

Network Theory

by Matthew Lewington

FILE	4069_MATTHEW_LEWINGTON_NETWORK_THEORY_227510_1330916823 .DOCX (792.03K)		
TIME SUBMITTED	11-JAN-2020 05:54PM (UTC+0000)	WORD COUNT	3554
SUBMISSION ID	117976038	CHARACTER COUNT	19277



JANUARY 16, 2020

NETWORK THEORY

NETWORKED APPLICATIONS (ACCA6015)

MATTHEW LEWINGTON
BSC SOFTWARE ENGINEERING
University of Wales Trinity St David

Contents

1	Introduction	1
2	Network Communication Models	2
2.1	OSI Model	2
2.2	TCP/IP Model.....	3
3	Packet Encapsulation	4
3.1	TCP Segment	4
3.2	IP Datagram.....	5
4	Transport Layer Protocols.....	6
4.1	Transmission Control Protocol	6
4.1.1	Three-way Handshake	6
4.2	User Datagram Protocol.....	10
5	Maximum Transmission Unit.....	11
5.1	IP Fragmentation	13
6	Synchronicity and Asynchronicity in Network Protocols.....	15
6.1	Data Transmission Channels	15
6.2	Synchronous and Asynchronous I/O Operations	15
6.2.1	I/O Control and Processing	16
7	Interoperability in Networked Applications	17
7.1	Interoperability Frameworks.....	17
7.2	Achieving Interoperability in Networked Applications.....	17
8	Conclusion	18
9	References.....	19

Table of Figures

Figure 3-1 TCP Header [7, p. 588]	4
Figure 3-2 IPv4 Header [7, p. 182]	5
Figure 4-1 Three-way handshake [6, p. 239]	6
Figure 4-2 Wireshark capture of three-way handshake	7
Figure 4-3 NSLookup for uwtsd.ac.uk	7
Figure 4-4 SYN flag set	8
Figure 4-5 SYN and ACK flags set	9
Figure 4-6 ACK flag set	10
Figure 5-1 Windows MTU size	11
Figure 5-2 Wireshark capture of packet fragmentation	12
Figure 5-3 TCP and IPv4 Header length	13
Figure 6-1 Synchronous and Asynchronous I/O difference [28]	16

Table of Tables

Table 1 OSI model and common protocols [1], [6, pp. 223–224], [7, p. 9]	2
Table 2 TCP/IP and OSI models	3

1 Introduction

This report has been produced to provide an overview of networked applications, including the protocols that operate on the network layer of the several network models which exist, how data is reliably passed through the different layers and sent from a client to a host through methods such as IP fragmentation, interoperability and finally how synchronous and asynchronous processes are utilised in networked applications.

2 Network Communication Models

There exist two widely known and similar layered models; the Open Systems Interconnection (OSI) model, which is more of a conceptual model, and the transmission control protocol / Internet protocol (TCP/IP) model (also known as the Internet protocol suite).

These models are split into layers as this allows logical viewing composed of a succession of layers with each layer wrapping lower layers to isolate them from the higher layers in the models [1].

These two models define frameworks which allow communication across a network [2], [3]. They will be briefly discussed to provide an outline, as only one layer will be discussed in detail which is related to the topic at hand.

2.1 OSI Model

The OSI model is a seven-layer model (Table 1) which was presented by the International Organisation for Standardisation (ISO) [4] in 1984 [5].

Table 1 OSI model and common protocols [1], [6, pp. 223–224], [7, p. 9]

OSI Model	Common Protocols
Application layer	HTTP, FTP, DHCP, DNS
Presentation layer	TLS, SSL, SSH
Session layer	SMB, NetBIOS, RPC
Transport layer	TCP, UDP
Network layer	ARP, RIP, ICMP, OSP
Data-link layer	ARP, CHAP, MAC
Physical layer	Ethernet, ISDN, DL

Layer 7 – Application layer: The application layer is the highest layer in the OSI model and is the “end-user” level of communication as it interacts with user data directly.

Layer 6 – Presentation layer: The presentation layer is accountable for the formatting of the data. This is also where encryption and decryption is usually accomplished.

Layer 5 – Session layer: The session layer manages, when necessary, the lower layers of the communication link by specifying methods for connections.

Layer 4 – Transport layer: The transport layer is where data is segmented or re-assembled, and this layer is responsible for flow and error control.

Layer 3 – Network layer: The network layer concerns itself with packet routing and completing packet transmission between hosts.

Layer 2 – Data-link layer: The data-link layer defines the protocol to both begin and terminate a client-host connection.

Layer 1 – Physical layer: The physical layer is the lowest layer in the OSI model. The responsibility of this layer is sending bits across physical hardware, data rates and sometimes how bits are encoded.

2.2 TCP/IP Model

The TCP/IP model, originally described by Cerf and Kahn in 1974 [8], is the model around which the Internet is developed. It shares similarities with the OSI model such as some layer names and common protocols, however the TCP/IP model only contains four layers rather than the seven layers from the OSI model (Table 2). This is because TCP/IP is under the assumption that anything beyond the transport layer will have been processed already, and it combines the physical and data-link layers from the OSI model [5].

Table 2 TCP/IP and OSI models

TCP/IP Model	OSI Model
Application layer	Application layer
	Presentation layer
	Session layer
Transport layer	Transport layer
Internet layer	Network layer
Network access layer	Data-link layer
	Physical layer

3 Packet Encapsulation

Packet encapsulation is the process of processing data and adding a header on each layer of the given network model [6, p. 226]. This section will primarily be involved with the TCP/IP model.

If everything above the transport layer has been processed, the packet of data will be encapsulated (wrapped) further in headers. At the transport layer, this will be a TCP segment, at the network layer it is an IP datagram, then lower on the network access layers, frames are added.

The reverse of packet encapsulation is de-multiplexing, or de-encapsulation, works in reverse where first the frames are removed, then the IP header, TCP header, and up to the application layer.

3.1 TCP Segment

The TCP segment, or header (Figure 3-1), is comprised of the source and destination ports, sequence and acknowledgement numbers (TCP connection establishment), a checksum (data integrity), the data itself and flags:

- **URG** urgent,
- **ACK** acknowledge,
- **PSH** push,
- **RST** reset,
- **SYN** synchronise,
- **FIN** finish

Source Port						Destination Port					
Sequence Number											
Acknowledgement Number											
Data Offset	Reserved	U	A	P	R	S	F	Window			
		R	C	S	S	Y	I				
		G	K	H	T	N	N				
Checksum						Urgent Pointer					
Options										Padding	
Data											

Figure 3-1 TCP Header [7, p. 588]

3.2 IP Datagram

The IP datagram, or header (Figure 3-2), encapsulates the previous TCP segment and contains various information such as the source and destination IP addresses which are required for a TCP connection to be established.

Version	Length	Service Type	Total Length	
Identifier			Flags	Fragment Offset
Time to Live	Protocol		Header Checksum	
Source IP Address				
Destination IP Address				
Options and Padding				

Figure 3-2 IPv4 Header [7, p. 182]

4 Transport Layer Protocols

On the transport layer, there exist two main protocols; the transmission control protocol (TCP) and the user datagram protocol (UDP). This section will primarily focus on TCP, and a brief mention of UDP.

4.1 Transmission Control Protocol

The TCP protocol provides basic data transfer via continuous streams of octets and offers reliability, multiplexing and flow control [7, pp. 583–585], [9], [10, pp. 3–5].

TCP operates as a connection-oriented protocol, meaning that transmission of data between end hosts must have obtained and acknowledged TCP connection between each other [7, p. 585], [11, p. 178]. Said connection requires both a source and destination IP address (wrapped in the IP header) and port number (wrapped in the TCP header), and is established by a sequence known as the three-way handshake [12].

4.1.1 Three-way Handshake

The three-way handshake is a process to establish, or terminate a TCP connection and it occurs between a client and a server (Figure 4-1).

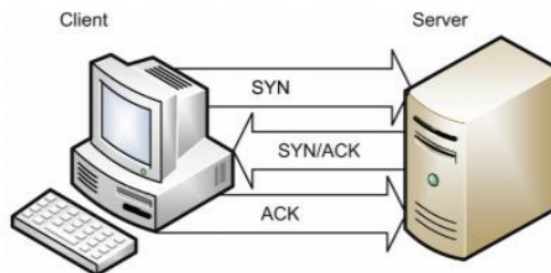


Figure 4-1 Three-way handshake [6, p. 239]

By utilising a packet analyser (Wireshark [13]), to monitor a wireless network interface and browse to uwtsd.ac.uk, the three-way handshake can be visualised (**Error! Reference source not found.**Figure 4-2) and the process outlined below showing the HTTP GET request to display the website. NSLookup of uwtsd.ac.uk provided for IP address verification (Figure 4-3).

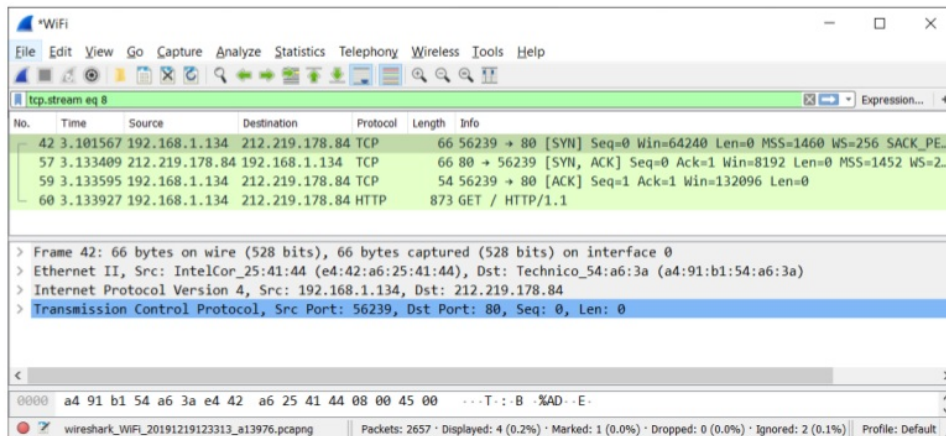


Figure 4-2 Wireshark capture of three-way handshake

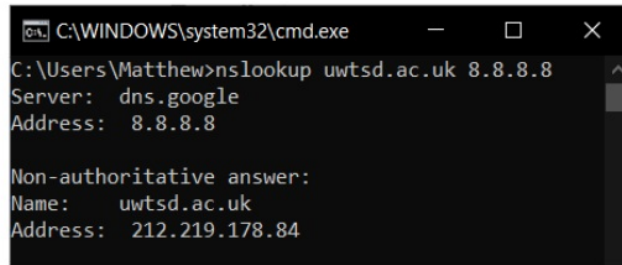


Figure 4-3 NSLookup for uwtss.ac.uk

4.1.1.1 SYN

The client first sends a packet with a synchronisation request to the server, specifying its initial sequence number. The SYN flag in the TCP header is also set to true here. (Figure 4-4).

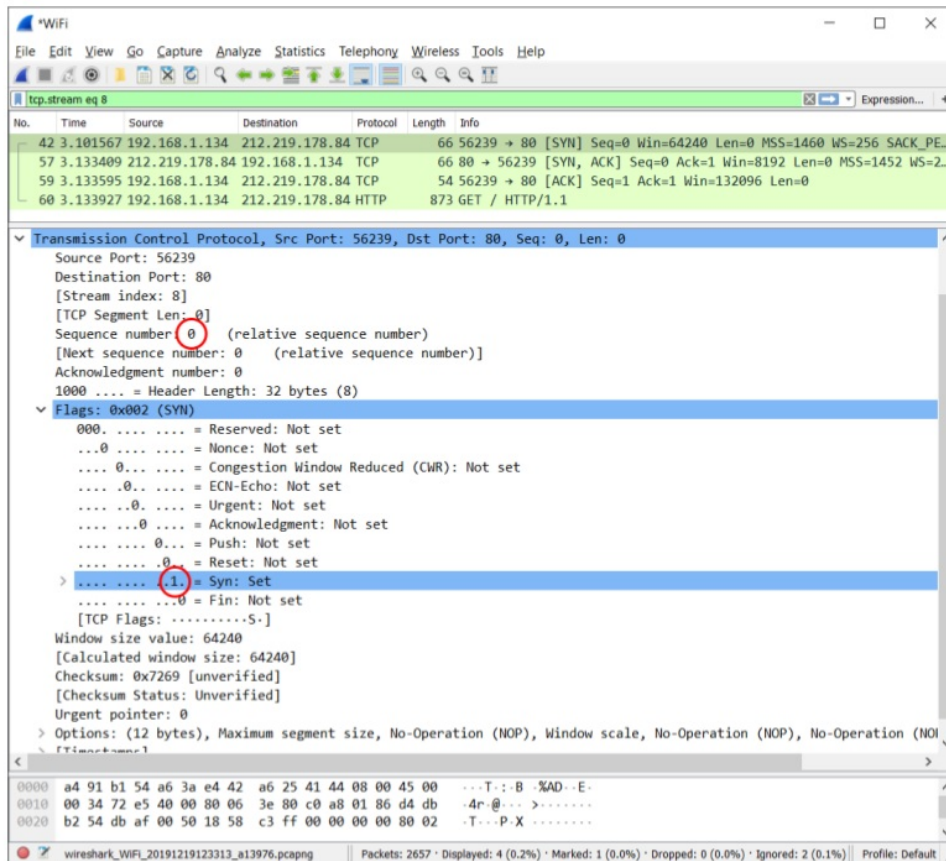


Figure 4-4 SYN flag set

4.1.1.2 SYN ACK

The server then responds, setting both the SYN and ACK flags to true (Figure 4-5), acknowledging the request of the client and transmitting an acknowledgement number which is specific to the SYN. This allows the server to increment the clients sequence number by one.

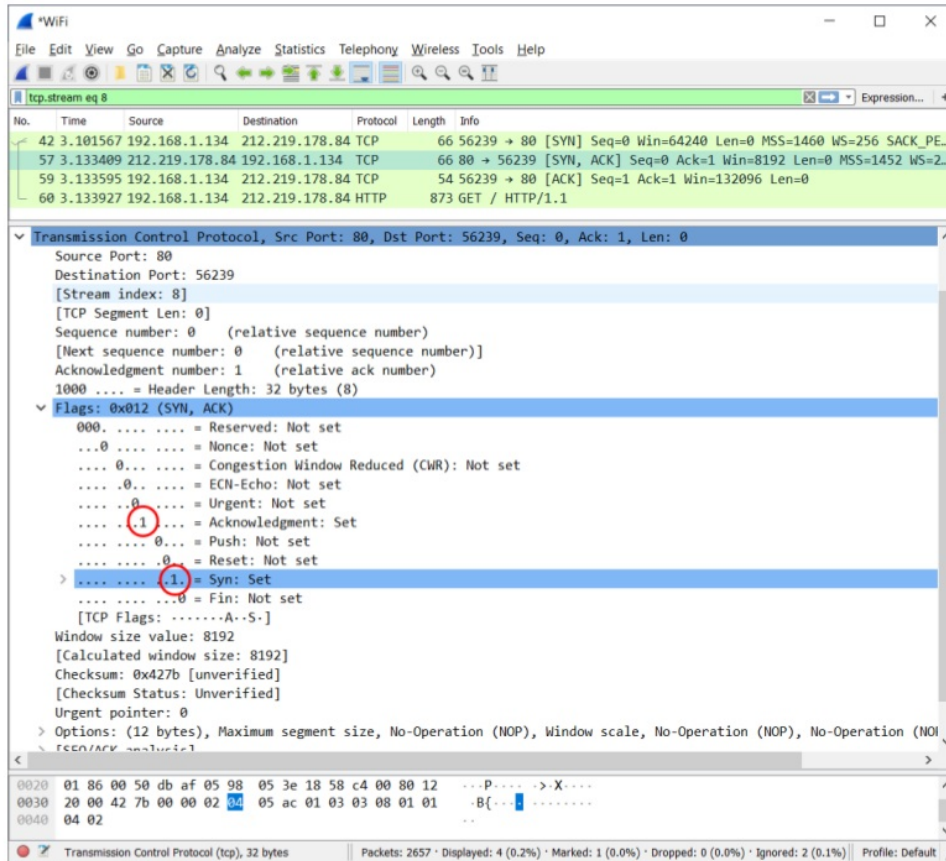


Figure 4-5 SYN and ACK flags set

4.1.1.3 ACK

The client further responds with an acknowledgement (ACK) to the servers' request for synchronisation (Figure 4-6), completing the three-way handshake. The HTTP GET request can then be completed and the website displayed.

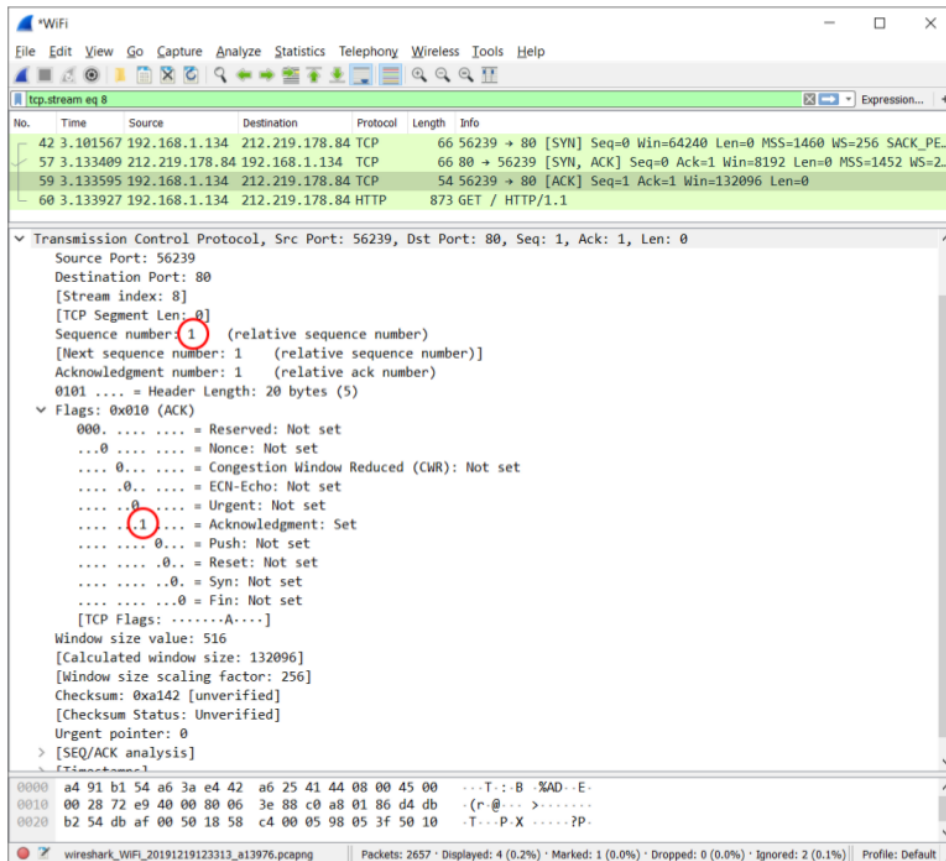


Figure 4-6 ACK flag set

4.2 User Datagram Protocol

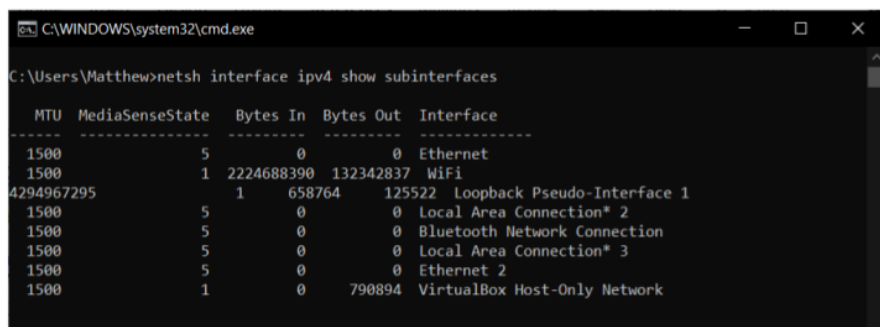
UDP is another transport layer protocol (defined by RFC 768 [14]) that is used on the IP network layer. However contrasting TCP, UDP is an unreliable and connectionless protocol with no concept of windowing, packet re-transmission, packet acknowledgement or flow control [15].

It can be argued that due to the lack of acknowledgement and re-transmission, the efficiency can be improved as UDP was developed for time-sensitive applications, however, on the other hand the reliability is greatly reduced and the lack of data integrity may increase due to the possibility of packet loss [16, pp. 176–177].

5 Maximum Transmission Unit

The maximum transmission unit (MTU) is the maximum protocol data unit (PDU) size, without requiring fragmentation, that may be sent across a link in one datagram [6, p. 228], [17, p. 206], [18].

There is a typical size of 1500 bytes for the Ethernet MTU (Figure 5-1), but the headers applied on the lower layers need to be taken into consideration when calculating the actual transmission size (MTU – (total size of headers)).



MTU	MediaSenseState	Bytes In	Bytes Out	Interface
1500	5	0	0	Ethernet
1500	1	2224688390	132342837	WiFi
4294967295	1	658764	125522	Loopback Pseudo-Interface 1
1500	5	0	0	Local Area Connection* 2
1500	5	0	0	Bluetooth Network Connection
1500	5	0	0	Local Area Connection* 3
1500	5	0	0	Ethernet 2
1500	1	0	790894	VirtualBox Host-Only Network

Figure 5-1 Windows MTU size

For example, if trying to utilise ICMP for network diagnostics, the maximum payload would be 1472, as an IPv4 header is 20 bytes, and an ICMP header is 8 bytes. As shown in Figure 5-2, the packet size is 1473 so this packet has been fragmented for that 1 byte over. Taking the case of TCP, as per Figure 5-3 the IPv4 header is again 20 bytes and the TCP header is also 20 bytes, so the maximum payload would be 1460 bytes.

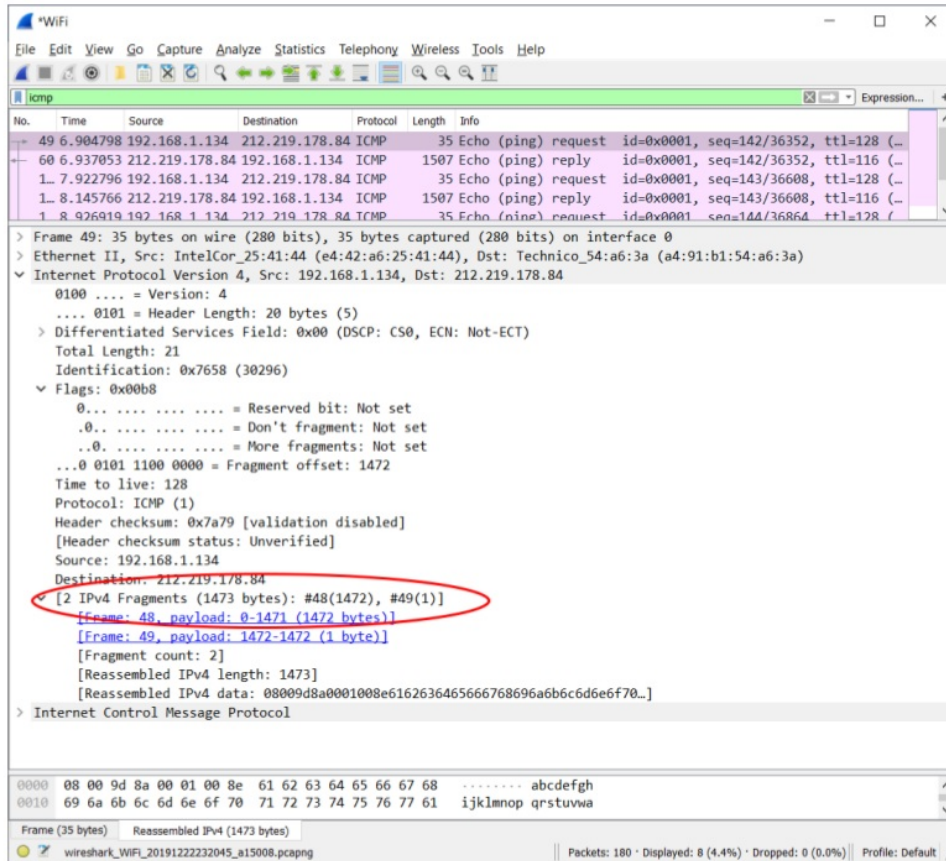


Figure 5-2 Wireshark capture of packet fragmentation

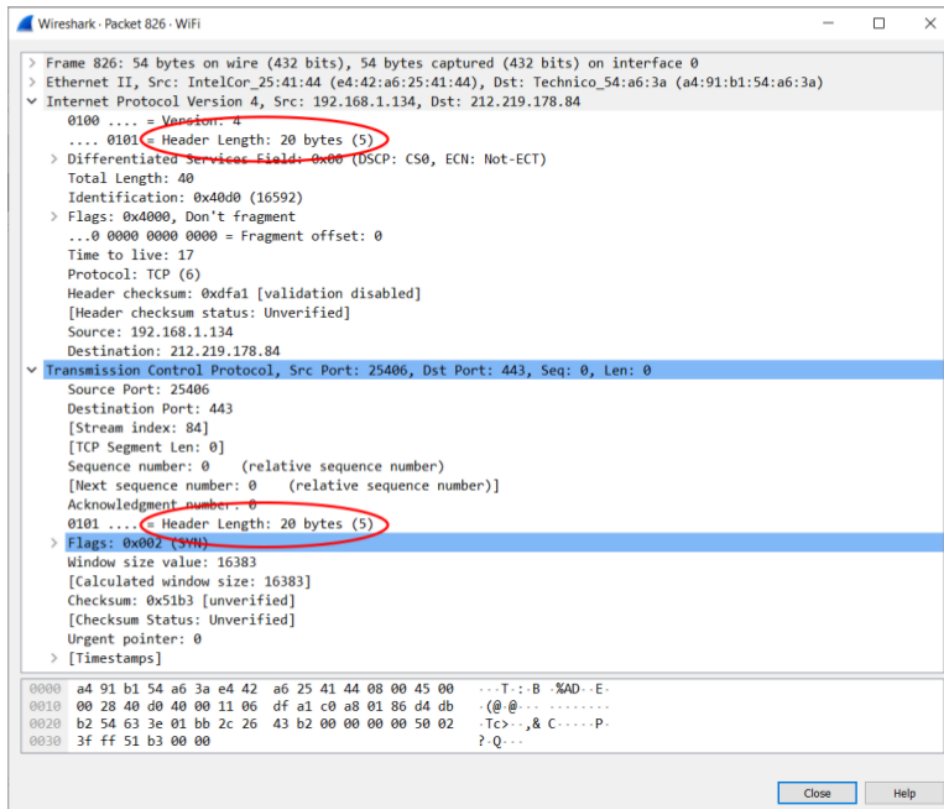


Figure 5-3 TCP and IPv4 Header length

5.1 IP Fragmentation

IP fragmentation will take place if the packet size is larger than the MTU (as above in Figure 5-3 TCP and IPv4 Header length) and can occur at either the originating device, or any router as the packet transitions across the network.

Fragmentation is the process of splitting the datagram up into smaller fragments, so each fragment is smaller than the specified MTU size. The reason fragmentation exists for example is that, as previously stated, an Ethernet typical MTU is 1500 bytes, whereas an IPv6 packet on an FDDI (Fibre Distributed Data Interface) network can be up to 4500 bytes as per RFC 2467 [19], [20, p. 140]. Either every IP datagram would need to be small enough to fit inside one packet, or there would need to be some means of splitting the datagram down to a usable size [21, pp. 210–211].

When a datagram is fragmented, it is re-assembled only once it reaches its end host as these fragmented packets may not all take the same route so

there is no need for them to be re-assembled during network traversal [7, p. 488]. During re-assembly if there are any missing fragments the end recipient will still attempt to re-assemble the datagram, though it will fail. This is a prime reason as to why it is recommended to avoid IP fragmentation, and another process is encouraged called PMTUD (Path MTU Discovery) [21, p. 213] which can help avoid the increase in computational overheads on the source end and routers or devices, as well as avoid the potential security risks that come with IP fragmentation [22].

PMTUD simply determines the lowest MTU across the network path going from the packet's source to its destination, and that value can instead be used for the MTU size to try to combat IP fragmentation.

6 Synchronicity and Asynchronicity in Network Protocols

One way to characterise protocols can be defined by whether it is asynchronous or synchronous. The definition of these terms can vary depending on the layer where they are used, but in the transport layer they can be thought of as two opposite protocols with opposing data transmission channels [21, p. 439].

6.1 Data Transmission Channels

There are three data transmission channels offered which are simplex, half duplex and full duplex [23, pp. 337–338], [24, pp. 701–703]:

Simplex: Data transmission is always in one direction with the receiving device having no option of transmitting data back to the originating device. An example of this would be a radio or television broadcasting to its audience, but the audience cannot respond to it.

Half duplex: Data is able to flow in both directions, but only in one direction at a time. A walkie talkie is a half-duplex device where either the transmitter is on and receiver is off, or the transmitter is off and the receiver is on.

Full duplex: Data is able to flow in both directions simultaneously, such as a telephone where both parties are able to communicate at the same time.

6.2 Synchronous and Asynchronous I/O Operations

I/O operations can be either synchronous or asynchronous. The main difference between the two (Figure 6-1) is that with a synchronous operation (thread), once the operation starts it will enter a wait state until the I/O operation is completed. With an asynchronous operation, the thread performs an `DeviceIoControl` request (an `IOCTL` interface where an application can communicate directly with a driver [25]) to the I/O subsystem of the kernel [26], [27] and, if successful, carries on processing other jobs and does not wait until the I/O operation is completed until the kernel signals said I/O operation is complete.

Asynchronous processes would be well suited to an application that is potentially dealing with several connections and/or processes at the same time such as a database performing CRUD operations with large datasets.

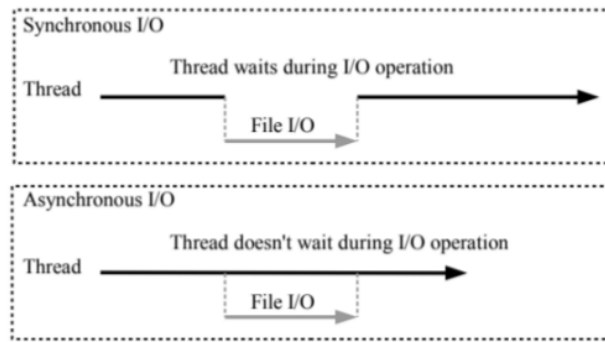


Figure 6-1 Synchronous and Asynchronous I/O difference [28]

6.2.1 I/O Control and Processing

The I/O manager which is part of the I/O subsystem controls the I/O request packets (IRP) which are created in response to the I/O requests. The IRP controls the processing of all I/O operations [29] and is handled by a dispatch routine [26], [30].

7 Interoperability in Networked Applications

The definition of Interoperability is the ability for two or more systems, services or components (whether these are of the same organisation or from different organisations, allowing collaboration and encouraging co-creation, irrespective of their location) to connect, exchange data, and to process the exchanged data between each other in a distributed and heterogenous environment [31]–[33].

7.1 Interoperability Frameworks

There exist multiple interoperability frameworks. Two of which are the Interop Network of Excellence framework for Enterprise Interoperability [34] and ATHENA (Advanced Technologies for Interoperability of Heterogenous Enterprise Networks and their Applications) Interoperability Framework [35].

There are four layers to these frameworks [36]:

Semantic Layer: The semantic layers focus is on the format that the information is to be defined and exchanged by, and the information semantics.

Technical Layer: The technical layer is concerned with the applications and infrastructure that links the services and systems together such as interface specifications, interconnection services, data integration and data presentation.

Organisational Layer: The way that system processes, responsibilities and procedures are aligned.

Legal Layer: The primary concern of the legal layer is involved with describing legislations of the organisational layer.

7.2 Achieving Interoperability in Networked Applications

Achieving interoperability comprises of various prerequisites that need to be satisfied which define the needs of stakeholders and describe the systems which must comply for interoperability consideration [36], [37].

The largest challenge of interoperability is sustainability; the interoperability that convenes the needs of the present without compromising the ability of future changes, the ability to meet future system requirements, and being able to adapt to new elements [38].

Developed networked applications should ideally meet the requirements to be sustainable to allow easy integration and for future updates to be initiated without causing problems to any current or legacy code.

8 Conclusion

There exist various protocols on different layers that achieve different things, however TCP is a protocol that can allow networked application developers to establish and utilise reliable connections. The handshake is an important factor to achieve reliability along with additional parameters such as sequence numbering, checksums, etc...

There is a discernible difference between synchronous and asynchronous operations, and although asynchronous processes seem to be a choice for large problems, it can potentially be excess for smaller problems.

Code that is written, and applications that are produced should be sustainable by allowing integration into already existing systems in order to help achieve interoperability in networked applications.

9 References

- [1] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. Commun.*, vol. 28, no. 4, pp. 425–432, 1980.
- [2] S. Orzen, "Interaction understanding in the OSI model functionality of networks with case studies," in *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2014, pp. 327–330.
- [3] S. Al-Fedaghi, A. Alasqa, and Z. Fadel, "Conceptual Model for Communication," *Int. J. Comput. Sci. Inf. Secur.*, vol. 6, no. 2, pp. 29–41, 2009.
- [4] "ISO - 35.100 - Open systems interconnection (OSI)." [Online]. Available: <https://www.iso.org/ics/35.100/x/>. [Accessed: 17-Dec-2019].
- [5] P. Ravali, "A Comparative Evaluation of OSI and TCP/IP Models," *Int. J. Sci. Res.*, vol. 4, no. 7, pp. 514–521, 2015.
- [6] E. Conrad, S. Misenar, and J. Feldman, *CISSP Study Guide*, 3rd ed. Syngress, 2015.
- [7] K. Fall, R. Stevens, and G. Wright, *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd ed. Addison-Wesley, 2011.
- [8] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Trans. Commun.*, vol. 22, no. 5, pp. 637–648, 1974.
- [9] D. Clark, "The design philosophy of the DARPA Internet Protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 1, pp. 102–111, 1995.
- [10] IETF, "Transmission Control Protocol (RFC 793)." RFC Editor, 1981.
- [11] D. Hanes, G. Salguiero, P. Grosettete, R. Barton, and J. Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press, 2017.
- [12] F.-H. Hsu, Y.-L. Hwang, C.-Y. Tsai, W.-T. Cai, C.-H. Lee, and K. Chang, "TRAP: A Three-Way Handshake Server for TCP Connection Establishment," *Appl. Sci.*, vol. 6, no. 11, p. 358, 2016.
- [13] "Wireshark." [Online]. Available: <https://www.wireshark.org/>. [Accessed: 19-Dec-2019].
- [14] IETF, "User Datagram Protocol (RFC 768)." RFC Editor, 1980.

- [15] S. H. Ali, S. A. Nasir, and S. Qazi, "Impact of router buffer size on TCP/UDP performance," in *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*, 2013, pp. 1–6.
- [16] L. Huawei Technologies Co., Ed., *HCNA Networking Study Guide*. Singapore: Springer Singapore, 2016.
- [17] W. Goralski, *The Illustrated Network: How TCP/IP Works in a Modern Network*, 2nd ed. Morgan Kaufmann, 2017.
- [18] A. Custura, G. Fairhurst, and I. Learmonth, "Exploring Usable Path MTU in the Internet," in *2018 Network Traffic Measurement and Analysis Conference (TMA)*, 2018, pp. 1–8.
- [19] IETF, "Transmission of IPv6 Packets over FDDI Networks (RFC 2467)." RFC Editor, 1998.
- [20] S. Hagen, *IPv6 Essentials*, 2nd ed. O'Reilly Media, 2006.
- [21] L. Peterson and B. Davie, *Computer Networks: A Systems Approach*, 5th ed. Morgan Kaufmann, 2011.
- [22] M. Göhring, H. Shulman, and M. Waidner, "Path MTU Discovery Considered Harmful," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 866–874.
- [23] S. Mishra, N. K. Singh, and V. Rousseau, *System on Chip Interfaces for Low Power Design*, 1st ed. Morgan Kaufmann, 2015.
- [24] P. Zhang, *Industrial Control Technology*, 1st ed. William Andrew, 2008.
- [25] "Device Input and Output Control." [Online]. Available: <https://docs.microsoft.com/en-us/windows/win32/devio/device-input-and-output-control-ioctl->.
- [26] T. Ni, Z. Yin, Q. Wei, and Q. Wang, "High-Coverage Security Testing for Windows Kernel Drivers," in *2012 Fourth International Conference on Multimedia Information Networking and Security*, 2012, pp. 905–908.
- [27] "Synchronous and Asynchronous I/O." [Online]. Available: <https://docs.microsoft.com/en-us/windows/win32/fileio/synchronous-and-asynchronous-i-o>.
- [28] D. Wellisch, S. Kunze, and R. Poschl, "A modular software implementation for smart charging stations," in *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, 2015, pp. 254–259.
- [29] JingLv Zhou and Hongyu Zhang, "Study of IRP for windows driver model,"

in *2007 International Conference on Communications, Circuits and Systems*, 2007, pp. 521–524.

- [30] "Writing IRP Dispatch Routines." [Online]. Available: <https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/writing-irp-dispatch-routines>.
- [31] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. Shams Aliee, "Interoperability evaluation models: A systematic review," *Comput. Ind.*, vol. 65, no. 1, pp. 1–23, Jan. 2014.
- [32] F. B. Vernadat, "Technical, semantic and organizational issues of enterprise interoperability and networking," *Annu. Rev. Control*, vol. 34, no. 1, pp. 139–144, 2010.
- [33] A. Tolk, "Interoperability, Composability, and Their Implications for Distributed Simulation: Towards Mathematical Foundations of Simulation Interoperability," in *2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, 2013, pp. 3–9.
- [34] H. Panetto, M. Scannapieco, and M. Zelm, "INTEROP NoE: Interoperability Research for Networked Enterprises Applications and Software BT - On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops," 2004, pp. 866–882.
- [35] A.-J. Berre *et al.*, "The ATHENA Interoperability Framework - Enterprise Interoperability II," 2007, pp. 569–580.
- [36] G. da Silva Serapião Leal, W. Guédria, and H. Panetto, "Interoperability assessment: A systematic literature review," *Comput. Ind.*, vol. 106, pp. 111–132, Apr. 2019.
- [37] G. da Silva Serapião Leal, W. Guédria, and H. Panetto, "An ontology for interoperability assessment: A systemic approach," *J. Ind. Inf. Integr.*, vol. 16, Dec. 2019.
- [38] C. Agostinho *et al.*, "Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology," *Comput. Ind.*, vol. 79, pp. 64–76, Jun. 2016.

Network Theory

GRADEMARK REPORT

FINAL GRADE

92/100

GENERAL COMMENTS

Instructor

A large number of appropriate sources used. An excellent report overall, which does a very good job of digging into the core concepts.

The presentation of your report is particularly nice - the use of images and diagrams is excellent. Very well done Matthew, an extremely high quality of work, worthy of a high first.

Engagement with Literature Skills (20): 19

Knowledge and Understanding Skills (30): 28

Cognitive and Intellectual Skills (30): 27

Transferable Skills for Life and Professional Practice (20): 18

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25
