*? This was fundamental to the analysis I spent so much time on building. Moreover, what specifically is meant by 'Poisson Arrivals'? The Poisson process is a counting process so what does it tell is about* N *packets arriving over a given interval of time* t*? If the arrival process is Poisson what can you say about the distribution of* inter-arrival times*? What is an inter-arrival time (be careful—it is easy to make a mistake on this one)?*

Problem Three: Voice-over-IP

Consider a packet voice system similar to the one on the problem given in the midterm. The system consists of two *packet voice* (VoIP) channels that are multiplexed over a single T-1 line.

Important note: do not be confused by the use of of T-1 in this problem: T-1 was developed by the phone company to support TDM voice, and is still used in that capacity by service providers and by end-users with large private phone systems. However, it is common to offer T-1 as a 'service' to end-users that can be used as a more flexible 'bit-pipe' to carry arbitrary forms of packet data (e.g. IP, Frame-Relay, ATM, etc.). As such, the notion of TDM is no longer directly relevant. The packets **are not** transmitted with TDM. From the perspective of the attached device it is a transmission line with a given transmission rate. There is, however, an important consideration. In some systems the carrier may offer what is called *clear-channel* T-1 service, wherein the only limitation is the framing. This can be done by offering T-1 service over SONET or through modification of line equipment. Other carriers either charge more or do not offer clear-channel T-1. As such, the end-user is further limited by the in-band signaling system designed in conjuction with the *super-frame* for TDM voice (See T-1 notes and handout). In this problem we consider a system that is carries data (VoIP)—it does not use TDM but does not have clear-channel service.

(A) The transmission rate of a T-1 line is 1.5444 Mbps; however there are two factors that limit the amount of usable bandwidth: (1) framing and (2) signaling overhead. These limitations arise from the original application of T-1 as a carrier of TDM voice (24 PCM channels) and the mechanisms used to achieve framing and the in-band transfer of signaling information. Considering these factors what is the effective transmission rate for carrying *general* data traffic (packet data)?

Solution

The effective transmission rate without a clear channel service is the basic rate: 1.544 *Mbps minus the overhead for framing (one bit per frame) and the low order bit of each 'time-slot' which is vulnerable to robbed-bit signaling in every sixth frame. The easiest way to compute this is to assume only* 56 *Kbps per channel; hence,* $R = 56000 * 24 = 1.344$ *Mbps.*

(B) Current VoIP systems packetize voice in fixed 30 ms blocks. The packet length may vary depending on the encoding and compression schemes being used. Assume standard PCM is used to convert the voice from analog to digital and that no compression is being used. Assume the total protocol overhead per-packet (due to IP etc.) is 40 octets (bytes). What is the resulting fixed packet length L?

Solution

A 30 *ms voice interval using standard PCM consists of* $\frac{30}{0.125} = 240$ *8-bit samples. Given the* 40 *bytes of overhead the total packet length will be fixed at* 280 *bytes or* $L = 2240$ *bits.*

(C) Assume that samples are collected and packetized simultaneously from both channels—one packet for channel 1 and a second packet for channel 2—and then moved instantaneously to the transmission buffer. Let the end-to-end propagation delay be 250μs and assume processing delays are negligible. Is the system *stable* as defined—i.e. will the transmission queue be cleared prior to the arrival each subsequent batch of packets? If the system is *stable* what will the long-term average end-to-end delay (from sampling to play-out) be for each channel? If the system is *unstable* what will the delays be?

Solution

At the given transmission rate and packet lengths it will take $600\mu s$ *to transmit each VoIP packet. Hence, it will take* $1.2ms$ *to clear the buffer. Since packetization takes* $30ms$ *there will be* $28.8ms$ *of time when the transmitter is idle during each cycle. As such, the system is stable. The end-to-end delay consists of the sampling/packetization delay, the queuing delay, the transmission time and the propagation delay:*

$$
\begin{aligned}
D_{ch1} &= 30\ ms + 600\ \mu s + 250\ \mu s = 30.85\ ms \\
D_{ch2} &= 30\ ms + 2 * 600\ \mu s + 250\ \mu s = 31.45\ ms
\end{aligned} \tag{2}
$$

Problem Four: Shortest-Path Route Computation

Consider the network shown in Figure 1. In this problem we use the distance vector (DV) routing approach to find shortest paths. DV routing is based in the Bellman-Ford algorithm which iterates on the number of hops—thus, it is also known as a depth-first search. Note that by *iterating on the number of hops* we mean that on each iteration $k = 1, 2 \ldots$ the algorithm finds the least cost path of $\leq k$ hops. Thus, under fixed conditions (no changes in link costs or topology—meaning no link or node failures or activations), in the worst case it will require $N - 1$ iterations to converge (find all the shortest paths). In practice the algorithm terminates following any iteration in which no cost changes occurred.

Use the following notation and consider the centralized/synchronous operation of the DV routing algorithm:

- $D_i^{(m)}(k)$: the estimated least cost from node m to node i after the k^{th} iteration of the algorithm.
- $D_i^{(j)}(k)$: the estimated least cost from node j to node i as reported in its routing update after the k^{th} iteration of the algorithm.
- N_m: the set of routers that are directly adjacent (neighbors) of node m.
- $d_{m,j}$: the fixed link cost between node m and node $j \in N_m$.
- $d_{m,j}(k)$: the dynamically varying link cost between node m and node $j \in N_m$ as seen at the k^{th} iteration of the algorithm (it is the link cost used by both nodes during the k^{th} iteration. For fixed costs $d_{m,j}(k) = d_{m,j} \forall k$.

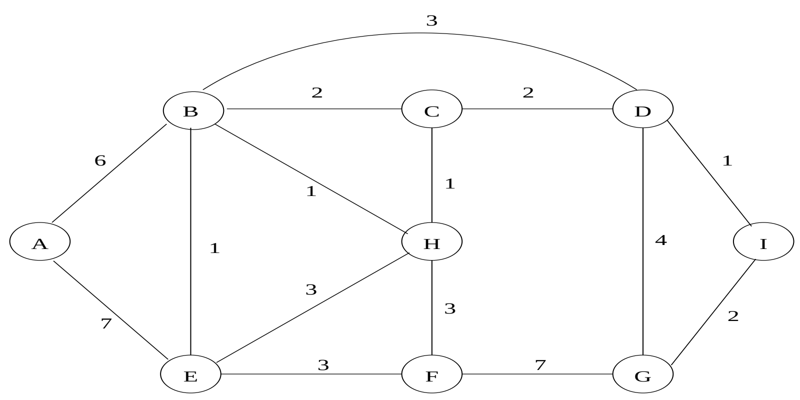

Figure 1: Sample Network with Initial Link Costs

(A) Using the notation above and assuming that link costs are fixed state the minimization step that is performed at each iteration of the DV algorithm. State it according to the following: $D_i^{(m)}(k+1) = \min_{j\in W}(X(\alpha), Y + Z(\beta))$ (identify $\{W, X, Y, Z, \alpha, \beta\}$)

Solution

$$D_i^{(m)}(k+1) = \min_{j\in N_m}(D_i^{(m)}(k), d_{m,j} + D_i^{(j)}(k))$$

(B) Assume the link costs are fixed based on the values in the figure. Use the DV algorithm to compute the routing table for node A. Specifically, determine the node ID, total cost, and next hop for the shortest cost path from node A to all other nodes. It *may* be easier to look at the problem as finding the shortest path from all node back to node A since that involves only a single destination.

Solution

Here is the routing table for node A (dst, cost, next-hop):
(A,0,-),(B,6,B),(C,8,B),(D,9,B),(E,7,A or E),(F,10,E),(G,12,B),(H,7,B),(I,10,B).

(C) How many iterations did it take for the algorithm to terminate—converge to shortest paths?

Solution

There is the initialization stage 0 (not an iteration) and 5 *iterations, wherein there are no changes during the last iteration.*

(D) In this case assume that some of the link costs vary dynamically. When computing path costs always use the most recent link cost, i.e. in iteration k use $d_{m,j}(k)$ in the minimization step. Again, use the DV algorithm to compute the routing table for node A. The following list enumerates the changing link values. All other links are assumed to keep their fixed cost (from the figure):

1. $d_{A,B}(k+1) = \min(2, d_{A,B}(k) - 1)$
2. $d_{D,I}(k+1) = \min(5, d_{D,I}(k) + 1)$
3. $d_{F,G}(5) = \infty$

(E) Did any temporary routing table loops form during the convergence of the algorithm? Justify your answer as follows: for each iteration show for each destination that the path (induced by the routing table entries) is either loop-free or includes at least one cycle.