15-441 Question Set #4 Solution

May 4, 2007

1 Quality of Service Examples

In the Quality of Service discussions there were 3 different models for applications' bandwidth needs. Provide 2 examples of applications conforming to each model and briefly explain (1 sentence) why they fit the model.

(Elastic) Transferring files, browsing the web, sending e-mail are all elastic as it only requires a tiny bit of bandwidth for them to work, but the more bandwidth the faster they go and the happier the user.

(Realtime) A remote control surgical robot is realtime as it is imperative the the robots motions not be jerky and there is some fixed amount of bandwidth required to transmit the motions.
VOIP is realtime as missing data bits make the conversation unintelligible and any serious latency makes the conversation impossible.

(Delay-Adaptive) Many video games are delay-adaptive, they require some amount of bandwidth before the game is functional and then they become "laggy" with some minimal amount of bandwidth. With more bandwidth the annoying lag goes away and the user gets more utility.
Streaming video is delay-adaptive as there is some amount of bandwidth required for the minimal number of frames to play the video. As more frames get through, the quality of the video greatly improves.

NB: The line between delay-adaptive and realtime is somewhat fuzzy and can depend on the implementation. A streaming video that dropped the connection and forced you to start over if a frame were late might be considered to be realtime.

2 Quality of Service Reservations

In RSVP why are bandwidth reservations granted on the way back from the destination to the requester rather than on the way from the requester to the destination?
The process of making a reservation is expensive, this allows us to wait until we know there is enough bandwidth to complete the reservation before attempting it.

3 Mobile Latency

(a) Suppose you and your partner transfer a large file between your adjacent laptops in the cluster

on CMU's network. How many trips across the internet will each packet and it's corresponding ACKs take?
Zero trips, both machines are on the same network.

(b) Now suppose you both go to an onsite interview at your favorite company and the company kindly allows your laptops on the corporate network and, even better, lets your keep your CMU ip addresses using Mobile IP. Suppose you and your partner, once again, transfer a large file between your adjacent laptops in a conference room. How many trips across the internet will each packet and it's corresponding ACKs take?
Four trips. The sender needs to send the packets to CMU which forwards them to the receiver. The receiver then sends the ACKs back to CMU which forwards them to the sender.

4 Public vs Private Key

With the advent of public key encryption it might seem that private key encryption should be obsolete as we can have private, authenticated conversations by signing messages with our private key and encrypting them with our correspondent's public key. Frequently, however, this technology is used as a way of exchanging private keys and then a conversation continues using the private keys. Why do we do this? Give two or three plausible reasons.
1) Private key encryption is significantly less time consuming than public key, especially since we need to both encrypt and sign each message.
2) If one were to have a conversation with more than 2 entities, each message would need to be encrypted once for every recipient whereas one group could share a private key allowing us to encrypt and send the message once.
3) Signing documents is typically done by signing a hash of the document. While it's believed that finding another document that hashes to the same value is difficult, signing additional things gives an adversary more signatures to work with and find a potential match. As a general rule in cryptography, the more examples we give our adversaries of encryption in action, the easier it is to break it.

5 Evading Security

A replay attack is an attack on a server where we first observe a legitimate client authenticate with the server and then replay some of the messages we observed to gain control of the server ourselves.

Alice and Bob have designed the following protocol for authenticating with their mp3 server:

K_{USER} is a private key shared by USER and the mp3 server

client to server: $\{\text{USER}, \text{NONCE}1\}$
server to client: $\{\{\text{NONCE}1\}_{K_{\text{USER}}}, \text{NONCE}2\}$
client to server: $\{\text{SONGNAME}1, \{\text{NONCE}2\}_{K_{\text{USER}}}\}$
server to client: $\{\text{SONG}1\}$
client to server: $\{\text{SONGNAME}2, \{\text{NONCE}2\}_{K_{\text{USER}}}\}$
server to client: $\{\text{SONG}2\}$
client to server: $\{\text{SONGNAME}3, \{\text{NONCE}2\}_{K_{\text{USER}}}\}$
server to client: $\{\text{SONG}3\}$
etc.

(a) How can we use a replay attack to gain access to their music library?

We can use the encrypted nonce2 to request additional songs from the server once Alice or Bob has authenticated.

(b) How can we adjust the algorithm to prevent this attack?

The server could send back a different nonce each time that needs to be used for the next request, or we could get just one song before needing to authenticate again. There are other solutions as well.