

# Analys av överföringshastigheter över olika länktyper

En studie som undersöker prestandan hos olika nätverkstyper

Kandidatarbete inom institutionen Data- och informationsteknik

JOHNY AFREM  
DAVID AZIZI  
NIK ERLANDSSON  
SEBASTIAN RIZK GUSTAVSSON  
MOHAMMAD MOURAD  
OSKAR SWÄRD

**INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK**

CHALMERS TEKNISKA HÖGSKOLA | GÖTEBORGS UNIVERSITET

Göteborg 2024

[www.chalmers.se](http://www.chalmers.se)

[www.gu.se](http://www.gu.se)



KANDIDATARBETE 2024

# Analys av överföringshastigheter över olika länktyper

En studie som undersöker prestandan hos olika nätverkstyper

JOHNY AFREM  
DAVID AZIZI  
NIK ERLANDSSON  
MOHAMMAD MOURAD  
SEBASTIAN RIZK GUSTAVSSON  
OSKAR SWÄRD



GÖTEBORGS  
UNIVERSITET

---



**CHALMERS**

Institutionen för Data- och informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA | GÖTEBORGS UNIVERSITET  
Göteborg 2024

Analys av överföringshastigheter över olika länktyper  
En studie som undersöker prestanda på olika nätverkstyper  
JOHNY AFREM  
DAVID AZIZI  
MOHAMMAD MOURAD  
NIK ERLANDSSON  
SEBASTIAN RIZK GUSTAVSSON  
OSKAR SWÄRD

© JOHNY AFREM, 2024.  
© DAVID AZIZI, 2024.  
© MOHAMMAD MOURAD, 2024.  
© NIK ERLANDSSON, 2024.  
© SEBASTIAN RIZK GUSTAVSSON, 2024.  
© OSKAR SWÄRD, 2024.

Handledare: Romaric Duvignau, Institutionen för Data- och Informationsteknik  
Rättande lärare: Arne Linde, Institutionen för Data- och Informationsteknik  
Examinator: Patrik Jansson och Arne Linde, Institutionen för Data- och Informationsteknik

Kandidatarbete 2024  
Institutionen för Data- och Informationsteknik  
Chalmers Tekniska Högskola  
Göteborgs Universitet  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagsbild: Trädtopologi av ett ethernetanslutet nätverk med 38 Raspberry Pi enheter.

Skriven i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2024

# Abstract

With the rapid increase in internet connected devices it's become more and more important to use the best link type for the application. The network link types Ethernet, Wi-Fi and 5G have their own characteristic attributes which gives them their different uses and qualities. The aims of this report is to analyze the download speeds that these link types enable in a network composed of IoT devices, and to see how well the link types scale as the network grows larger. Experimentation with the devices occurred in incremental steps by first testing 4 devices, then 16 devices, then 32 devices and so forth up to 73 devices. The result showed that using Ethernet to transfer data in a network would be the fastest option and that 5G was the best option if wireless connectivity is needed. This result confirms the notion that using Ethernet over other link types for connectivity in a demanding network is the preferred option. However, due to limits regarding the technology used it was not possible to test mmWave-5G or a pure 5G network. The work developed during this thesis can be expanded to a distributed computing setting and can allow for the establishment of computational heavy applications such as Federated Learning algorithms.

Keywords: 5G, Wi-Fi, Ethernet, Internet of Things, IoT.

# Sammanfattning

Med den snabba ökningen av internetanslutna enheter har det blivit viktigare att använda den bästa länktypen för applikationen. Nätverkslänktyperna Ethernet, Wi-Fi och 5G har sina egna karaktäristiska attribut som ger dem deras olika användningsområden och kvaliteter. Syftet med denna rapport är att analysera nedladdningshastigheterna som dessa länktyper möjliggör i ett nätverk bestående av IoT-enheter och att se hur väl länktyperna skalas när nätverket växer. Experiment med enheterna skedde stegvis genom att först testa 4 enheter, sedan 16 enheter, sedan 32 enheter och så vidare upp till 73 enheter. Resultatet visade att användning av Ethernet för att överföra data i ett nätverk skulle vara det snabbaste alternativet och att 5G var det bästa alternativet om trådlös anslutning krävs. Detta resultat bekräftar uppfattningen att användning av Ethernet över andra länktyper för anslutning i ett krävande nätverk är det föredragna alternativet. På grund av begränsningar i tekniken som användes var det inte möjligt att testa mmWave-5G eller ett rent 5G-nätverk. Arbetet som utvecklats under detta projekt kan utökas till en distribuerad datormiljö och kan möjliggöra etableringen av en federerad inlärnings-algoritm.

Nyckelord: 5G, Wi-Fi, Ethernet, Internet of Things, IoT.

## Förord

Vi skulle vilja tacka vår handledare Romaric Duvignau för all hjälp vi har fått under detta kandidatarbete. Med ditt stöd har vi kunnat navigera oss runt alla svårigheter i arbetsprocessen. Vi vill även tacka Arne Linde för att ha gett oss tillgång till utrustningen vi har behövt för experimenten under alla tidiga morgnar. Ett stort tack till Institutionen för Elektroteknik på Chalmers för att vi fick använda CASE-labbets utrustning under hela projektets gång. Detta arbete använder resurser som utvecklades av Samuel Bach, Albin Karlsson, Jan Körner och Linus Sundkvist under deras arbete: *Monitoring IoT/CPS*. Vi tackar dem för inspirationen. Vi vill även tacka Eben Upton, VD för Raspberry Pi Foundation, för att vi får använda Raspberry Pi-loggan.





# Ordlista

Nedan är en lista av begrepp, akronymer och förkortningar som används i rapporten:

5G	Inom detta projekt åsyftar 5G på den 5:e generationens mobilnätverk.
Accesspunkt	Hårdvara som möjliggör för Wi-Fi enheter att koppla upp sig till ett nätverk.
Bandbredd	Överföringshastigheten för en länk.
Bash	Bourne Again Shell, en kommandotolk för Linux, med stöd för kortare program.
Datarensning	Borttagning av korrupt och eller orepresentativ data
HTTP	Hyper Text Transfer Protocol, ett protokoll för att skicka filer över internet.
Hyper-V	Virtualiseringsplattform utvecklad av Microsoft.
IP adress	Internet Protokoll adress, ett unikt nummer som identifierar en dator eller ett nätverk.
Kantnoder	Ibland även kallat Slutnoder eller Ändnoder, Är de datorer eller servrar som nätverket kopplar ihop.
Router	En enhet som dirigerar datapaket mellan olika nätverk. Den används för att koppla samman olika enheter i ett nätverk och möjliggör kommunikation mellan dem.
Subnät	Ett prefix till ett nätverk som används för att urskilja ett nätverk från resten av internet.

---

SSH	Secure Shell, ett protokoll som möjliggör att ansluta datorer med varandra över ett nätverk.
Wi-Fi channel	En del av Wi-Fi spektrumet som används för ett visst nätverk.
WLAN	Wireless local area network. Ett lokalt nätverk där enheterna kommunicerar utan fysisk anslutning.

# Innehåll

<b>Ordlista</b>	<b>ix</b>
<b>Figurer</b>	<b>xiii</b>
<b>1 Inledning</b>	<b>1</b>
1.1 Bakgrund . . . . .	1
1.1.1 Betydelsen av länktyper . . . . .	2
1.1.2 Introduktion till 5G . . . . .	2
1.2 Syfte och mål . . . . .	2
1.3 Avgränsningar . . . . .	3
1.3.1 Säkerhet . . . . .	3
1.3.2 Länktyper . . . . .	3
<b>2 Teknisk Bakgrund</b>	<b>5</b>
2.1 Länktyper . . . . .	5
2.1.1 Ethernet . . . . .	5
2.1.2 Wi-Fi . . . . .	6
2.1.3 5G . . . . .	6
2.2 Federerad inlärning . . . . .	7
2.3 Mjukvara . . . . .	8
2.3.1 Git . . . . .	8
2.3.2 Virtual Private Server (VPS) . . . . .	8
2.3.3 Python och viktiga bibliotek . . . . .	8
2.3.4 WSL . . . . .	9
2.3.5 Linux-distributioner . . . . .	9
2.3.6 Comma-separated values-filer (CSV) . . . . .	10
2.4 Hårdvara . . . . .	10
2.4.1 5G-router . . . . .	10
2.4.2 Wi-Fi-router . . . . .	10
2.4.3 Switchar . . . . .	10
2.4.4 Kablar . . . . .	11
2.4.5 Raspberry Pi . . . . .	11
2.5 Nätverkstopologi . . . . .	11
2.5.1 Stjärntopologi . . . . .	11
2.5.2 Linjetopologi . . . . .	12
2.5.3 Trädtopologi . . . . .	12

<b>3</b>	<b>Genomförande</b>	<b>15</b>
3.1	Efterforskning . . . . .	15
3.2	Implementering av applikation . . . . .	15
3.2.1	Val av testdata . . . . .	16
3.2.2	Utveckling av Bash-skript . . . . .	17
3.3	Testning . . . . .	18
3.4	Verktyg och materiel . . . . .	21
3.4.1	Mjukvara och versionhantering . . . . .	21
3.4.2	Hårdvara . . . . .	21
<b>4</b>	<b>Resultat</b>	<b>23</b>
4.1	Samlad data från Ethernet . . . . .	23
4.2	Samlad data från Wi-Fi . . . . .	27
4.2.1	2.4GHz Wi-Fi . . . . .	27
4.2.2	5GHz Wi-Fi . . . . .	27
4.3	Samlad data från 5G . . . . .	30
4.4	Data för upskalning av nätverket . . . . .	32
4.5	Jämförbar data för en fast konfiguration . . . . .	34
<b>5</b>	<b>Diskussion</b>	<b>37</b>
5.1	Ethernet . . . . .	37
5.2	Wi-Fi . . . . .	37
5.3	5G . . . . .	38
5.4	Uppskalning av nätverket . . . . .	39
5.5	Fast konfiguration . . . . .	39
5.6	Jämförelse mellan länktyperna . . . . .	39
5.7	Felkällor . . . . .	40
5.7.1	5G-nätverket . . . . .	40
5.7.2	Varierande skrivhastigheter . . . . .	40
5.7.3	Varierande nätverkshastigheter . . . . .	40
5.8	Utmaningar . . . . .	41
5.8.1	Portabilitet av Bash-skriptet . . . . .	41
5.8.2	Problem vid testning via Wi-Fi . . . . .	41
5.8.3	För få SIM-kort . . . . .	42
5.8.4	5G-nätverket . . . . .	42
5.9	Samhälleliga och etiska aspekter . . . . .	42
5.9.1	Integritet och IoT . . . . .	43
5.9.2	Klasskillnader . . . . .	43
5.9.3	Positiva aspekter . . . . .	43
5.10	Användningsområden . . . . .	43
5.11	Reflektioner och framtida utvecklingsområden . . . . .	44
<b>6</b>	<b>Slutsats</b>	<b>45</b>
	<b>Litteraturförteckning</b>	<b>50</b>

# Figurer

2.1	Exempel på struktur av ett nätverk med stjärntopologi. . . . .	12
2.2	Exempel på struktur av ett nätverk med linjetopologi. . . . .	12
2.3	Exempel på struktur av ett nätverk med trädtopologi. . . . .	13
3.1	Ethernet experiment på 73 Raspberry Pi. Varje switch har 4 enheter kopplade till sig, utom en som har 1 enhet. . . . .	19
3.2	5G experiment på 64 Raspberry Pi-datorer. . . . .	20
4.1	Ethernet, 1Gbps, Datamängd, Trädtopologi, 38 enheter. . . . .	24
4.2	Ethernet, 1Gbps, Datamängd, Linjetopologi, 38 enheter. . . . .	24
4.3	Ethernet, 1Gbps, Skivavbild, Trädtopologi, 38 enheter. . . . .	25
4.4	Ethernet, 1Gbps, Skivavbild, Linjetopologi, 38 enheter. . . . .	25
4.5	Ethernet, 1Gbps, Datamängd, Trädtopologi, 73 enheter. . . . .	26
4.6	Ethernet, 1Gbps, Datamängd, Trädtopologi, 4 enheter. . . . .	27
4.7	Wi-Fi, 2.4GHz, Datamängd, 37 enheter. . . . .	28
4.8	Wi-Fi, 5GHz, Datamängd, 37 enheter. . . . .	28
4.9	Wi-Fi, 5GHz, Datamängd, 19 enheter. . . . .	29
4.10	Wi-Fi, 5GHz, Skivavbild, 19 enheter. . . . .	29
4.11	Wi-Fi, 5GHz, Datamängd, 19 enheter. . . . .	30
4.12	5G, Sub 6GHz, FR1, Datamängd, 63 enheter. . . . .	31
4.13	5G, Sub 6GHz, FR1, Datamängd, 32 enheter. . . . .	32
4.14	Linjediagram för olika länktyper med total nedladdningstid på y- axeln och antal enheter på x-axeln . . . . .	33
4.15	Ethernet, 1Gbps, Datamängd, Trädtopologi, 8 enheter. . . . .	34
4.16	5G, Sub 6GHz, FR1, Datamängd, 8 enheter. . . . .	34
4.17	Wi-Fi, 5GHz, Datamängd, 8 enheter. . . . .	35



# 1

## Inledning

Under inledningen beskrivs kort 5G-teknologin och dess sammanhang i samhället som en bakgrund för rapporten. Detta följs med kandidatarbetets syfte och efter det nämns avgränsningarna.

### 1.1 Bakgrund

Världen har blivit alltmer beroende av internet och digitala tjänster [1]. Detta har lett till att det finns fler smarta enheter i hushållen som är kopplade till internet. Dessa enheter kan vara rörelsesensorer, kamerövervakning, och smarta säkerhetssystem. Dessa är i sin tur kopplade till ett lokalt nätverk, där data kan skickas mellan enheterna. Eftersom utvecklingen med smarta mobiltelefoner, datorer och kommunikation skett snabbt dök begreppet sakernas internet upp eller på engelska *internet of things* (IoT). Detta begrepp åsyftar just vardagliga saker såsom klockor, lampor och dylika föremål som är sammankopplade och utbyter information [2].

Då teknologisektorn fortsätter att växa, medföljer detta en ökande konsumtion samt användning av elektronik [3]. Statistik visar att under år 2022 använde cirka 66% av världens befolkning internet. Detta kan jämföras med år 2005 där denna procentsats ungefär var 16% [4][5]. Allt fler tjänster digitaliseras som till exempel bankärenden och kommunikation. Därför är det viktigt att internet alltid fungerar som förväntat och att information och funktioner finns tillgängliga. Med ökad användning ökas även mängden trafiken på nätverken vilket kan leda till överbelastning. Därmed är det i vår samtid viktigt att se till att metodiken för datautbyte mellan enheter är effektiv för att således inte överbelasta nätverken [6].

Under 1980-talet introducerades 1G, den första generationens mobilnät, vilket gjorde det möjligt att ha mobila telefonsamtal. Sedan dess har nya generationer av nätverk utvecklats och idag är den femte generationens mobilnät, 5G, den mest moderna versionen. Med populariseringen av IoT-enheter krävs det att kommunikation mellan enheter fungerar bra. Detta är något som 5G kan vara användbart till då 5G fokuserar på överföringshastighet, tillgänglighet och effektivitet [7].

5G-teknologin är lovande och kan neutralisera framträdande problem inom mobiltelefoni när fler får tillgång till det [8]. Ett sätt att öka mottagligheten av teknologin är att visa konkret data där tidigare teknik jämförs med 5G. De två följande underrubrikerna beskriver 5G närmare och varför det är ett intressant område att utforska.

### 1.1.1 Betydelsen av länktyper

Allt fler människor använder sig av internet. Tjänster och ärenden som tidigare hanterades fysiskt görs nu digitalt [1]. Det är därför önskevärt att ha en stabil anslutning till internet och ett kännetecken på en god anslutning är upp- och nedladdningshastighet. Ett studieområde inom optimeringen av en snabbare uppkoppling är användningen av en lämplig länktyp mellan enheterna i nätverket. De tre länktyperna som kommer att studeras i denna rapport är: Ethernet, Wi-Fi och 5G.

### 1.1.2 Introduktion till 5G

5G står för femte generationens mobilnät, och är menad att vara en förbättring till den föregående generationens mobilnät, 4G. Förbättringarna innebär att 5G förväntas förse fler användare med en snabbare samt mer pålitlig uppkoppling kontra 4G.

Vid utveckling av nästa generations mobilnät har det lagts till funktionalitet vilket innebär att 4G är en delmängd av 5G [9]. Övergången från 4G av 5G blir då smidigare, eftersom äldre enheter inte behövs bytas ut mot nya vilket underlättar samhällets digitaliseringsprocess [10]. Äldre enheter kommer fortfarande eventuellt behövas bytas ut men alla enheter kommer inte behövas byta ut samtidigt.

År 2022 använde företagen Ericsson, Huawei och Nokia sammanlagt ungefär 32,5 miljarder amerikanska dollar för forskning och utveckling inom 5G [11]. En stor anledning bakom intresset till utvecklingen är den ökade efterfrågan efter snabbare och mer effektiv nätverkskommunikation. Efterfrågan beräknas fortsätta öka med populariseringen av ”smarta” apparater som kräver internetuppkoppling [12].

Med ankomsten av 5G förväntas fortare upp- och nedladdningshastigheter, stabil prestanda under tung belastning och lägre latens. Dess kapacitet ska även kunna tillåta storskalig introduktion av IoT-enheter, vilket kan innebära nya innovationer inom till exempel sjukvården och säkerhetsövervakning [13][14]. Flera operatörer runt om världen har investerat i installationer av 5G nätverk [15].

## 1.2 Syfte och mål

Syftet med detta kandidatarbete är att undersöka hur nedladdningshastigheten ser ut på länktyperna Ethernet och Wi-Fi jämfört med 5G och hur väl dessa länktyper presterar då mängden slutnoder skapas upp. Målet är att kunna dra en slutsats kring vilken länktyp som presterar bäst i olika sammanhang. Slutsatsen kommer att baseras på mätdata som samlas in vid laborationer, som huvudsakligen kommer bestå av tidsåtgång vid nedladdning av filer av olika storlekar.



## 1.3 Avgränsningar

Detta projekt täcker Ethernet, Wi-Fi och 5G-nätverk med Raspberry Pi-datorer som kantnoder. På grund av projektets tidsrestriktioner kommer följande avgränsningar att göras.

### 1.3.1 Säkerhet

Nätverketssäkerheten har inte prioriterats. Extra tillägg på mjukvara för ökad digital säkerhet leder till högre belastning på nätverket. Dataöverföringen är redan intensiv och krävande, utan säkerhetsåtgärder som brandvägg och kryptering. Eftersom känsligt data inte hanteras i nätverket finns det inte något behov för säkerhetsåtgärder som inte redan är inbyggd på system som används under projektet.

### 1.3.2 Länktyper

Länktypena 5G, Wi-Fi, och Ethernet har undersökts under projektets gång. 5G valdes eftersom det är den nyaste generationens mobilnät, Ethernet och Wi-Fi valdes för att ha något att jämföra 5G med och för att de är väldigt vanliga inom både konsument- och företagsnätverk. För laborationerna med Wi-Fi har Wi-Fi 5 (IEEE 802.11ac) använts eftersom det är den standard som stöds av Raspberry Pi 4-serien. För Ethernet användes en länkhastighet på 1Gbps (1000BASE-T). 5G-routarna som gruppen har till sitt förfogande har inte stöd för mmWave-5G, därav testades endast "vanlig" 5G (Frequency Range 1). Se sektion 2.4 för fler detaljer om vilken hårdvara som har använts.



# 2

## Teknisk Bakgrund

I det tidigare kapitlet introducerades länktyperna 5G, Wi-Fi och Ethernet. Under detta kapitlet beskrivs kortfattat länktypernas historia, karakteristiska egenskaper samt deras relation till varandra. Kapitlet börjar med att berätta om de tre länktyperna som kommer att testas, och därefter går den över information gällande den mjukvara och hårdvara som har använts i detta arbete. Det kompletteras sedan med relevant teori kopplat till arbetsområdet. Beskrivningarna är endast ur perspektivet av detta arbetes syfte då helheten kan innehålla information som inte är relevant i detta sammanhang.

### 2.1 Länktyper

De tre länktyperna har olika egenskaper vilket ger olika fördelar och nackdelar. Länktyperna Wi-Fi och 5G är båda trådlösa teknologier som har överlappande egenskaper, medan Ethernet ansluter enheter genom kablar.

#### 2.1.1 Ethernet

Ethernet är ett samlingsnamn för olika kabelteknologier som används för att ansluta nätverksenheter med varandra inom ett lokalt nätverk, *Local Area Network* (LAN). Tekniken släpptes ursprungligen på tidigt 1970-tal och det använde sig av koaxiala kablar som är gjorda av koppar med ett isolerande plaströr. Ethernet framställdes för att tillåta högre dataöveringshastigheter än radiokommunikation samt att undvika utmaningarna som uppstod då signaler överlappade med varandra [16].

En av anledningarna till varför Ethernet kan överföra data fortare än trådlös kommunikation, som till exempel Wi-Fi, är att Ethernet är en direkt anslutning till enheten och är därmed inte en delad media mellan flera enheter. Eftersom signalen färdas via en direktanslutning är den mer resistent mot störningar som Wi-Fi behöver åtgärda. Störningar innefattar till exempel fysiska hinder, egna speglade signaler och radiovågor från andra källor som skapar interferens. Moderna Ethernet-varianter kommunicerar via full-duplex, vilket innebär att det går att skicka och ta emot data samtidigt, och har betydligt mindre störningskällor än kommunikation via radiovågor [17][18].

Den första versionen av Ethernet hade en maxhastighet på 10 megabit per sekund (Mbps). Idag är gigabit Ethernet en vanlig standard, med hastigheter upp till 1

gigabit per sekund (Gbps). Det finns också snabbare varianter som 10 gigabit Ethernet och 100 gigabit Ethernet för företag och datacenter som kräver ännu högre hastigheter [19].

### 2.1.2 Wi-Fi

Wi-Fi är en samling av olika standarder för trådlöst anslutna enheter som är baserade på standarden IEEE 802.11. Standarderna är ratifierade av den amerikanska delorganisationen *Institute of Electrical and Electronics Engineers*, IEEE [20].

Den senaste Wi-Fi standarden som har fått stor spridning, Wi-Fi 6, fungerar inom frekvensområdena 2.4GHz och 5GHz [21]. Den tillåter teoretiskt maximala dataöverföringshastigheter på 10-12Gbps. Med framtida generationer utvecklar man för att tillåta fler frekvenser och högre överföringshastigheter, upp till 40Gbps [20]. Högre hastigheter uppnås genom att öka frekvensen på signalerna som sänds från routern till enheterna. Den höga frekvensen innebär nödvändigtvis att pulsens period förkortas vilket gör signalen känsligare mot störningar då en liten avvikelse i pulsen kan felaktigt tolkas av mottagaren. Önskas en högre överföringshastighet på Wi-Fi är det nödvändigt att åtgärda denna utmaningen [18].

### 2.1.3 5G

Trots att både 5G och Wi-Fi är begrepp för teknologier som tillåter trådlös kommunikation är skillnaden mellan dem markant. Wi-Fi är framställt för att möjliggöra digital kommunikation i ett lokalt nätverk, medan 5G är designat för att täcka större områden. Detta möjliggörs genom 5G-master som 5G-enheter kommunicerar genom [22].

5G har två övergripande frekvensområden, FR1 och FR2. FR1 använder sig av lågfrekventa vågor, 4-7GHz [23]. FR2 använder vågor med millimeterlånga våglängder som kan göra det möjligt att förse många enheter med hög dataöverföringshastighet. Dessa frekvenser ligger mellan 30-300GHz. Den korta våglängden är känsligare mot fysiska hinder och andra störningar. Teknik som *Massive Multiple-Input Multiple-Output*, Massive MIMO, planeras användas med *Beamforming* som är en teknik där en antens signal riktas för bättre signalstyrka och stabilitet [24][25].

För enheter som inte kan direktkommunicera med 5G-antennerna krävs istället en mottagare som tar emot 5G-signalerna. Även att mottagaren sedan överför datan till enheten via ett protokoll som den är kompatibel med.

Under projektets gång har ett 5G-nätverk som tillhandhålls av Ericsson använts. 5G-nätverket sänds genom två antenner som finns i CASE-labbet på Chalmers.

## 2.2 Federerad inlärning

Federerad inlärning är en maskininlärningsteknik som skapades av Google 2015 [26]. Maskininlärning (ML) är en delmängd av artificiell intelligens (AI) och använder sig av maskininlärningsalgoritmer som bearbetar speciell indata för att sedan producera en önskvärd utdata. Dessa algoritmer kan lära sig från tidigare resultat (erfarenhet) och sedan ändras därefter. Därmed är de inte hårdkodade för uppgiften. Maskininlärning har på senare år ökat i en väldigt hög takt [27]. Det finns olika formella definitioner av maskininlärning men Oxford Languages definierar det enligt: “a type of artificial intelligence in which computers use huge amounts of data to learn how to do tasks rather than being programmed to do them” [28].

Som tidigare nämnt, tränas en maskininlärningsalgoritm på specifik indata för att lära sig se mönster. Denna data kan variera, allt från beskrivande statistik till bilder. När man tränar en algoritm är det även fråga om huruvida man gör det genom *övervakad* eller *oövervakad* träning. *övervakad* innebär att modellen vet när den har rätt (då den korrekt förväntade utdatan är märkt) och *oövervakad* är träning på icke-märkt data. Maskininlärning som fenomen har visat sig väldigt framgångsrik inom att kunna detektera mönster i framförallt bilder och har således kunnat användas i detektering av bland annat cancer [29].

Det som utgör träningsmiljön inom traditionell maskininlärning är en server som samlar all träningsdata från respektive slutnod i systemet. Till skillnad från detta bygger federerad inlärning istället på en decentraliserad idé där träningen alltid sker lokalt i varje ändnod. Processen börjar med att en maskininlärningsmodell skickas till varje ändnod. Modellen tränas därefter lokalt på varje enhet och den nu uppdaterade modellen kan sedan skickas tillbaka från ändnoderna till servern. Denna process är iterativ, menat att när de globala parametrarna från systemet har uppdaterats, skickas den uppdaterade modellen ut igen till en ny grupp av ändnoder [26].

Genom användande av federerad inlärning medkommer många fördelar men även nackdelar. En av dem främsta nackdelarna är att det förekommer mycket extra overhead när det kommer till kommunikationen från slutenhet till server. Det existerar lösningar som beprövats samt verifierats. Bland annat att öka träningstiden lokalt på slutenheterna för att inte skicka information lika frekvent. Även om teknologin som sådan inte är fulloptimerad, är den utvecklat på ett sätt som gör den mindre benägen för risk av säkerhetsattacker jämfört med traditionell maskininlärning. Därmed finns det applikationsområden där federerad inlärning kan vara värt att använda [30].

Då federerad inlärning är så pass nytt, är det ett aktuellt ämne som ständigt utvecklas. Det är en innovativ idé som har potential att kunna effektivisera mycket och speciellt när det kommer till IoT:s. Detta innebär att det finns flera områden där principen är applicerbar [31].

### 2.3 Mjukvara

Den mjukvara som använts för detta arbete är bland annat Git för versionhantering och Python vilket är ett programmeringsspråk som hårdvaran som kommer att användas i detta projekt är kompatibel med. Med hjälp av en *VPS* går det att lagra filer på nätet så att alla som behöver hämta filerna kan göra det smidigt. En lämplig programmiljö där mjukvaran kan exekveras är ett av Linux-distributionerna och med hjälp av systemet *WSL* går det även att exekvera linuxprogram i operativsystemet Windows. All mätdata i detta arbete sparas i formattet *CSV* för läsbarhet.

#### 2.3.1 Git

Programvaran *Git* används för versionhanteringen av källkod. Den främsta fördelen med Git är dess decentraliserade natur, vilket innebär att varje utvecklare har en fullständig sammansättning av projektets källkodshistorik på sin egen dator [32]. Detta möjliggör att flera utvecklare i en grupp parallellt kan arbeta på olika delar av projektet utan att ständigt behöva vara uppkopplade till en central server.

Git använder sig av *branches*, eller grenar, för att möjliggöra samtidig utveckling av separata funktioner, vilket gör det enkelt att testa och integrera ändringar innan koden sammanställs i huvudgrenen [32].

#### 2.3.2 Virtual Private Server (VPS)

En Virtuellt Privat Server (VPS) är en instans av en virtuell maskin som en fysisk server är värd för, där VPS-instanserna har en viss delmängd av den fysiska serverns resurser allokerad till sig [33]. En VPS är isolerad från andra VPS-instanser på en och samma fysiska server, och användningen av en VPS-instans kommer inte att påverka användningen av en annan VPS-instans. Detta möjliggör att en fysisk server kan agera server för flertalet användare samtidigt utan att användarna måste tävla om resurser, då alla användare kan använda sin max-kapacitet av VPS:ens resurser.

#### 2.3.3 Python och viktiga bibliotek

Python är ett mångsidigt hög-nivå programmeringsspråk som lanserades i början av 1990 talet. Språkets syntax var definierat med enkelhet som syfte och detta är en av anledningarna till varför språket har så många användare [34]. Dess popularitet har lett till att många bibliotek och ramverk har utvecklats av användarbasen. Det finns bibliotek som gör det enklare att framställa olika typer av applikationer bland annat nätverksapplikationer [35].

Ett av biblioteken för nätverksprogrammering i Python är socket-biblioteket. Socket-biblioteket tillhandahåller funktioner för att skapa och hantera nätverksanslutningar, inklusive TCP/IP- och UDP-anslutningar. Med socket-biblioteket kan utvecklare enkelt skapa klient- och serverapplikationer för att kommunicera över nätverket.

Ett annat bibliotek är `paramiko`, vilket tillåter ett program att öppna en SSH-anslutning till en enhet. Med hjälp av SSH kan man skicka kommandon till en server, hämta utdata, skicka filer och hantera autentisering med användarnamn och lösenord eller nycklar. Genom att kombinera Python och SSH kan man bygga kraftfulla verktyg för att automatisera hanteringen av nätverk och servrar, vilket kan spara tid och minska risken för felaktigheter vid manuella uppgifter.

För att hantera datan som samlades in under experimenten användes *Pandas* biblioteket och för att rita upp grafer användes biblioteket *MathPlotLib*. Python är förinstallerat i Raspberry Pi OS vilket även gör programmeringsspråket mer lämpligt än andra för detta projekt.

### 2.3.4 WSL

Windows Subsystem for Linux (WSL) är en funktion i Windows-operativsystem som tillåter användning av funktioner och program gjorda för Linux på en dator med Windows [36]. WSL finns i två versioner, WSL1 och WSL2. Både möjliggör användningen av Linux specifik mjukvara i Windows, men har olika tillvägagångssätt [37]. Enbart WSL2 används i detta projekt, på grund av anledningar som tas upp nedan, och WSL2 kommer således att refereras som WSL.

WSL1 använder sig av ett kompatibilitetslager som "översätter" exekverbara Linux-filer till ett format som kan köras på Windows. WSL1 behöver således inte använda sig av en virtuell maskin [37]. Detta innebär dock att Linux-program körs långsammare i och med detta extra lager, och att full kompatibilitet av systemanrop saknas.

Till skillnad från WSL1 använder WSL2 sig av en Linux-kärna, och behöver därför köras i en virtuell maskin [37]. Denna virtuella maskinen är lättviktig och baserad på Hyper-V. I och med att WSL2 använder sig av en Linux-kärna är kompatibilitet bättre. Dessutom körs program snabbare i och med att inget mellanlager används.

### 2.3.5 Linux-distributioner

En Linux-distribution (ibland kallad GNU/Linux-distribution) är ett operativsystem som bygger på Linux-kärnan [38]. I projektet användes två olika Linux-distributioner; Raspberry Pi OS på samtliga Raspberry Pi-enheter, och Ubuntu som användes via WSL på Windows-datorer.

Raspberry Pi OS, ofta kallat Raspbian då detta var dess tidigare namn, är ett operativsystem som är förinstallerat på alla Raspberry Pi-enheter som används i experimenten. Raspberry Pi OS är en specialanpassad Linux-distribution ämnad för Raspberry Pi-enheter, och är baserad på Debian. Raspberry Pi OS erbjuder en stabil plattform för olika projekt och experiment. Det inkluderar även ett brett utbud av förinstallerade program, vilket gör det enkelt för användare och utvecklare att komma igång i projekt. Bland de förinstallerade programmen finns språkplattformarna

Python 2 och 3 för programmering, samt verktyg som Mathematica och Scratch, som erbjuder möjligheter till matematisk modellering och visuell programmering [39].

Ubuntu, likt Raspberry Pi OS, är en Debian-baserad Linux-distribution. Till skillnad från Raspberry Pi OS är inte Ubuntu specialanpassad för någon specifik hårdvara, och behöver således inte ta hänsyn till vissa begränsningar som Raspberry Pi OS gör i och med att Raspberry Pi OS är inriktat mot Raspberry Pi-enheter. Detta innebär att Ubuntu kan komma med fler program förinstallerat, då Ubuntu generellt installeras på hårddisk, till skillnad från Raspberry Pi OS som brukar installeras på SD-kort, och är därmed mer platsbegränsad [40].

### 2.3.6 Comma-separated values-filer (CSV)

Komma-separerade värden (CSV-filer), är filer i textformat för att lagra data. Kolumnerna är separerade med kommatecken, och varje post står på sin egna rad [41]. I projektet används CSV-filer för att lagra data insamlad från Raspberry Pi-enheter, som sedan skickas till en dator för vidare analys. Hur dessa filer konstrueras underlättar lagring och hantering av data.

## 2.4 Hårdvara

I detta delkapitel diskuteras hårdvaran som varit en essentiell del av arbetet.

### 2.4.1 5G-router

5G-routrar av typen *E-lins H685* användes. Dessa routrar har bland annat två Ethernet-portar, varav en kan konfigureras som WAN port. Båda stöder en dataöverföringshastighet på 1Gbps. Dessutom kan fyra antenner monteras på varje router för att ansluta till 5G-nätet. Vissa routrar stödjer även Wi-Fi och för dessa routrar finns det möjlighet att montera ytterligare två antenner för Wi-Fi-nätverket. Ett SIM-kort kan monteras för att möjliggöra anslutning till 5G-nätet [42].

### 2.4.2 Wi-Fi-router

Den router som används är av typen *ASUS RT-AX82U AX5400* och har stöd för Wi-Fi 6. Denna router har fem Ethernet-portar, varav en är för anslutning till WAN. Routers Wi-Fi har två frekvensband att välja mellan vilka är 2,4 och 5GHz med en processor med klockfrekvens på 1,5GHz [43].

### 2.4.3 Switchar

Switchar av typ *D-Link DGS-105* används i alla experiment förutom de som fokuserar på Wi-Fi. Denna switch har fem Ethernet-portar som alla stödjer gigabithastighet för snabb och effektiv dataöverföring mellan anslutna enheter. Dessutom stödjer switchen *plug-and-play-installation*, vilket innebär att ingen konfigurering av inställningar krävs. Denna switch stödjer en avkänningsmekanism som automatisk känner



av den maximala dataöverföringshastigheten som stöds av de anslutna enheterna för att säkerställa optimal prestanda inom nätverket [44].

På grund av brist på 5-ports switchar användes även switchar av typen *Linksys LGS124* i flera stora experiment. Denna switch har 24 Ethernet-portar med Gigabit dataöverföringshastighet och stöder dessutom *Plug-and-Play-teknik* [45].

#### 2.4.4 Kablar

Under projektet har Ethernet-kablar av typ Cat 6 och Cat 6a använts för att ansluta switcharna med varandra, samt att ansluta Raspberry Pi-enheterna till switcharna. Dessa kablar finns i olika längder beroende på behov och den längsta som användes i testerna var tio meter lång. Den maximala dataöverföringshastigheten i Cat 6-tekniken når 10Gbps och har en maxfrekvens på 500MHz [46].

#### 2.4.5 Raspberry Pi

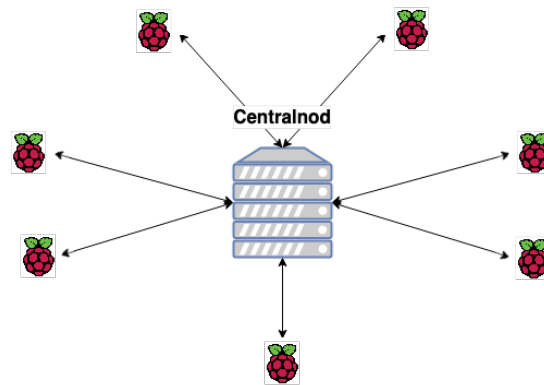
Raspberry Pi är en serie av mångsidiga enkortsdatorer som stödjer operativsystemet Linux, i en liten formfaktor [47]. Enheten kan anslutas till olika sensorer, kameror och andra enheter via *generell-purpose-input/output*-stift (GPIO-stift). I och med dess kompakta storlek, låga strömförbrukning, och relativt låga inköpspris är valet av Raspberry Pi ett lämpligt val för IoT-projekt, eller andra projekt där en dator behöver byggas in [48]. De Raspberry Pi-modeller som användes i experimenten var Raspberry Pi 4B och Raspberry Pi 400. Dessa Raspberry Pi-enheter har en SD-kortplats då ett SD-kort kommer utgöra sekundärminnet för Raspberry Pi-enheten. Dessutom stödjer Raspberry Pi-enheterna IEEE 802.11ac, numera kallad Wi-Fi 5, på 2.4GHz och 5GHz frekvensbanden för trådlös kommunikation, och de är utrustade med en Gigabit Ethernet-port för trådbunden kommunikation [49][50].

### 2.5 Nätverkstopologi

Ett nätverks topologi har en icke-trivial påverkan på hur snabbt data överförs bland nätverksnoderna. Olika val av topologier medför styrkor och svagheter som är karaktäristiska för det valet och i detta arbete studeras linje-, stjärn- och trädtopologierna [51].

#### 2.5.1 Stjärntopologi

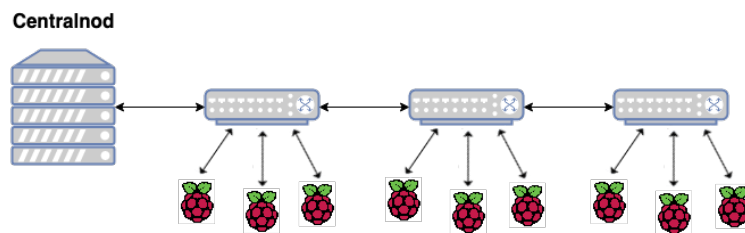
I en stjärntopologi är varje enhet i nätverket direkt ansluten till en gemensam centralnod, vanligtvis en server eller en router, se figur 2.1. Centralnoden agerar som en knutpunkt och dirigerar trafiken mellan enheterna i nätverket. Fördelarna med en stjärntopologi inkluderar enkel installation och felsökning. Går en av kantnoderna ur drift påverkas inte de andra vilket gör det enklare att lägga till eller ta bort enheter utan att sätta stopp på hela systemet. Nackdelarna inkluderar en potentiell flaskhals vid centralnoden och de tekniska svårigheterna med att implementera ett sådant nätverk med ett högt antal noder [52].



**Figur 2.1:** Exempel på struktur av ett nätverk med stjärntopologi.

### 2.5.2 Linjetopologi

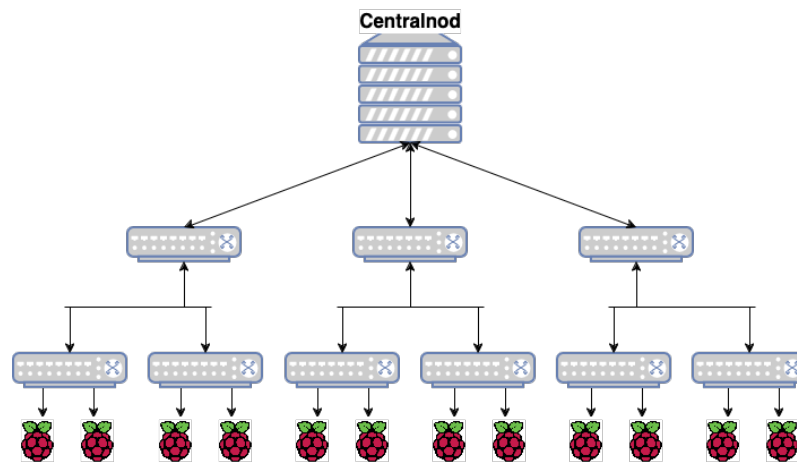
I en linjetopologi, engelska *bus topology*, finns det en gemensam nätverksbuss som delas av alla enheter. I detta projekt har nätverksswitchar kopplats ihop i serie med Raspberry Pi-enheter under dem, se figur 2.2. Fördelarna med en linjetopologi inkluderar enkelhet i uppsättning och konfiguration. Nätverket kan enkelt utvidgas genom att lägga till fler enheter längs serien. En nackdel med linjetopologi är den tunga belastningen som läggs på enheterna närmast centralnoden. Ett fel på den första noden påverkar hela nätverket vilket utesluter linjetopologin om enheterna som används har en tendens att sluta fungera. Denna nätverksarkitektur kan vara långsammare än stjärntopologin på grund av bristen av en direkt anslutning till centralnoden. När den centrala noden vill kommunicera med någon enhet i nätverket måste meddelandet först passera alla enheter som ligger före den önskade noden [52].



**Figur 2.2:** Exempel på struktur av ett nätverk med linjetopologi.

### 2.5.3 Trädtopologi

Trädtopologi har en hierarkisk struktur med centralnoden längst upp och resten av enheterna i olika undernivåer, se figur 2.3 [53]. En fördel med trädtopologier är att den är mycket mer skalbar än stjärn- och linjetopologierna. Utökning av nätverket är trivialt då det endast behövs läggas till en ny undernivå till hierarkin för att kunna stödja de nya noderna. Likt stjärntopologi är en nackdel med trädtopologi är att om centralnoden går sönder eller slutar fungera av någon anledning, kan det leda till att hela nätverket havererar [54].



**Figur 2.3:** Exempel på struktur av ett nätverk med trädtopologi.



# 3

## Genomförande

För att få ett strukturerat och effektivt arbetsflöde på projektet delades det upp i olika faser. Detta kapitel ägnas åt att beskriva dessa faser vilket innefattar efterforskning, implementation av applikationen och avslutningsvis tester.

### 3.1 Efterforskning

Forskningsfasen ägnades åt att läsa på om relevant information gällande projektet. Denna information åsyftar hårdvara, mjukvara, liksom allmän teori kring teknologier som ansågs relevanta för arbetet. Således lästes relevanta undersökningar, böcker och vetenskapliga artiklar. Som grund för projektet tilldelades gruppen även ett arbete som tidigare hade producerats av andra studenter vid institutionen för Data- och informationsteknik på Chalmers Tekniska Högskola [55]. Detta arbete studerade olika typer av algoritmer vid hämtning av data till IoT-enheter. Liksom de ovannämnda källorna, studerades detta arbete som källa för inspiration och för att stärka gruppens kunskaper.

När detta var utfört kunde en bild målas upp av vad som skulle kunna åstadkommas med projektet. Initialt övervägdes det att fortsätta utvecklingen baserat på tidigare studenternas arbete, med tanken på de intressanta utvecklingsområdena gällande projektet. Istället bestämdes det att skifta projektets fokus till att undersöka prestandan av olika nätverkslänktyper. Detta gjordes på grund av tidsbrist eftersom undersökningen av länktypernas prestanda sågs som den viktigaste delen av projektet. Sedan sattes målen upp och nästa fas ”Implementering av applikation” kunde påbörjas.

### 3.2 Implementering av applikation

Detta kapitel går igenom hur implementeringen av själva applikationen har gått till. Den här delen av projektet har framförallt fokuserat på hur målen skulle uppnås, se delkapitel 1.2, för att stärka kvalitet på projektet. Kopplat till detta formulerades ett antal funderingar kring processen, dessa innefattade:

1. Vilken testdata som kunde användas för att göra testresultaten så verklighetstrogna som möjligt?
2. Vilka parametrar som var av intresse att mäta under testerna och hur ska de samlas?

3. Hur resultaten skulle hanteras för att få testen att spegla verkligheten?
4. Vilka avgränsningar borde göras för att se till att projektets huvudsyfte hålls i fokus?

I hänseende till de ovannämnda frågeställningarna gjordes det några avgränsningar för att underlätta processen rent praktiskt. Det första som togs i åtanke var vilka mätvärden som skulle extraheras från testerna. Här gjordes även några begränsningar då parametrar som exempelvis energianvändning valdes bort och detta var eftersom mätningen av dessa parametrar skulle kräva mätinstrument som inte var tillgängliga. På grund av tidsbrist lades istället fokuset på nedladdningstiden.

Det var klart från början att antalet kantnoder i nätverket skulle öka väsentligt under arbetsprocessen och därmed skulle för stora filer leda till alldeles för långa tester. Således anpassades filstorlekarna på test datan enligt detta. Typerna på filerna valdes sedan så att de skulle kunna appliceras på ett verklighetstroget scenario. Därmed valdes en skivavbildnings-fil och en fil som innehåller mätdata. Se sektion 5.10 för reflektioner kring dessa filer kopplat till resultaten.

För att hålla fokuset på undersökningen på länktyperna valdes enkla nätverkstopologier. När Ethernet och 5G mättes var noderna inte direktansluta till den centrala noden på grund av begränsningar på hårdvaran. Därför mättes prestandan för både linje- och trädtopologi. Wi-Fi hade inte dessa begränsningar eftersom att de kommunicerade trådlöst med centralnoden.

#### 3.2.1 Val av testdata

Då tester påbörjades tidigt under projektets gång krävdes redan då en idé om vilken relevans testdatan skulle ha i hänsyn till frågeställningen kring användningsområdet. Den datamängd som användes var försedd av projektets handledare. Denna datamängd består av flertalet filer i olika format. Datamängden ansågs vara av lämplig storlek då den var stor nog för att simulera ett användningsområden. Datafilen skulle dock inte vara så stor att varje experiment skulle ta över en timme att utföra.

Experiment gjordes även med en enda fil som var ungefär samma storlek som hela den ovan nämnda datamängden. Detta för jämföra hur nätverken hanterade flera små filer med hur de hanterade en stor fil när testerna skalas upp med flera kantnoder. Filen som i detta syfte var en skivavbildning av en Linux-distribution, då de är lättillgängliga, finns i passande storlekar, och får distribueras fritt.

Vid tester med datamängden och/eller skivavbildningen användes både en komprimerad och en icke-komprimerad version av dessa, storlekarna på dessa filer var:

- Datamängd komprimerad: 216MB
- Datamängd okomprimerad: 546MB
- Skivavbild komprimerad: 242MB

- Skivavbild okomprimerad: 900MB

### 3.2.2 Utveckling av Bash-skript

Under utveckling av Bash-skriptet som skulle användas på samtliga Raspberry Pi-enheter, har skriptet ändrats ett flertal gånger under projektets gång. Den slutgiltiga versionen som användes för testerna använder sig av `wget`-kommandot för nedladdning av filer, `unzip`-kommandot för att extrahera de komprimerade filerna, och `time`-kommandot för att få tidsåtgången av nedladdningen och extrahering av filerna. Utöver dessa användes flertalet kommandon för att få ut data från det som gjorts, och för att formatera datan till ett format som sedan kunde föras in i en CSV-fil som sparades lokalt på den Raspberry Pi-enheten som körde skriptet. Då den mätdata som kan kopplas direkt till målet var nedladdningshastigheten var det av intresse att få ut den. Detta värde kunde fås genom att ta storleken på datan som laddats ner och dividera detta med tidsåtgången för nedladdning. Skriptet skapade två filsökvägar (**data**, **downloads**) lokalt på den enheten som exekverade skriptet. Filer som laddats ner placerades i `downloads` och slutgiltiga mätvärden sparades i `data`. Dessa mätvärden innefattar filstorlek, nedladdningshastighet, nedladdningstid, tid för att extrahera data samt total nedladdningstid.

En version av Bash-skriptet använde sig av `perf`-kommandot, och tog tidsåtgång från detta istället. `Perf`-kommandot gjorde även så att antalet klockcykler och antalet instruktioner det tog för att utföra nedladdning via `wget`-kommandot och extrahering av data via `unzip`-kommandot kunde samlas in. Denna version användes inte vid laborationerna då flest tester redan hade utförts med en äldre version av Bash-skriptet som inte använde sig av `perf`-kommandot, och skulle således kräva att tidigare tester gjorde om. Denna process hade varit tidskrävande.

Då testdata som tidigare nämnts inkluderade komprimerade filer, i zip-format, behövde dessa extraheras på den lokala enheten när de väl var nedladdade. Därmed implementerades funktionalitet som extraherar de komprimerade filerna om de var komprimerade, men ignorerar detta steg om filerna var okomprimerade. Den totala tidsåtgången för komprimerade filer gavs av att addera tidsåtgången för nedladdning med tidsåtgången för extrahering av filerna.

Vid utvecklingen av Bash-skriptet gjordes övervägande om skriptet skulle ta tider som `wget`-kommandot rapporterade eller om den skulle ta tider från `time`-kommandot. Tiderna som `wget`-kommandot rapporterade är den tidsåtgång som den faktiska nedladdningen tar, och räknar inte med *overhead*, tidsåtgång som inte relaterar direkt till uppgiften som skall utföras. `Time`-kommandot tar med denna *overhead*-tiden, och ger oss den verkliga tidsåtgång för exekvering av `wget`-kommandot. `Time`-kommandot valdes då detta kommando måste användas för att få tiden för extrahering av data då `unzip`-kommandot inte rapporterar tidsåtgång. Det ansågs inte vara en rättvis jämförelse att få ut tidsåtgång på två olika sätt varav en av metoderna inte tar *overhead*-tid i hänsyn. Utöver detta ansågs det att den enda tidsåtgång som var relevant var den verkliga tidsåtgång med *overhead* in-

räknad då att ignorera overhead-tiden kan ses som en "försköning" av datan.

För att underlätta testningsfasen var det viktigt att på ett effektivt sätt kunna distribuera detta skript till alla Raspberry Pi-enheter. Till förfogande användes mjukvara från ett tidigare arbete som modifierades för att göra anpassa den till detta arbetet samt för att parallellisera den mer. Detta innefattar två program som var skrivna i python (**deploy.py**, **startup.py**) och som exekveras från en värddator. Det första skriptet, deploy-skriptet, tar två argument. Det första argumentet är en CSV-fil som innehåller IP-adresser, användarnamn och lösenord till de Raspberry Pi-enheter som skulle anslutas till genom SSH. Det andra argumentet som deploy-skriptet tar är en filsökväg till den mapp som skall skickas över till Raspberry Pi-enheterna. Denna mapp innehåller de filer och skript som skall användas lokalt på Raspberry Pi-enheterna. Det andra skriptet, startup-skriptet, tar på liknande sätt två argument. Det första argumentet är samma CSV-fil som har använts för deploy-skriptet. Det andra argumentet är namnet på den fil eller skript som skall exekveras lokalt på Raspberry Pi-enheterna. Med dessa två program till hands kunde Bash-skriptet anges som argument och kunde således distribueras samt köras på ett effektivt sätt.

## 3.3 Testning

Med fungerande mjukvara kunde experiment genomföras för att testa prestandan av de olika nätverkstyperna och nätverkstopologierna. Nätverkstyperna innefattade Ethernet, Wi-Fi och 5G. Med Ethernet testades både träd- och linjetopologi. När det kom till filer som skulle laddas ner användes "datamängd" och "skrivavbildnings"-filen. Nedladdningen startades på alla enheter samtidigt för att belasta nätverken så mycket som möjligt. För att få en struktur på arbetsflödet och säkerställa att det var enkelt att upptäcka oväntade fel, användes ett arbetssätt där antal enheter skalades upp efter varje test. Arbetsmodellen såg ut enligt följande:

1. X antal Raspberry Pi-enheter togs fram.
2. Nätverket kopplades upp med hänseende till att alla X antal enheter skulle få plats på eventuella switchar och dylikt. Beroende på länktyp som skulle testas, kopplades nätverket upp med länktyp Y.
3. Enheterna förbereddes genom att arbetsmappen för testet rensades med ett rensnings-skript.
4. Parametrar i huvud-skriptet justerades för att hämta fil Z. Skriptet distribuerades till samtliga Raspberry Pi-enheter på nätverket och exekvering påbörjades därefter.
5. Exekveringssteget av steg 4 repeterades ytterligare 2 gånger.
6. Resultaten (som låg lokalt på kantnoderna) samlades in till värd-datorn.

Denna arbetsmodellen följdes under hela arbetets gång. Antalet Raspberry Pi-enheter skalades successivt upp under projektets gång då resultat för nätverk i större skala skulle uppnås. För att säkerställa att jämförbar data hela tiden erhöles under projektets gång, försökte det upprätthållas att tester genomfördes med samma antal

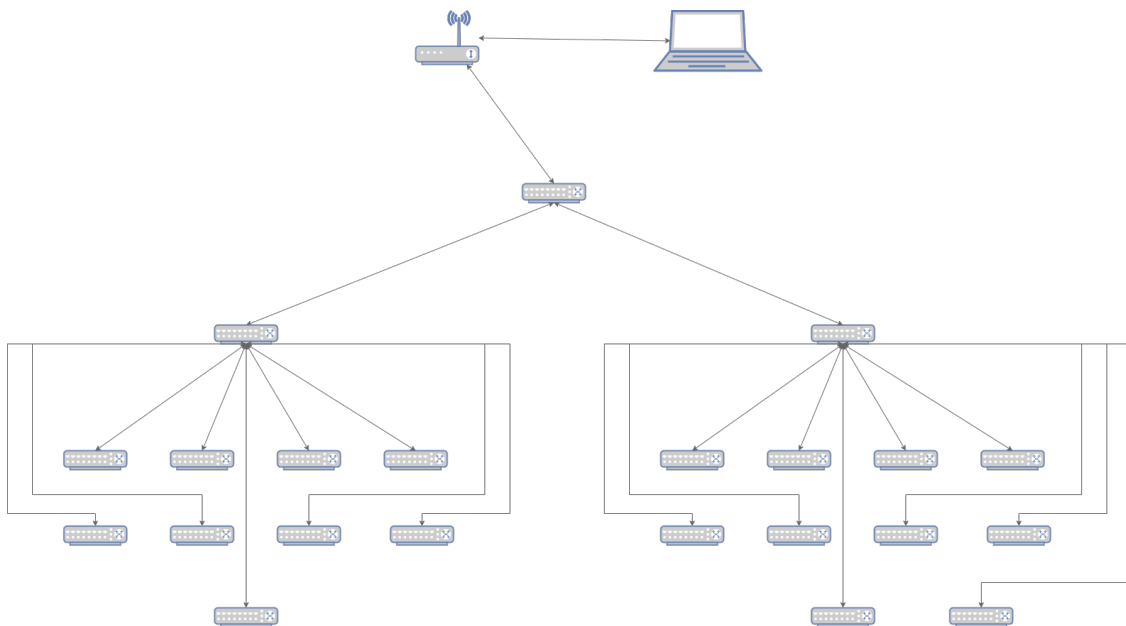


enheter för samtliga nätverkstyper. På så sätt fanns det alltid någon data som grund för att kunna dra en eventuell slutsats. Detta var viktigt då tester, framförallt för de i stor skala, inte var garanterade i mån av tid då stora utrymmen till exempel ett stort laborations rum krävdes för att sätta upp testmiljön.

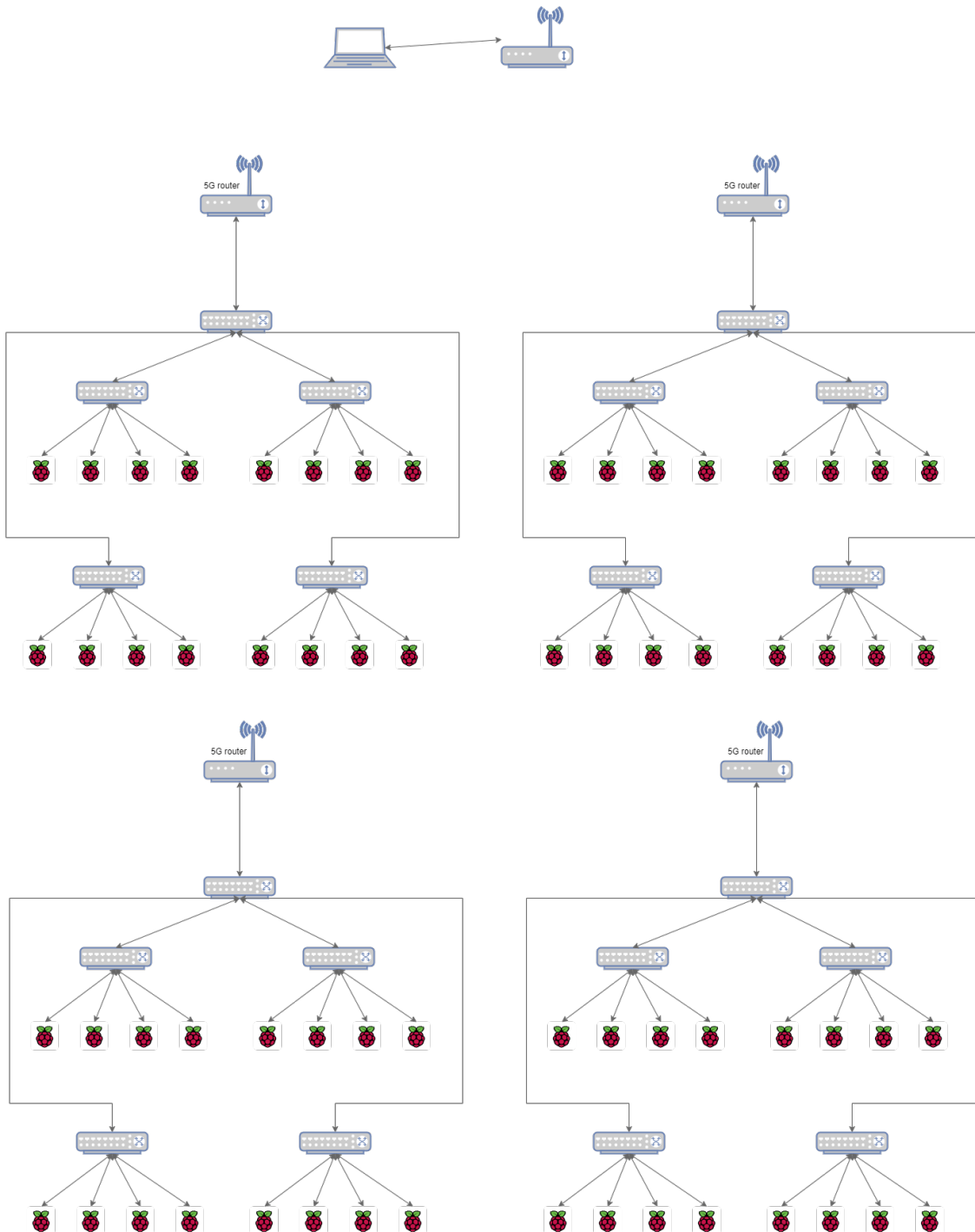
Skivavbildnings-filen var endast en del av Ethernet-testerna då det insågs att det tog för mycket tid att utföra samma test för 5G och Wi-Fi. Därmed behövde inte filtyp ändras förutom tester gällande Ethernet. Tester utfördes inte på enskilda komprimerade och okomprimerade filer separat eftersom resultaten för båda typerna redan ingick i testerna.

Resultaten som hade hämtats till värddatorn användes för att rita upp grafer som kan ses i kapitel 4. I de flesta fall gjordes tre försök av samma experiment så för att rita upp graferna användes genomsnittet av dessa. I vissa fall rensades datan för att ta bort korrupta och eller orepresentativa datapunkter. Detta gjordes särskilt för linjegrafen som jämförde uppskalning av nätverken, se figur 4.14.

Ethernetnätverket med 73 enheter kopplades upp med trädtopologi enligt figur 3.1. Wi-Fi-nätverket kopplades upp med stjärntopologi enligt figur 2.1. 5G-nätverket kopplades upp med en hybrid mellan träd- och stjärntopologi enligt figur 3.2



**Figur 3.1:** Ethernet experiment på 73 Raspberry Pi. Varje switch har 4 enheter kopplade till sig, utom en som har 1 enhet.



Figur 3.2: 5G experiment på 64 Raspberry Pi-datorer.

## 3.4 Verktyg och materiel

I den här sektionen nämns all material och mjukvara som användes under projektets gång för att realisera målet för projektet.

### 3.4.1 Mjukvara och versionhantering

För att hålla reda på de olika versionerna användes versionhanterings-programmet Git. Git var hjälpsam då det gav en tydlig struktur på processen. Det var även väsentligt i situationer som exempelvis när skulle återställa en fil till en tidigare version. Git valdes även för att gruppens medlemmar hade tidigare erfarenheter av programmet till en sådan grad som krävdes för detta projekt. När det gäller skrivande av rapporten användes *Overleaf*. Overleaf möjliggör enkelt samarbete för skrivning av texter i L<sup>A</sup>T<sub>E</sub>X [56].

### 3.4.2 Hårdvara

Upp till 73 stycken Raspberry Pi-enheter användes i det största experimentet. Det användes som maximalt 97 stycken Ethernet-kablar, 20 stycken switchar med 5 portar var och två stycken switchar med 24 portar samt deras strömadaptrar, 73 stycken Raspberry Pi-enheter samt SD-kort och strömadaptrar till dessa, en router med tillhörande strömadapter, 4 stycken 5G-routrar med tillhörande strömadaptrar och 4 stycken SIM-kort som stödjer 5G.

När det kommer till hanteringen av hårdvara har processen av att packa upp och packa ner enheterna varit väldigt tidskrävande. Alla Raspberry Pi-enheter var lagda i separata lådor och det behövdes många grenuttag så att all enheter kunde få ström. När det gällde uppackning av enheterna har all utrustning lagts på ett stort bord och sorterats efter typen (4B eller 400). När det sedan kom till att packa ner enheterna, placerades dem tillbaka i respektive låda med tillhörande materiell. Eftersom omfattningen av experimenten gällde så pass många enheter, behövdes cirka 8 timmar för uppackning, nedpackning samt körande av experimentet tillsammans. Utöver detta tog det även tid att transportera enheterna från platsen där de förvarades till testlokalerna.



# 4

## Resultat

I detta kapitel redovisas de resultat som samlats in från experimenten som utförts under projektets gång. Som tidigare nämnts är målet med detta projekt att undersöka hur olika länktyper presterar i olika sammanhang. Kapitlet behandlar tester som omfattar högst 73 stycken Raspberry Pi-enheter. Datan som presenteras nedan är, om inget annat nämns, genomsnittet av tre tester.

### 4.1 Samlad data från Ethernet

Det utfördes fyra experiment med Ethernet där skillnaden var vad som laddades ner och vilken nätverkstopologi som användes. Figur 4.1 och 4.2, som innehåller resultaten från testerna som laddade ner datamängden, uppvisar mer variation i tidsåtgången för nedladdningen än vad som visas i figur 4.3 och 4.4. Dessa har motsvarande data för test med skivavbildningen.

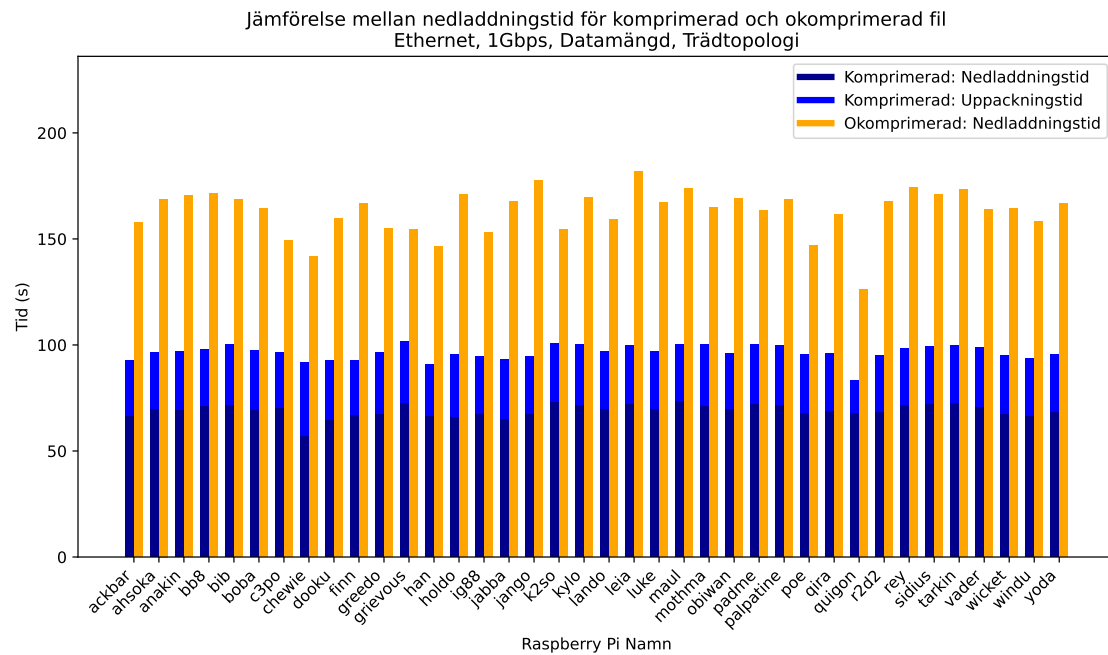
I experimenten med nedladdning av datamängden uppvisar testet som utfördes med en linjär nätverkstopologi, figur 4.2, mer variation i tid än testet som gjordes med en trädtopologi, figur 4.1. Ingen skillnad syns i testerna med nedladdning av en skivavbildningen, figur 4.3 och 4.4.

Båda testen med skivavbildningen, figur 4.3 and 4.4, har en enhet, "Quigon", som konsekvent blev klar snabbare än resten av enheterna. Detta är extra märkbart då varje datapunkt som presenteras här är genomsnittet av tre experiment. Den här avvikelser observerades inte till lika stor grad i testerna med datamängden och observerades inte alls i tester med fler enheter.

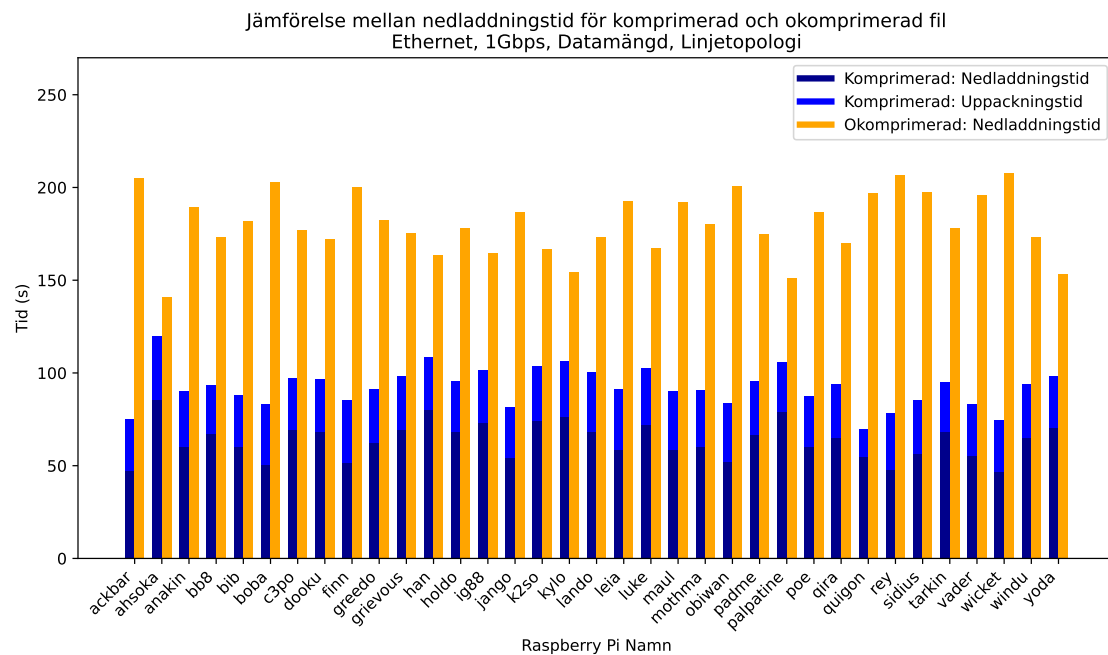
Figur 4.5 redovisar resultaten från ett test där 73 enheter laddade ner datamängden. Grafen visar att Ethernet är stabilt och pålitligt även med över 70 aktiva enheter. Variationen mellan de olika enheterna är låg vilket även observerades i tidigare Ethernet tester med färre enheter.

Som kan ses i figur 4.6 så finns det tillfällen där nedladdning och uppackning av en komprimerad fil är snabbare än nedladdning av en okomprimerad version av samma fil. Detta inträffar dock bara regelbundet vid ett lågt antal kopplade enheter.

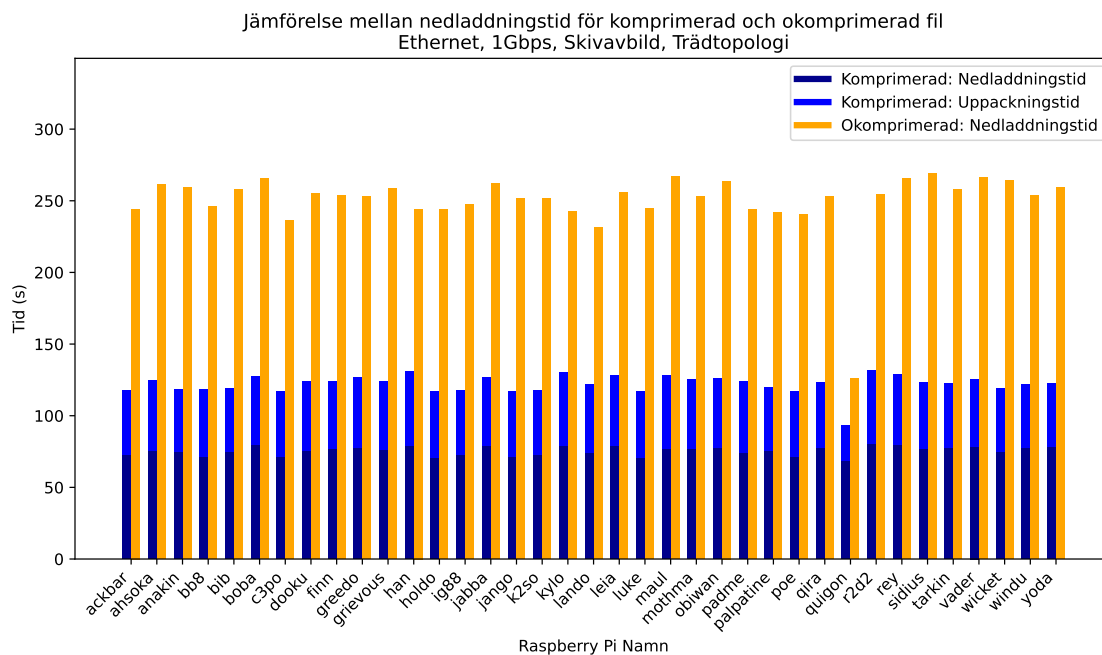
## 4. Resultat



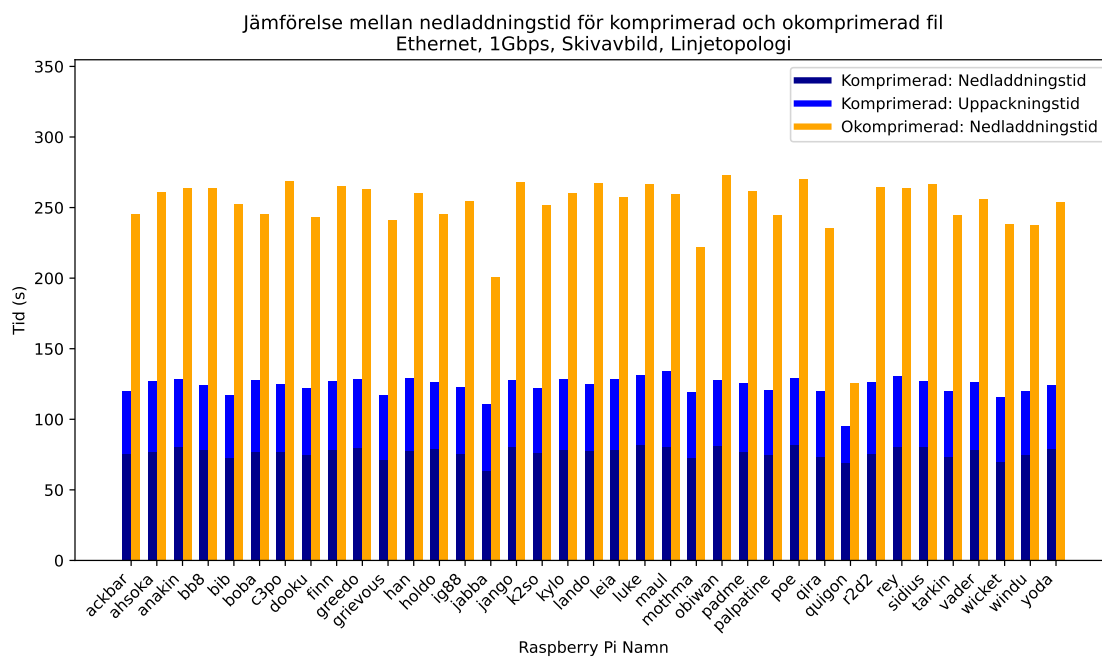
**Figur 4.1:** Ethernet, 1Gbps, Datamängd, Trädtopologi, 38 enheter.



**Figur 4.2:** Ethernet, 1Gbps, Datamängd, Linjetopologi, 38 enheter.



**Figur 4.3:** Ethernet, 1Gbps, Skivavbild, Trädtopologi, 38 enheter.

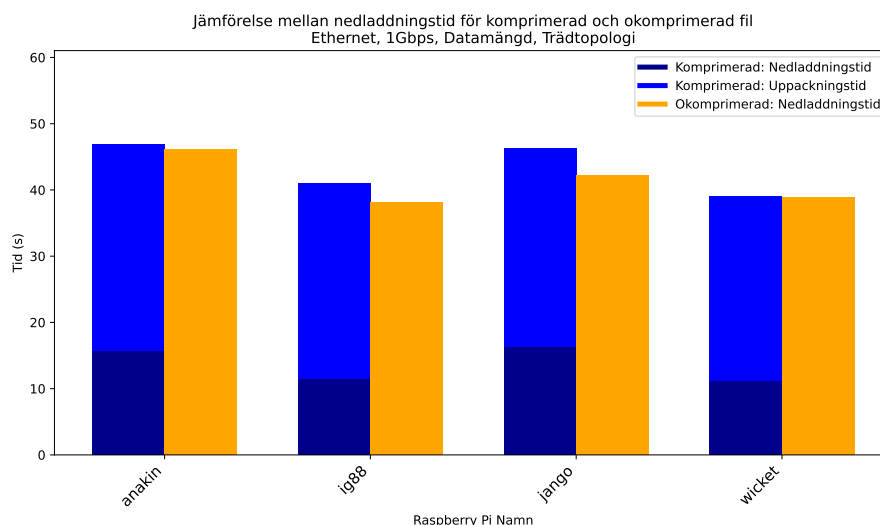


**Figur 4.4:** Ethernet, 1Gbps, Skivavbild, Linjetopologi, 38 enheter.



26





**Figur 4.6:** Ethernet, 1Gbps, Datamängd, Trädtopologi, 4 enheter.

## 4.2 Samlad data från Wi-Fi

I denna sektion presenteras den data som samlats in under experimenten med Wi-Fi, både 5GHz Wi-Fi och 2.4GHz Wi-Fi har testats.

### 4.2.1 2.4GHz Wi-Fi

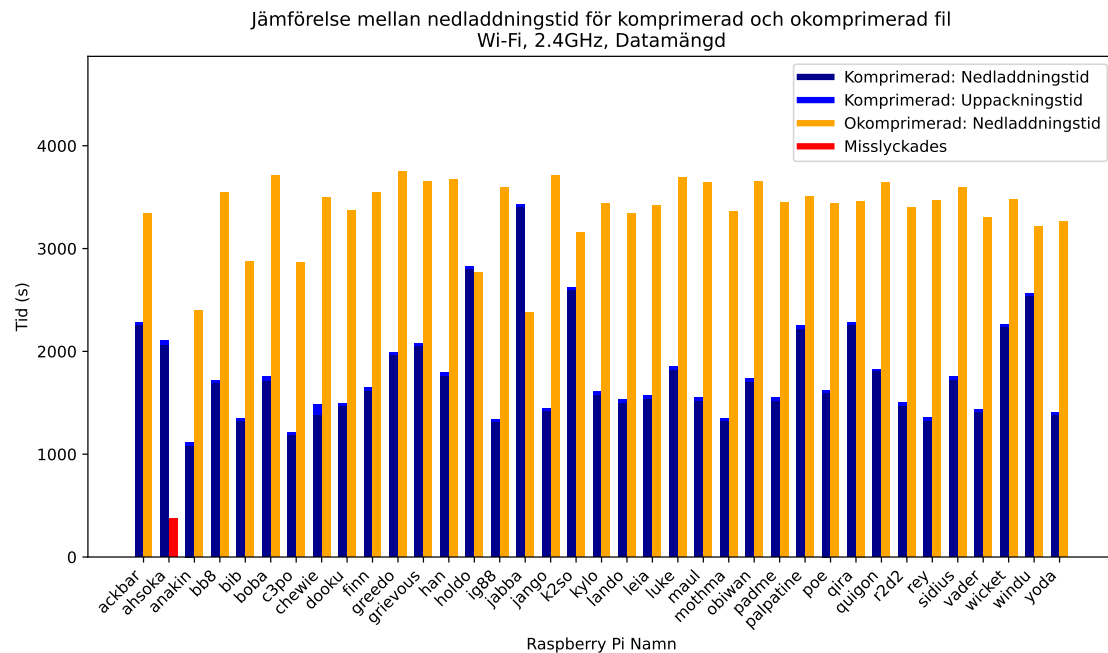
Endast ett experiment gjordes över 2.4GHz Wi-Fi eftersom det tog nästan 2 timmar att genomföra. I figur 4.7 framgår det att resultaten är relativt konsistenta jämfört med 5GHz Wi-Fi med bara ett fåtal enheter som tar mycket längre tid än genomsnittet. Dock är de fortfarande väldigt spridda jämfört med Ethernet. Av de 37 enheter som testades var det bara en, "Ahsoka", som misslyckades med nedladdningen.

### 4.2.2 5GHz Wi-Fi

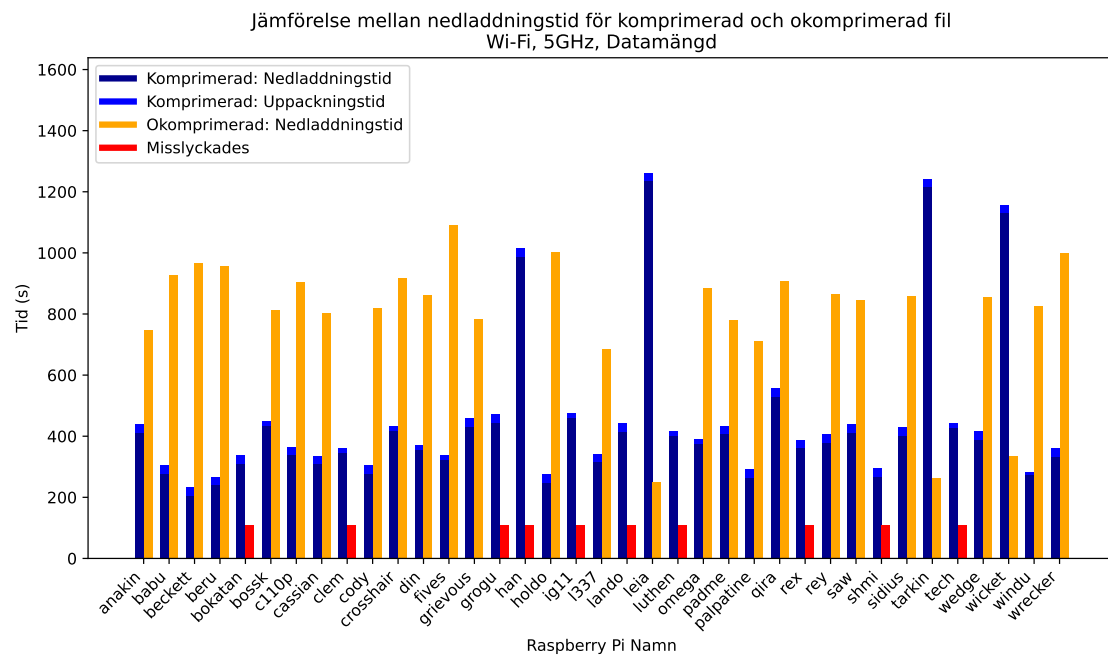
I figur 4.8 redovisas resultaten från ett test med 37 enheter. Det fanns tio enheter som misslyckades med en del av nedladdningen, och fyra andra som tog väldigt lång tid på sig. Detta är en stor försämring i jämförelse med 2.4GHz Wi-Fi, se figur 4.7, där endast en enhet misslyckades med nedladdningen. På grund av detta utfördes senare 5GHz tester med endast 19 enheter. Men även där misslyckas ibland någon en eller två enheter, se figur 4.11

Resultaten från nedladdning av skivavbildningen, figur 4.10, fanns det flera enheter som tog längre tid att ladda ner den komprimerade filen än en okomprimerade. Detta är märkbart eftersom den komprimerade filen är en fjärdedel av storleken av de okomprimerade filen. Resultaten från nedladdning av datamängden, figur 4.9, är mer rimliga i det att den komprimerade filen laddas ner snabbare än den okomprimerade i alla fall utom ett där det tog ungefär lika lång tid för två enheter.

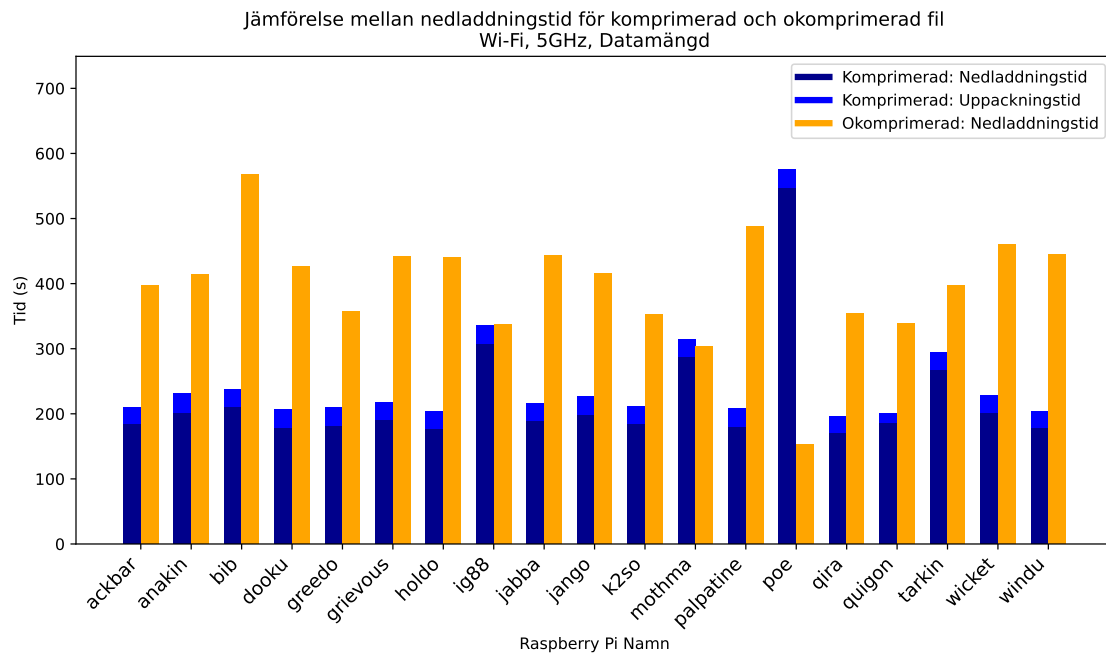
## 4. Resultat



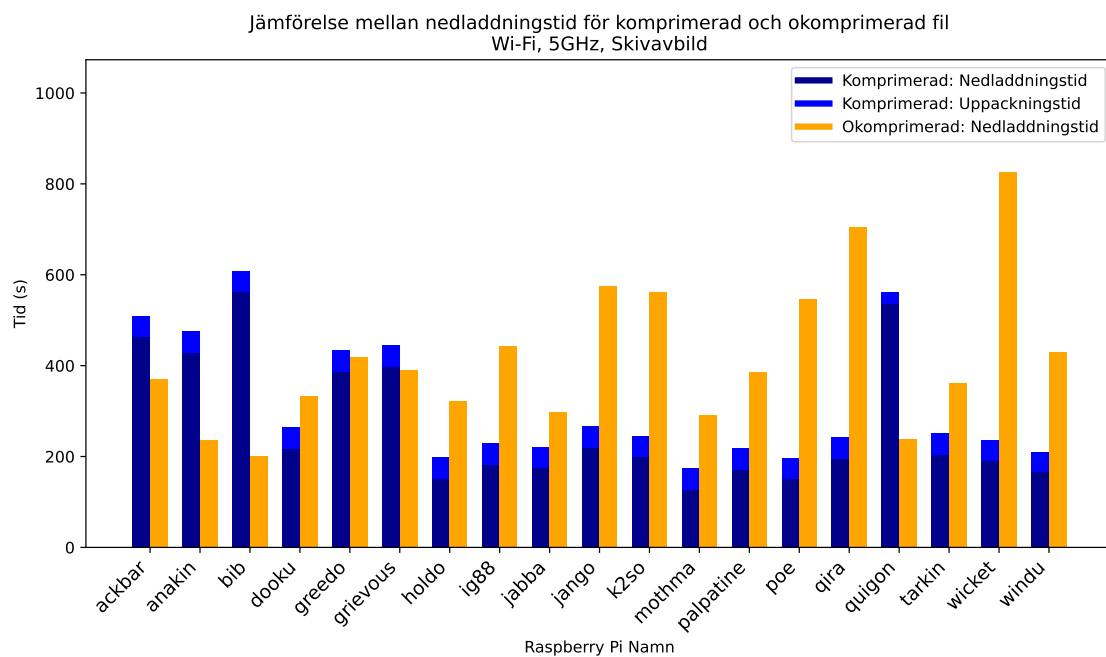
Figur 4.7: Wi-Fi, 2.4GHz, Datamängd, 37 enheter.



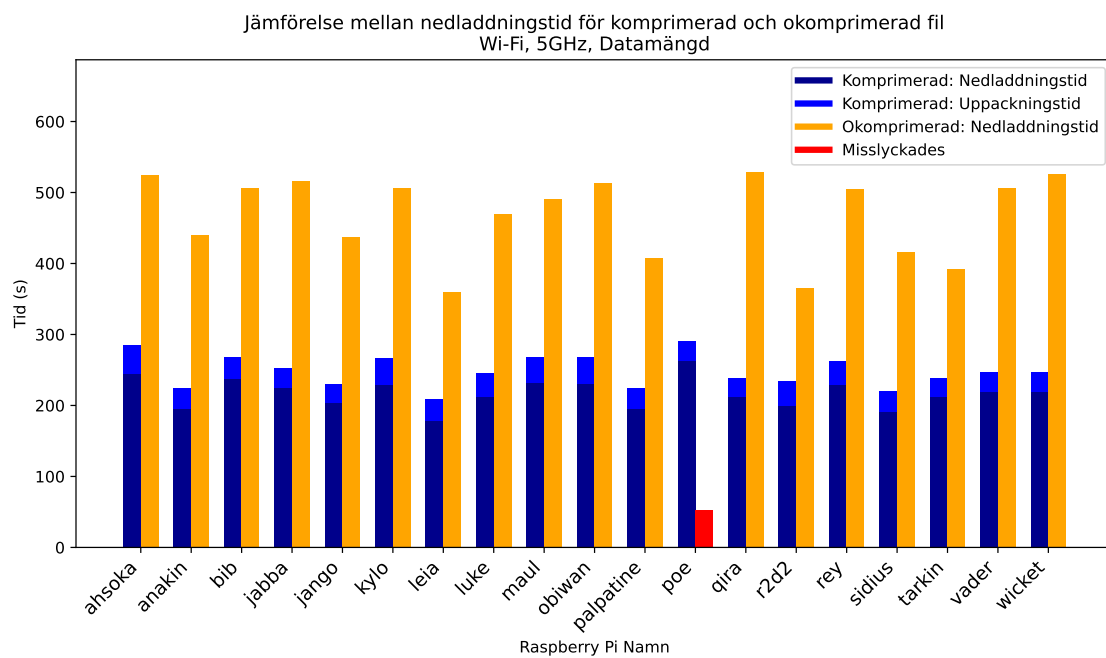
Figur 4.8: Wi-Fi, 5GHz, Datamängd, 37 enheter.



**Figur 4.9:** Wi-Fi, 5GHz, Datamängd, 19 enheter.



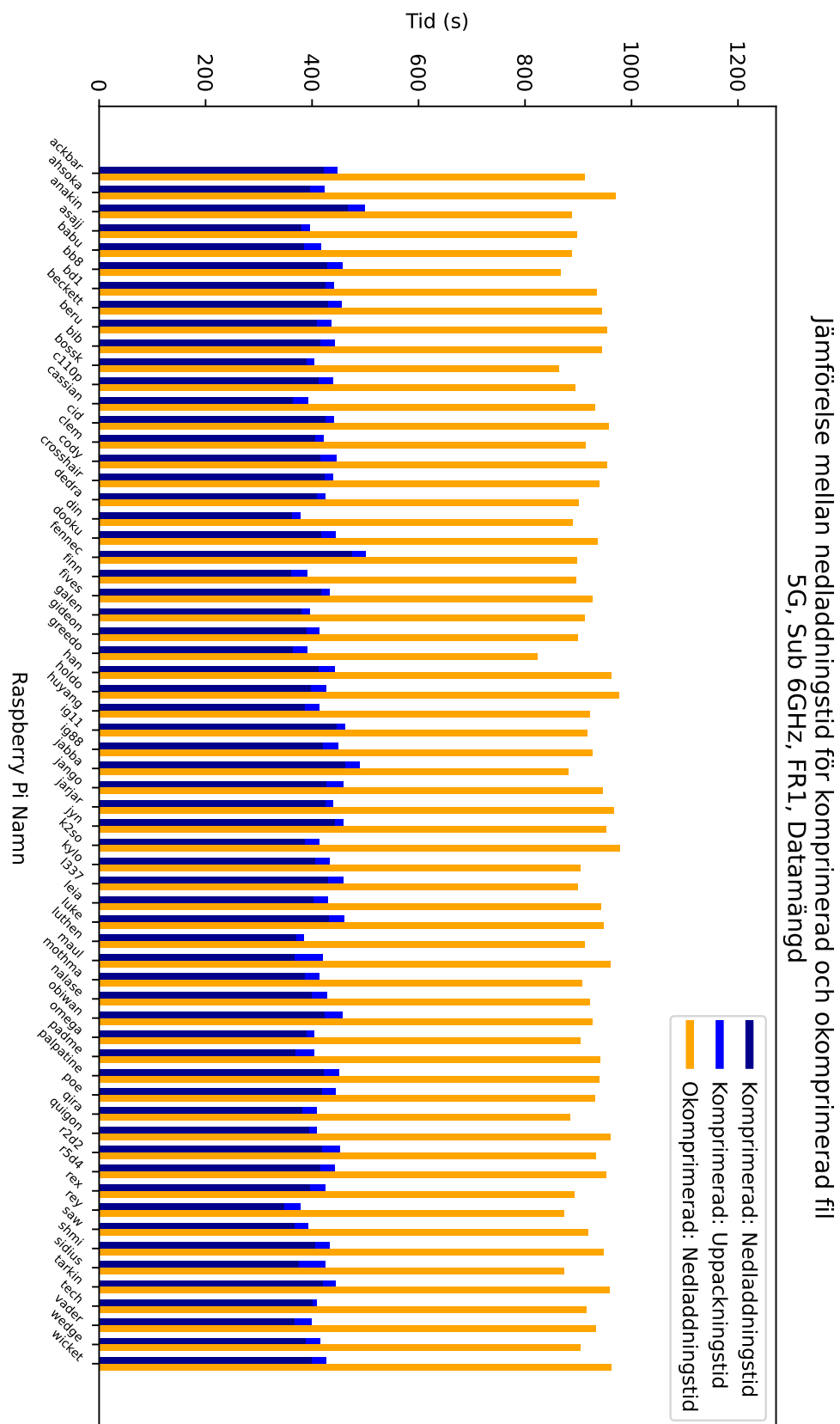
**Figur 4.10:** Wi-Fi, 5GHz, Skivavbild, 19 enheter.



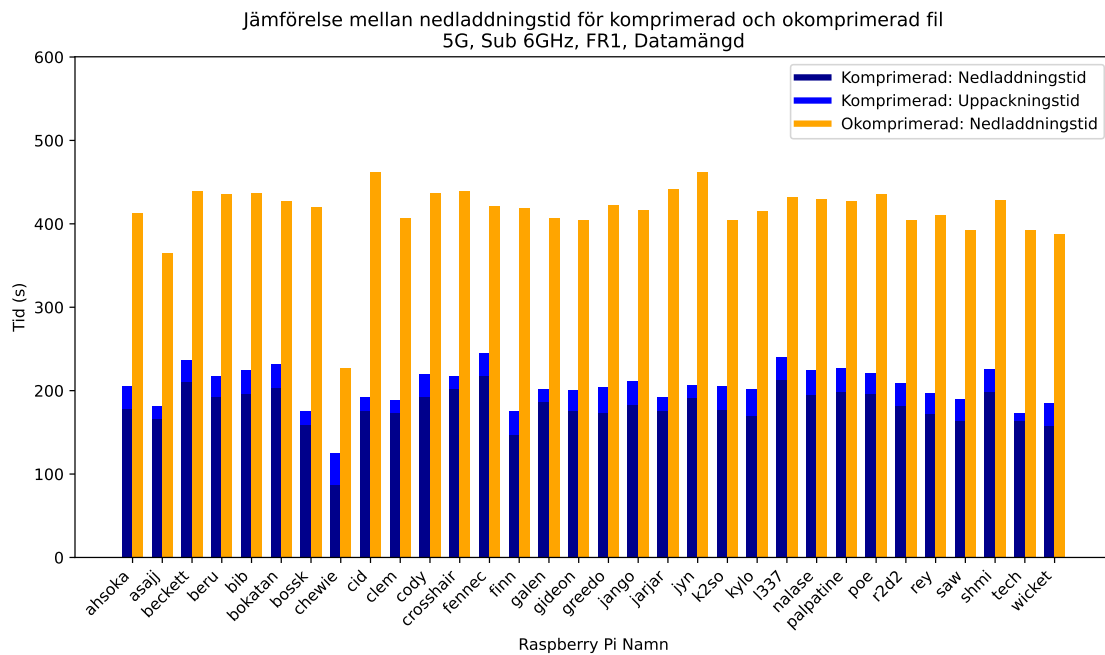
Figur 4.11: Wi-Fi, 5GHz, Datamängd, 19 enheter.

### 4.3 Samlad data från 5G

5G testerna utfördes med fyra stycken 5G-routrar över Sub-6GHz 5G. I figur 4.13 syns det att 5G är konsistent i och med att variationen mellan enheterna är liten och bandbredden är relativt hög, särskilt jämfört med Wi-Fi. Dock finns det en enhet, "Chewy", som blir klar avsevärt snabbare än de resterande. Även med 63 enheter, se figur 4.12, visar sig 5G vara nästan lika stabil som Ethernet. Under testerna användes fyra 5G routrar med ett jämnt antal enheter kopplade till varje router. På grund av tidsbrist testades endast nedladdningen av datamängden med 5G.



Figur 4.12: 5G, Sub 6GHz, FR1, Datamängd, 63 enheter.

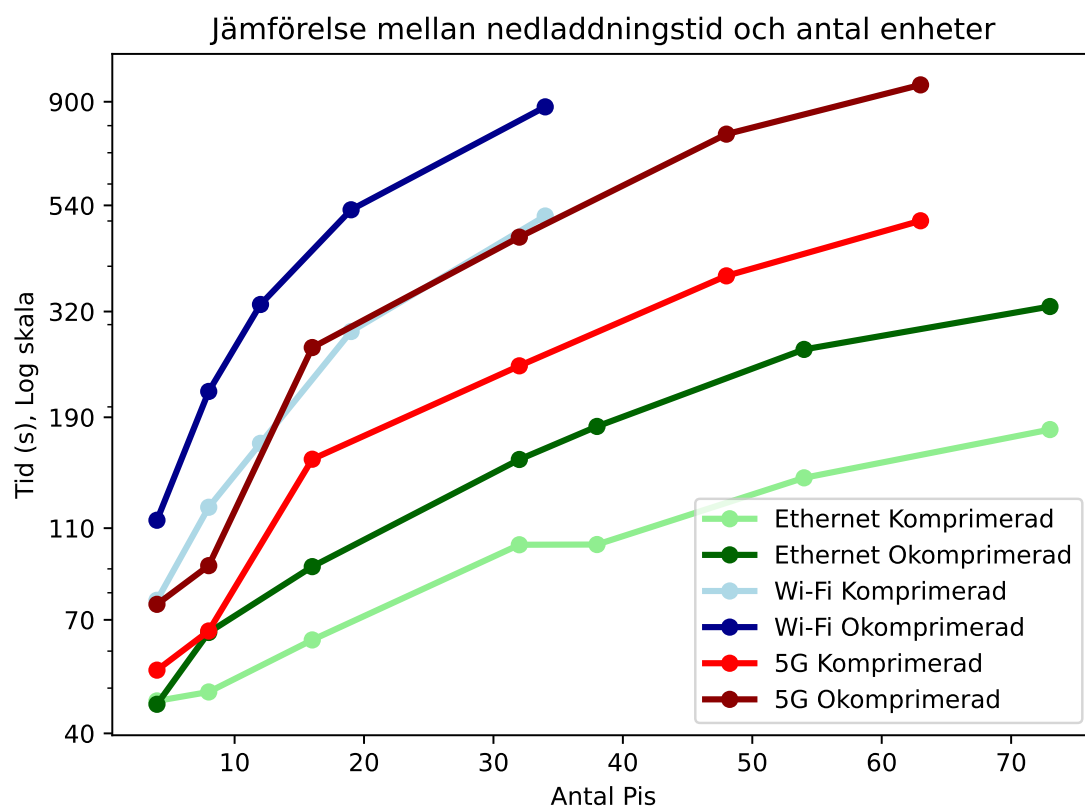


Figur 4.13: 5G, Sub 6GHz, FR1, Datamängd, 32 enheter.

## 4.4 Data för upskalning av nätverket

Eftersom Wi-Fi visade sig vara opålitligt med över cirka 20 uppkopplade enheter behövdes att två datapunkter tas bort i testet med 34 Pi. Dessa var extremt avvikande, tre till fyra gånger högre än genomsnittet för det testet, och bidrog således till att datan blev missvisande. Även testet med 63 enheter över 5G behövde rensas lite. En av enheterna rapporterades ta över 3000s, vilket var omöjligt då testet bara hade tagit cirka en halvtimme. Denna datapunkten togs bort.

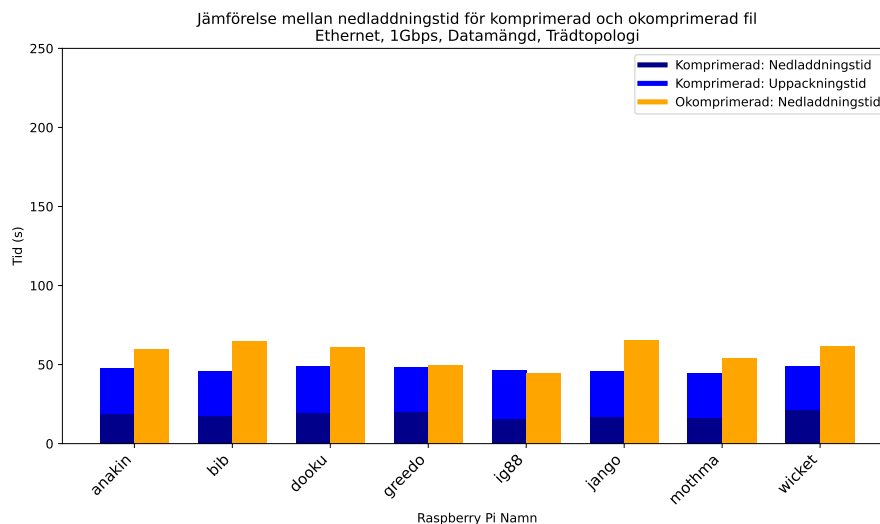
Efter att ha rensat datan kan grafen i figur 4.14 visa hur de olika länktypernas prestanda skalar när nätverket utvidgas med allt fler enheter. Wi-Fi datan uppvisar en logaritmisk kurva, vilket i en linjärt skalad graf hade motsvarat ett linjärt förhållande mellan tidsåtgång och antalet enheter. Datan för Ethernet verkar också följa ett linjärt förhållande, dock inte lika starkt. Datan för 5G verkar inte vara lika linjär, mellan 8 och 16 enheter hoppar grafen upp högt för att sedan stabilisera sig vid ett högre antal enheter. Nedladdningen av en okomprimerad fil med 5G tar vid ett högt antal enheter ungefär lika lång tid som nedladdningen av en komprimerad fil på Wi-Fi.



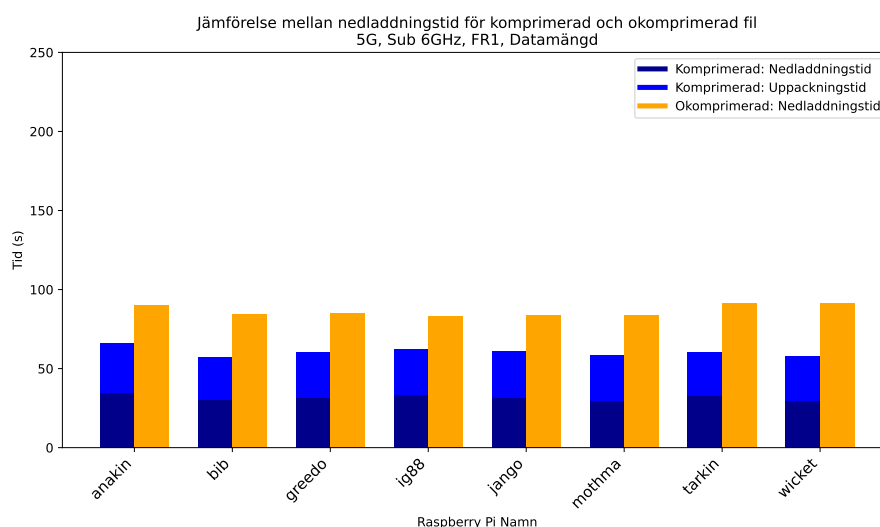
**Figur 4.14:** Linjediagram för olika länktyper med total nedladdningstid på y-axeln och antal enheter på x-axeln

## 4.5 Jämförbar data för en fast konfiguration

Figurerna 4.15, 4.16, och 4.17 visar resultaten av tester med 8 enheter över Ethernet respektive 5G och Wi-Fi där enheterna laddar ner datamängden. Från graferna framgår det tydligt att Ethernet är snabbast med cirka 50s för både komprimerad och okomprimerad fil. 5G är näst snabbast med cirka 60s för komprimerad fil och cirka 100s för okomprimerad. Wi-Fi är långsammast med cirka 100s för komprimerad fil och cirka 200s för okomprimerad.

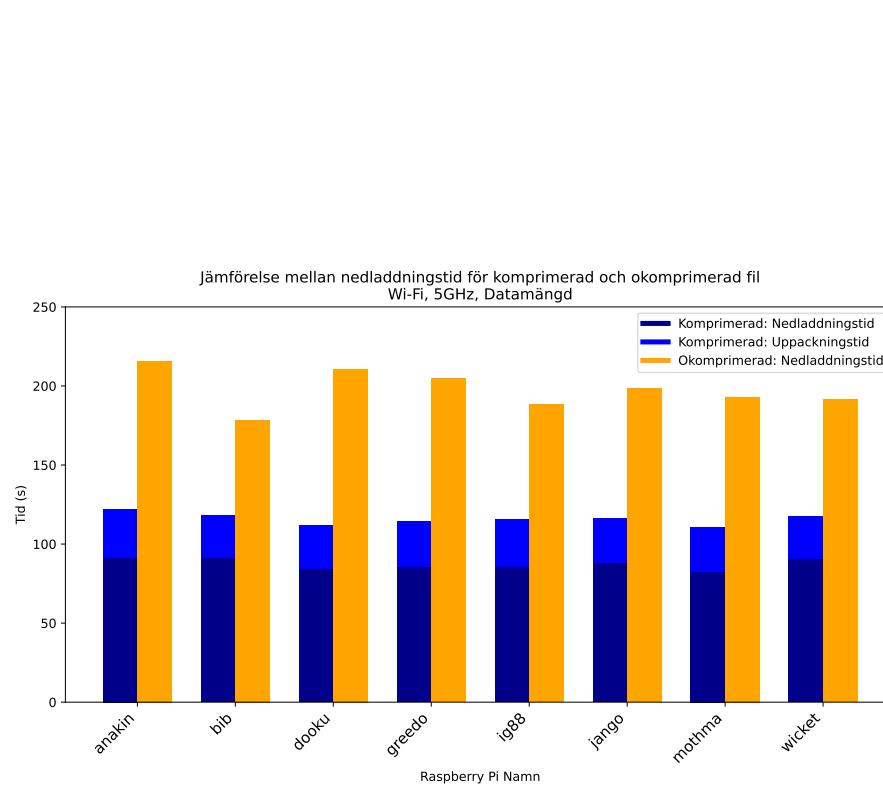


**Figur 4.15:** Ethernet, 1Gbps, Datamängd, Trädtopologi, 8 enheter.



**Figur 4.16:** 5G, Sub 6GHz, FR1, Datamängd, 8 enheter.





**Figur 4.17:** Wi-Fi, 5GHz, Datamängd, 8 enheter.



# 5

## Diskussion

I detta kapitel diskuteras resultaten som redovisades i kapitel 4. Först diskuteras de olika länktyperna individuellt och sedan jämförs de med varandra. Felkällor och utmaningar under projektets gång kommer också att diskuteras samt användningsområden för denna forskning.

### 5.1 Ethernet

Eftersom Ethernet är ett trådat nätverk är det inte lika känsligt för störningar som trådlösa nätverk. Detta bidrar till att göra Ethernet till det mest stabila nätverk vi testat. Variationen mellan nedladdnings tiderna för de olika enheterna var låg, även vid ett högt antal enheter. Inga enheter misslyckades med testerna.

När det kommer till nätverkstopologi verkar trädtopologi vara mer pålitligt än linjetopologi då resultaten från Ethernet-tester inte fluktuerar lika mycket, se sektion 4.1. Detta antas bero på att trädtopologin inte är lika känslig för störningar. I en linjetopologi delar alla enheter på en central buss vilket kan leda till mer kollisioner då alla paket måste ta sig igenom hela denna delade buss. I en trädtopologi å andra sidan är det bara den översta kabeln som kopplar trädet till routern som är delad, efter det skickas paketen ner i olika grenar av trädet och de har således inte lika stor möjlighet att störa varandra.

I testen med nedladdning av en skivavbild fanns det en enhet, "Quigon", som var mycket snabbare än resten av Raspberry Pi:erna. Efter en närmare inspektion visade det sig att den hade ett annat SD-kort än de andra enheterna. En hypotes är att detta SD-kortet hade högre läs- och skrivhastigheter och gav därför bättre resultat. Uppackning av en komprimerad fil är väldigt beroende av läs- och skrivhastigheter på disken som används och detta förklarar att "Quigon" packade upp den komprimerade filen nästan dubbelt så snabbt som de andra enheterna. Då detta skedde för båda testerna anser vi att datan från "Quigon" inte är representativ i den här instansen. Detta verkade dock inte vara något problem vid test med fler enheter och vid test med andra länktyper.

### 5.2 Wi-Fi

Experimenten med Wi-Fi visar att 5GHz bandet inte är lika pålitligt som 2.4GHz bandet med många aktiva enheter. I figur 4.8 framgår det att flera av enheterna

misslyckades med nedladdningen, och de som lyckades hade väldigt spridda resultat jämfört med 2.4GHz Wi-Fi som visas i figur 4.7. Det verkade inte spela någon roll var i rummet enheterna var placerade, de uppvisade alla stora variationer och lika stor risk att misslyckas.

Ytterligare tester visade att gränsen för en stabil koppling på 5GHz Wi-Fi låg på runt 20 enheter. Denna slutsats kunde dras då vid testning med olika antal Raspberry Pi-enheter så misslyckades enheter först när tester med 19 enheter kördes. Vid tester med 19 enheter så misslyckades ibland en av enheterna. Då vi gjorde testerna med 20 enheter eller fler så ser vi konsekvent att minst en misslyckas, och att andelen av enheter som misslyckas växer med antalet kantnoder som testas. Till exempel ser vi att vid testning av 37 enheter så misslyckades 10 av dessa, se figur 4.8.

Anledning till detta tros vara att enhetens anslutning bröts på grund av stockning i nätverket. Följaktligen togs beslutet att utföra de resterande 5GHz testerna med endast 19 enheter. Trots det låga antalet enheter som användes är resultaten väldigt spridda jämfört med Ethernet och 2.4GHz Wi-Fi testerna, se figur 4.9 och 4.10.

Vid tester på 2.4GHz Wi-Fi skalades detta bättre med hänseende till antalet kantnoder, och vi kunde köra fler enheter utan att de misslyckades. Vid testning av 37 Raspberry Pi-enheter misslyckades bara en, se figur 4.7. Men med fler enheter blev hastigheterna blev väldigt låga vilket medför att tidsåtgången blev väldigt stor. Detta på grund av att vi observerade en bandbredd på 50Mbps för 2.4GHz Wi-Fi. Vid tester på 37 stycken enheter tog en testning ungefär 2 timmar. Därför tog vi beslutet att inte fortsätta skala upp testerna på 2.4GHz bandet.

Anledningen till varför 2.4GHz skalar upp bättre än 5GHz, med hänseende till andelen enheter som lyckas, tros vara att 5GHz är mer känslig för störningar. Detta var något som introducerades i väldigt stora mängder då vi testade många Raspberry Pi-enheter parallellt.

### 5.3 5G

Experimentet med 5G visade sig vara mer pålitligt och stabilt jämfört med Wi-Fi. För 5G testerna användes trädtopologi i kombination med stjärntopologi, se figur 3.2. Vid tester på 32 enheter visade enheten "Chewie" ett intressant resultat, då komprimerad nedladdningstiden för denna enhet var mindre än 100 sekunder vilket visar mindre tid än de andra enheter som varierade mellan 150-200s. När det gäller okomprimerad nedladdningstid visade "Chewie" återigen en lägre nedladdningstid på ca 250s jämfört med alla andra enheter som hade mellan 380-460s. Efter kontroll av enheten "Chewie" visades inga skillnader i hårdvaran, därför tros det att felkällor från denna enhet kan ha gett missvisande resultat. På grund av detta användes inte "Chewie" i följande tester.

Nätverket skalades sedan upp till det högsta möjliga antalet noder som routern klarade av, 64 enheter. Ett av resultaten togs bort eftersom det var korrupt, efter

det hade vi data från 63 enheter, se figur 4.12. Resultatet visades sig vara stabila för alla enheterna och hade en komprimerad nedladdningstid mellan 350-450s och okomprimerad nedladdningstid mellan 800-975s.

För att dra en slutsats om hur uppskalning av nätverket påverkade nedladdningstiderna tog vi fram genomsnittstiden. För 32 enheter visade sig att det tog cirka 204s per enhet för den komprimerade nedladdningen, medan nedladdningstiden för de okomprimerade filerna tog cirka 414s per enhet. När nätverket skalades upp till 63 enheter, tog det cirka 430s för nedladdning av den komprimerade filen, och 922s för de okomprimerade filerna. Detta visar att en uppskalning till cirka dubbelt så många enheter tar lite längre än dubbelt så lång tid i både de komprimerade och de okomprimerade fallen. Detta tyder på att uppskalning av nätverket inte är helt linjärt.

## 5.4 Uppskalning av nätverket

För att göra jämförelsen mellan hur länktyperna skalade med antalet enheter i nätverket ritades en linjefgraf upp, se figur 4.14. Från grafen framgår det tydligt att nedladdning och uppackning av en komprimerad fil är snabbare än nedladdning av de okomprimerade filerna för alla länktyper och för alla antal enheter över cirka fyra stycken. Ethernet är klart den snabbaste länktypen, både vid nedladdning av en komprimerad och okomprimerad fil. Wi-Fi är långsammast och 5G ligger någonstans däremellan, men är närmare Wi-Fi än Ethernet sett till hastigheterna. Alla länktyperna följer relativt logaritmiska kurvor vilket innebär ett linjärt samband, då grafen är logaritmiskt skalad.

## 5.5 Fast konfiguration

En fast konfiguration med åtta enheter användes för att få helt jämförbar data mellan länktyperna, se sektion 4.5. Än en gång syns det att Ethernet är snabbast, 5G näst snabbast och Wi-Fi långsammast. Detta kan bero på att Ethernet har en fysisk uppkoppling, och inte är lika känslig för störningar.

## 5.6 Jämförelse mellan länktyperna

Ethernet verkar vara den mest stabila och pålitliga länktypen. Variationen mellan enheterna är liten och den skalar väl med allt fler kopplade enheter. 2.4GHz Wi-Fi visade sig också ha relativt små variationer och kunde hantera många kopplade enheter, dock var bandbredden väldigt begränsad med runt 50Mbps jämfört med Ethernets 1000Mbps. 5GHz Wi-Fi å andra sidan hade bättre bandbredd, runt 250Mbps, men fungerade väldigt dålig med över cirka 20 enheter kopplade då väldigt många enheter misslyckades. Nätverket blev helt överbelastat till den punkten att det var svårt att skicka kommandon över SSH eftersom så många paket tappades.

5G nätverket var ungefär lika stabilt som Ethernet. Dock kan detta ha berott på att ett "hybridnätverk" mellan Ethernet och 5G behövde användas eftersom gruppen bara hade fyra SIM-kort till sitt förfogande, mer om detta i sektion 5.8.3. Detta hybridnätverket bestod av fyra oberoende trådnätverk som alla hade en 5G router som centralnod. Detta var en hypotes, men eftersom det inte var möjligt att testa ett rent 5G nätverk går det inte att helt bevisa den. Det är möjligt att nätverket hade hanterat fler enheter som skickar data lika bra som Ethernet, men det hade också kunna bli värre likt det som hände med Wi-Fi. Laboration med fler 5G-routrar krävs för att veta säkert.

## 5.7 Felkällor

Vid laboration hämtades mätdata från ett flertal Raspberry Pi-enheter. Då mycket av denna mätdata är beroende på variabler vi inte haft kontroll över, har flera felkällor introducerats i mätdata.

### 5.7.1 5G-nätverket

Då Ericsson tillhandahåller 5G-nätverket, fanns det flertalet saker vi inte kunde konfigurera eller hade möjlighet att få ut data på. Vi kunde bland annat inte bekräfta vilka 5G-routrar som var kopplade jämnt mot vilken av de två 5G-antennerna som fanns i CASE-labbet. Detta kan ha inneburit att samtliga 5G-routrar var kopplade mot en och samma antenn. Detta kan ha lett till överbelastning av antennen och därmed sämre hastigheter. Vi kunde inte heller bekräfta att 5G-nätverket hade skalat bättre med flera kantenoder om båda 5G-antennerna används på grund av detta. Raspberry Pi:arna anslöts till 5G-routrarna via Ethernet. Detta utgjorde ingen felkälla eftersom Ethernet inte var en flaskhals i informationsflödet.

### 5.7.2 Varierande skrivhastigheter

Skrivhastigheten varierade mellan Raspberry Pi-enheter, vilket kan tillskrivas användningen av SD-kort som var av olika fabrikat. Dessa variationer har en direkt påverkan på både skriv- och läshastighet. Eftersom uppackning av en komprimerad fil nästan enbart är begränsad av läshastigheten på lagringsmediumet som filen ligger på så blev skillnaden extra stor när vissa enheter hade snabbare SD-kort.

### 5.7.3 Varierande nätverkshastigheter

Då nätverkshastigheter varierar med tiden på grund av flera variabler, bland annat hur belastat nätverket är, så varierar mätdata. Detta är något som förväntas, men bör generellt sett inte ge allt för stor variation på mätdata. Eftersom vi utgår från att nätverken är belastat ganska jämnt, borde det alltså inte ge en för stor variation i hastigheter. Undantagsfallet, där mätdata varierade kraftigt på grund av nätverkshastigheter, tas upp i sektion 5.8.4.

## 5.8 Utmaningar

Under projektets gång har ett flertal problem och utmaningar uppstått. I detta kapitel presenteras dessa och vilka komplikationer de medfört.

### 5.8.1 Portabilitet av Bash-skriptet

Då vi arbetade i olika lokaler så arbetade vi ofta utan Raspberry Pi-enheterna. Detta medförde att testning av Bash-skriptet gjordes via WSL som körde Ubuntu. Då Linux-distribution och systemspråket inte är detsamma på Raspberry Pi-enheterna och WSL-instansen uppstod flera problem vid körning av Bash-skriptet.

Ett problem var att med svenska som systemspråk används decimalkomma, vilket blev problematiskt när Bash-skriptet lägger in mätdata i en CSV-fil, då kommatecken i en CSV-fil innebär en ny kolumn. Då samtliga Raspberry Pi-enheter är inställda på svenska så valde vi att lösa detta genom att lägga till en funktion i Bash-skriptet som byter ut decimalkomma mot decimalpunkt. Vi fick liknande problem med så kallade *Narrow No-Break Space*, NNBS, som är specialtecken som läggs in som avgränsare till tusental om man har svenska som systemspråk. Vi löste detta genom att ta bort dessa vid körning av Bash-skriptet.

När vi skrev skriptet så hade vi i åtanke att skriptet skulle kräva så få extra program som möjligt, och att det helst bara skulle använda sig av de program som redan finns förinstallerade. Då programmen som är förinstallerade varierar mellan Raspberry Pi OS och Ubuntu så fanns det delar av Bash-skriptet som inte kunde köras. Detta kunde åtgärdas genom att göra ett nytt skript som installerar de program som saknades. Installations-skriptet kunde sedan köras på liknande sätt som vi kör Bash-skriptet.

### 5.8.2 Problem vid testning via Wi-Fi

Vid experiment med Wi-Fi fick vi problem när vi skalade upp till 20 eller fler enheter. Förutom att varje körning av Bash-skriptet tog väldigt lång tid på grund av de relativt låga nedladdningshastigheterna, så tappade vissa Raspberry Pi-enheter anslutningen. Detta innebar att körningen av testet på dessa enheter misslyckades då de inte kunde återuppta nedladdningen. Detta problem blev alltmer frekvent och uppträdde på en större andel av Raspberry Pi-enheterna ju fler enheter vi testade med. Ett annat problem som vi inte hade räknat med var att routern vi använde oss av har en gräns på 64 stycken Wi-Fi-uppkopplade enheter.

Under 5GHz testerna blev Wi-Fi nätverket överbelastat till den grad att det var svårt att övervaka testet. Detta eftersom så många paket tappades att SSH blev extremt långsamt. SSH visade sig vara extra känslig mot överbelastning av nätverket eftersom den behöver skicka fyra paket för varje symbol som skall skickas mellan datorerna.

### 5.8.3 För få SIM-kort

För de storskaliga experimenten med 5G-nätverket var det planerat att vi skulle använda fler 5G-routrar än vad vi gjorde. Tanken var att vi skulle koppla 4 stycken Raspberry Pi-enheter till varje 5G-router. Detta innebar att för det största 5G-testet med 63 kantnoder, så skulle 16 stycken 5G-routrar användas. Istället användes bara 4 stycken 5G-routrar, detta då vi hade för få SIM-kort till vårt förfogande. I och med att vi endast tilldelades 4 stycken SIM-kort ledde det till att vi enbart kunde använda oss av 4 stycken 5G-routrar. Detta innebar även att vi använde oss av en hybridtopologi, som bestod av en kombination av stjärntopologi och trädtopologi. Detta gav oss intressanta resultat men fler slutstater hade kunnat dras om fler 5G-routrar kunde användas.

### 5.8.4 5G-nätverket

Som tidigare nämnts så fanns det flertalet saker vi inte kunde konfigurera på 5G-nätverket som Ericsson tillhandahöll. Detta ledde till problem vid vissa tillfällen då experiment skulle utföras via 5G-nätverket. Bland annat så var FR1-banden avstängda vid vissa tillfällen när tester skulle utföras. Detta resulterade i låga hastigheter, cirka 20-25Mbps, vilket gjorde att testerna inte var representativa då övriga tester gjordes med cirka 400-500Mbps.

En stor utmaning med 5G-nätverket var att mobila nätverk är så kallat "isolerade", vilket betyder att enheter på nätverket inte kan kommunicera med varandra. Detta utgjorde ett problem eftersom tidigare tester med Ethernet och Wi-Fi startades genom att en dator direktkopplades till nätverket och skickade startkommandot via SSH. Detta var inte möjligt att göra via 5G-nätverket. För att komma runt detta problem kopplades därför alla enheter också upp via Wi-Fi och det var via denna länken som testkoden distribuerades och startades. Detta fungerade tillfredsställande men bidrog till mer komplexitet och fler saker som kunde gå fel. Ett problem som uppdagades först när testet med 75 enheter skulle utföras var att Wi-Fi-routern som användes hade en begränsning på 64 stycken Wi-Fi uppkopplade enheter. Eftersom det inte fanns tid att undersöka en annan lösning ledde detta till att 5G-testerna begränsades till 64 enheter.

## 5.9 Samhälleliga och etiska aspekter

Eftersom detta främst är ett tekniskt arbete som ämnar ta reda på vilken länktyp som presterar best i olika sammanhang finns det vad vi kunde se inga direkta etiska problem. Datamängden som användes i experimenten var testdata från företag som vi hade tillåtelse att använda. Den innehöll inga personuppgifter. Experimenten i sig gick ut på att koppla upp Raspberry Pi-enheter i olika nätverkskonfigurationer och ladda ner den ovannämnda datamängden, vi kan inte se några etiska problem med detta. Det finns dock vissa sekundära samhälleliga och etiska aspekter med IoT i allmänhet och vissa av länktyperna i synnerhet som kommer att diskuteras i denna sektion.



### 5.9.1 Integritet och IoT

Med mer och mer smarta IoT enheter i våra liv får företagen som säljer dessa mer och mer möjligheter att samla in data om våra beteenden. Denna data säljs ofta för att användas i marknadsföringssyfte vilket kan vara väldigt integritetskränkande.

### 5.9.2 Klasskillnader

De nya teknologierna som förs med implementation av 5G mobiltelefoni kräver investering i ny infrastruktur. Denna investering kommer att behöva komma från telekommunikationsbolag vars mål är att gå med vinst samtidigt med utan att bli utkonkurrerade. Ett möjligt utfall av att införa 5G ur detta perspektivet är att förmögna klienter och grannområden får teknologin före resten av samhället. Bolagen kan till exempel bedöma att det inte är värt att satsa på 5G infrastruktur i fattigare områden till samma grad än rika. En avvikelse som detta kommer att leda till att de förmögna klienterna får en mycket större del av internet-överföringshastighet medan de flertal icke-förmögna får det som finns kvar. Skillnaden mellan de olika samhällsklasserna kommer då att ökas och kan leda till vidare ojämlikhet. Vem får tillgång till 5G teknologin, endast de rika?

### 5.9.3 Positiva aspekter

Detta projektet har undersökt vilka länktyper som som passar bäst i olika sammanhang. Med hjälp av denna forskning kan individer och organisationer lättare välja den länktypen som passar bäst för deras specifika applikation. Detta kan leda till effektivare nätverk som sparar både tid och energi.

## 5.10 Användningsområden

Det finns flera användningsområden för det som undersökts i detta projektet. Som nämnt i sektion 3.2.1 valdes testdata som skulle kunna representera ett verkligt scenario. Ponera då att man har en stor uppsättning med IoT-enheter där man behöver köra en fabriksåterställning på samtliga enheter. Då behöver man skicka just en skivavbild till dessa enheterna och hastighet kan vara essentiellt för att effektivisera processen. Tyvärr fanns det ej tid att genomföra tester med denna filtyp förutom på Ethernet men testresultaten som erhållits kan ändå ligga som grund för att se hur lång tid något sådant skulle kunna tänkas ta.

Ett annat exempel kopplat till vårt val av testdata är federerad inlärning. Där måste stora dataset skickas till de medverkande enheterna som sedan använder datan för att träna en AI-modell. Denna modell måste sedan skickas tillbaka till den koordinerande servern för att kombineras med de andra enheternas modeller. Detta kräver att många enheter skickar och tar emot mycket data samtidigt, vilket kan optimeras genom att välja den bästa länktypen för sammanhanget. Därmed har resultaten för vårt projekt kunnat ge en bild av vad som passar bäst i just detta sammanhang.

Federerad inlärning är trots allt en ny, innovativ teknik som har svårigheter när det kommer till just aspekten av att effektivisera.

### 5.11 Reflektioner och framtida utvecklingsområden

Direkta jämförelser mellan de tre olika länktyperna visade sig inte vara helt lämpligt i fallet då antalet Raspberry Pi enheter översteg 20, eftersom vissa enheter då misslyckades med testet via Wi-Fi. I framtida experiment bör en bättre Wi-Fi-router användas. För 5G-testerna behöver en bättre metod för distribution och exekvering av testkoden hittas, alternativt behövs en bättre Wi-Fi-router som kan hantera fler Wi-Fi uppkopplade enheter. Jämförelser av överföringshastigheterna kan endast göras inom den mängd enheter som finns inom samtliga länktypers kapacitet.

Sent under projektets gång påbörjades utvecklingen av en version av Bash skriptet som kunde samla in data för antalet CPU cykler och antalet instruktioner som behövdes för att exekvera skriptet. På grund av tidsbrist var det inte möjligt att utföra några tester med denna versionen av skriptet, men i framtida experiment kan detta vara något att titta på.

Ett utvecklingsområde för projektet hade varit att tillämpa detta på något praktiskt, som till exempel federerad inlärning. Detta hade gett mer intressanta resultat i och med att de hade varit mer tydligt förankrade med verklig data, och där inte alla kantnoder behöver ladda ner all data samtidigt.

# 6

## Slutsats

Ethernet visade sig vara den snabbaste och mest effektiva länktypen med en hastighet upp till 1000Mbps, och som skalas väl med fler enheter. Vid behov av en trådlös länktyp är 5G den mest effektiva länktypen med en hastighet mellan 400-500Mbps, hur väl den skalas med fler kantnoder är svårt att säga på grund av begränsningarna vi haft när vi utfört tester via 5G. Wi-Fi med 5GHz skalas inte upp väl, och vid för stor mängd uppkopplade enheter så kan det ses som obrukbart. Wi-Fi med 2.4GHz skalas upp något bättre än Wi-Fi på 5GHz-bandet, men på grund av den låga bandbredden så kommer tidsåtgången vid nedladdning av större filer vara väldigt hög.

Vid tester med Ethernet så är trädtopologi snabbare än linjetopologi. Detta kan tillskrivas att vid linjetopologi så måste en väldigt stor mängd data passera genom varje switch, vilket leder till flaskhalsar.

Testerna visade att nedladdning och uppackning av en komprimerad fil nästan alltid är snabbare än nedladdning av en okomprimerad version av samma fil. Det fanns ett fåtal tillfällen när detta inte stämde så som vid test via Ethernet med väldigt få enheter då uppackningen tog väldigt lång tid relativt till den totala tidsåtgången. Detta observerades även vid vissa av Wi-Fi testerna, där ett fåtal enheter tog mycket längre tid på sig än de resterande. Detta kan vara på grund av instabilitet i nätverket, men fler experiment krävs för att veta säkert.



# Litteraturförteckning

- [1] J. Lindner, “Dependence On Technology Statistics [Fresh Research],” *Git-nux*, 2023. [Online]. Available: <https://gitnux.org/dependence-on-technology-statistics/>
- [2] J. Schulze. (2024, Jan) What is the internet of things? (IoT). Accessed: 2024-02-06. [Online]. Available: <https://www.ibm.com/topics/internet-of-things>
- [3] A. McCain, “How Fast Is Technology Advancing? [2023]: Growing, Evolving, And Accelerating At Exponential Rates,” *Zippia*, 2023.
- [4] ITU, “Number of internet users worldwide from 2005 to 2023 (in millions),” *Statista*, 2023. [Online]. Available: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>
- [5] Worldbank, “Population, total - World,” *Worldbank*, 2022. [Online]. Available: <https://data.worldbank.org/indicator/SP.POP.TOTL?end=2022&locations=1W&start=2000>
- [6] IoTSverige, “IoT – så funkar det,” retrieved 2024-01-30. [Online]. Available: <https://iotsverige.se/om-oss/iot-sa-funkar-det>
- [7] Ericsson, “Discover the power of 5G,” *Ericsson*, 2024. [Online]. Available: <https://www.ericsson.com/en/5g>
- [8] WPI, “5G technology, Explained,” <https://www.wpi.edu/news/explainers/5g-technology>, [Accessed 12-05-2024].
- [9] A. K. M. Bahalul Haque, M. Oahiduzzaman Mondol Zihad, and M. Rifat Hasan, *5G and Internet of Things–Integration Trends, Opportunities, and Future Research Avenues*. Singapore: Springer Nature Singapore, 2023, pp. 217–245, doi:10.1007/978-981-99-3668-7\_11.
- [10] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and D. Soldani, “SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility,” *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 1, pp. 82–92, 2015, doi:10.1002/ett.2915.
- [11] A. J. Ruoff. (2023, Apr.) Telecom R&D; Who’s at the top? Silverlinings. Accessed : 2024-03-29. [Online]. Available: <https://www.fierce-network.com/5g/here-are-telecoms-spending-most-and-least-rd>
- [12] D. Kridel, P. Rappoport, and L. Taylor, *The Demand for High-Speed Access to the Internet*. Boston, MA: Springer US, 2002, pp. 11–22, doi: 10.1007/978-1-4615-0861-8\_3.
- [13] A. Ahad, Z. Ali, A. Mateen, M. Tahir, A. Hannan, N. M. Garcia, and I. M. Pires, “A Comprehensive review on 5G-based Smart Healthcare Network Security: Taxonomy, Issues, Solutions and Future research directions,” *Array*, vol. 18, p. 100290, 2023, doi: 10.1016/j.array.2023.100290.

- [14] A. Malik, V. Parihar, B. Bhushan, R. Chaganti, S. Bhatia, and P. N. Astya, *Security Services for Wireless 5G Internet of Things (IoT) Systems*, bookTitle="5G and Beyond. Singapore: Springer Nature Singapore, 2023, pp. 169–195, doi: 10.1007/978-981-99-3668-7\_9.
- [15] R. White, “115 mobile operators now investing in 5G Standalone networks globally,” Aug. 2023, accessed: 2024-05-09. [Online]. Available: <https://gsacom.com/press-release/115-operators-investing-in-5g-standalone/>
- [16] C. E. Spurgeon and J. Zimmerman, *Ethernet: The Definitive Guide*, 2nd ed. O’Reilly Media, Inc., 2014, doi:10.5555/2614415.
- [17] Veeravalli, Venugopal V. and El Gamal, Aly, *Interference Management in Wireless Networks: Fundamental Bounds and the Role of Cooperation*. Cambridge University Press, 2018, doi:10.1017/9781316691410.
- [18] I. Grigorik, *Introduction to Wireless Networks*. O’Reilly Media, 2013, ch. 5, pp. 79–88, isbn:9781449344764.
- [19] Britannica, The Editors of Encyclopaedia. Ethernet. Encyclopedia Britannica. Accessed: 10 April 2024. [Online]. Available: <https://www.britannica.com/technology/Ethernet>
- [20] IEEE Standards Association, “The Evolution of Wi-Fi Technology and Standards,” retrieved 2024-04-24. [Online]. Available: <https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/>
- [21] B. Bellalta, “IEEE 802.11ax: High-efficiency WLANs,” *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016, doi: 10.1109/MWC.2016.7422404.
- [22] C. Cox, *Architecture of the Radio Access Network*. John Wiley & Sons, Ltd, 2021, ch. 3, pp. 55–71, doi:10.1002/9781119602682.ch3.
- [23] R. Dilli, “Analysis of 5G wireless systems in FR1 and FR2 frequency bands,” in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, Mar. 2020, doi: 10.1109/ICIMIA48430.2020.9074973.
- [24] T. Bogale, X. Wang, and L. Le, “Chapter 9 - mmWave communication enabling techniques for 5G wireless systems: A link level perspective,” in *mmWave Massive MIMO*, S. Mumtaz, J. Rodriguez, and L. Dai, Eds. Academic Press, 2017, pp. 195–225, doi:10.1016/B978-0-12-804418-6.00009-1.
- [25] M. Pieska, A. J. Kassler, H. Lundqvist, and T. Cai, “Improving TCP Fairness over Latency Controlled 5G mmWave Communication Links,” in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, 2018, pp. 1–8. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8385490>
- [26] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, “Federated learning review: Fundamentals, enabling technologies, and future applications,” *Information Processing & Management*, vol. 59, no. 6, p. 103061, 2022, doi:10.1016/j.ipm.2022.103061.
- [27] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 3, p. 160, Mar 2021, doi:10.1007/s42979-021-00592-x.
- [28] Oxford Learners Dictionary. Machine Learning noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner’s Dictionary at OxfordLearnersDictionaries.com. Oxford University Press. Accessed :

- 2024-03-24. [Online]. Available: <https://www.oxfordlearnersdictionaries.com/definition/english/machine-learning>
- [29] J. A. Nichols, H. W. Herbert Chan, and M. A. B. Baker, “Machine learning: applications of artificial intelligence to imaging and diagnosis,” *Biophys. Rev.*, vol. 11, no. 1, pp. 111–118, Feb. 2019.
- [30] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, “A review of applications in federated learning,” *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020, doi:10.1016/j.cie.2020.106854. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835220305532>
- [31] T. R. Gadekallu, Q.-V. Pham, Huynh-The, Thien, S. Bhattacharya, P. K. R. Maddikunta, and M. Liyanage, “Federated learning for big data: A survey on opportunities, applications, and future directions,” 2021, doi:10.48550/arXiv.2110.04160.
- [32] A. Skoulikari, *Learning Git: A Hands-On and Visual Guide to the Basics of Git*. O’Reilly, 2023, isbn:9781098133917.
- [33] A. Almurayh, “Virtual Private Server,” 2010. [Online]. Available: [http://cs.uccs.edu/~cs526/studentproj/projS2010/aalmuray/doc/Almurayh\\_VPS.pdf](http://cs.uccs.edu/~cs526/studentproj/projS2010/aalmuray/doc/Almurayh_VPS.pdf)
- [34] StackScale, “Most popular programming languages in 2023 [ranking],” 2023, accessed: 2024-05-09. [Online]. Available: <https://www.stackscale.com/blog/most-popular-programming-languages/>
- [35] M. Lutz and D. Ascher, *Learning Python*. O’Reilly Media, 2003, isbn:9780596551933.
- [36] (2023, November) What is the Windows Subsystem for Linux? [Online]. Available: <https://learn.microsoft.com/en-us/windows/wsl/about>
- [37] (2023, December) Comparing WSL Versions. [Online]. Available: <https://learn.microsoft.com/en-us/windows/wsl/compare-versions>
- [38] R. Stallman. Linux and the GNU System. [Online]. Available: <https://www.gnu.org/gnu/linux-and-gnu.html>
- [39] T. Reidt. (2022) Which Raspberry Pi 4 OS should you choose? [Online]. Available: <https://emteria.com/blog/raspberry-pi-4-os>
- [40] Ubuntu. The story of Ubuntu. [Online]. Available: <https://ubuntu.com/about>
- [41] Y. Shafranovich, “Common Format and MIME Type for Comma-Separated Values (CSV) Files,” RFC 4180, Oct. 2005, doi: 10.17487/RFC4180.
- [42] E-lins, *H685f Gigabit 5G Router Datasheet*. [Online]. Available: <https://www.e-lins.com/EN/download/H685-5G-Router-Datasheet-Eng.pdf>
- [43] Dustin, *RT-AX82U AX5400 Dual Band WiFi6 Gaming Router 802.11AX*. [Online]. Available: <https://www.asus.com/networking-iot-servers/wifi-routers/asus-gaming-routers/rt-ax82u/>
- [44] D-Link. DGS-105/108: 5/8-Port Gigabit Unmanaged Desktop Switch. [Online]. Available: [https://www.dlink.com/uk/en/-/media/business\\_products/dgs/dgs-105/datasheet/dgs\\_105\\_108\\_e1\\_datasheet\\_en.pdf](https://www.dlink.com/uk/en/-/media/business_products/dgs/dgs-105/datasheet/dgs_105_108_e1_datasheet_en.pdf)
- [45] Linksys. Linksys LGS124. [Online]. Available: <https://www.linksys.com/on/demandware.static/-/Sites-master-product-catalog/default/dw5c9b4669/Data-sheet-Linksys/linksys-switches-lgs116-lgs124-data-sheet.pdf>

- [46] (2024) Nätverkskabel / Patchkabel LSZH RJ-45 RJ-45 CAT 6a 5m Vit. [Online]. Available: <https://www.dustin.se/product/5011153031/tp-cable>
- [47] S. C. L. Cheah Wai Zhao, Jayanand Jegatheesan, “Exploring IOT Application Using Raspberry Pi,” *International Journal of Computer Networks and Applications*, feb 2015. [Online]. Available: <https://ijcna.org/Manuscripts/Volume-2/Issue-1/Vol-2-issue-1-M-04.pdf>
- [48] T. Bräunl, *Embedded Robotics*, 4th ed. Springer Singapore, 2022.
- [49] Raspberry Pi Foundation, *Raspberry Pi 400*, January 2022. [Online]. Available: <https://datasheets.raspberrypi.com/rpi400/raspberry-pi-400-product-brief.pdf>
- [50] —, *Raspberry Pi 4 Model B*, December 2023. [Online]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>
- [51] E. Kurt, *Network Topology: The Physical and Logical Structure of a Network Connection Between Model and Nodes*. Independently published, 2019, iISBN: 9781672463591.
- [52] D. Vaidya, S. Mukherjee, M. A. Zagrodnik, and P. Wang, “A review of communication protocols and topologies for power converters,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Oct. 2017, pp. 2234–2235, doi: 10.1109/IECON.2017.8216376.
- [53] P. P. A. Santanu Santra, “A Study And Analysis on Computer Network Topology For Data Communication,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 1, January 2013. [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=73b5e4626684f4814360a0d2c58374dbb917211b>
- [54] S. V. Pankaj Singh, “Network Topologies,” *International Journal of Research & Development Organisation*, vol. 1, no. 5, May 2015. [Online]. Available: <http://13.234.104.160/index.php/cse/article/view/1184/1110>
- [55] S. Bach, A. Karlsson, J. Körner, and L. Sundkvist, “Monitoring in IoT/CPS,” Chalmers Tekniska Högskola, Tech. Rep., Oct. 2023.
- [56] (2024) Overleaf. [Online]. Available: <https://www.overleaf.com/learn>





GÖTEBORGS  
UNIVERSITET

---



**CHALMERS**