CS538 Take-home Submission

Siting Chang, schang13

October 8, 2014

1 Problem 1: Data Transmission in Internet

1.1 part 1

Advantages of Packet Switching v.s. Circuit Switching Transmission are:

- Packet switching is able to achieve higher utility in terms of bandwidth. The key feature of circuit switching is that it reserves a channel for a pair of hosts for a certain period of time, with and without actual data or messages being transmitted. In other words, there are times the channel sends nothing while keeping other hosts connect to each other.

  However, packet switching allows packets between various hosts to use the same link during the same session. It increases the utility by allowing messages sent between one pair of hosts to take advantage of the dull time when the other pairs are not transmitting any messages.

- Packet switching is able to resend only damaged or lost packets while circuit switching has to resend the entire message. Packets in packet switching have sequence numbers which help to identify which packet got damaged or lost, so the specific packet could be resent. However, circuit switching does not keep such sequence numbers. Therefore, if a piece of packet gets lost, the entire message or file needs to be resent.

Disadvantages of packet switching are:

- Packet switching cannot guarantee a bandwidth while the circuit switching can. The delivery under packet switching is best-effort delivery since the link is not reserved for a specific pair of hosts, which means other hosts are competing for the bandwidth as well.

- The forwarding in circuit switching would be much simpler compared to packet switching. Because a channel is reserved in circuit switching, routers visited by the transmitted messages forward messages to the same destination. Therefore, as long as a channel is alive, the routers can forward the messages to the same outgoing link without checking routing table.

However, in packet switching, packets are being forwarded to multiple destinations which requires the routers to identify which outgoing inks to send these packets. So more effort is made when forwarding packets in packet switching.

1.2 part 2

1.2.1 part a

NCP was not sufficient because it had no end-to-end host error control. To be more clear, NCP was designed only to serve ARPANET – the only existing network which was so reliable that no error control was needed, which leads to the design that NCP had no end-to-end host error control. However, when network enlarged, network reliability became one critical issue which NCP could not accommodate. Therefore, NCP was not sufficient anymore.

The missing end-to-end host error control also restricts the size of Internet. When Internet dramatically grows, the reliability of Internet decreases which introduces corrupted packets. NCP does not provide the service to identify these corrupted packets. Another missing feature of NCP is congestion control. Since growing number of computers are trying to send data packets to the network, it is easy to forecast that there will be a high potential of traffic congestion. Therefore, having some protocol to restrict the amount of data being sent into the network is critical, which NCP did not have.

1.2.2 part b

TCP/IP is a replacement protocol for NCP. The new features are:

1. TCP/IP provides reliable transmission by including a sequence number in its header. The sequence number is used to identify the order of received data packets and reconstruct the original message/file regardless of any disordering.

2. TCP/IP provides error detection since it has a checksum field in its header. After receiving a data packet, a checksum procedure will be performed to ensure the correctness of the received data.

3. TCP/IP provides sliding window flow control to avoid letting senders sending data too fast that receivers cannot properly receive and process. The flow control protocol lets receivers specify restrictions on the maximum amount of data a sender could send, and the sender could only move forward to sending more data after it received an acknowledgment from the receiver.

4. TCP/IP provides congestion control by adjusting the speed senders send data into the network. Generally speaking, if a potential traffic congestion is detected, senders decrease their sending rates to less the traffic goes into the network. If no sign of traffic

congestion is detected, senders are allowed to increase their sending rates to take fully advantage of the network.

1.3 part 3

1. Domain Name System (DNS) was introduced to address the naming and addressing issue with the development of scale of Internet. The early stage network has a limited number of hosts and their names and addresses are able to be stored in a single table. However, as the number of hosts in network increases, the single table is not sufficient anymore. Therefore, the hierarchical distributed naming system, DNS, was developed to solve this issue by associating various information with domain names assigned to each of entities and it translates easily domain names to the numerical IP addresses to locate computers.

2. A hierarchical model of routing using an Interior Gateway Protocol (IGP) and an Exterior Gateway Protocol (EGP) was developed to connect the networks and regions together. Each region can run their selected IGP to deliver packets inside their region, while using EGP to route packets among different regions.

2 Problem 2: Secure and Reliable Data Transmission Service

In this problem, there are two main aspects,reliability and security, we focus on about the transmission.

- One possible fault is data packets got lost or delayed during transmission. Causes could be bad or broke connections and traffic congestion inside network. Another possible fault is that data packets got damaged or corrupted during transmission.

  In terms of attacking, the confidential file could be tampered by intruders and the file the client received would not be the original file anymore. Intruders could as well send a make up confidential file to clients claiming they are the storage server. Also, if we do not properly process the data file, intruders would be able to access the content of the file which leads to a leaking issue. After the confidential file being received by the client, it is possible that virus on the clients computer modifies the file. We also anticipate a potential security issue that an intruder would be able to detect the sender and the receiver transmitting the confidential file which results in identity exposure of the client.

- To ensure reliable data transfer, we would like to build several disjoint UDP connections between the server and the client, and asking client acknowledges the data receiving status by sending ACKs back to the server to avoid server resending the file. This service involves both application-level and network-level reliability.

Since the confidential file is a short file, which implies that making and sending a certain number of copies of the confidential file won't overload the network as it will with a huge file. Therefore, we choose to send copies of the confidential file to increase the reliability in terms of data transferring. We choose to build UDP instead of TCP connections because the cost of building TCP connections is high. Even though UDP does not provide reliable transfer as TCP does, however, the multiple copies of the confidential file increases its chance of successfully transferring the file to some extent. Also, by asking client acknowledges the file receiving status we could make sure that server knows when it is necessary to resend the file.

To secure the data transfer process, we would like to perform digital signature authentication, public key encryption as well as adopting an application to protect the file from being damaged on client's computer. By performing these function: 1) we would be able to authentic the sender of the file is the expected server, 2) we could make sure the file was not modified, 3) intruders do not have access to the confidential file, 4) file is safe from being damaged after receiving by the client. This service involves both application-level and network-level reliability.

Denote the confidential file as m. We let the server and the client both obtain their public and secret key. Upon sending m, the server signs the file with its secret key, s_s, and use the public key of the client, p_c, to encrypt both m and the signed file $\{m\}_{s_s}$ to get $\{m, \{m\}_{s_s}\}_{p_c}$. After receiving the encrypted data, the client uses its secret key, s_c, to decrypt it. Since intruders don't have the client's secret key, they are not able to read the content of the confidential file. Next, we use the server's public key to verify signature to check authentication and integrity.

Furthermore, if we would like to achieve the goal of disabling intruders from detecting the fact a file is being transmitted from the server to a client, we could take advantage of the tor project to provide privacy for the client.

- The advantages of my approach are:
  - Taking advantage of the size of the confidential file, which is small, to perform multi-path routing to insure reliable data transfer instead of keeping high cost TCP.
  - Provide authentication and integrity check for the confidential file.
  - Provide client activity and identity privacy.
  - Prevent file access from intruders.

  The disadvantages of my approach are:
  - It is expensive to keep the PKI infrastructure.
  - Intruders could perform DDoS attack to the client by sending huge amount of malicious files since it is expensive to decrypt files.

3 Problem 3: High Speed Routers

3.1 part 1

1. Each forwarding engine has a complete set of the routing tables. Traditionally, routers keep a central master routing table and the satellite processors each keep only several latest used routes. If a route information is not available in the satellite processors, they request the information from the central master routing table. Therefore, at high speeds, the cost of requesting routing table multiple times is much higher than processing the packet header. By letting each forwarding engine has a complete set of routing tables would overcome the bottleneck issue.

2. The MGR uses a switched backplane. Switched backplane allows parallelism of a switch compared to the traditionally applied shared bus mechanism.

3. The MGR includes Quality of Service (QoS) processing in the router by splitting the QoS function. The forwarding engine classifies packets and a specialized processor called QoS processor takes charge of the scheduling of the packets. This design proves the possibility of building a router that includes line-speed QoS.

3.2 part 2

3.2.1 part a

The Ethernet-used ARP does not work for the MGR architecture is because the pipelined MGR does not have a convenient place in the forwarding engine to store datagrams awaiting an ARP reply.

3.2.2 part b

The ARP is implemented following a two-part strategy. The first part is the router ARP's for all possible addresses on each interface to collect link-layer addresses for the forwarding tables at a low frequency. And the second part is datagrams for which the destination link-layer address is unknown are passed to the network processor, which does the ARP and, once it gets the ARP reply, forwards the datagram and incorporates the link-layer address into future forwarding tables.

3.3 part 3

3.3.1 part a

The reason IP header checksum is not checked is due to its high cost. In the best situation, it would require 17 instructions to be spread over a minimum of 14 cycles which increase the time to perform the forwarding code about 21%. It is considered as high cost to check for a rare error that can be caught end-to-end.

3.3.2 part b

1. If the destination in a header of a data packet is missed in the route cache, the packet will not be handled in fast path code which could result in reordered packets.

2. Since the forwarding engine is designed to instruct the inbound line card to discard the errored packets, therefore packets whose headers have errors will be discarded and appear as lost.

3. For datagrams whose headers has IP options, and the datagrams that must be fragamented, they are sent to the network processor for further processing which could results in reordered packets.

4. To deal with multicast datagrams, the processor needs to write out copies of the header in order to dispatch copies of the datagram. The routing process is done by multicasting code which could also leads to delay and packet reordering.

3.4 part 4

The advantage of switched architecture is it does not have the problem of head-of-line blocking since each input keeps its own FIFO and bids separately for each output. And it was shown that such a switch can achieve 100% throughput.

The disadvantage of switched architecture is that it is a point-to-point switch without the function of one-to-many, so it does not support multi-casting.

3.5 part 5

One option is as described in the "A 50-Gb/s IP Router" paper. Split the forwarding table memory on forwarding engines into two parts, call them A and B. When getting the latest forwarding table from network processor, only one or the two memories is used. Let's assume the updated information is feeding in to part A. At the same time, part B is still being used by forwarding engines to forward packets. As soon as the complete forwarding information finished updating in part A, forwarding engines start to use the forwarding information from A. Therefore, in the described process, there is no time that is spent on updating forwarding table solely. In other words, while updating the forwarding table, packets are still being sent by forwarding engines which makes the updating process seems as no designated time is required.

Another option is in the situation when bandwidth is a high limitation which means updating forwarding table for all forwarding engines would cause congestion in the network. Furthermore, it would slow down the updating process for all forwarding engines. Therefore, we could divide the network into different zones and let each zone has a master forwarding engine. When forwarding information needed to be pushed from network processor to forwarding engines, the forwarding engines will feed the forwarding table to those master forwarding engines. After the table in master forwarding engines got updated, they will push the table to other forwarding engines in their zones respectively.

4 Problem 4

4.1 Source routing algorithm

To simplify the notations when describing paths, labels are assigned to paths as shown in Figure 4.1.

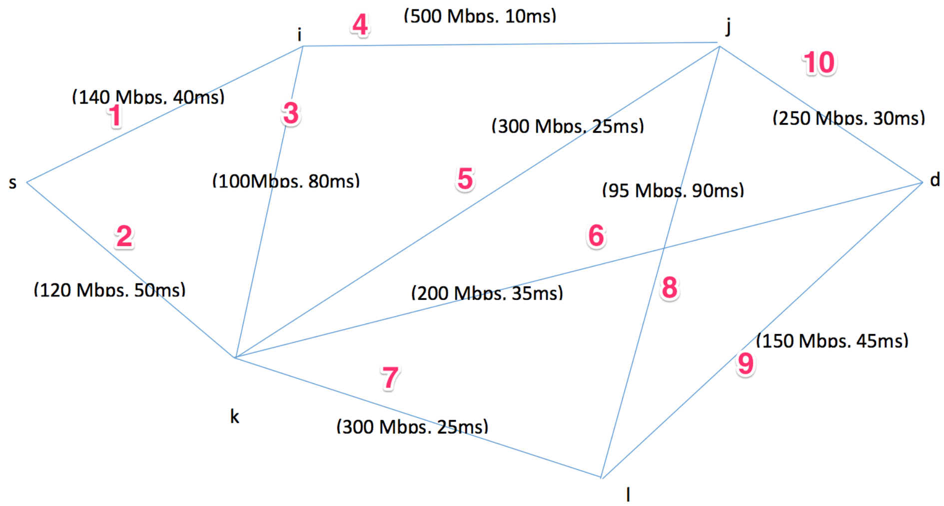

Figure 4.1: Graph G with links labeled.

Step 1: generate all paths from s to d that satisfy the bandwidth requirement.
Noticed that there is only one link, Link 8, in the graph that has a bandwidth under 100Mbs. Therefore, any paths including Link 8 will not satisfy the bandwidth requirement. When generating candidate paths, we exclude those paths which include Link 8. And all other generated paths satisfy the bandwidth requirement.

The paths generated are:

1. s-1-3-5-10-d
2. s-1-3-6-d
3. s-1-3-7-9-d
4. s-1-4-5-6-d
5. s-1-4-5-7-9-d
6. s-1-4-10-d
7. s-2-3-4-10-d
8. s-2-5-10-d

9. s-2-6-d
10. s-2-7-9-d

Step 2: keep paths that satisfy the latency requirement.

What we do is, for each path listed above, sum up the total latency and exclude those with a total latency higher than 100ms.

The remained paths are:

1. s-1-4-10-d
2. s-2-6-d

Comparing the delay of these two paths, we see the first path has delay of 80ms and the second one has 85ms. In conclusion, the best path that satisfy both requirements s-i-j-d.

4.2 Hop-by-Hop routing Algorithm

The link preference is to choose link with the highest bandwidth. The steps are listed as follows:

1. Start from origin s.
2. Choose Link 1 over Link 2.
3. Now we reach node i. Choose Link 4 over Link 3.
4. Reach node j. Link 5 and Link 10 are both valid for the next step.
5. If choose Link 5. Reach node k. Choose Link 7 over Link 6. Reach node l. Choose Link 9 over Link 8. Reach destination d.
6. If choose Link 10. Reach node d.

Both of these links has the highest bandwidth of 140Mbs. However, the total latency of the first path is 145ms which is greater than second path of 80ms. Therefore, the best path is s-i-j-d.

5 Problem 5: BGP Routing

5.1 part 1

We use notation S to represent source and D for destination.

First example: S first sends a file to D through routers A and B. The path is described as S-A-B-D. However, when D try to send a file back to S follow the reverse path, it finds

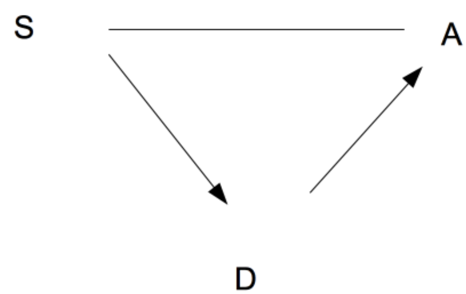

Figure 5.1: Client customer relationship. Arrow points from client to provider.

that the link from A to S is highly congested. Therefore, to avoid congestion, D sends file to B and then routes the file around A, say B-E-S. In such case, asymmetric routing happens.

Second example: assume S is D's client and D is A's client. S and A are peers. The relationship is presented in Figure

When S try to send a file to D, it will choose to route through A instead of directly send the file to D since it prefers peer. The route will be S-A-D. When D try to send a file to S, it will choose to send it directly to S since S is its customer. And we have an asymmetric routing situation.

5.2 part 2

1. One possible attack is that both E and F not broadcasting D's information. If, as shown in the problem that D is only connected to E and F, then D will not be able to receive ant data packets from any other domain. In other words, D is being blocked. In order to limit the impact of D from getting blocked and get aware of the situation sooner, a server located in domain other than E and F could be used to periodically sends data packets to D to see if D is able to receive any. If D is not receiving any packets from the server, then clearly D is being attacked.

2. Another possible attack is that E broadcasts an invalid route for E to send data packets to another domain, say A. In such case, E could get data packets from D and discard them and the data packets will be lost. To protect D from such attack, Secure BGP could be applied to prevent such routing manipulation. Secure BGP validates attributes in BGP update messages between ASes through the use of digital signatures and associated public key certificates.

3. The third possible attack is a DDoS attack from E to D by sending huge amount of data packets, such as BGP updates, to overload D. What D could do to protect itself against this attack is running a filtering algorithm at its gateway which is connected to E. The filtering algorithm identifies the prefixes and restricts the amount of BGP updates gets sent into D in a fixed period of time.