Projektet

Kaan Bür och Maria Kihl

LUND
UNIVERSITY

Mål med projektet

- Målet med projektet är att du ska sätta in dig i ett tekniskt system som använder datakommunikation samt lära ut detta till dina kursare.
- Du gör detta genom att antingen undersöka en applikation med **Wireshark** eller skriva ett **Whitepaper** om ett aktuellt Internet-relaterat ämne.
- EITF45 Datorkommunikation ingår i progressionsplanen för skrivande på D.

Projektspår

- Wireshark
  - Här ska du analysera en applikation med hjälp av Wireshark enligt de riktlinjer som finns i projektbeskrivningen.
- Whitepaper
  - Här ska du läsa in dig på ett spännande teknikområde och skriva en rapport som lär ut detta område till dina kursare.

För alla gäller att ni ska arbeta med en vetenskaplig metodik.

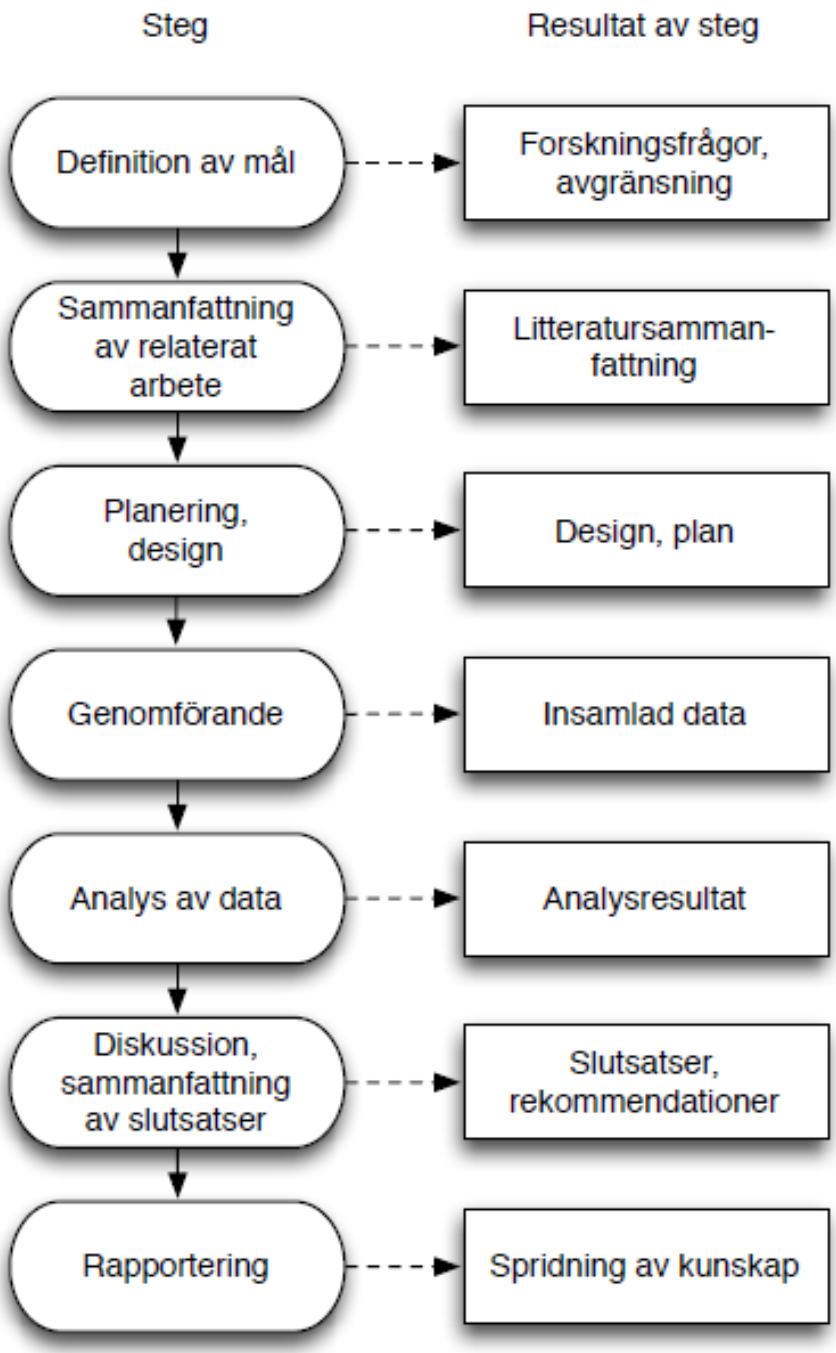
Vad är Wireshark?

- Ett verktyg för att mäta ett flertal parametrar för datatrafiken som transporteras på en länk.
- Mer information och tutorial kommer på övningarna denna veckan.
- Du kan ladda ner Wireshark via https://www.wireshark.org/download.html

Steg i en vetenskaplig undersökning

1. Definition av mål med studien.
2. Sammanfattning av relaterad forskning och litteratur.
3. Planering av forskningen.
4. Genomförande av datainsamling. (för Whitepaper insamlas kunskap om området)
5. Analys av data (kunskap).
6. Diskussion och sammanfattning av slutsatser

Källa: Martin Höst, EDAA35 Utvärdering av programvarusystem


Steg
Resultat av steg
Definition av mål
Forskningsfrågor, avgränsning
Sammanfattning av relaterat arbete
Litteratursamman-fattning
Planering, design
Design, plan
Genomförande
Insamlad data
Analys av data
Analysresultat
Diskussion, sammanfattning av slutsatser
Slutsatser, rekommendationer
Rapportering
Spridning av kunskap

Formella krav på rapporten

1. Rapporten ska vara skriven på svenska.
2. Formatet ska följa ”IEEE Transactions” (finns att ladda hem från hemsidan till Latex och Word).
3. Rapporten ska vara minst 3 sidor exklusive referenser.
4. Ett lämpligt antal bilder ska inkluderas i rapporten men de får inte finnas där bara för att fylla ut sidorna.
5. Referenslistan ska följa IEEEs mall för referenser.

Ytterligare krav på rapporten

- **Rapporten** får endast **väldigt kortfattat** ta upp protokoll/system/tekniker som ingår i föreläsningar och övningar i kursen. Dvs, ni får inte fylla ut rapporten med sånt som era kursare redan kan.
- **Wireshark**: Applikationen som undersöks måste vara tydligt Internet-baserad, dvs applikationen måste ha Internet-access för att fungera.
- **Whitepaper**: Ämnet måste vara tillräckligt brett för att fylla 3-4 sidor utan ”fluff” och det ska finnas forskning inom ämnet.

En teknisk rapport

(Källa: Martin Höst, EDAA35 Utvärdering av programvarusystem)

- **Abstract**
  - Ca 200 ord som marknadsför rapporten.
- **Inledning**
  - Sätter rapporten i sitt sammanhang så att läsaren vet vad det handlar om.
- **Bakgrund och relaterat arbete**
  - Översikt av systemet som undersöks, och sammanfattning av tidigare forskning/arbete (ej sånt som ingår i kursen).

En teknisk rapport forts.

- **Frågeställning**
  - Sammanställning av de frågeställningar eller forskningsfrågor som behandlas i rappporten.
- **Metodbeskrivning**
  - Beskrivning av de metoder du använt för att besvara frågeställningarna. Du ska även förklara varför du valt dessa metoder. Om du kör Wireshark ska din uppställning och experiment beskrivas här.

En teknisk rapport forts.

- **Validitet/Begränsingar**
  - Vilka begränsningar har du gjort i din studie? Finns det några validitetshot?
- **Resultat och Diskussion**
  - Beskrivning av det arbete du utfört samt en sammanställning och diskussion av resultaten. Resultat och Diskussion kan delas upp på olika sätt beroende på studie.

En teknisk rapport forts.

- **Slutsatser**
  - Presentation av svaren på dina frågeställningar. Detta avsnitt brukar även innehålla en sammanfattning av din studie. Om du gjort ett Whitepaper så kan detta avsnitt innehålla dina personliga åsikter om ämnets framtid.
- **Fortsatt arbete**
  - Kan vara en del av avsnittet med slutsatser. Vad är lämpligt att fortsätta med?
- **Referenser**

Förslag på disposition för Wireshark

1. Abstract / Sammanfattning (kan skrivas på engelska eller svenska)
2. Introduktion och bakgrund
   - Ska inkludera frågeställning
   - Får endast väldigt kort ta upp teori som ingår i kursen
3. Metodbeskrivning
   - Ska inkludera valididet/begränsningar
4. Experiment
5. Resultat och diskussion
6. Egna slutsatser
7. Referenser

Förslag på disposition för Whitepaper

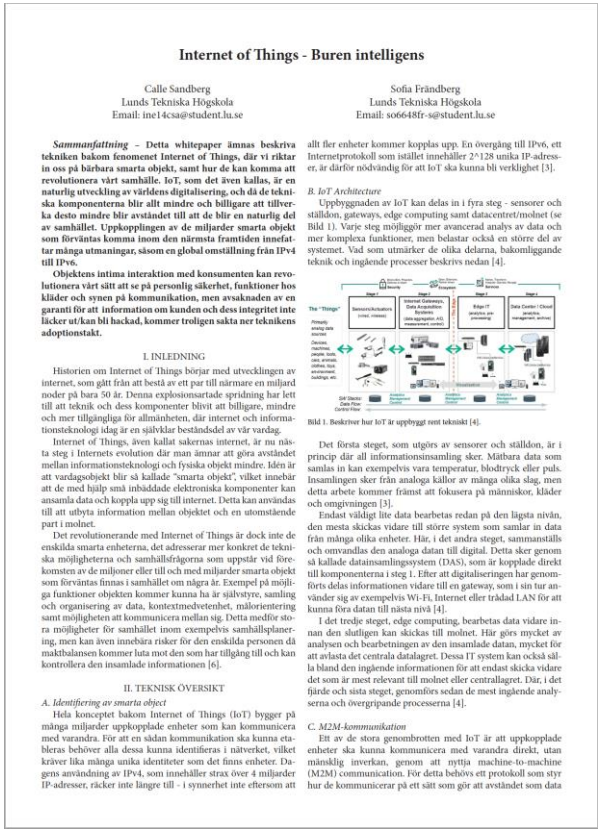
1. Abstract/Sammanfattning (kan skrivas på engelska eller svenska)
2. Introduktion och bakgrund
   - Ska inkludera frågeställning och metodbeskrivning
3. Teknisk översikt
   - Ska inkludera valididet/begränsningar
   - Får endast väldigt kort ta upp teori som ingår i kursen.
4. Nutida tillämpningar
5. Framtida och möjliga tillämpningar
6. Konkurrerande teknologier och standarder. Fördelar och nackdelar
7. Egna slutsatser
8. Referenser

Men kom ihåg

Välj de rubriker/avsnitt du anser passar bäst för just ditt ämne och arbete!

Rapporten: Hur ska den se ut?

- Kolla hur de bästa gjort i fjol!
  - https://www.eit.lth.se/index.php?ciuid=1271&coursepage=9278&L=1



Internet of Things - Buren intelligens

Calle Sandberg
Lunds Tekniska Högskola
Email: ine14csa@student.lu.se

Sofia Frändberg
Lunds Tekniska Högskola
Email: so6648fr-s@student.lu.se

Sammanfattning **– Detta whitepaper ämnas beskriva tekniken bakom fenomenet Internet of Things, där vi riktar in oss på bärbara smarta objekt, samt hur de kan komma att revolutionera vårt samhälle. IoT, som det även kallas, är en naturlig utveckling av världens digitalisering, och då de tekniska komponenterna blir allt mindre och billigare att tillverka desto mindre blir avståndet till att de blir en naturlig del av samhället. Uppkopplingen av de miljarder smarta objekt som förväntas komma inom den närmsta framtiden innefattar många utmaningar, såsom en global omställning från IPv4 till IPv6.**

Objektens intima interaktion med konsumenten kan revolutionera vårt sätt att se på personlig säkerhet, funktioner hos kläder och synen på kommunikation, men avsaknaden av en garanti för att information om kunden och dess integritet inte läcker ut/kan bli hackad, kommer troligen sakta ner teknikens adoptionstakt.

I. INLEDNING

Historien om Internet of Things börjar med utvecklingen av internet, som gått från att bestå av ett par till närmare en miljard noder på bara 50 år. Denna explosionsartade spridning har lett till att teknik och dess komponenter blivit att billigare, mindre och mer tillgängliga för allmänheten, där internet och informationsteknologi idag är en självklar beståndsdel av vår vardag.

Internet of Things, även kallat sakernas internet, är nu nästa steg i Internets evolution där man ämnar att göra avståndet mellan informationsteknologi och fysiska objekt mindre. Idén är att vardagsobjekt blir så kallade "smarta objekt", vilket innebär att de med hjälp små inbäddade elektroniska komponenter kan ansamla data och koppla upp sig till internet. Detta kan användas till att utbyta information mellan objektet och en utomstående part i molnet.

Det revolutionerande med Internet of Things är dock inte de enskilda smarta enheterna, det adresserar mer konkret de tekniska möjligheterna och samhällsfrågorna som uppstår vid förekomsten av de miljoner eller till och med miljarder smarta objekt som förväntas finnas i samhället om några år. Exempel på möjliga funktioner objekten kommer kunna ha är självstyre, samling och organisering av data, kontextmedvetenhet, målorientering samt möjligheten att kommunicera mellan sig. Detta medför stora möjligheter för samhället inom exempelvis samhällsplanering, men kan även innebära risker för den enskilda personen då maktbalansen kommer luta mot den som har tillgång till och kan kontrollera den insamlade informationen [6].

II. TEKNISK ÖVERSIKT

A. Identifiering av smarta object

Hela konceptet bakom Internet of Things (IoT) bygger på många miljarder uppkopplade enheter som kan kommunicera med varandra. För att en sådan kommunikation ska kunna etableras behöver alla dessa kunna identifieras i nätverket, vilket kräver lika många unika identiteter som det finns enheter. Dagens användning av IPv4, som innehåller strax över 4 miljarder IP-adresser, räcker inte längre till - i synnerhet inte eftersom att allt fler enheter kommer kopplas upp. En övergång till IPv6, ett Internetprotokoll som istället innehåller 2^128 unika IP-adresser, är därför nödvändig för att IoT ska kunna bli verklighet [3].

B. IoT Architecture

Uppbyggnaden av IoT kan delas in i fyra steg - sensorer och ställdon, gateways, edge computing samt datacentret/molnet (se Bild 1). Varje steg möjliggör mer avancerad analys av data och mer komplexa funktioner, men belastar också en större del av systemet. Vad som utmärker de olika delarna, bakomliggande teknik och ingående processer beskrivs nedan [4].

Bild 1. Beskriver hur IoT är uppbyggt runt teknisk [4].

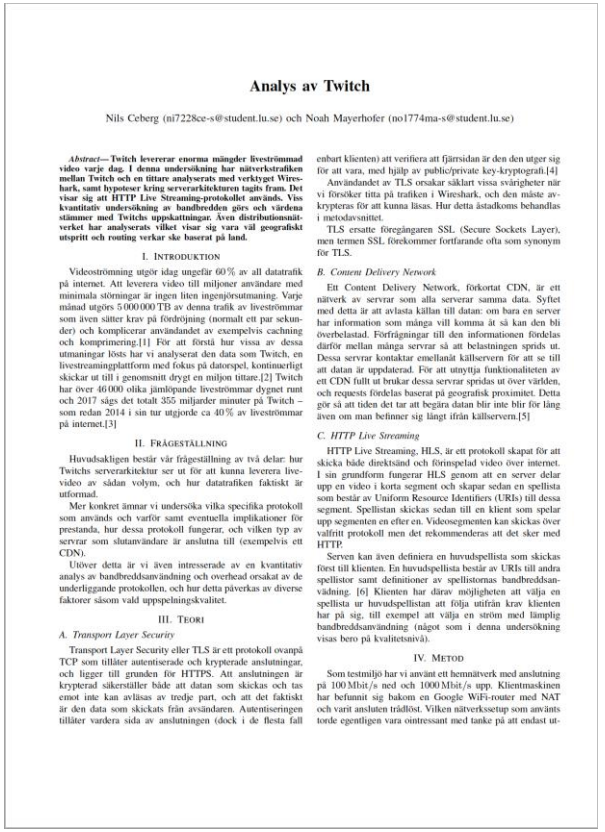
Det första steget, som utgörs av sensorer och ställdon, är i princip där all informationsinsamling sker. Mätbara data som samlas in kan exempelvis vara temperatur, blodtryck eller puls. Insamlingen sker från analoga källor av många olika slag, men detta arbete kommer främst att fokusera på människor, kläder och omgivningen [3].

Endast väldigt lite data bearbetas redan på den lägsta nivån, den mesta skickas vidare till större system som samlar in data från många olika enheter. Här, i det andra steget, sammanställs och omvandlas den analoga datan till digital. Detta sker genom så kallade datainsamlingssystem (DAS), som är kopplade direkt till komponenterna i steg 1. Efter att digitaliseringen har genomförts delas informationen vidare till en gateway, som i sin tur använder sig av exempelvis Wi-Fi, Internet eller trådad LAN för att kunna föra datan till nästa nivå [4].

I det tredje steget, edge computing, bearbetas data vidare innan den slutligen kan skickas till molnet. Här görs mycket av analysen och bearbetningen av den insamlade datan, mycket för att avlasta det centrala datalagret. Dessa IT system kan också sålla bland den ingående informationen för att endast skicka vidare det som är mest relevant till molnet eller centrallagret. Där, i det fjärde och sista steget, genomförs sedan de mest ingående analyserna och övergripande processerna [4].

C. M2M-kommunikation

Ett av de stora genombrotten med IoT är att uppkopplade enheter ska kunna kommunicera med varandra direkt, utan mänsklig inverkan, genom att nyttja machine-to-machine (M2M) communication. För detta behövs ett protokoll som styr hur de kommunicerar på ett sätt som gör att avståndet som data



Analys av Twitch

Nils Ceberg (ni7228ce-s@student.lu.se) och Noah Mayerhofer (no1774ma-s@student.lu.se)

***Abstract*— Twitch levererar enorma mängder liveströmmad video varje dag. I denna undersökning har nätverkstrafiken mellan Twitch och en tittare analyserats med verktyget Wireshark, samt hypoteser kring serverarkitekturen tagits fram. Det visar sig att HTTP Live Streaming-protokollet används. Viss kvantitativ undersökning av bandbredden görs och värdena stämmer med Twitchs uppskattningar. Även distributionsnätverket har analyserats vilket visar sig vara väl geografiskt utspritt och routing verkar ske baserat på land.**

I. INTRODUKTION

Videoströmning utgör idag ungefär 60 % av all datatrafik på internet. Att leverera video till miljoner användare med minimala störningar är ingen liten ingenjörsutmaning. Varje månad utgörs 5 000 000 TB av denna trafik av livesträmmar som även sätter krav på fördröjning (normalt ett par sekunder) och komplicerar användandet av exempelvis cachning och komprimering.[1] För att förstå hur vissa av dessa utmaningar löstes har vi analyserat den data som Twitch, en livestreamingplattform med fokus på datorspel, kontinuerligt skickar ut till i genomsnitt drygt en miljon tittare.[2] Twitch har över 46 000 olika jämlöpande livesträmmar dygnet runt och 2017 sägs det totalt 355 miljarder minuter på Twitch – som redan 2014 i sin tur utgjorde ca 40 % av livesträmmar på internet.[3]

II. FRÅGESTÄLLNING

Huvudsakligen består vår frågeställning av två delar: hur Twitchs serverarkitektur ser ut för att kunna leverera live-video av sådan volym, och hur datatrafiken faktiskt är utformad.

Mer konkret ämnar vi undersöka vilka specifika protokoll som används och varför samt eventuella implikationer för prestanda, hur dessa protokoll fungerar, och vilken typ av servrar som slutanvändare är anslutna till (exempelvis ett CDN).

Utöver detta är vi även intresserade av en kvantitativ analys av bandbreddsanvändning och overhead orsakat av de underliggande protokollen, och hur detta påverkas av diverse faktorer såsom vald uppspelningskvalitet.

III. TEORI

A. Transport Layer Security

Transport Layer Security eller TLS är ett protokoll ovanpå TCP som tillåter autentiserade och krypterade anslutningar, och ligger till grunden för HTTPS. Att anslutningen är krypterad säkerställer både att datan som skickas och tas emot inte kan avläsas av tredje part, och att det faktiskt är den data som skickats från avsändaren. Autentiseringen tillåter vardera sida av anslutningen (dock i de flesta fall enbart klienten) att verifiera att fjärrsidan är den den utger sig för att vara, med hjälp av public/private key-kryptografi.[4]

Användandet av TLS orsakar såklart vissa svårigheter när vi försöker titta på trafiken i Wireshark, och den måste avkrypteras för att kunna läsas. Hur detta åstadkoms behandlas i metodavsnittet.

TLS ersatte föregångaren SSL (Secure Sockets Layer), men termen SSL förekommer fortfarande ofta som synonym för TLS.

B. Content Delivery Network

Ett Content Delivery Network, förkortat CDN, är ett nätverk av servrar som alla serverar samma data. Syftet med detta är att avlasta källan till datan: om bara en server har information som många vill komma åt så kan den bli överbelastad. Förfrågningar till den informationen fördelas därför mellan många servrar så att belastningen sprids ut. Dessa servrar kontaktar emellanåt källservern för att se till att datan är uppdaterad. För att utnyttja funktionaliteten av ett CDN fullt ut brukar dessa servrar spridas ut över världen, och requests fördelas baserat på geografisk proximitet. Detta gör så att tiden det tar att begära datan blir inte blir för lång även om man befinner sig långt ifrån källservern.[5]

C. HTTP Live Streaming

HTTP Live Streaming, HLS, är ett protokoll skapat för att skicka både direktsänd och förinspelad video över internet. I sin grundform fungerar HLS genom att en server delar upp en video i korta segment och skapar sedan en spellista som består av Uniform Resource Identifiers (URIs) till dessa segment. Spellistan skickas sedan till en klient som spelar upp segmenten en efter en. Videosegmenten kan skickas över valfritt protokoll men det rekommenderas att det sker med HTTP.

Serven kan även definiera en huvudspellista som skickas först till klienten. En huvudspellista består av URIs till andra spellistor samt definitioner av spellistornas bandbreddsanvädning. [6] Klienten har därav möjligheten att välja en spellista ur huvudspellistan att följa utifrån krav klienten har på sig, till exempel att välja en ström med lämplig bandbreddsanvändning (något som i denna undersökning visas bero på kvalitetsnivå).

IV. METOD

Som testmiljö har vi använt ett hemnätverk med anslutning på 100 Mbit/s ned och 1000 Mbit/s upp. Klientmaskinen har befunnit sig bakom en Google WiFi-router med NAT och varit ansluten trådlöst. Vilken nätverkssetup som använts torde egentligen vara ointressant med tanke på att endast ut-

Rapporten: Några tips för skrivandet

- Använd inte talspråk i en teknisk rapport.
- **Varje avsnitt ska inledas med en eller flera meningar som talar om vad avsnittet handlar om.**
- Du bestämmer själv om du vill skriva i passiv eller aktiv form. Det är oftast enklare att skriva i aktiv form. Ett exempel är ”Vi undersökte systemet genom att…”, istället för ”Systemet undersöktes genom att…”.
- Använd korrekta värden istället för förstärkningsord, tex ”ca 10ms” istället för ”kort tid”.
- Alla figurer och tabeller ska numreras och tydligt refereras till och beskrivas i texten.

Rapporten: Vad ska man helst undvika?

- **Fel template!**
- För kort inledning eller bakgrund
- Sektioner som är för korta
- Svag koppling mellan sektionerna (röd tråd)
- Frågeställning som angivs endast i ’bullet points’
- Figurer som inte nämns i text
- För få och icke-vetenskapliga referenser (Wikipedia är ok).

Wireshark: Förslag på ämnen

- **Strömningstänster för musik**: Spotify, iTune
- **Strömningstjänster för video**: Netflix, HBO, Viaplay, Youtube, Twitch.tv, Vimeo, Popcorn Time.
- **Meddelandetjänster**: Messenger, Skype, Kik
- **VoIP**: Skype, Viber, Discord
- **Online-spel**: WoW, CS, Dota 2, Minecraft, LoL
- **Realtidsapplikationer**: Pages, SubEthaEdit
- **Molnlagring**: Dropbox, Box, OneDrive, Google Drive
- **Anti-virus**: Avast, Symantec, Norton
- **Sociala medier**: Facebook, Twitter, Instagram, Snapchat, Tinder
- **P2P Fildelning**: BitTorrent, eDonkey, Gnutella, Freenet

Wireshark: vad måste man undersöka?

- Kommunikationens aktörer
- Protokoll
- Paketinnehåll
- …

IP	Protokoll	Land	Företag
216.98.54.23	TCP	USA	Ubisoft
216.98.62.46	TCP	USA	Ubisoft
216.98.62.72	TCP	USA	Ubisoft
216.98.55.85	UDP	USA	Ubisoft
40.68.201.140	UDP	Holland	Microsoft Azure
40.113.149.253	UDP	Holland	Microsoft Azure
74.201.107.45	RTP	USA	Vivox
74.201.107.23	RTP	USA	Vivox

TABLE I

IP-ADRESSER

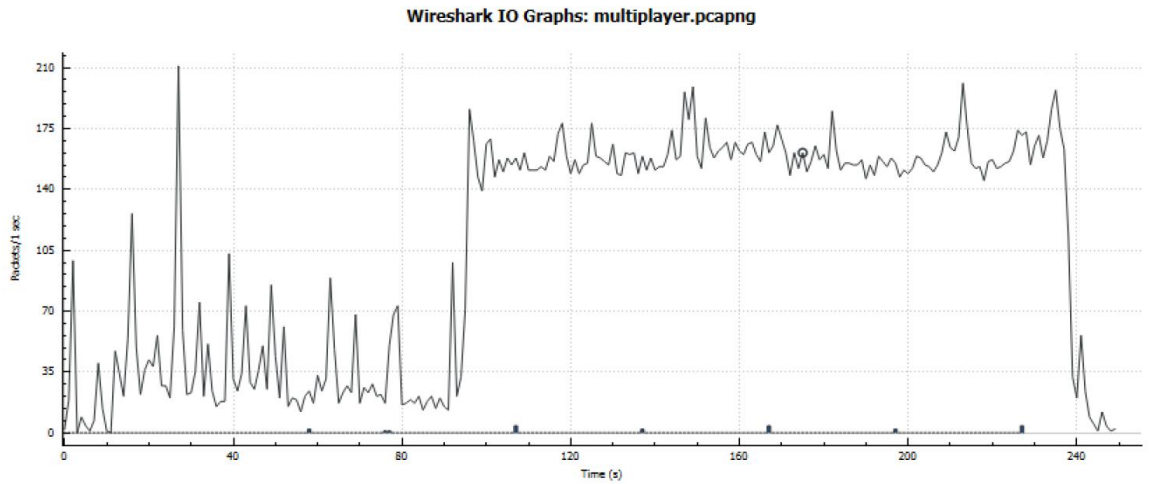
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.105	208.117.253.88	TCP	66	53474 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.016792	208.117.253.88	192.168.0.105	TCP	66	80 → 53474 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_P…
3	0.016893	192.168.0.105	208.117.253.88	TCP	54	53474 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0

Wireshark: vad måste man undersöka?

- Prestanda
- Bandbredd
- Trafikprofil
- …

TABELL III: Bandbreddsanvändning för olika videokvaliteter i kbit/s

Video kvalitet	Min	Genomsnittlig	Max
1080p 60FPS	0,5	6385	13750
1080p	1995	2798	3423
720p 60FPS	4	3458	7144
720p	5	2428	5461
480p	0,6	1566	3371
360p	0	773	1762
160p	10	356	736

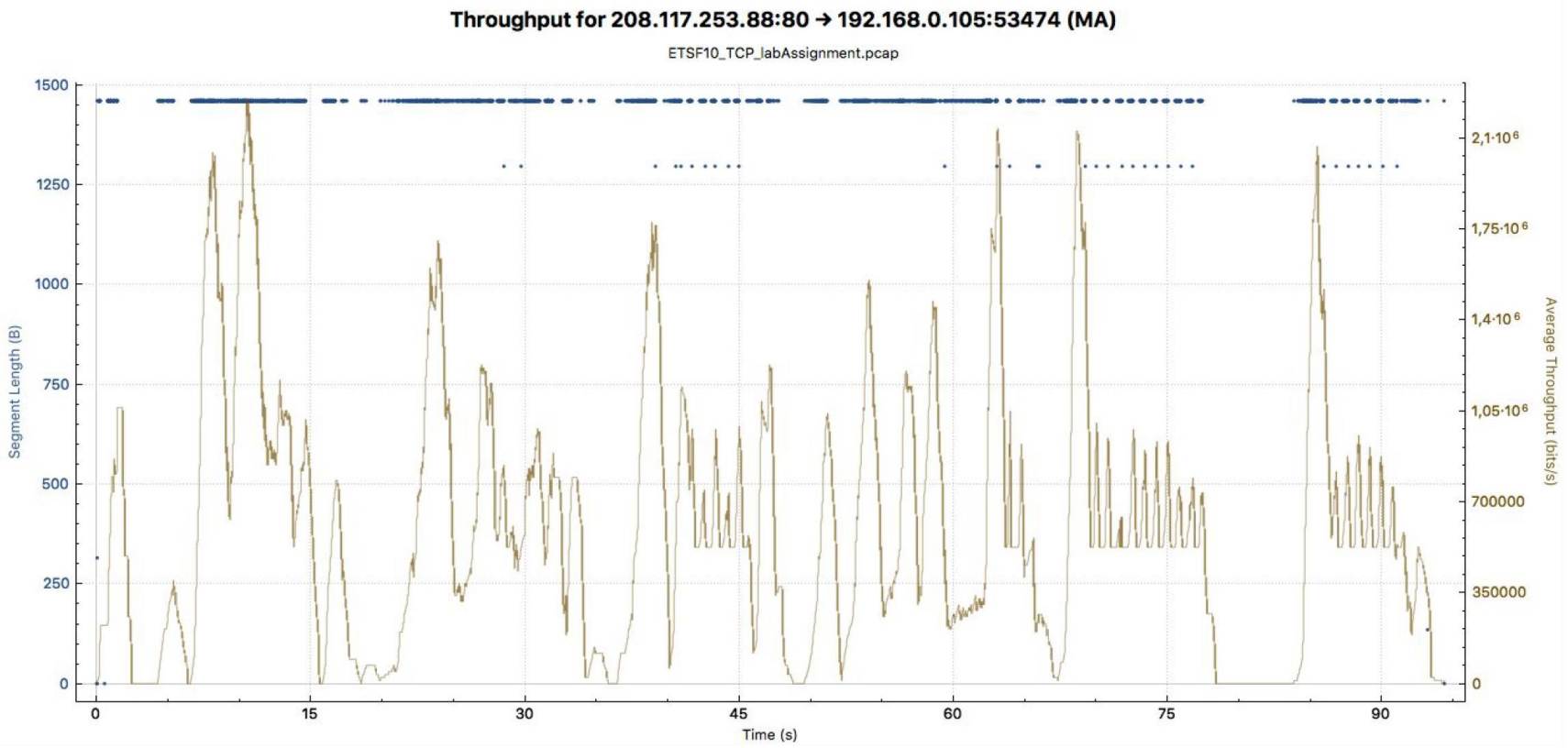


Wireshark: vad måste man undersöka?

- Nätverk-setup
- Applikationens beteende, funktionalitet
- Molntjänster? CDN?
- Säkerhetsprotokoll?

Wireshark: några exempel

- Genomströmning
- …



Whitepaper: Förslag på ämnen

- VANET, Självkörande fordon (autonomous vehicles)
- LTE, 5G & Massive MIMO
- Passive Optical Network (PON)
- Cloud Computing, Fog computing, Edge clouds
- Near Field Communication (NFC)
- Green Networking
- Software Defined Network (SDN)
- Internet of Things (IoT)
- Webbsäkerhet
- Smart cities
- E-health
- Cloud robotics
- Resursoptimering av datacenter
- Digitala valuator
- Nätsäkerhet (måste specifieras)
- Industri 4.0

Tidsschema

- Läsvecka 2: Val av projektpartner och ämne
  - Projektpartner inom **samma redovisningsgrupp**
  - Val av ämne genom CANVAS (3 ämnen ska väljas)
- Beslut av ämnen
  - Inom varje redovisningsgrupp ska alla ämnen vara unika
- Läsvecka 3: Inlämning av projektbeskrivning
  - Inlämning genom CANVAS (ca. 250 ord)

Tidsschema forts.

- Läsvecka 5-6: Handledermöte
  - Mer info kommer senare
- Läsvecka 6: Inlämning av rapport (v1) för kamratgranskning
  - Mer info kommer senare
- Läsvecka 7: Kamratgranskning
  - Mer info kommer senare
- Läsvecka 7: Presentation (19-20/12)
  - Mer info kommer senare
- 2020 KV1: Inlämning av slutlig rapport (6/1)

CANVAS

https://canvas.education.lu.se/courses/740

- Nytt system för inlämning av uppgifter
- Alla uppgifter är registrerade med slutdatum och krav
- Alla ladokstudenter har tillgång till kursen på **CANVAS**
- Alla uppgifter ska lämnas in individuellt
  - Dvs förutom val av partner och ämne.
  - Rapportens slutversion även via **URKUND** till kaan.bur.lu@analys.urkund.se av ***endast*** en partner i varje grupp.

Redovisningsgrupp 1:
19/12 torsdag, 08.15 - 10.00

Redovisningsgrupp 2:
19/12 torsdag, 08.15 - 10.00

Redovisningsgrupp 3:
19/12 torsdag, 15.15 - 17.00

Redovisningsgrupp 4:
19/12 torsdag, 15.15 - 17.00

Redovisningsgrupp 5:
20/12 torsdag, 10.15 - 12.00

Redovisningsgrupp 6:
20/12 torsdag, 10.15 - 12.00

Redovisningsgrupp 7:
20/12 torsdag, 13.15 - 15.00

Redovisningsgrupp 8:
20/12 torsdag, 13.15 - 15.00