LiTH, Linköpings Tekniska Högskola
IDA, Institutionen för Datavetenskap
Prof. Dr. Christoph Schuba

Final Exam & Answer Key

TDTS04/TDTS43 (2007 - vt1)
Computer Networks and Distributed Systems

Thursday, March 15, 2007
8:00 - 12:00 (Linköping)

Student ID/name:_____

Question #	*Max. points*	*Actual points*
1	4	*1+1+1+1*
2	8	*3+2+3*
3	5	*1+1+1+2*
4	6	*6*
5	3	*3*
6	6	*4+2*
7	2	*2*
8	6	*2+1+1+2*
Total:	*40*	*40*

Instructions:

1. Write your name and ID# onto **every** page. Do this **now**!

2. Check this exam for completeness. There should be 19 **numbered** pages total.
3. Answers can be given in English or Swedish.
4. Read all questions very carefully and only answer the questions**!**
   Do not waste time by writing down unrelated text!!!
5. You are **not allowed** to use any helping *materials* such as books, laptops, lecture notes, neighbors, etc. with the following exceptions:
   - English/Swedish dictionary
   - Basic (non-programmable) pocket calculator
6. Write your answers into the space provided. If that's not enough, use the reverse side or the extra pages in the back and make it clear which question you are answering.
7. Breaking the rules above will be considered cheating and will be dealt with according to University policy.

8. Have you written your name or student ID onto **EVERY** page?

1. TCP - Congestion Control (5p.)

(a) Describe what **TCP slow start** means. (1 p.)

ANSWER:

When a TCP connection begins, the value of the congestion window is typically initialized to 1 maximum segment size (MSS). Because the available bandwidth to the connection may be much larger, the TCP sender increases its rate of sending exponentially by doubling its value of the congestion window rather than increasing it linearly Once the first packet loss is detected , the congestion window is cut in half and subsequently grows linearly.

(b) When a TCP sender receives a **triple duplicate ACK**, its congestion window (`CongWin`) is cut in half, then grows linearly. However, when a TCP sender experiences a **timeout** event, it employs a new slow start.
Explain the philosophy behind this different treatment of observed data loss. (1 p.)

ANSWER:

Triple duplicate ACKS indicate that the network is capable of delivering at least some segments, while a timeout indicates a much more alarming congestion scenario. Therefore, the TCP sender backs off in the latter scenario much more aggressively.

(c) Explain the terms **additive increase** and **multiplicative decrease** in the context of TCP congestion control. (1 p.)

ANSWER:

TCP modifies its transmission rate (i.e., window size) probing for usable bandwidth until loss occurs. In additive increase, TCP increases the `CongWin` by 1 MSS every RTT until loss is detected. In multiplicative decrease, TCP cuts its `CongWin` in half after loss is observed. This results in a saw tooth behavior while probing for bandwidth.

(d) Explain what will happen when two applications (one using TCP, the other using UDP) that want to send as much data as possible *compete* for bandwidth. (1 p.)

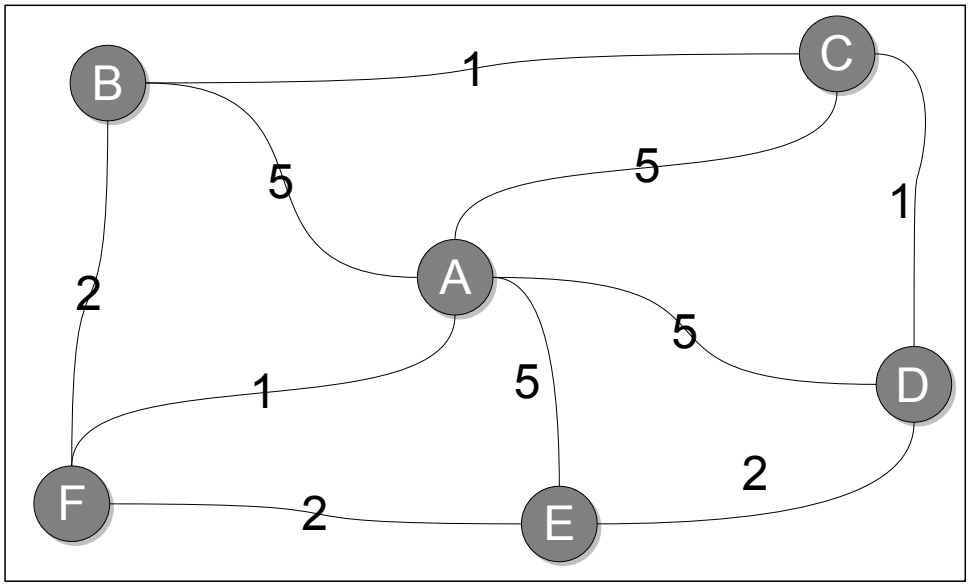
ANSWER:

Because TCP adjusts to the amount of resources that are available and UDP sends as much data as possible, the TCP connection will back off, giving up almost the entire network bandwidth to the application using UDP.

2. Routing (8 p.)

(a) Link State Routing (3 p.)

Complete the following table, using Dijkstra's algorithm.
Compute the shortest path from node A to all network nodes.
Note: Possible ties **must be** broken in favor of the leftmost column.



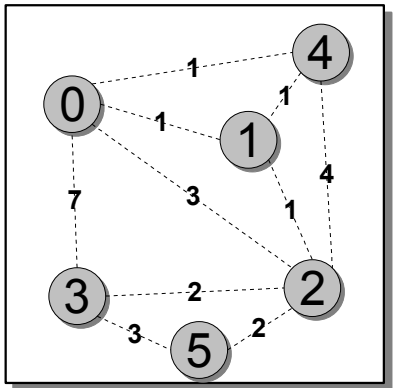
Step	*N'*	*D(B),p(B)*	*D(C),p(C)*	*D(D),p(D)*	*D(E),p(E)*	*D(F),p(F)*
0	A					
1						
2						
3						
4						

ANSWER:

Step	*N'*	*D(B),p(B)*	*D(C),p(C)*	*D(D),p(D)*	*D(E),p(E)*	*D(F),p(F)*
0	A	5, A	5, A	5, A	5, A	1, A
1	AF	3, F	5, A	5, A	3, F	done
2	AFB	done	4, B	5, A	3, F	
3	AFBE		4, B	5, A	done	
4	AFBEC		done	5, A		

(b) Distance Vector Routing (2 p.)

Consider the network shown below and assume that each node initially knows the costs to each of its neighbors. Use the distance vector algorithm and complete the entire distance table below as it would look like at node ***2*** after the algorithm has converged.
Make sure to consider the Poison Reverse algorithm in your calculation!! That means some of the entries might have the cost ∞!



Distance Vector table at node 2		**Cost to destination node**					
		0	***1***	***2***	***3***	***4***	***5***
Distance Vector from neighbor	***0***						
	1						
	3						
	4						
	5						

ANSWER:

Distance Vector table at node 2		**Cost to destination node**					
		0	***1***	***2***	***3***	***4***	***5***
Distance Vector from neighbor	***0***	0	1	2	4	1	4
	1	1	0	∞	∞	1	∞
	3	∞	∞	∞	0	∞	3
	4	1	1	2	4	0	4
	5	∞	∞	∞	3	∞	0

(c) Comparison Link State and Distance Vector algorithms (3 p.)

Compare the link state and distance vector algorithms with respect to the following categories. Complete the table below.

Criterion	***Link State Algorithm***	***Distance Vector Algorithm***
Message complexity		
Speed of convergence		
Robustness (what happens if router malfunctions?)		- Any node can advertise incorrect path cost - Each node's table is used by others. Therefore, errors propagate through the network!

ANSWER: (give ½ point for each correct entry.)

Criterion	***Link State Algorithm***	***Distance Vector Algorithm***
Message complexity	with n nodes, E links: O(nE) messages sent	- exchange between neighbors only - convergence time varies
Speed of convergence	- O(n2) algorithm requires O(nE) messages - may have oscillations	- convergence time varies - may create routing loops - count-to-infinity problem
Robustness (what happens if router malfunctions?)	- node can advertise incorrect link cost - each node computes only its *own* table	- Any node can advertise incorrect path cost - Each node's table is used by others. Therefore, errors propagate through the network!

3. Network Address Translation and Peer-to-peer Computing (5 p.)

Suppose a peer with user name Alice discovers through querying that a peer with user name Bob has a file she wants to download. Also suppose that Bob is behind a NAT whereas Alice is not. Let 138.76.29.7 be the WAN-side address of the NAT and let 10.0.0.1 be the internal IP address for Bob. Assume that the NAT is not specifically configured for the P2P application.

(a) Explain why Alice's peer cannot initiate a TCP connection to Bob's peer, even if Alice knows the WAN-side address of the NAT, 138.76.29.7. (1 p.)

ANSWER:

Consider what happens when Alice attempts to establish a TCP connection with Bob.. Alice might send a TCP SYN packet with destination address 138.76.29.7 and some destination port number, say, x. When the NAT receives this TCP SYN packet, it doesn't know to which internal host it should direct the packet, since it doesn't have an entry for a connection initiated from the WAN side. Thus the NAT will drop the SYN packet.

(b) Now suppose that Bob has established an ongoing TCP connection to another peer, Cindy, who is not behind a NAT. Also suppose that Alice learned from Cindy that Bob has the desired file and that Alice can establish a TCP connection with Cindy.

Describe how Alice can use these two TCP connections (B-C and A-C) to instruct Bob to initiate a direct TCP connection (without passing through Cindy) back to Alice. (1 p.)

ANSWER:

There is an existing TCP connection between Alice and Cindy and between Cindy and Bob. Via these two TCP connections, Alice can send a message to Bob. In particular, Alice can ask Bob to initiate a direct TCP connection from Bob to Alice. Since Bob is initiating this TCP connection, it can be established through Bob's NAT. Once this direct TCP connection is established between Alice and Bob, Alice can ask Bob to send the file over the direct TCP connection.

(c) Suppose both Alice and Bob are behind different NATs. Is it possible to devise a technique that will allow Alice to establish a TCP connection with Bob without application-specific NAT configuration? Explain your answer. (1 p.)

ANSWER:

It is not possible to devise such a technique. In order to establish a direct TCP connection between Alice and Bob, either Alice or Bob must initiate a connection to the other. But the NATs covering Alice and Bob drop SYN packets arriving from the WAN side. Thus neither Alice nor Bob can initiate a TCP connection to the other if they are both behind NATs.

(d) Give at least two arguments why Network Address Translation technology is still quite controversial, in spite of having been adopted widely. (2 p.)

ANSWER:

Possible answers include:

- Routers should process only up to layer 3 - layer violation
- NAT violates the end-to-end argument, because application designers must take into account that NAT boxes might be in the communications path between participating applications.
- IP address shortage should have been solved with IPv6 instead of NATs

4. Link Layer (6 p.)

Describe the remaining six services provided by the Link Layer. The description for *Framing* serves as an example for the level of detail you are expected to provide in your answers.

Link Layer Service	***Description***
Framing	Almost all link layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header or trailer fields. The structure of the frame is defined by the link-layer protocol.
Link access	
Reliable delivery	
Flow control	

Link Layer Service	***Description***
Error detection	
Error correction	
Half-duplex and full-duplex	

ANSWER:

Link Layer Service	*Description*
Framing	Almost all link layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header or trailer fields. The structure of the frame is defined by the link-layer protocol.
Link access	A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple (or nonexistent) - the sender can send the frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link - the so-called multiple access problem. Here the MAC protocol serves to coordinate the frame transmissions of the many nodes.
Reliable delivery	Reliable delivery service guarantees to move each network-layer datagram across the link without error. Similar to a transport-layer reliable delivery service, a link-layer reliable delivery service is achieved with acknowledgments and retransmissions. A link-layer reliable delivery service is often used for links that are prone to high error rates, such as a wireless link, with the goal of correcting an error locally - on the link where the error occurs - rather than forcing an end-to-end retransmission of the data by a transport or application layer protocol.
Flow control	The nodes on each side of a link have a limited amount of frame buffering capacity. This is a potential problem, because a receiving node may receive frames at a rate faster than it can process them. Without flow control, the receiver's buffer can overflow and frames can get lost.
Error detection	A node's receiver can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduced by signal attenuation or electromagnetic noise. Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism to detect the presence of one or more errors. This is done by having the transmitting node set error-detection bits in the frame, and having the receiving node perform an error check.
Error correction	Error correction goes one step beyond error detection in that it detects not only that an error has occurred, but exactly where in the frame it has occurred and then correct these errors. Some protocols provide error correction only for packet header/trailer fields.
Half-duplex and full-duplex	With full-duplex transmission, the nodes at both ends of a link may transmit packets at the same time. With half-duplex transmission, a node cannot both transmit and receive at the same time.

5. Traceroute (3 p.)

Explain in detail how the traceroute program works. (3 p.)

ANSWER:

Traceroute works by increasing the *time-to-live* value of each successive batch of ICMP Echo Request packets sent. The first three packets have a time-to-live (TTL) value of one (implying that they make a single hop). The next three packets have a TTL value of 2, and so on. When a packet passes through a host or router, normally the TTL value is decremented by one, and the packet is forwarded to the next host. When a packet with a TTL of one reaches a host, the host discards the packet and sends an ICMP time exceeded (type 11) packet to the sender. The traceroute utility uses these returning packets to produce a list of hosts that the packets have traversed en route to the destination. Traceroute may not list the real hosts, it indicates that the first host is at one hop, the second host at two hops. IP does not guarantee that all the packets take the same route. On modern Unix and Linux-based operating systems, the traceroute utility by default uses UDP datagrams with a high destination port number that is unlikely to be in use.

6. Network Security (6 p.)

(a) When we ask what it means for entities to communicate securely, we specify a number of desired properties, called *security goals*. Fill in the blanks below, specifying the five security goals together with a description of what they mean. Be as precise as possible. (4 p.)

Network Security Goals	***Description***

ANSWER:

Grading guide:

- Because this is a repeat question from the last exam, we removed the example entry.
- The first two correctly filled in rows score 1/2 point each.
- Every additional correctly filled in row scores 1 point each.

Network Security Goals	***Description***
Confidentiality	The concept that only the sender and intended receiver should be able to understand the contents of a transmitted message.
Authenticity	The concept that communicating parties should be able to confirm the identity of the other party involved in the communication, to confirm that indeed the other party is who or what they claim to be.
Integrity	The concept that any modification to messages/data by unauthorized entities can be detected; in other words, messages should be received exactly as they were sent, with nothing missing, nothing added, nothing changed.
Availability	The concept captures that two or more parties need to be able to communicate in the first place, e.g., resources, such as a network service, should be accessible. Counterpart to Denial-of-Service.
Non-repudiation	The concept of ensuring that a contract, especially one agreed to via the Internet, cannot later be denied by one of the parties involved.

(b) Message authenticity, integrity, and privacy. (2 p.)

Suppose both Alice and Bob have their own public/private key pairs. Furthermore, assume that both know each others' correct public keys and that their private keys are not compromised.

Now, Alice wants to send a message *m* to Bob and wants to make sure that its authenticity, integrity, and confidentiality are assured.

Alice sends $\{\{m\}K^{+}_{B}\}K^{-}_{A}$ to Bob. (That is, she first encrypts *m* with Bob's public key, then with her own private key, before sending the result to Bob.)

Explain if this approach accomplishes Alice's stated security goals.
If it doesn't do so, suggest a better approach and explain why it is better.

ANSWER:

The given protocol does not protect the authenticity of the message, because anyone who has Alice's public key can remove the outer layer of encryption from the message, including Trudy. Trudy could then re-encrypt the result with a different private/public key pair. The message would then look like it had originated with Trudy, not with Alice.

To correct the problem, simply apply the two encryption steps in the opposite order, i.e., first encrypt *m* with Alice's private key, then with Bob's public key. This is a much better approach, because only Bob can unwrap the outer layer of encryption.

Alice would therefore send $\{\{m\}K^{-}_{A}\}K^{+}_{B}$ to Bob.

7. CORBA (2 p.)

Imagine that you are hired as a consultant by a high-tech company's legal department to write a client-server CORBA program that can be used by their employess to submit patent invention disclosures electronically. As part of this system you are asked to specify an interface for a routine that determines if the specified co-authors indeed valid employees of the corporation. You need to know the following:

- Employees are uniquely identified by their six digit employee id (e.g., "AB0001")
- The submitting author needs to be identified as the primary author
- It needs to be specified how many co-authors there are in total
- Each co-author's employee ID needs to be specified
- The invention disclosure needs a name
- The routine goes through the list of each submitted employee ID, attempts to validate that the employee ID is in the human resources database and returns if the lookup has succeeded or not.
- Finally, if all co-authors were successfully found in the human resources database, the routine returns a unique identifier for the new invention disclosure (e.g., "MSC07-0539")

Specify an interface description in IDL (file **InventionDisclosure.**`idl`) for the preceding client-server interaction by filling in the blanks. Use the most appropriate data types (see table below) and use argument names that make it clear what is meant. (2 p.)

```
interface InventionDisclosure
{
  _____________ checkAuthors(_____________________________________,
                             _____________________________________,
                             _____________________________________,
                             _____________________________________,
                             _____________________________________);
}
```

IDL	C++	IDL	C++
boolean	bool	double	double
char	signed char	any	(no native equivalent, similar to void *)
octet	8 bits	object	(closest to class)
enum	enum	string	class string (bounded or unbounded length)
short	short	struct	struct (like C, rather than C++)
unsigned short	unsigned short	union	union (a type-tagged C struct)
long	long	array	[ ]
unsigned long	unsigned long	sequence	(a parametrized array of bounded or unbounded length)
float	float		

ANSWER:

Grading guide: start with 2 p. Subtract ½ p. for mistakes in the return type, the first three string arguments, the string array, and the boolean array.

```
interface InventionDisclosure
{
  string checkAuthors(in string         primaryAuthor,
                      in string         numberAuthors,
                      in string         inventionTitle,
                      in string array   authorIDs,
                      out boolean array successFlags);
}
```

8. Distributed Coordination-Based Systems (6 p.)

(a) Taxonomy of coordination models (2 p.)

Complete the following taxonomy by filling in the shaded table entries.

		Temporal	
Referential			Mailbox
	Uncoupled		

ANSWER:

		Temporal	
		Coupled	**Uncoupled**
Referential	**Coupled**	Direct	Mailbox
	Uncoupled	Meeting oriented	Generative Communication

(b) What is the key idea in generative communication? (1 p.)

ANSWER:

The key idea in generative communication is that a collection of independent processes make use of a shared persistent dataspace of tuples. Processes can put any type of record into the shared dataspace and find tuples through an associative search mechanism.

(c) Explain the mechanism called *Leasing*. (1 p.)

ANSWER:

Leasing is a mechanism to manage references to objects. In Leasing, referenced objects keep track of who is referring to them, leading to what is known as reference lists. When that list becomes empty, the object can safely destroy itself.

(d) Implementation of JavaSpaces (2 p.)

Assume you have a multicomputer system for which reliable broadcasting is available. A serious candidate for an implementation architecture is to replicate all the subspaces in full on all machines. Explain what needs to happen for the WRITE, the READ, and the TAKE operations. To illustrate your answers, you may add diagrams to your explanations.

WRITE operation:

ANSWER:

When a WRITE operation is done, the new tuple instance is broadcast and entered into the appropriate subspace on each machine.

READ operation:

ANSWER:

To do a READ operation, the local subspace is searched. If the element exists, it can be read.

TAKE operation:

ANSWER:

Similar to a READ operation, the local subspace is searched. However, since successful completion of a TAKE operation requires removing the tuple instance from the JavaSpace, a delete protocol is required to remove it from all machines. To prevent race conditions and deadlocks, a two-phase commit protocol can be used.

Note: For illustrations see Figure 12-17 in the course textbook [TvS2002].

Grading guide: ½ p. each for the correct explanations for the WRITE and READ operations. 1 p. for the TAKE operation.