Redes de computadoras, 5 de diciembre de 2024

Transferencia confiable, TCP y control de congestión

Martínez Buenrostro Jorge Rafael

correo, molap96@gmail.com
Universidad Autónoma Metropolitana
Unidad Iztapalapa, México

Knowledge Checks

Connectionless Transport: UDP

UDP header files

- Which fields are in a UDP segment header?

  Source port, destination port, length (of UDP header plus payload), internet

- Why is the UDP header length field needed?

  Because the payload section can be of variable length, and this lets UDP know where the segment ends

- Over what set of bytes is the checksum field in the UDP header computed over?

  The entire UDP segment, except the checksum field itself, and the IP senders and receive address fields

- The next statements are true about a checksum:

  - A checksum is computed at a sender by considering each byte within a packet as a number, and then adding these numbers (each number representing a bytes) together to compute a sum (which is known as a checksum)
  - The sender-computed checksum value is often included in a checksum field within a packet header
  - The receiver of a packet with a checksum field will add up the received bytes, just as the sender did, and compare this locally-computed checksum with the checksum value in the packet header. If these two values are **different** then the receiver **knows** that one of the bits in the received packet has been changed during transmission from sender to receiver.

- Compute the Internet checksum for these two 16-bit words: 11110101 11010011 and 10110011 01000100

  To compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. This means that the Internet checksum is **01010110 11100111**

- Compute the Internet checksum for these two 16-bit words: 01000001 11000100 and 00100000 00101011

  To compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. This means that the Internet checksum is **10011110 00010000**

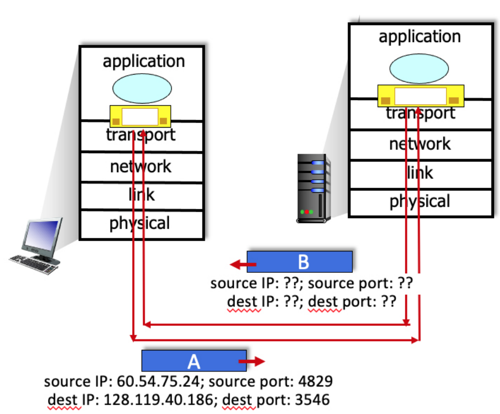
NOTE https://traductordebinario.com/calculadora-de-sumas-binario/

NOTE https://www.allmath.com/es/calculadora-del-complemento-a-uno.php

- When computing the Internet checksum for two numbers, a single flipped bit (i.e., in just one of the two numbers) will always result in a changed checksum

- When computing the Internet checksum for two numbers, a single flipped bit (i.e., in just one of the two numbers) will NOT result in a changed checksum.

- Now consider the UDP datagram (and the IP datagram that will encapsulate it) sent in reply by the application on host 128.119.40.186 to the original sender host, labeled B in the figure above.



- The source port number of the UDP segment (B) sent in reply is: **3546**

- The source IP address of the IP datagram containing the UDP segment (B) sent in reply is: **128.119.40.186**
- The destination port number of the UDP segment (B) sent in reply is: **4829**
- The destination IP address of the IP datagram containing the UDP segment (B) sent in reply is: **60.54.75.24**

UDP segment length field

- Why is the UDP header length field needed?

  Because the payload section can be of variable length, and this lets UDP know where the segment ends

- Over what set of bytes is the checksum field in the UDP header computed over?

  The enire UDP segment, except the checksum field itself, and the IP sender and recieve address fields

- The next statements are true about a checksum:

  - A checksum is computed at a sender by considering each byte within a packet as a number, and then adding these numbers (each number representing a bytes) together to compute a sum
  - The sender-computed checksum value is often included in a checksum field within a packet header
  - The receiver of a packet with a checksum field will add up the received bytes, just as the sender did, and compare this locally-computed checksum with the checksum value in the packet header. If these two values are **different** then the receiver **knows** that one of the bits in the received packet has been changed during transmission from sender to receiver

- Compute the Internet checksum for these two 16-bit words: 11110101 11010011 and 10110011 01000100

  To compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. This means that the Internet checksum is **01010110 11100111**

- Compute the Internet checksum for these two 16-bit words: 01000001 11000100 and 00100000 00101011

To compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. This means that the Internet checksum is **10011110 00010000**

- When computing the Internet checksum for two numbers, a single flipped bit will always result in a changed checksum

- When computing the Internet checksum for two numbers, a single flipped bit in each of the two numbers will NOT result in a changed checksum

Internet Checksum and UDP

-

What is checksum?

Computing the internet Checksum (1)

Computing the internet Checksum (2)

UDP Checksum: how good is it?

UDP Checksum: how good is it?

IP addresses and port numbers in a UDP segment sent in reply

Principles of Reliable Data Transfer

Reliable data transfer protocol mechanisms

Cumulative ACK

Stop-and-wait: channel utilization

Channel utilization with pipelining

Channel utilization with pipelining (more)

Pipelining

Packet buffering in Go-Back-N

Packet buffering in Go-Back-N (more)

Receiver operation in Selective Repeat

Receiver operation in Selective Repeat (more)

Connection-oriented Transport: TCP

Reliable data transfer protocol mechanisms

Cumulative ACK

Stop-and-wait: channel utilization

Channel utilization with pipelining

Channel utilization with pipelining (more)

Pipelining

Packet buffering in Go-Back-N

Packet buffering in Go-Back-N (more)

Receiver operation in Selective Repeat

Receiver operation in Selective Repeat (more)

Principles of Congestion Control

Congestion control versus flow control

Two congested senders

Different approaches towards congestion control

Ejercicios interactivos