

Socket_prog_web_service_report

by Anthony Smith

FILE	3739_ANTHONY_SMITH_SOCKET_PROG_WEB_SERVICE_REPORT_349477_1928907059.DOCX (113.7K)		
TIME SUBMITTED	08-MAY-2021 08:22PM (UTC+0100)	WORD COUNT	3446
SUBMISSION ID	152297602	CHARACTER COUNT	20043



Socket programming and Concurrency

ACCA6028

Web service report

Submitted by:

Anthony Smith 1704559

Submitted to:

Dr Tim Bashford

May 2021

Table of Contents

1. Introduction	2
2. TCP and guaranteed delivery of data	2
3. Impact of maximum transmission unit size	3
4. Synchronicity and asynchronicity	5
4.1 Synchronous transmission	5
4.2 Asynchronous transmission	5
4.3 I/O dispatching	6
4.3.1 Blocking I/O model	6
4.3.2 Non-blocking I/O model	7
5. Interoperability for networked applications	7
6. Conclusion	8
References	9

1. Introduction

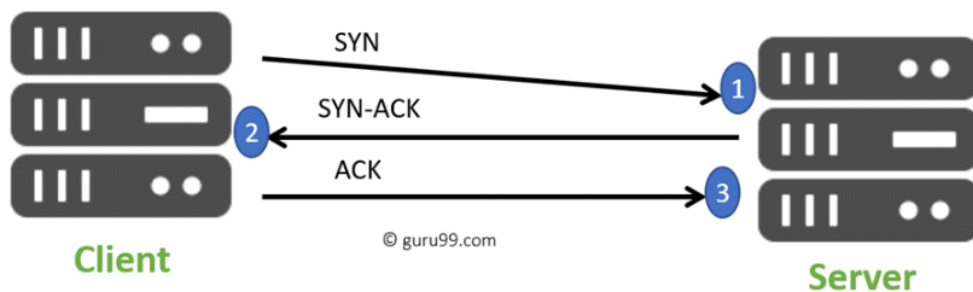
Client-server architecture is utilised on a daily basis and for most part, operates without the knowledge of the user. The system is divided into two sections, 'client' and 'server', allowing applications on a network to connect, one to the other. While the server side acts as the central functionality of the system: i.e., allowing several connections to the server, the client side performs the initial request to join the server and awaits the server response. The client-server model can be employed for global use such as the internet, or for business organisations to communicate via the local area network. With emphasis on communication between two clients, the moment the source of the data sends correspondence to the target system, the message is routed from server to server. This is only possible by IP addressing, and the inclusion of information that states the origin and destination of the message. To ensure data will be delivered successfully, there are certain protocols that aid in that success, the TCP protocol being one of them. There are also other factors that have been discussed within this report that have bearing on the success of communication. Such considerations are: MTU and the transmission size of the data packets, the communication mode that relates to the transmission of data and the interoperability of the system that is utilised by the user. [1] [2]

2. TCP and guaranteed delivery of data

Transmission Control Protocol (TCP) is a part of the internet protocol suite and is a process that is required after the initial IP protocol has been carried out. IP is the system that has the main function of delivering packets of data from one user to a target user. But as those packets can be sent and delivered in any order, there has to be a system that can handle packet ordering while also checking for any errors that has occurred to those packets. This is where the TCP protocol comes into play.

TCP has several operations that are carried out before a message is deemed successfully sent. It has to collect the packets and assemble the packets in the correct order. If any packets are lost or contain errors, TCP requires the packet to be resent and has to maintain connection between the sender and receiver until the message has been reassembled correctly and received by the target device.

Whenever an email is sent, a connection known as the three-way handshake is made. This three step process requires a synchronization exchange and acknowledgment packets to be passed, before real the communication process begins [Fig 1].



[Fig 1] Three way handshake in operation [1]

1. SYN – This is the initial request from the sender device to establish the connection.
2. SYN/ACK – The target device sends an ACK packet to confirm that the original SYN had been received.
3. ACK – The source of the communication then sends an ACK packet to confirm the process is ready to begin. After this has occurred, messages can be sent.

When successful communication between the sender and target device occurs, and each message is received correctly, both systems know what operation to perform next. But a problem can occur if one of those messages does not reach the destination.

This dilemma is known as the Two Generals Problem and happens when one system is unaware of what the other is doing. Although TCP is very reliable, so far it has been unable to solve this highlighted challenge.

Several approaches have been researched to try and solve the Two Generals Problem with one main ideology, that there will also be uncertainty on the communication channel. From this base, the first premise is to send more than one message to the target device and vice versa. Each message will be given a serial number and based on the success of the messages the reliability of the communication can be gauged. The other option is to send messages at time intervals in which the absence of a message will be gauged. This will indicate communication reliability as the interval will be agreed by both devices. [1] [2] [3] [4]

3.Impact of maximum transmission unit size

MTU is the measurement of the largest data packet that a device can accept while connected to the network. If internet packets exceed the MTU they will not be passed through the network. To combat this problem, the data packets are broken up into smaller chunks to appease the size restriction and are then delivered to the destination. This is known as IP fragmentation and allows for:

- Transport layer protocols to be unaware of the underlying network architecture, reducing overheads.
- IP and higher layer protocols can work across variable and diverse network paths and mediums without the need of a path discovery protocol.

Although the idea behind IP fragmentation first seems logical, on investigation several drawbacks can be identified, which result in the avoidance of this process wherever possible:

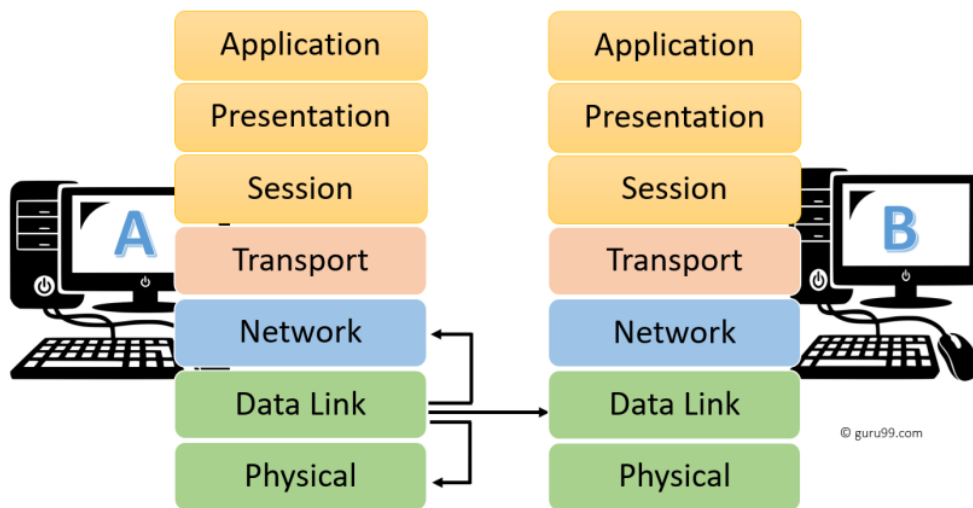
- The loss of a fragment can result in all fragments having to be resent.
- If the sender resends the same packet, fragmentation can happen all over again.
- Only the first fragment contains the high layer headers, which can cause issues with firewalls and routers that rely on inspecting headers, mainly the NAT function.
- Adds a degree of latency and inefficiency to communications.

When the exchange of data packets occur, it is not just the MTU of the devices that look at the size of the packets. But every device that is involved with the communication such as routers, switches and servers. Each has its own MTU that the packet has to contend with and depending on connection, sizes of MTU can differ. At any point on the network path the packets can be fragmented.

IP fragmentation is not allowed in IPv6 but is still occurring because IPv4 is still widely used. If routers in operation support IPv6, they will automatically drop packets that exceed the MTU, as they cannot be fragmented [5] [6].

The OSI model (Open Systems Interconnection Model) is the framework that describes the functionality of a networking system. This model is split into 7 layers in which the IP packets can travel from the source device to the target device. The data units (PDU - protocol data unit) describes the type of data unit that will be transported in the OSI model and within layers 1 to 4.

The data unit and description of the four layers are as follows:



[Fig 2] 7 layer OSI model [7]

1. Physical Layer – This includes the equipment that is needed to transfer the data such as cables and switches. The data transmits raw bit data made up of 1s and 0s.
2. Data Link Layer – Very similar to the network layer, except this facilitates data transfer between two devices on the same network. The data link layer gathers the packets from the network layer and breaks them into smaller chunks called frames. Frames contain the source and destination addresses, the message to be delivered, error detection and correction bits, and flags that mark the start and end of the frame.
3. Network Layer – Transfers data between different networks. This layer separates the segments from the transport layer, into smaller units called packets. The MTU also takes effect on this layer. A packet contains the two sections: the header, which contains the IP addresses of the source and target device. The payload contains parts of the data that require delivery.
4. Transport Layer -this layer is responsible for the end to end communication between the source and target devices. Depending on which direction the data is going, this layer breaks up data from the session layer into segments or reassembles the segments for the session layer. This layer is where encapsulation begins, and the 3-way handshake establishes a connection. Segments contain the header and the data to be sent to the target device. [7] [8] [9]

The main dilemma of the PDU travelling through the OSI layer is that data from the source device gets encapsulated with a further protocol. This protocol then increases the overall frame size, which can lead to the packet exceeding the MTU. This can then lead to loss of data or to non-functioning applications.

4. Synchronicity and asynchronicity

Synchronicity and asynchronicity relate to the transmission of data from one device to another through a network. This procedure can also be known as the communication mode.

4.1 Synchronous transmission

The synchronous transmission transfers data in a continual or “real-time” stream and is accompanied by timing signals. This signal ensures that the transmitter and the receiver are synchronised with each other. The main purpose of this communication mode is to transfer large amounts of data from the source device to the target device.

Synchronous transmission employs separate clocking lines if the distance between the data terminal and data communications equipment is small. Devices that are linked synchronously can use either of the separate clocking channels. The data sent through synchronous connection are within blocks or frames of data and each byte is transmitted without space between the following byte.

This type of communication can occur in several different ways, either in-person or virtual, scheduled or impromptu, and due to the large data transferal it is best utilised when the communication is:

- Video Conferences
- Online class discussions
- Phone calls

Further functionality of synchronous transmission are:

1. A waiting period for the transmission to finish, before getting a response from the server.
2. Broadband channels use synchronous transmission
3. Can be used in low-speed communication.

Using synchronous transmission is not totally without pitfalls, and can suffer from the following drawbacks:

- Accuracy of the source data depends on the ability of the target device. If the receiving system lacks the ability to count incoming bits adequately the quality of communication will suffer.
- The system sending the data and the target system can only operate simultaneously if both systems have measurable clock frequencies. [11] [12]

4.2 Asynchronous transmission

Unlike synchronous transmission that occurs in real-time, asynchronous happens “out of sync”. This means that there is no time-sensitivity and correspondence does not happen while the interaction is occurring.

The process of transmission is initiated by two bits, (0 bit) to start the communication and (1 bit) to stop the transmission. As communication is not time reliant, both the source system and the target system can have different clock frequencies.

Asynchronous transmission provides the ability for the participant to respond at their own convenience and is widely used within organisations and widespread public interaction. Examples of this type of communication are:

- Emails
- Online forums
- Text messages

Further functionality of asynchronous transmission are:

1. No waiting time for response.
2. Data is sent in bytes.
3. Transmission method is slow.
4. There is a gap between data transmission.

Asynchronous transmission can also suffer from negative implementation when a system uses this type of communication:

- Timing errors may occur as it is difficult to determine synchronicity.
- It has a slower transmission rate.
- Noise on the channel can cause false recognition of bits.
- Start and stop bits are required. [11] [12] [13]

4.3 I/O dispatching

When the employed system is required to process requests an I/O queue is created and a dispatching method framework for that queue is given. The two main I/O dispatch methods are sequential and parallel dispatching.

Depending on the capability of the device, will determine which model to utilise. If only one I/O request can be processed from the queue at a time, then a sequential dispatch will be better suited. Sequential or synchronous dispatch uses a framework that delivers requests one at a time. This framework will not allow another request to be delivered until the request at hand is completed, cancelled or placed back in the queue.

To process multiple I/O requests at the same time, the system should utilise parallel dispatching or as it is better known Asynchronous dispatch. The framework used by parallel dispatching delivers requests as soon as they are available in the queue, this will enable the device to process several requests at once [11].

4.3.1 Blocking I/O model

When a device makes a request to connect to a server, the thread that operates the connection is blocked until data is ready. Until the data has been completed, the thread will be in waiting mode. If the need to fulfil concurrent requests is essential, then a new thread will be allocated for each connection.

The drawback of this approach is that each thread will require a stack of allocated memory as the number of connections increase. Switching between multiple threads will become cumbersome. Another problem, may be the amount of threads just waiting for requests to be sent, this would be a waste of resources.[15]

4.3.2 Non-blocking I/O model

With a socket set to non-blocking the operating system is told to return an error when an I/O operation cannot be completed, instead of putting the process in sleep mode. Non-blocking frameworks use infinite loops to verify data has been sent from the I/O. This process is known as the Event Loop. Implementation of non-blocking I/O can be achieved in other ways, such as signals and hardware interrupts.

Signals are required for inter-process communication and could also be employed to signal the ready I/O data. However, catching signals is an expensive process and not a preferred method over the more common event loop application.

Hardware interrupts are used to halt the current executing thread, allows the section of code that was running to execute, and return to the thread. This process is not ideal for applications that are event intensive, as any new request will interrupt the CPU and slow it down.

Both interrupts and signals do not offer flexibility or scalability, but for embedded systems are the preferred framework. If flexibility and scalability are required, implementing event loops are the best course of action [16].

5. Interoperability for networked applications

Network interoperability is the capacity to allow different networks or applications to communicate without needing additional tools or interfaces. By continually sending and receiving data through the network will provide a greater quality level and limit any negative impact on the network. More specifically, network interoperability refers to the functioning innerworkings of network systems across multi-users, multi-servers and multi-connections. All those operations need to occur fluidly, not only under normal circumstances, but also under tense conditions.

Network interoperability becomes vital if successful end to end connectivity is to be achieved. As varying networks exist, it is more important that they can interoperate in order to make communication possible. This ability for two networks to connect and communicate is achieved by:

- Networks conform to a common protocol standard.
- Define a standard interface, to which all networks adhere to.
- Provide a gateway that translates the two protocols between the two networks.

An example of such an application is TCP/IP that has become available on every operating system, computer and is used by almost every network. It is very important for Networks to be able to communicate with other networks as isolation and a decrease in efficiency could occur. [16] [17]

Benefits to Interoperability:

1. Lower costs, as resources and maintenance is less likely to be required.
2. Data quality is improved as more sources are available.
3. Data connectivity and transferal between systems will become standardised and allow for widespread communication between IOT's and other applications. [17]

6. Conclusion

Network communications between client and server are a culmination of factors that are important to ensure quality and success of those interactions. With protocols in place and systems that enable positive requests and responses, current frameworks illustrate the effectiveness of the methods involved. However, improvements to the transmission of data could be made, to enhance the possibility of successful communication. With advancements in technology, computer programmers have to keep abreast of the relevant approaches when it comes to design decisions. Any insufficient research may impact on the development of network applications.

A reliable solution to the transmission control protocol requires further research, to avoid the two generals problem. Network packets that have been unidentified by the target source can get lost in transit, and although certain avenues have been addressed, no system that maintains performance and keep costs to a minimum has been found.

With IOT's and further advances in file resolution, a developer will have to take in to account the MTU (maximum transmission unit). Network packets have a limited data size before further fragmentation is initiated, this can result in latency, inefficiency and data loss. IPv6 has removed the option of packet fragmentation, which stops the data from advancing if the packet is too large. With ever increasing file sizes such as video, high quality images and real-time rendering, the implementation of jumbo frames will have to become common practise. The only possible downfall is that every device between the source to the target device has to be configured using jumbo frames. If a switch or the receiver has standard MTU, then packet fragmentation may occur [21].

Dependant on the service provided by the network application, Synchronicity or asynchronicity will be chosen from the outset. If real-time services are desired, then synchronicity will offer constant data streaming allowing for online meetings, video calls and phonecalls. But to maintain the quality of this service will only be possible, if both participants in communication have similar facilities. If synced communication is not important to the application, asynchronicity will be more viable. As async does not require connection while the interaction is occurring i.e., email or text message, the system hardware in operation does not have to be equivalent. Developers will need to exercise this option with the client while exploring the development process.

More and more devices contain technology that utilise a network, whether it is organisational infrastructures or home equipment. With IOT growing evermore popular, the need for interoperability is increasingly important. Interoperability allows different devices on different networks to conform, giving a universal appeal. This reduces think time on compatibility issues and ensures end to end connections can be achieved. Interoperability functionality is what a developer really needs to consider when designing a system, as compatibility with every network can be the difference between offering excellent services and communication or a system that has become obsolete from the start.

References

- [1] Open University, "An introduction to web applications architecture," 2020. [Online]. Available:
] <https://www.open.edu/openlearn/science-maths-technology/introduction-web-applications-architecture/content-section-1.1>. [Accessed 4 May 2021].
- [2] B. Mitchell, "Introduction to client server networks," 31 Oct 2019. [Online]. Available:
] <https://www.lifewire.com/introduction-to-client-server-networks-817420#:~:text=The%20client%2Dserver%20model%20organizes,request%20and%20returning%20the%20results..> [Accessed 4 May 2021].
- [3] Guru99, "TCP 3-way Handshake," 2021. [Online]. Available: <https://www.guru99.com/tcp-3-way-handshake.html>. [Accessed 14 Apr 2021].
- [4] B. Lutkevich, "TCP (Transmission Control Protocol)," 2020. [Online]. Available:
] [https://searchnetworking.techtarget.com/definition/TCP#:~:text=TCP%20\(Transmission%20Control%20Protocol\)%20is,of%20data%20to%20each%20other..](https://searchnetworking.techtarget.com/definition/TCP#:~:text=TCP%20(Transmission%20Control%20Protocol)%20is,of%20data%20to%20each%20other..) [Accessed 29 Apr 2021].
- [5] Jakub, "Two Generals Problem," 2 May 2018. [Online]. Available: <https://finematics.com/two-generals-problem/>. [Accessed 29 Apr 2021].
- [6] CloudFlare, "What is TCP/IP," 2021. [Online]. Available: <https://www.cloudflare.com/en-gb/learning/ddos/glossary/tcp-ip/>. [Accessed 29 Apr 2021].
- [7] CloudFlare, "What is MTU (maximum transmission unit)?," 2021. [Online]. Available:
] <https://www.cloudflare.com/en-gb/learning/network-layer/what-is-mtu/>. [Accessed 1 May 2021].
- [8] S. Iveson, "IP Fragmentation in Detail," 26 Nov 2019. [Online]. Available:
] <https://packetpushers.net/ip-fragmentation-in-detail/>. [Accessed 1 May 2021].
- [9] Guru99, "Layers of OSI Model Explained," 2021. [Online]. Available:
] <https://www.guru99.com/layers-of-osi-model.html>. [Accessed 1 May 2021].
- [1] ComputerNetworkingNotes, "Data Encapsulation and De-encapsulation Explained," 20 Jan 2019.
0] [Online]. Available: <https://www.computernetworkingnotes.com/ccna-study-guide/data-encapsulation-and-de-encapsulation-explained.html>. [Accessed 2 May 2021].
- [1] CloudFlare, "What is the OSI Model?," 2021. [Online]. Available:
1] <https://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/>. [Accessed 1 May 2021].
- [1] HighTeck, "OSI Network Layer," 2021. [Online]. Available:
2] http://www.highteck.net/EN/Network/OSI_Network_Layer.html. [Accessed 2 May 2021].
- [1] Guru99, "Difference between Synchronous and Asynchronous Transmission," 2021. [Online].
3] Available: <https://www.guru99.com/difference-between-synchronous-and-asynchronous-transmission.html>. [Accessed 2 May 2021].

- [1] J. Szatar, "When to choose Synchronous Vs Asynchronous communication," 21 Apr 2020.
- 4) [Online]. Available: <https://www.loom.com/blog/synchronous-vs-asynchronous>. [Accessed 3 May 2021].
- [1] RingCentral, "Asynchronous communication," 2021. [Online]. Available:
- 5) <https://www.ringcentral.co.uk/gb/en/blog/definitions/asynchronous-communication/>. [Accessed 3 May 2021].
- [1] Microsoft, "Dispatching Methods for I/O Requests," 20 Apr 2017. [Online]. Available:
- 6) <https://docs.microsoft.com/en-us/windows-hardware/drivers/wdf/dispatching-methods-for-i-o-requests>. [Accessed 2 May 2021].
- [1] R. Athapathu, "Blocking I/O and non-blocking I/O," 31 Aug 2017. [Online]. Available:
- 7) <https://medium.com/coderscorner/tale-of-client-server-and-socket-a6ef54a74763>. [Accessed 3 May 2021].
- [1] H. De Vries, "How does non-blocking IO work under the hood?," 20 Nov 2019. [Online].
- 8) Available: <https://medium.com/ing-blog/how-does-non-blocking-io-work-under-the-hood-6299d2953c74#:~:text=APIs%20that%20use%20blocking%20IO,comes%20after%20calling%20the%20API..> [Accessed 3 May 2021].
- [1] Comviva, "All you need to know about Network Interoperability," 23 May 2016. [Online].
- 9) Available: <https://www.comviva.com/blog/references/all-you-need-to-know-about-network-interoperability/>. [Accessed 3 May 2021].
- [2] S. Lewis, "Interoperability," 2021. [Online]. Available:
- 0) <https://searcharchitecture.techtarget.com/definition/interoperability>. [Accessed 3 May 2021].

Socket_prog_web_service_report

GRADEMARK REPORT

FINAL GRADE

53/100

GENERAL COMMENTS

Instructor

A good effort overall. The report is fairly well written, covering key topics. The sources trend a little too heavily towards non-authoritative web sources, but granted it's not the easiest topic to find modern academic material for.

You've made life a little hard for yourself, implementing from scratch - but you've had a fair go, a sound use of principles.

Report (100): 65

Implementation (100): 41

Module average: 53

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

