CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

Problem	Points	Lost	Gained	Running Total	TA
1	1				
2	24				
3	15				
4	20				
5	20				
6	20				
Total	100				

You may ask for clarification but you are ultimately responsible for the answer you write on the paper. If you make any assumptions state them.

Please look through the entire test before starting. WE MEAN IT!!!

NOTE: M = 10^6 K = 10^3 Mi = 2^{20} Ki = 2^{10}
Illegible answers are wrong answers.

Show your work in the space provided to get any credit for problem-oriented questions.

Good luck!

1. (1 point, 1 min) (don't worry you get 1 point regardless of what you say!)
The last time Jackets beat the Bulldogs in football
(a) 2013
(b) 2012
(c) 2008
(d) 2009
(e) 1999
(f) at the time of big bang

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

Cache

2. (24 points, 20 min)

a) (3 points) A page fault occurs for a process when the CPU cannot find a page table entry in the page table to translate the virtual address generated by the process. Fill in the table below indicating by writing "hardware" or "software" as to who is responsible for dealing with the events shown to continue with the execution of the process:

Event	**Who handles it?**
TLB Miss	hardware +1
Cache Miss	hardware +1
Page Fault	Software +1

b) (6 points)

Given:

C0	C1

- 8 total cache blocks
- 2-way set-associative organization
- Cache initially empty
- LRU replacement policy
- Memory blocks A, B, C, and D all map to the same cache line.
- The processor performs a total of eight accesses, to memory blocks A, B, B, A, C, B, A, and C, in that order. For each of these accesses, specify (by filling in the table below) whether it is a cache hit or a cache miss, type of miss (cold/capacity/conflict), and the memory block evicted (if any). **Note: capacity miss dominates over conflict; cold dominates over capacity**.

Memory Access	Hit/miss	Type of miss	Block evicted from cache
A	Miss	cold +0.5	-
B	Miss	cold +0.5	-
B	Hit +0.5	-	-
A	Hit +0.5	-	-
C	Miss	Cold +0.5	B +0.5
B	Miss	Conflict +0.5	A +0.5
A	Miss	Conflict +0.5	C +0.5
C	Miss	Conflict +0.5	B +0.5

(Area for rough work)

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

c) (2 points) Spatial locality suggests that… (circle the correct choice)

(i) Once brought into the cache, we should keep the data around as long as possible

(ii)On a miss, we should bring in adjacent memory locations into the cache

(iii) The memory location being brought in due to a miss is not likely to be referenced in the future

(iv) None of the above

d) (2 points) Temporal locality suggests that… (circle the correct choice)

(i) Once brought into the cache, we should keep the data around as long as possible

(ii)On a miss, we should bring in adjacent memory locations into the cache

(iii) The memory location being brought in due to a miss is not likely to be referenced in the future

(iv) None of the above

e) (2 points)Drop in miss rate with increasing blocksize of the cache stops after a point dues to (circle the right choice)

(i) bugs in the cache implementation

(ii) the cache miss penalty

(iii) the changes in the working set of the program

(iv) the number of stages in the pipelined processor

f) (2 points) Virtually indexed and physical tagged cache is attractive because (circle the correct choice)

(i) it results in a better cache hit ratio for a given cache organization

(ii) it eliminates the memory aliasing problem with physically indexed physically tagged caches

(iii) it enables building bigger first level caches than physically indexed and physically tagged caches

(iv) it removes the address translation through the TLB out of the critical path of the cache access

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

g) (2 points)Page coloring is a … (circle the correct choice)

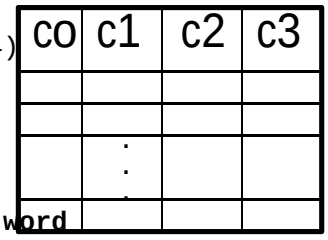
(i) type of page replacement algorithm

(ii) technique for increasing the hit rate of the first level cache

(iii) technique to allow the hardware to increase the size of the first level cache

(iv) technique to reduce the TLB lookup time

h) (5 points) Consider a **4-way set associative** cache:



- Total **data size** for the cache = **256 KiB** (note: Ki = 1024)
- CPU generates 32-bit byte-addressable memory addresses
- Each memory **word** consists of **4 bytes**
- The cache **block size** is **64 bytes**
- The cache has **one valid bit per block**
- The cache uses **write-back** policy with **one dirty bit per word**
- The cache **protects the MOST RECENTLY USED** cache block from being replaced

Answer the following questions (**you have to show your work for ANY credit**):

(i) How many index bits are needed for this cache organization?
Total number of cache blocks = data size of cache/block size
= 256 KiB/ 64 B = 4 Ki = 4096
Number of cache lines = number of blocks/associativity
= 4096/4 = 1024
Number of index bits needed = $\log_2$(number of cache lines) = **10** +1

(ii) How many tag bits are needed in each cache block?
Number of bits for block offset = $\log_2 64$ = 6
Number of tag bits per block = (32-10-6) = **16** +1

(iii) How many dirty bits are needed in each cache block?
Number of dirty bits per block = **16** (one per word in a block) +1

(iv) How much meta data is needed per cache block?
Total number of bits of metadata per block
= Tag + valid + dirty = 16+1+16 = **33** +1

(v)How much meta data is needed per cache line?
Number of bits for preserving MRU block in a line = $\log_2 4$ = **2**
Total number of bits of metadata per cache line =
= number of blocks in a cache line * 33 + MRU
= 4 * 33 + 2 = **134** +1

Each part all or nothing

No double jeopardy

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

Input/Output and Disk

3. (15 points, 20 min)

a) (2 points) Memory mapped I/O is a … (circle the right choice)

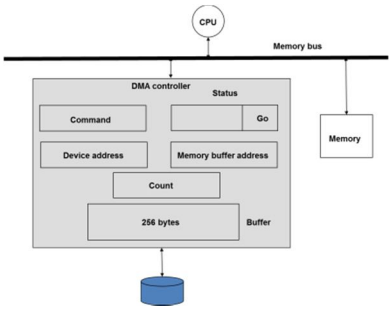
(i) technique for interfacing slow speed devices to the CPU

(ii) technique for interfacing high speed devices to the CPU

(iii) technique that allows the CPU to use Load/Store instructions to access the device registers

(iv) technique that allows the CPU to quickly find the location of the handler code for a device

b) (3 points) What is the role of the buffer in a DMA controller?



+1 - Controller communicates synchronously with the disk (streams data to/from)

+1 - Controller communicates asynchronously using the memory bus (moving a block of data at a time)

+1 - The buffer in the controller is there to smooth this dichotomy of a synchronous device and memory asynchronous bus.

c) (2 points) Zoned bit recording … (circle the right choice)

(i) has the same number of sectors on all the tracks

(ii) has more sectors on the outer tracks compared to the inner tracks

(iii) has more sectors on the inner tracks compared to the outer tracks

(iv) has different number of sectors on different tracks of the same cylinder

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

d) Given the following specification for a disk drive:

Average seek time	=	8 ms
Rotational speed	=	12000 RPM
Platters	=	2
Surface per platter	=	2
Tracks per surface	=	1024
Sectors per track	=	256
Recording density	=	256 bytes per sector

(i) (1 point)How much time is needed to get to a random sector on the disk?

Time to get to a random sector = seek time + average rotational latency
= 8 ms + 0.5 * 1/rotational speed
= 8 ms + 0.5 * 60 * 1000 /12000 ms
= 10.5 ms

+1

(ii) (1 point) How much time is needed to read one random sector from the disk when the head is already positioned on the desired sector?

+1 Time for one revolution = 60*1000/12000 = 5 ms
Time to read one sector = time for one revolution/# of sectors per track
= 5/256 ms

(iii) (2 points) If the disk gets a request to read 6 random sectors, how much total time will that request take to complete?

+1 Time to read one random sector = time to get to sector + time to read
= 10.5 ms + 5/256 ms
+1 Time to read 6 random sectors = 6 * (10.5 + 5/256) ms
= 6 * (10.5+0.019) ms
= 63.18 ms

(iv) (2 points) If the disk gets a request to read 6 consecutive sectors, how much total time will that request take to complete?

Time to read 6 consecutive sectors +1 +1
= time to get to first sector + time to read 6 sectors
= 10.5 ms + 6 * 5/256
= 10.617 ms

(v) (2 points) What is the transfer rate of the disk?

Transfer rate = number of bytes in one track/rotational latency
= (256 * 256 bytes / 5 ms)
+2 = 13.1 MB/Sec

All or nothing for each part
No double jeopardy

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

File Systems

4. (20 points, 20 min)

a)

Notes:

- Unix "**touch <file>**" command creates a zero byte new file or updates the timestamp of the named file
- Unix "**ln <file1> <file2>**" command creates a hard link
- Unix "**ln –s <file1> <file2>**" command creates a sym link
- Unix "**rm** <file>" removes the named file

(i) (8 points) In the following table, assume **none of the files exist to start with** in the current directory. Fill in the table. The reference count in the table pertains to the i-node that is affected by the command in that row. If a new i-node is created, show the old reference count for that i-node as 0.

Command	New i-node created (yes/no)	Reference count old	new
touch f1	Yes	0	1
touch f2	Yes	0	1
ln –s f2 f3	Yes	0	1
ln –s f1 f4	Yes	0	1
ln –s f4 f5	Yes	0	1
ln f4 f6	No	1	2
rm f4	No	2	1
rm f2	No	1	0

Use this area for rough work for this question.

+1 for each correct row

–0.5 for each incorrect slot in the table

(ii) (2 points) After all the above commands are executed which file names will result in successful access?

+1 +1

Only file names f1 and f6 are valid after the above commands.
Since f4 is removed, f5 won't work
Since f2 is removed, f3 won't work

–0.5 for each incorrect file named for successful access

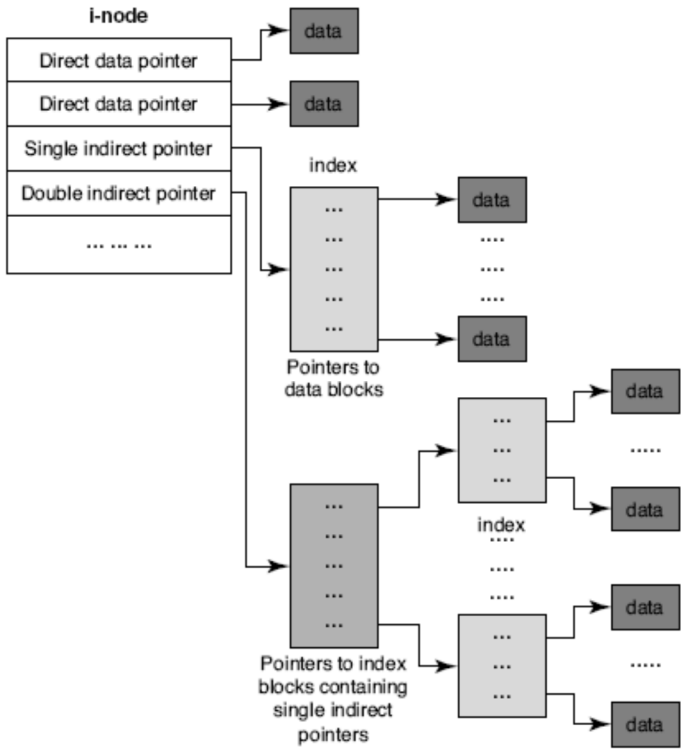
CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

b) (**NOTE Mi = 2^{20}**)

Given the following:
Size of index block = 256 bytes
Size of Data block = 1024 bytes
Size of pointer = 8 bytes (to index or data blocks)
The i-node consists of
2 direct data block pointers,
1 single indirect pointer, and
1 double indirect pointer.

Note that the index blocks and data blocks are allocated on a need basis. An index block is used for the top-level i-node as well as for the index blocks that store pointers to other index blocks and data blocks (see Figure).



i. (2 points) How many pointers does each index block contain?
Num pointers = size of index block / size of pointer
= 256/8 = 32 pointers +2

ii. (2 points) How many data blocks are used to store a 40 KiB file?
Num data blocks = size of file/size of data block
= 40 KiB/ 1 KiB
= 40 data blocks +2

iii. (2 points) How many index blocks (including the i-node for the file) are needed to store a 258 KiB file?
We need totally 258 data blocks for the file
- i-node gives 2 direct blocks +0.5
- single indirect index node gives 32 data blocks +0.5
- double indirect index node can hang 32 single level index blocks +0.5
- The number of single indirect index nodes needed for the remaining 224 data blocks = 224/32 = 7 +0.5

So total number of index blocks
= 1 i-node + 1 single indirect index block
+ 1 double indirect index block + 7 single indirect index block
= 10 index blocks

Name:_____Kishore_____GT Number: _____

iv. (2 points) What is the largest file size that can be supported in this file system?
Max file size in data blocks
= number of direct blocks
+ number of data blocks from single indirect
+ number of data blocks from double indirect
= 2 + 32 + 32*32
= 1058 data blocks = 1058 KiB

} +0.5
+0.5 for each term

c) (2 points) Multilevel indexed allocation results in (circle ONE choice that captures ALL the TRUE statements in the following list)

(i) External Fragmentation

(ii) Internal fragmentation

(iii) Ability to grow the file easily

(iv) Inefficiency for accessing small files

+0.5 for each of (ii) (iii) or (iv)

(v) {1 and 2}

(vi) {2 and 3}

(vii) {2, 3, and 4}

(viii) None of the above

Parallel Systems

5. (20 points, 20 mins)
a) (2 points) Deadlock (circle the right choice)

(i) Is a condition where threads are not using mutex locks

(ii) Is a condition where all the locks variables are in use

(iii) A lock variable that is dead

(iv) Is a condition where one or more threads are waiting for an event that will never happen

b) (2 points) User level threads with process level scheduling … (circle the right choice)

(i) Can take advantage of the hardware concurrency available in multiprocessor

(ii) Is impossible to implement on a true multiprocessor

(iii) Will have no performance advantage on a multiprocessor compared to a uniprocessor

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

b) Given the following code:

```
#define BUSY 1
#define NOT_BUSY 0
int res_state = NOT_BUSY; /* init */
mutex_t cs_mutex;
cond_var_t res_not_busy;

// A thread calls this function to acquire access to a shared resource
acquire_shared_resource()
{
  thread_mutex_lock(cs_mutex);  /* function provided by the OS */
    L1: while (res_state == BUSY)
            thread_cond_wait (res_not_busy, cs_mutex);  /* function provided by the OS */
    L2:
         res_state = BUSY;
  thread_mutex_unlock(cs_mutex); /* function provided by the OS */
}

// A thread calls this function to release access to a shared resource
release_shared_resource()
{
  thread_mutex_lock(cs_mutex);
    res_state = NOT_BUSY;
    thread_cond_signal(res_not_buys);
  thread_mutex_unlock(cs_mutex);
}
```

(i) (3 points) In the statement labeled L1, why do we need a "while" loop?

A thread that wakes up from the thread_cond_wait should ensure that the predicate which it wants to be true to execute the statement at label L2 is valid. This predicate is res_state = NOT_BUSY.

+2

+1

(ii) (2 points) If a thread is at label L2 (i.e., fallen through the while loop) what are the invariants?

It has cs_mutex and res_state is NOT_BUSY

+1 +1

(iii) (2 points) Who ensures that the invariants are satisfied when a thread is at label L2?

OS ensures that the thread has cs_mutex when it schedules after the cond_wait. +1

The program has to ensure on its own that res_state is NOT_BUSY. +1

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

c) (4 points) In the following code, buflock, bufavail,and frame_buf are shared variables. Each of the functions (digitizer and tracker) are executed by two distinct threads. What is the problem (if any) with the following code?

```
digitizer()
{
 image_type dig_image;
 int tail = 0;

 loop {
  grab(dig_image);
  thread_mutex_lock(buflock);
   while (bufavail == 0) do nothing;
  thread_mutex_unlock(buflock);
  frame_buf[tail mod MAX] =
                    dig_image;
  tail = tail + 1;
  thread_mutex_lock(buflock);
   bufavail = bufavail - 1;
  thread_mutex_unlock(buflock);
 }
}
```

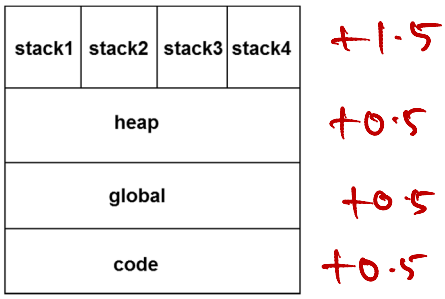
```
tracker()
{
 image_type track_image;
 int head = 0;

 loop {
  thread_mutex_lock(buflock);
   while (bufavail == MAX) do nothing;
  thread_mutex_unlock(buflock);
  track_image = frame_buf[head mod
                          MAX];
  head = head + 1;
  thread_mutex_lock(buflock);
   bufavail = bufavail + 1;
  thread_mutex_unlock(buflock);
  analyze(track_image);
 }
}
```

The arrows show the problem:

1. Tracker could end up waiting for mutex lock (red arrow) while the digitizer is spinning for bufavail to become non-zero holding the mutex lock. +2
2. Digitizer could end up waiting for mutex lock (purple arrow) while the tracker is spinning for new work holding the mutex lock. +2

d) (3 points) Pictorially show the memory footprint of a multithreaded process.



+1.5
+0.5
+0.5
+0.5

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

e) (2 points) Shown below is code for implementing mutual exclusion lock.

```
lock(L):
    L1: if (L == 0) L = 1;
        else
            while (L == 1); /* spin */
            go to L1;

unlock(L):
    L = 0;
```

If you have only atomic read and write instructions, will the above code work for implementing mutual exclusion lock? If not, why not?

It will not work. +0.5

The sequence "if (L == 0) L = 1" amounts to a minimum of three machine instructions:

read of L into a register;
Tst register to see if it is 0;
Write 1 into L (if L was originally 0);

+0.5

+1

The above 3 instructions have to be executed "atomically" for the correctness of the mutual exclusion lock algorithm.

Networking

6. (20 points, 20 min)

a) (2 points) In stop-and-wait protocol, what are the fundamental assumptions that make it possible to use a 1-bit sequence number?

- Packets may be lost in transit between the sender and the receiver.
- Packets will never get re-ordered between the sender and the receiver.
- Packets never get arbitrarily delayed between the sender and the receiver.

b) (2 points) The sequence number in a packet … (circle the right choice)

(i) Gives the destination address

(ii) Is needed for message reconstruction at the destination

(iii) Assures the integrity of the packet

(iv) Is computed using cyclic redundancy check (CRC) algorithm

(v) Is the same for every packet in a given message

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

c) (2 points) In the sliding window protocol,

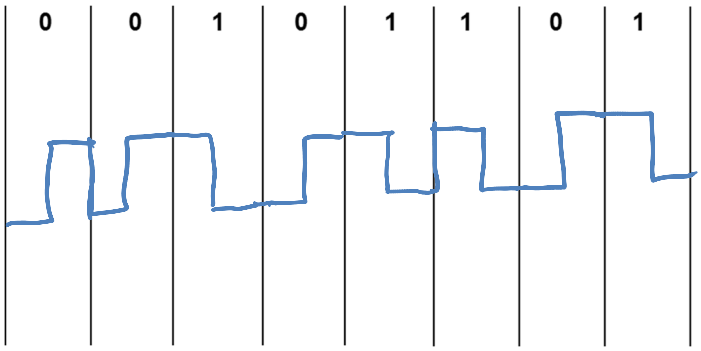
(i) when is the window size decreased?

When there is congestion observed in the network (no timely acks for packets, too many retransmissions) +1

(ii) when is the window size increased?

When the congestion eases (no loss of packets, no retransmissions) +1

d) (4 points) Show the wave form for the following packet with Manchester encoding (the space between the vertical lines represent time per bit):



+0.5 for each bit

e) (2 points) Token ring … (circle the correct choice)

(i) Is as collision prone as Ethernet

(ii) Uses IP addresses

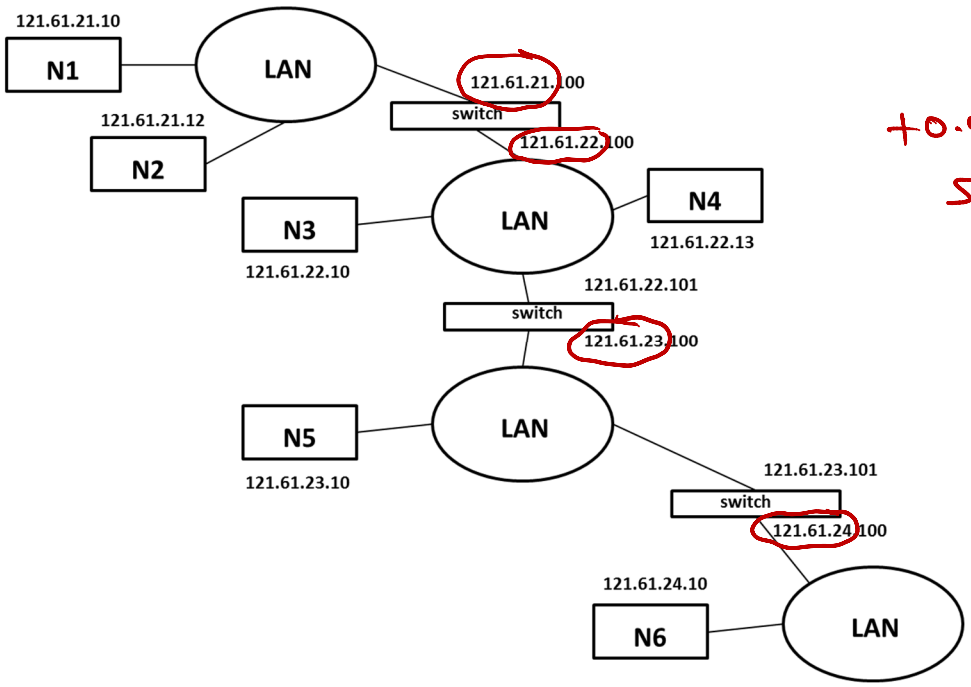
(iii) Results in better throughput under high load compared to Ethernet

(iv) Results in less average latency per transmission compared to Ethernet

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

f) (2 points) How many IP networks in the following figure?



+0.5 for each subnet

Four

g) (3 points)
Given the following:

Message size	=	10,000 bits
Bandwidth on the wire	=	100,000 bits/sec
Time of flight	=	10 msecs
Sender overhead	=	1 ms
Receiver overhead	=	1 ms

Compute the throughput.

Message transmission time
= sender overhead + wire delay time of flight + receiver overhead
= 1 ms + ((10000 * 10^3)/100,000) ms + 10 ms + 1 ms
= 1 ms + 100 ms + 10 ms + 1 ms
= 112 ms

+1 +1

Throughput
= bits transmitted/transmission time
= 10000 * 10^3/112 bits/sec
= 89.285 K bits/sec

+1

CS 2200 Fall 2013 Final Exam 8 AM to 10 AM

Name:_____Kishore_____GT Number: _____

h) (3 points)
Given the following:

Message size	=	90,000 bits
Header size per packet	=	1000 bits
Packet size	=	10,000 bits
Bandwidth on the wire	=	400,000 bits/sec
Time of flight	=	2 secs
Window size	=	10
Sender overhead	=	0
Receiver overhead	=	0
Size of ACK message	=	negligible (take it as 0)

Assuming a 10% packet loss on DATA packets (no loss on ACK packets), how many total DATA packets are transmitted by the sender to accomplish the above message delivery?

Payload in each packet
= packet size – header size
= 10000 – 1000 bits = 9 K bits] +1

Number of packet needed to complete the transmission (without data loss)] +1
= 90000/9000 = 10 packets

Sent	Received
10	9
1	1

] +1

Total number of packets sent = 11