Ariel Stolerman

Total: 96

Computer Networks (CS544) - Final Examination

Professor Mike Kain

Due Saturday, June 15th, 2013 at 11:59pm

By turning in this exam, each one of you is explicitly making the following pledge of honesty: "I understand that this exam is to be done individually. The exam is open book (text book and course notes, and lectures – NO other materials may be consulted or used). I have taken this exam individually and have not discussed any question or examined anyone else's exam or used any other materials than documented above. I will not copy anyone else's written work, or any copyrighted work (including the text book), nor will I have anyone other than myself prepare any portion of my work. Copying of the solutions of others is expressly forbidden. I will not allow any other person to create, nor to copy, any part of my assignment handed in with my name on it."

The exam must be submitted via Drexel Learn within the 4 hour time limit.

1. Name and describe the 7 goals of security and give an example of each goal in a network protocol (15 points).
2. The two major classes of networks are virtual circuits and datagrams. Explain the basic operation of each (how each does routing, addressing, segmentation & reassembly, flow control, and robustness). The concept of a "flow" brings together these two concepts. Describe the concept of a "flow" of packets across the Internet. Give two examples of possible flows. (15 points)
3. The goal of a routing table is to help the routing algorithm find the best path to the destination. What are the three major columns of a routing table? Describe the basic operation of the three major types of routing algorithms and how they use the routing table in their operation. Include the two subtypes of adaptive algorithms (DVR and LSR). Now expand the goal of the routing algorithm to find 2 or more paths (rather than one) to every destination that are disjoint (the paths don't share any common links). What information do you now have to know about the network? Which algorithm would be better (DVR or LSR) and why? (25 points)
4. Think and describe of the major functions of the Internet Protocol (such as routing, etc.). Sublayer the IP protocol into as many sublayers as you can determine and describe each sublayer and the reason that it is a separate sublayer. Could you add congestion control as a sublayer (if IP would support that)? (10 points)
5. What is Media Access Control? Describe the concepts of contention and collision and how they affect the Media Access Control sublayer. (15 points)
6. The Internet uses a hierarchical routing model – how do we route across the Internet with IPv4? How does the addressing model used to represent this model (how is the address split into pieces which each contain a layer of a hierarchy). Describe the three different techniques to represent this hierarchical addressing in the IPv4 address. (15 points)
7. What is a security association and what does it cover? Describe two protocols which use a security association and the security association contained. Why would a network protocol use a security association? (15 points)

Answers

1)

Following is a description of the 7 goals of security, along with examples in network protocols:

1. Authentication and verification: to ensure the connecting parties are who they claim they are.
   Authentication, like outside the digital world, is done by trusting a third party that issues some type of authentication identifier. A physical world example would be a state-issued ID card (where you trust the state), DragonCard (where you trust Drexel) etc.
   An example in network protocols: Kerberos – a centralized authentication mechanism which includes an authentication and ticket issuing server; clients authenticate themselves in front of that server, which then issues them a token that contains authentication information of the client and the KDC (the authenticating server). Then the client uses the token issued to him in order to log in the server it desires. In this example, the server from the last step trusts the authentication server.
   Authentication is done in network protocols usually via keys – passwords known only to the authenticated entities. An example of a mechanism: private/public key infrastructure, used by certificates – a client can authenticate a server by its digitally signed certificate, issued by a trusted certificate authority.
2. Access Control: access control mechanisms ensure that (authenticated) entities are allowed access only to resources they are approved for. For instance, administrator permissions.
   An example in network protocols: firewalls have access control properties embedded in their design – they have the ability to filter and block access to internal network by characteristics like ip address, port, protocol etc. An example would be blocking some ports on a local network – a connecting entity may be authenticated and allowed access to other applications (other ports) on the server, but blocked for accessing a certain application - certain port.
3. Data integrity: making sure that the data hasn't been modified since it was created or issued. An example from network protocols: HMACs – cryptographic digital signatures, which involve a key and a hard (irreversible) hash function, like HMAC-SHA1. With digital signature, the sent data will be digitally signed by the sender by hashing it with such a method with a private key, or hash it and then encrypt it with the private key. On the other end, the signature can be extracted using the public key, and hashing the data can be matched to the attached hash,

making sure nothing changed. This way the integrity of the data is checked to be both unchanged since it was sent, and sent by the actual sender we expect.

4. Confidentiality: ensures that the data can be understood only by the sender and the receiver, and no one else in the middle. Confidentiality is achieved in network protocols by cryptographic algorithms, keys (shared symmetric key or private/public key infrastructure) and hash/message digest. For instance, DES/3DES/AES are types of common symmetric crypto algorithms, that can be selected for usage in SSH, e.g. by selecting RSA_WITH_AES_256_CBC_SHA1 – which means, use RSA (private/public key infrastructure) with AES for the symmetric encryption/decryption phase, 256-bit keys, CBC (block ciphers) and SHA1 message digests for crypto checksum.
5. Availability: ensures resources are always available to legitimate users who wish to use them. An example from network protocols: mechanisms laid out to prevent DoS or DDoS attacks – denial of services done by flooding the server with requests and by that crashing it / blocking it for legitimate users. Mechanisms to prevent DoS attacks can include filtering and blocking susceptible incoming communication (e.g. blacklists), preventing more than a predefined fixed number of authentication attempts etc.
6. Non-repudiation: ensures a certain transaction is carried out once and only once, and neither the sender nor the receiver can prove that it didn't occur. An example from the real world is a purchase receipt, but in network protocols: reply attacks are examples of attacks against non-repudiation, where a middle man that listens to the communication may try to reply a message to invoke the receiver to respond twice to a client transaction. Network protocols use mechanisms like sequencing (attaching sequence numbers), like done by TCP, in order to prevent these types of attacks.
7. Trust: this goal means that each protocol has to have an inbuilt an idea of trust with the other devices in the protocol. The more a communication with a certain party occurs, the more the trust in that party evolves. An example from network protocols (although not necessarily a good one): accepting authentication of a server that we first login to via SSH. Upon initial connection, the user is asked whether he trusts the server – and may select yes and remember the selection for the next time. This way the trust is embedded in the application, so that next time we know the server can be trusted and immediately connect to it (the initial user decision may be dumb; however this example does illustrate trust).

Excellent 15/15

2)

Following is an explanation of the basic operation of virtual circuits and datagrams:

Virtual circuits:

- Routing: a fixed path is configured in advance, before any data is transferred. Then, all data is communicated over the configured path – the virtual circuit.
- Addressing: it follows from the routing method that addressing is done during the circuit configuration phase, and all packets follow that addressing.
- Segmentation & reassembly: all routers along the virtual circuit are predefined during setup, therefore segmentation and reassembly can be performed at every one of them.
- Flow control: flow control is possible, since all points in the circuit are predefined we can propagate a "slow down" message, or retransmit lost packets from sources along the path.
- Robustness: since all routers along the path are predefined, the path is as good as the nodes it consists of – so if a router goes down, the path needs to be redefined. Therefore robustness is not as good as with datagrams (as seen next).

S&R only done at sender and recv

4.5/5 for VC

Datagrams:

- Routing: unlike VCs, here no path is predefined, and each packet is routed independently of the others.
- Addressing: since no path is predefined with datagrams, each packet has to have the addressing information – similar to mailing letters via US-mail (each letter contains the address).
- Segmentation & reassembly: having no predefined path means the intermediate routers are unknown, therefore operations like segmentation and reassembly can be done only at the source or destination.
- Flow control: applied only by the source/destination; unlike VCs, here if a packet gets lost, for instance, it must be retransmitted from the source.
- Robustness: better than VCs, since if a bad node is encountered along the way, another path is chosen to bypass it – no predefined path that relies on the routers it contains. Therefore datagrams are much more robust than VCs and have better durability for router failures.

Segmentation can be done anywhere, but reassembly only at receiver

Flows:

4.5/5 for DG

Flows are used to get the best of both worlds – VCs and DGs. The informal definition of a flow is: all packets going in the same direction. That is, packets that are all sent to the same destination and

share some part of the network. This is applied on datagrams: we look for commonalities across datagrams so we can group similar datagrams together, so that for the period of transfer they all share some information – essentially addressing a group of datagrams as a virtual circuit, to some extent. So flows are basically groups of datagrams that follow the same path, and therefore treated as a virtual circuit; they may be valid only for parts of the network, and not necessarily from source to destination.

Following are two examples of possible flows, based on source/destination ip/port and protocol:

- (*,*,127.0.0.1,8080,*) – a flow defined by a filter that takes all communication to localhost on port 8080 (e.g. a local webserver)
- (*,*,*,*,UDP) – all UDP communic

Somewhat. Using loopback means that they don't share any of the same path through the network (unrouted) - and how is all UDP one flow? 4 out of 5 for flow. 13/15 for Q2.

3)

The three major columns of the routing table are:

- Destination – a description of the destination address, or in general – the network id, usually derived from an ip address along with a netmask.
- Gateway – the actual node to point the data to, the gateway through which the data needs to go to get to the desired destination.
- Cost – the cost of using this path (some measurement of time/distance)

Following is a description of the three major types of routing algorithms, and how they use routing tables in their operation:

1) Static (non-adaptive) algorithms:

In static routing algorithms, data is transferred by flooding using the shortest-path algorithm (Dijkstra) to determine the routes. Flooding means that the data is passed to everyone except the node that gave it to me. Selective flooding is used to known when to stop: sequence numbers are assigned such that if a node receives a number it already propagated, it does not flood it further. Flooding makes static algorithms very wasteful.

Sel flooding is flooding to selective hosts - not using seq numbers 4/5

2) Adaptive algorithms:

With adaptive algorithms, a routing table is maintained at every node with every potential destination we care about. Some parts of the table are given to everyone, so nodes see what their neighbors see. There's a redundancy of data, however since nodes see the same information, problems can be propagated back so different decisions can be made.

Ariel Stolerman

2.1) Distance-Vector Routing (DVR):

In DVR, periodically and at certain network events, every node propagates its routing table information to its neighbors, and upon receipt the nodes recalculate their routing tables. That way each node has an updated routing table (calculated using the Bellman-Ford algorithm) with shortest paths to all relevant destinations. This algorithm may suffer from a count-to-infinity problem, where a split in the horizon and a non-updated routing table at some node may cause a loop path that tries to reach an unreachable destination through the calculated path. This problem is overcome using loop detection.

2.2) Link State Routing (LSR):

With LSR, shortest paths are calculated using global information about the network. First, each node discovers its active neighbors; then it measures the cost to those neighbors; the data it passes onward is only what it calculated, what it discovered. This data is passed on to all routing devices and then the new routing table is calculated. This method is quicker than the previous since it uses flooding, however it is therefore requires passing more data (more traffic). In addition, it doesn't scale as well since in a large network it requires figuring out a network from lots of little bits sent by the components of the network.

Dynamic 5/5, I don't see hierarchical 0/5

Expanding the algorithm:

If we now expand the routing algorithm to include not only the shortest path, but 2 or more disjoint paths, then now the information we have to know about the network is not just a routing table – but an ordered routing matrix – each previous entry is not a vector of entries, sorted by cost, with their corresponding gateway.

Out of DVR and LSR, I think DVR would be better, since it is more efficient in propagating data as it much less data is transferred over the network. With LSR, flooding is used to propagate what every node calculates locally, and this results with a lot of traffic and requires complex calculations at every node. Another advantage of DVR is the fact it reacts to "good" news fast – so if a shortest path / several paths are to be updated, this change will be propagated quickly in the network.

Overall there's a tradeoff between DVR and LSR, and similar effects we had before when dealing with just one shortest path to maintain, we have also now. However, in my opinion, the major increase in the size of the routing table (to a matrix of routes) favors DVR.

What is needed is more than a routing matrix - who is connected to whom (topology). DVR doesn't know about it - LSR already does. 2/3 for info - 4/7 for answer and reasons. 15 out of 25 for Q3.

4)

The internet protocol (IP) is a datagram based protocol that is constructed of autonomous subnetworks and does not provide reliability. IP is in fact a network of networks, which is expressed in the IP addressing scheme, that defines a hierarchical model of the internet. The major functions of IP are to provide transfer ability of data from source to destination based only on IP addresses. In addition to the addressing methods (differ between IPv4 and IPv6, but essentially consist of network and host ids), IP provides datagram structure – the basic pieces to be transferred over the network. These two functions – addressing and datagram structure – are the major functions defined by IP.

As for sublayering IP, we can divide it into two types of sublayers:

- Structural sublayers: sublayers that address the structure of the IP packet, the different components that theoretically can be separated and are independent of one another. For instance, a sublayer can be deinfed for each of:
  - Header – that contains the IP variables
  - Source address
  - Destination address
  - data

  Theoretically, we can define the source address differently than the destination address (though no logic in doing so), which means we can "plug and play" these components – which makes them suitable for a sublayer.
- Functional sublayers: sublayers that are defined by the functionality they provide, like those we can derive from the helper protocols we learned about in class:
  - ICMP – IP control messages, like echo/echo reply, source quench etc.
  - IGMP – multicast address management
  - ARP – address resolution protocol to map IP address to MAC
  - DHCP – dynamic allocation of IP addresses
  - NAT – network address translation, IP/port change

  All the classes of functions above are defined as sublayers (although they can be defined as separate layers from the internet layer) since they provide functionality for IP, and potentially can be provided different implementations – therefore suitable to be defined as sublayers (again, the "plug and play" principle).

True, but where do we maintain the routing table?
6 out of 8.

As for congestion control, IP does not provide any, and relies and that of the transport layer (like in TCP). However, if IP would have support for congestion, we could add a sublayer with flags and data to provide the functionality necessary for congestion control, similar to what's implemented in TCP. For instance, if a closed/explicit congestion control scheme is applied, this sublayer could have included, among others, functionality to send choke packets to indicate congestion to the other side.

Yes, it would be the topmost sublayer and pull it down from TCP. 2 out of 2 - total of 8 out of 10 for Q4.

5)

Media Access Control, or MAC, is used to handle the physical layer and its characteristics. These are methodologies and algorithms to deal with shared mediums, like cell-phones, Ethernet, or wireless, either statically or dynamically, as opposed to a point-to-point connection. It can be addressed as a sublayer of the data link layer. There are two major categories of MACs:

- Static: specific slots for different nodes are allocated with no overlap, which prevents collision and contention (described later). Examples: TDMA, CDMA.
- Dynamic:
    - Scheduled: a scheduling algorithm is used to divide media usage among nodes, like poll based or token based.
    - Random: with random MACs, each station has to figure out when to send without any common central station telling them. Examples: ALOHA (try to send after waiting a certain time period), CSMA.

Contention:

Contention happens when two or more stations try to claim the right to send on the media simultaneously. This affects MACs in that shared media has to be dealt with multiple requesters that wish to send over that media. Example for dealing with contention: polling.

Contention cannot occur in static MACs, however it may occur in dynamic MACs.

Collision:

Collision happens when two or more stations send data at the same time, which results in garbage (due to mixing of the data) and no parties are able to send their data properly. This affects MACs in a way similar to contention, only one step worse further – MACs have to be able to prevent collisions, since it 1) creates noise 2) uses the network unefficiently 3) requires the sending parties to resend their data anyway.

Collision cannot occur in static MACs or in scheduled dynamic MACs, however it can occur with dynamic random MACs.

Excellent. 15 out of 15.

6)

IPv4 uses hierarchical routing model, where addresses contain components that can be sorted into trees and clusters with logical (and in non-mobile networks: geographical) associations. The IPv4 address contains 32 bits, and consist of the following components. Note that these components are represented in a classful manner (as described later as one of the IPv4 addressing methods), however it doesn't have to be exactly this way, and only the two major components are described, which are shared by all IPv$ addressing methods:

- Net-id: the most significant (leftmost) bits of the address are the network id itself, which is shared by all the hosts in that network. This part is the identifier of the network within over the internet. Outside the network (i.e. on every host that has a different netid), the destination is determined by this part of the address.
- Host-id:
    - Subnet-id: this part exists only in classful with subnetting addressing scheme - with subnetting, the host is broken up into pieces – a subnetwork id and the host id.
    - Host-id: the identifier of the host itself, within the network (or subnetwork). This part is used to identify the host within the network.

Either by a predefined division into netid and hosted, or by using a subnet mask – a bitwise masking of what part of the address identify the network and subnetwork – we can identify the network and the host. For instance, the address 192.168.1.1 with subnet mask 255.255.255.0, or in short 192.168.1.1/24 means that the 192.168.1 part is the network id, and the host is 1.

In summary, routing uses the information above to route to a destination by:

- Finding the network using the network id
- Within the network, finding the host with the host id

The 3 techniques to represent the hierarchical addressing the IPv4 addresses are:

- Classful: by the classful addressing scheme, there are A-E classes where each defines a fixed number of MSBs that are constant (A – 0; B – 10; …), next a predefined number of bits that represent the network id, and finally the rest – represents the host id.

- Classful w/ subnetting: as mentioned briefly above, with subnetting a sub-hierarchy is applied by defining parts of the host-id as subnet, and the rest as the actual host id. So the components are netid, subnet-id and host-id.
- Classless Inter-Domain routing (CIDR): with CIDR the network is described as a range of addresses, and allows subdividing networks over and over, in a classless fashion – the mask bits are used to determine the network and host separation. Levels of hierarchy and subdivision of networks are applied by simply adding another bit of mask, and divide the network under that bit. For instance, 192.168.0.0/16 can be divided into 1) 192.168.0.0/17 and 192.168.128.0/17 – meaning, CIDR allows arbitrary length subnet masks and flexible network addressing.

Excellent 15/15.

7)

Security association is a mechanism that enables us to quantify the security we want to apply between two endpoints. With security association, trust can be built between two communicating parties, as the security constraints defined by the association are established and maintained. The security association is important in understanding all aspects of the security of a protocol, as it defines a complete image of:

- What to secure – what exactly we want to maintain secured
- How to secure – what methods are to be used for security (encryption, keys, hashing etc.)
- How LONG to secure – what is the period defined for the association, in which the secured communication is valid (e.g. in order to prevent usage of legitimate but outdated credentials to masquerade as a legitimate user)

Network protocols should use a security association for those reasons above – if these questions can be answered in a complete way, we made sure our protocol is secured in a satisfying manner. The security of a protocol can be crunched down to its security association, which encompasses all that related to security for that protocol.

Following is a description of two protocols that use a security association, and their security association:

SSL/TLS:

- What: a security protocol for a secured session over TCP that ensures authentication, confidentiality and integrity of data passed between the client and the server – which has to be

secured from eavesdropping / modifications. So the answer to "what" is essentially: all TCP data.

- How: the SSL/TLS handshake contains negotiation of a cipher suite (encryption algorithm, hashing, key-size, parameters etc.) and symmetric key to be used for encryption.
- How long: in SSL/TLS, the connection is secured as long as the session is valid (could be an initial session / renegotiation over an existing session, but the idea is that during an active session the security applies).

Ipsec:

- What: the "what" is defined by the policy applied, for instance (*,*,*,80,TCP) means that the answer to "what" is: all outgoing web-traffic.
- How: AH – provides authentication; ESP – provides integrity and confidentiality. These are defined by the policy to use with Ipsec, and provide the corresponding traffic security.
- How long: as long as the Ipsec policies are valid, the security is active, as the data that passes through the traffic selectors of Ipsec is conformed to the policies defined by it.

Excellent 15/15.