

1703117

by Dan Thomas Rowley

FILE	4103_DAN_THOMAS_ROWLEY_1703117_349477_2035484836.DOCX (28.78K)		
TIME SUBMITTED	12-MAY-2021 11:52PM (UTC+0100)	WORD COUNT	1464
SUBMISSION ID	152953469	CHARACTER COUNT	7506

Contents

1. TCP	1
2. MTU [1]	2
3. Synchronicity and asynchronicity	3
5. Interoperability [2]	4
9. Bibliography	4

1. TCP

TCP also known as Transaction Control Protocol is the protocol used to transmit data between applications over the proverbial series of tubes. TCP implements numerous strategies to be sure that data is sent and received without error. These are known as “Layers”.

1.1. Layers:

1.1.1. Application

The application layer sees that the data is set out in a format which the receiving/sending application is able to read, which is done through protocols such as http.

1.1.2. Transport

The Transport layer prepares data to be transported. It does this by separating data into packets to later be sent over the network to their destination, via a unique route. Each packet is given a header which contains information for the receiving application on how the data should be put back together as well as error checking information to help the receiving application knows whether or not all of the data was sent correctly.

1.1.3. Internet

The Internet layer simply assigns the IP addresses of both the sender and recipient to each packet.

1.1.4. Network

Finally, the Network layer adds a MAC address to each packet to ensure that the packets are sent to the correct machine.

1.2. Three-way handshake

The three-way handshake describes how a session between two devices on a network would begin. The three steps are as follows, using the example of a client and a server:

1.2.1. Step 1

First of all, the client would send a request containing its SYN (sequence number) to the server querying whether or not a session between the two would be possible.

1.2.2. Step 2

When the server receives the request, it would send an acknowledgment (ACK) to the client, confirming that a session would be possible as well as a request containing a SYN of its own querying whether or not a session between the two would be possible.

1.2.3. Step 3

Finally, the client sends an acknowledgment confirming that a session would be possible. And thus, the session begins.

1.3. The two general's problem

Simply put, the two general's problem is a problem, within computer science, which occurs when two entities, a client and a server for instance, are attempting to initiate a conversation. The issue is as follows:

- Client Request
Any time the client sends a request to the server, a problem arises: how can the client be sure that the request has been delivered successfully? An ongoing worry, when communicating with another device, is that a communication may not be received by its intended recipient. It is for this reason that the client will wait for the server to respond with a confirmation that the request was received.
- Server response
The problem then is how can the server be sure that the confirmation has been delivered successfully? The server will wait for the client to respond with a confirmation that the confirmation was received. These entities will continuously trade confirmations.

1.3.1. Solutions

The two generals problem remains unsolvable although there are a number of solutions which, though aren't completely water tight, remain reliable enough that they are widely used to tackle this problem.

- Solution 1
One solution was to send a large number of confirmations, 100 for example, each with its own serial number. If the recipient received a large proportion of these, 97 out of 100 for example, then it is safe to assume that the communication channel is safe enough to continue.
- Solution 2
Another solution was for entity 1 to send confirmation to entity 2 at intervals one at a time until a confirmation was received in response at which point entity 1 would stop send confirmation. Entity 2 would then take the lack of incoming confirmations as a sign that it was safe to proceed.

2. MTU [1]

2.1. MTU

The MTU (maximum transmission unit) operates on the network layer of the OSI model and is a process on a network which outlines to maximum size of any packets transferred on the network, this size is usually 1500 bytes, ignoring the size of the header. In the case that the size of a package is too large, the MTU will fragment it into smaller packet (fragments) which comply with the size requirements.

This is unless the packet header contains a DF (don't fragment) bit. This is an instruction to any devices along the path of the packet, specifying that the packet should not be fragmented.

2.1.1. Disadvantages

- One downside of fragmentation is that it takes time. Given that many packets may need to be fragmented, this could seriously slow down the transaction process.
- Another downside is that post fragmentation, if a fragment was to be lost, then the entire packet would have to be resent only to be fragmented again.

- There is also a likelihood that packets may be delivered out of order in which case they would have to be reordered.

2.1.2. PDU

A PDU (protocol data unit) is the name of each unit of data used on layers 1 – 4 of the OSI model.

1. Physical layer – bits
2. Data link layer – a frame
3. Network layer – a packet
4. Transport layer – a segment

Layers 2 – 4 each add a header to a packet before it is sent off. The data within these headers is defined by the PDU.

3. Synchronicity and asynchronicity

3.1. Synchronicity

In computing, the phrase “Synchronised” is used to describe a program in which all processes are run in order, each taking place one after the other.

Advantages

One advantage of this kind of programing, when creating smaller programs, is that there is less hassle on the part of the developer as they needn’t worry about the headache of ensuring that all threads are used correctly and in a manner which does not cause race conditions.

Generally, many smaller programs run quickly enough that threading would not improve the programs speed a noticeable amount.

Disadvantages

The main downside of synchronous programming is the speed at which larger programs can run. Writing programs and not taking advantage of threading can slow them down exponentially.

3.2. Asynchronous

On the other hand, processes within an asynchronous program can sometimes run parallel to one another. This process is done through assigning tasks to different threads (no more than are available).

Advantages

The main advantage of writing code in this manner is that tasks can be processed awfully fast. For example, if you have 100 task which must be processed and assign them to be carried out by 4 different threads, then the program would execute 4 times faster than it was run by executing one task after another.

Disadvantages

One disadvantage of asynchronous programming is the amount of extra code one has to write to ensure that the code does not develop errors such as race conditions.

4. Interoperability [2]

These days, many different technologies are in use in many different areas in the world of IT. For this reason, it is imperative that data transaction is possible between different devices despite what platforms or operating systems may be in use.

Interoperability enables different systems to communicate with one another to transfer data of any format. Using interoperability, developers can do their job without having to worry about things like making sure their database, webserver and personal PCs all run on the same operating system, there is no fear these systems may be unable to communicate with one another. This is imperative for many of today's businesses and the health sector is no different, as shown by the follow quote from an article on the subject by EHR intelligence [2]: "Achieving true healthcare interoperability across the care continuum is a top priority for providers, payers, and other key industry stakeholders."

There are three levels of interoperability:

- Level 1: Foundational interoperability

Level 1 takes care of the exchange of data between two systems. The information which is received will not need to be interpreted upon arrival, only to recognise that the data has been received. Level 1 interoperability is the most basic level of interoperability.

- Level 2: Structural interoperability

Level 2 enables the receiving system to understand the data at the data link level. Level 2 is the intermediate level of interoperability.

- Level 3: Semantic interoperability

Level 3 allow systems to exchange data with one another as well as being able to both understand and use it. Level 3 is the most sophisticated level of interoperability.

5. Bibliography

[1] An article from Search networking titles "Maximum Transmission Unit (MTU)":

<https://searchnetworking.techtarget.com/definition/maximum-transmission-unit>

[2] An article from EHR Intelligence titled "How health data standards support healthcare interoperability":

"Achieving true healthcare interoperability across the care continuum is a top priority for providers, payers, and other key industry stakeholders."

<https://ehrintelligence.com/features/how-health-data-standards-support-healthcare-interoperability>

FINAL GRADE

11 / 100

GENERAL COMMENTS

Instructor

Unfortunately a long way from the level of detail required. A limited attempt, based on some sources of extremely low academic credibility. Topics barely covered.

No implementation provided.

Report (100): 22

Implementation (100): 0

Module average: 11

PAGE 1

PAGE 2

PAGE 3

PAGE 4