PreViS Samhandlingssystem

Av Sandra Smaaberg, Simen Bergo og Kristian Teppan

Sammendrag

Tittel: PreViS Samhandlingssystem

Dato: 06.12.2021

Deltakere: Simen Bergo, Sandra Smaaberg, Kristian Teppan.

Veileder: Emil Bakke.

Oppdragsgiver: Peder Andreas Stokke, Innovasjonsrådgiver fra Forsknings- og innovasjonsavdelingen ved Sykehuset Innlandet HF.

Stikkord: Prehospitalt, videokommunikasjon, helsetjenesten, innovativt.

Antall sider: 25

Antall vedlegg: 9

Dette prosjektet er en del av et omfattende samarbeid for å effektivisere prehospitale tjenester, mellom Forsknings- og innovasjonsavdelingen ved Sykehuset Innlandet HF og PICTA. Målet med prosjektet er å sørge for høy kvalitet på video overføringen mellom ambulanse og sykehus, som kan effektivisere bedømmingen av pasienters tilstand. Vårt arbeid er en teoretisk innsikt i hvilke teknologier som best mulig tilrettelegger for dette, og må anses som en del én av to, siden arbeidet fortsetter i bacheloroppgaven. Gjennom dybdeintervjuer, prototyping og innsiktsarbeid skal vi danne et grunnlag for å kunne utvikle et videooverføringssystem senere. Sammenligning av video komprimerings kodeker gjør at vi kan utvikle et system som gir klare bilder selv i områder med dårlig dekning, som er ett av problemene prosjektet møter. Resultatet viser hvor nyansert disse teknologiene er, og at det er mange faktorer som spiller inn. Mot slutten legges det vekt på fremtidig arbeid, drøfting og hvordan testingen og prototypingen som er utført kan forbedres.

Introduksjon

Helsetjenesten i dag er sterkt påvirket av den teknologiske utviklingen vårt samfunn går gjennom, og gir stadig nye muligheter til å forbedre samfunnskritiske systemer. Et eksempel på hvordan denne digitaliseringen kan være med på å redde liv er samhandlingssystemet vi har som oppgave, PreViS.

PreViS er et samarbeidsprosjekt mellom PICTA og Sykehuset Innlandet. PreViS er en grenseskridende satsing innen innovative miljøer i indre Skandinavia. Prosjektet skal bidra til forskning, teknisk utvikling, og innovasjon innen bruk av videoteknologi som bedømmingsstøtte i prehospitale tjenester, og slik sørge for at flest mulig pasienter får trygg og god behandling der de er. Fokuset er på optimalisering av videostøtte generelt, men også spesielt for pasienter innen traume og psykisk helsevern (Internt dokument fra oppdragsgiver, 2021).

Hensikten med denne oppgaven er å hjelpe Sykehuset Innlandet utvikle et godt samhandlingssystem for å raskere avgjøre behovene til pasienter, optimalisere bruken av ressurser og gjøre tilværelsen til pasienten best mulig. Mer spesifikt er målet vårt å gjøre overføringen av data mellom ambulanse og sykehus effektiv, slik at kommunikasjonen med spesialist optimaliseres under kritiske situasjoner. Dette vil si at man får god legehjelp på et tidligere stadie, som kan effektivisere oppdrag for paramedic og frigjøre flere ressurser.

Denne rapporten og funnene vi legger frem er en forløper til bachelorprosjektet, som skal ha sin oppstart etter nyttår. Vårt arbeid i dette prosjektet dreier seg derfor mest om innsiktsarbeid, teoretisk analyse og forberedelse, før vi går over til utvikling i bachelorprosjektet. Research og intervjuer er også en del av denne oppgaven, og vil legge grunnlaget for vårt valg av teknologi når vi har sammenlignet og vurdert de forskjellige alternativene. Vi har også laget en mindre prototype for å skaffe oss en bedre forståelse for hvordan de involverte teknologiene fungerer.

Problembeskrivelse

Sykehuset Innlandet jobber for å utrede og utvikle bruk av videokommunikasjon som beslutningsstøtte for helsepersonell i akuttkjeden. Ved å koble opp spesialister eller legevaktslege over video, vil ambulansepersonell ha bedre forutsetning for diagnostisering og

triagering, som igjen vil sikre at pasienten får raskere behandling (Internt dokument fra oppdragsgiver, 2021).

Samhandlingsutfordringene skaper både effektivitetsproblemer (koordinering, informasjonsflyt, rett tjenestenivå, etc.) og kvalitetsproblemer (opplevd fragmentering, informasjonstap, tilgang på rett kompetanse, etc.). Innbyggerne opplever frustrerende møter med for fragmenterte tjenester, og pasienter og pårørende må ofte ta ansvar for selv å mobilisere tjenester, og for å innhente og dele relevant informasjon. Samfunnet får høyere kostnader enn nødvendig; manuell koordinering er nødvendig, men også enormt tidkrevende, og evnen til å hjelpe innbyggere raskt og effektivt reduseres. (Christie, Hoholm, Mørk, 2018, avsnitt 3).

Dette sitatet belyser problemet som PreViS-prosjektet forsøker å løse, og gir god innsikt i hvorfor innovasjon er et viktig satseområde innenfor helsetjenester.

Prosjektet er godt i gang, og har en modifisert ambulanse som prototype. Hoveddelen av systemet er bevegende PoE kameraer (Power over Ethernet) som er strategisk plassert bak i ambulansen hvor det jobbes på pasienten. De er plassert i taket for å være minst mulig i veien for de i ambulansen, og gir samtidig gode vinkler ned på pasienten. Det har også vært prøvd ut bruk av såkalte kroppskameraer som er festet på brystet til paramedicen, for å gi et nærmere perspektiv fra deres øyne, både i og utenfor ambulansen. Spesialister eller legevaktsleger får i sanntid overført denne feeden fra alle kameraene til seg, slik at de kan bistå med undersøkelser av pasient, og gi presise vurderinger basert på pasientens behov. Dette vil også gi større trygghet for paramedicer, siden de kan få assistanse dersom dette er nødvendig. Noe som enda ikke er inkludert men svært ønsket er muligheten til å koble opp multimonitoren til feeden. “Multimonitoren overvåker blant annet livsviktige funksjoner som puls, EKG, blodtrykk og oksygenmetning på pasienten” (Oftedahl, 2014, avsnitt 2). Dette vil gi spesialistene og legevaktslegene mer detaljert informasjon om pasientens tilstand, som igjen kan resultere i bedre vurderingsevne.

Problemstilling

Problemstillingen vår er basert på oppdragsgivers behov, og formulert med hjelp av vår veileder:

Hvilken teknologi bør benyttes for best mulig overføring av data?

Vi har valgt å avgrense problemstillingen med fokus på hvilken teknologi som er best, med tanke på bevaring av kvalitet. Dette er fordi detaljer kan være kritiske under syketransport.

Del-problemstillingen vår ble da:

Hvilken teknologi bevarer kvaliteten best når det kommer til videooverføring?

Vi valgte dette som vår problemstilling basert på innsikt fra paramedic og oppdragsgiver, da dette ble identifisert som svært viktig for systemets effektivitet. Dersom spesialisten kan se pasienten klarere og høre vedkommende tydeligere, vil dette resultere i en mer effektiv og presis diagnostisering . Høy oppløsning og stabilt signal er de to viktigste faktorene for at dette skal fungere optimalt (Peder A. Stokke, Innovasjonsrådgiver, 27.10.2021, Microsoft Teams). For å oppnå denne høye oppløsningen selv om dekningen i enkelte områder kan bli dårlig, er det viktig å velge en teknologi som overfører videoen på mest effektiv måte. Dette blir derfor et stort fokus for å kunne svare på problemstillingen.

Teoretisk bakgrunn

Teorien som legger grunnlaget for vår forskning kommer fra flere steder. Interne dokumenter og kravspesifikasjoner vi har fått tilsendt av vår oppdragsgiver, gode kilder på nettet og noe fra bøker. Problemet ved å bruke utdaterte medier for å gjøre forskning på teknologi er at feltet endrer seg enormt raskt, og informasjon blir derfor fort utdatert. Vår forskning inneholder teknologi-løsninger, oppløsning, bildefrekvens, video- og lydkodeker, dekning og hastighet. Alle disse påvirker hvilken teknologi som egner seg best til prosjektet.

Hensyn i forhold til prosjektet

Når vi begynner å sammenligne de forskjellige teknologiene er det viktig å ta hensyn til antall enheter og utstyr involvert. Som beskrevet tidligere er det flere kameraer i ambulansen, og ytterligere er ønsket i fremtiden. For å presist kunne svare på vår problemstilling må vi ta

utgangspunkt i et realistisk scenario. Et høyere antall kameraer og data som skal sendes, vil resultere i en større mengde data, noe vi må ta i betraktning under sammenligningen.

Vi tar utgangspunkt i utstyret som er i ambulansen for øyeblikket, som er 3 PoE kameraer med en maks oppløsning på 3840 x 2160 (4K), og 30 bilder per sekund. I tillegg skal også lyd og dataen fra multimonitoren sendes. Vi har ikke tilgang på denne informasjonen, og vil derfor basere videre vurderinger ut ifra en estimert datamengde. Dette vil si at estimater og utregninger videre i rapporten er basert på dette som grunnlag, men som fortsatt vil gi et godt bilde på resultatet, selv om prosjektet senere skaleres opp med flere kameraer.

Foreløpig er det installert kommunikasjonsutstyr som bruker 5G dersom dette er tilgjengelig, som overføringsteknologi mellom ambulansene og sykehuset. Dette er den raskeste standarden som er tilgjengelig, og har i startfasen en forventet hastighet på 50 Mbps - 2 Gbps (Nkom, 2021). I ambulansen er det installert en IBR1700 Series Ruggedized Router, som har en hastighet på 940 Mbps. Det er også installert et MC400 modular modem med mulighet for to SIM-kort for sikkerhet, som betyr at den kan bytte til et bedre nettverk dersom det ene ikke har tilstrekkelig nok med dekning eller svikter. Dette betyr at ruteren ikke vil begrenses av hastigheten til 5G nettet.

Teknologi

Ved valg av teknologi har vi sett på noen forskjellige muligheter innenfor rammeverk, protokoller og kodeker. Det finnes mange nettsider og apper som tilbyr video- og lyd-deling. De mest populære som Facetime, Facebook Messenger og Zoom benytter seg av WebRTC teknologi (Hancke, 2021a; Levent-Levi, 2015b; Jain, 2021c;). WebRTC er en teknologi som støtter sending av video, lyd og generisk data mellom nettlesere og apper. Teknologien er tilgjengelig på alle moderne nettlesere og er en åpen web standard, som også er tilgjengelig som Javascript-APIer i de største nettleserne. Som åpen kilde er WebRTC støttet av blant annet Apple, Google, Microsoft og Mozilla (WebRTC, u.å.). Teknologien har også bibliotek tilgjengelig for populære applikasjons klienter som Android og iOS. Vi har ikke funnet noe konkurransedyktig alternativ for WebRTC som muliggjør sanntidskommunikasjon gjennom nettlesere og apper.

WebRTC er peer to peer (enhet-til-enhet) kommunikasjon, hvor nettleseren hos bruker én kommuniserer direkte med nettleseren til bruker to. Dette betyr at det ikke er behov for en

server eller API for selve kommunikasjonen, men for å opprette en tilkobling må det brukes en signal-server. Signal-serveren sender og mottar nødvendig informasjon slik at WebRTC-teknologien vet hvilken enhet, eller nettleser, den skal kobles sammen med (MDN, u.å.). For å kunne bruke WebRTC trenger vi altså en form for signal-server. Her finnes det forskjellige typer protokoller man kan benytte seg av. På nettsiden til WebRTC skriver de at teknologiens spesifikasjoner ikke har noen preferanse når det kommer til en signal-serverløsning (WebRTC, 2020). Dette betyr at vi har frihet til å velge uten å tenke på hva teknologien foretrekker.

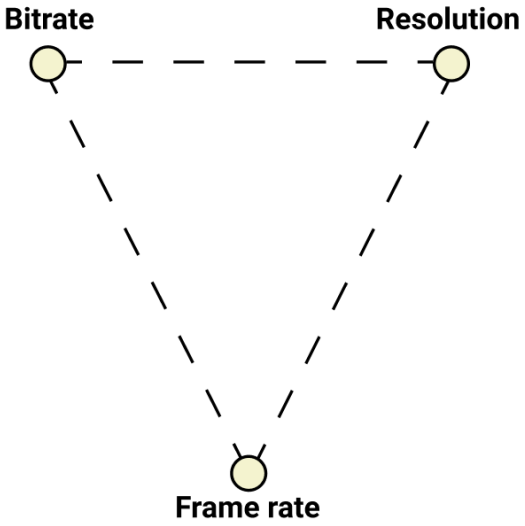
Mozilla Developer Network (MDN) bruker Websocket API på sin chat-server, derfor så vi nærmere på denne løsningen. Ved å undersøke Websockets fant vi en løsning som gjør det enkelt å opprette en tilkobling, kalt Socket.io. Socket.io er et javascript-bibliotek som hovedsaklig bruker websocket protokollen, som er en avansert teknologi som muliggjør en åpen interaktiv kommunikasjons-økt mellom klientens nettleser og en server (MDN, u.å.). Socetk.io har HTTP long-polling som sikkerhet. HTTP long-polling er også en måte å kommunisere med en server, og er nærmest universelt støttet av alle enheter og egner seg dermed godt som en reserveløsning (Kilbride-Singh, 2021).

Socket.io gjør det enkelt å opprette en websocket-forbindelse og muliggjør hendelsesbasert, toveis sanntidskommunikasjon mellom nettleseren og serveren. Det består av en Node.js server og et javascript klient bibliotek. Ved å bruke Socket.io vil vi få mer sikkerhet i forbindelsen, fordi den bruker HTTP long-polling dersom websocket-forbindelsen feiler (Socket.IO, 2021).

Vi fant også et alternativ til WebSockets, kalt WebTransport som også tilbyr toveis-kommunikasjon mellom klient og server med lav ventetid. WebTransport benytter seg av den nyeste HTTP-protokollen, HTTP/3. WebTransport er en del av et nytt spesifikasjons-utkast, og er dermed ikke like utbedret som WebSockets. WebSockets har bedre klient- og server-biblioteker enn WebTransport. Dette betyr en mye bredere støtte når det kommer til server-teknologier og nettlesere, og er for øyeblikket mer pålitelig (Posnick, 2021).

Bitrate

Illustrert i figur 1 er det tre variabler man må balansere for å oppnå god videokommunikasjon (Levent-Levi, 2021). Bitrate er mengden data (bit) som kan overføres samtidig, ofte oppgitt i kilobit per sekund (Kbps) eller megabit per sekund (Mbps). En kilobit er ett tusen bit, og en megabit er ett tusen kilobit (Rossen, 2021). Selv om en video har en gitt oppløsning og bildefrekvens, kan man justere bitraten opp eller ned. Man justerer ned for å spare filstørrelse men med større tap av data, og opp for å få mer detaljer i hvert bilde. Dette er relevant fordi mengden bitrate som videoen bruker vil bli påvirket av kvalitet, og teknologiens evne til å komprimere dataene.



Figur 1: Balansere god videokommunikasjon

Oppløsning

Oppløsning er antall piksler som vises på en dataskjerm eller i et digitalt bilde. Flere piksler betyr høyere oppløsning, og mindre kornete bilde. Som tidligere nevnt var dette et viktig mål i prosjektet, å oppnå en høy oppløsning slik at det blir tydeligere for leger eller spesialister å kunne tyde små detaljer på pasienter, eller zoome inn for å se nærmere på et spesifikt sted. Når ambulansen er i områder med høy nettverkshastighet (høy båndbredde) vil forhåpentligvis den fulle oppløsningen til kameraene kunne utnyttes (4K), som resulterer i veldig klare bilder. Avhengig av tilgjengelig båndbredde vil oppløsningen måtte kunne skrus opp og ned automatisk, slik at ikke bitraten overskrider båndbredden. 4K i oppløsning betyr at det horisontalt er rundt 4000 piksler, hvor det nesten er umulig å skille ut individuelle

piksler med sitt eget øye, selv på nær avstand. Dette er den moderne standarden på tv og strømmetjenester, men på grunn av den høye oppløsningen kreves det god videokompresjon for å overføre over nett eller mobilnett med vanlig hastighet (Myren, 2020).

HDR-video (High Dynamic Range) er en teknologi som forbedrer måten informasjon om kontrast og farge i en video presenteres. Det gjør at de hvite fargene fremstår lysere og mer detaljert, samtidig som de mørke fargene blir dypere og mer detaljert. Det muliggjør også et bredere spekter av farger (Amazon Web Services, u.å.). Dette kan være viktig dersom bildene i videoen for eksempel inneholder mye blod, som kan gjøre det mørkt og vanskelig å se viktige detaljer.

Bildefrekvens

Oppdragsgiver har poengtert at høy oppløsning er viktigere enn høy bildefrekvens. Dette er fordi det er mer kritisk å se tydelige og detaljerte bilder, i motsetning til lavere oppløsning og kvalitet med høy bildefrekvens, siden pasienten ligger som regel i ro på båre i ambulansen. Dette betyr at avhengig av kodeks, kan bildefrekvensen settes til å variere basert på den tilgjengelige båndbredden. For eksempel er den mest vanlige bildefrekvensen i dag 24 bilder per sekund, ofte omtalt som *fps* (frames per second). Dette er det som er standarden for filmer, og det mennesker i dag er vandt til å se på skjermer (MDN, 2021). Dette er viktig å tenke på ettersom antall bilder per sekund i stor grad påvirker datamengden som sendes. Oppdragsgiver har poengtert at høy oppløsning er viktigere enn høy bildefrekvens, så denne kan justeres ned for å spare data.

Kodek

En kodek er en algoritme som kan kode og dekode data. Kodeken komprimerer den gitte dataen ved hjelp av kode før den sendes til en mottaker, hvor den pakkes ut for å spare lagringsplass og nettverksbelastning (Nätt og Rossen, 2021). De fleste kodeker i dag er med tap, som betyr at noe informasjon går tapt under komprimering og kan føre til eksempelvis dårligere kvalitet på en video. Motparten er tapsfrie kodeker som komprimerer og pakker ut den fullstendige dataen uten tap. Ideelt sett vil vi ha tapsfrie kodeker for å bevare kvaliteten. Problemet med tapsfrie kodeker er at de ikke klarer å komprimere dataene like bra som kodekene med tap. Dette kan bli et problem dersom ambulansen befinner seg i områder med nedsatt nettverkshastighet, og dataene som skal sendes blir for store grunnet dårlig

komprimering. Derfor er ikke tapsfrie kodeker et godt alternativ i vårt prosjekt, og vi vil se bort fra disse.

Lydkodek

God lyd er viktig under kommunikasjonen mellom paramedic og spesialister, slik at det ikke oppstår tap av informasjon. I prototypen bruker vi WebRTC sin standard lydkodek, Opus. Denne er allerede godt tilpasset strømming og tilfredsstiller prosjektets behov når det gjelder lyd (*Opus Interactive Audio Codec,* u.å.). Etter tester med prototypen vår fant vi ut at lyd ikke krever mye bitrate. Under testing av prototypen var bitraten for lyd 32 Kbps, som er veldig lite selv om den totale båndbredden er nede i for eksempel 10 Mbps. Skjermbilde av testing med målinger ligger vedlagt i mappen.

Valg av videokodek

Noen av de mest brukte videokodekene i 2021 er H.264, H.265, VP8, VP9 og AV1 (MDN, 2021). Vi skal sammenligne disse for å finne hvilken som passer best til våre behov. Ved valg av videokodek vil vi først og fremst ha fokus på dens evne til å komprimere video av høy kvalitet, som 4K og HDR video. Vi vil også se på hvilke kodeker som har WebRTC-støtte og er kompatible med forskjellige nettlesere. I dette tilfellet ser vi på de mest brukte nettleserne, Chrome, Edge, Firefox, Internet Explorer, Opera og Safari. Det er også viktig at vi passer på at kameraene i ambulansen støtter kodeken.

H.264/AVC (Advanced Video Coding) er en av de påkrevde kodekene for WebRTC, og har derfor god kompatibilitet med både WebRTC og alle nettleserne. Den har mulighet for komprimering av både 4K video og HDR (MDN, 2021).

H.265/HEVC (High Efficiency Video Coding) er etterkommeren til H.264, og har bedre kompresjons hastigheter enn forgjengeren. Teoretisk sett, klarer H.265 å komprimere filer til halve størrelsen av det H.264 ville klart med samme kvalitet. Den krever altså mindre båndbredde og lagringsplass enn forgjengeren. H.265 muliggjør også komprimering av 4K og HDR video. H.265 er dog relativt nytt og er derfor ikke kompatibel med WebRTC, og det er kun Safari som har full kompatibilitet (MDN, 2021).

Kameraene i ambulansen støtter både H.264 og H.265 (Hikvision, u.å.). Videre fant vi ikke informasjon som tilsier at kameraene støtter noen annen kodek, verken fra oppdragsgiver eller kameraenes spesifikasjoner fra produsentens nettside.

VP8 er den andre påkrevde kodeken for WebRTC, og har også god kompatibilitet med både WebRTC og nettleserne. Den støtter komprimering av 4K video, men ikke av HDR video. Når det kommer til bevaring av kvalitet og komprimerings-hastighet, presterer VP8 på lik linje med H.264 (MDN, 2021).

VP9 er etterkommeren av VP8. VP9 komprimerer ca. like raskt som H.264, men bevarer kvaliteten bedre. VP9s komprimerte videokvalitet er på lik linje med det H.265 oppnår, gitt samme bitrates. VP9 kan komprimere video av høy kvalitet, både 4K og HDR. Den har kompatibilitet med WebRTC og nesten alle nettleserne, med unntak av Internet Explorer (MDN, 2021).

AV1 er, i likhet med H.265, en relativt nyutviklet kodek, men oppnår høyere komprimerings-hastigheter enn både H.265 og VP9. Den støtter komprimering av høykvalitets-videoer, både 4K og HDR. Som følge av at AV1 er såpass ny, er den i prosessen med å bli integrert i nettleserne. På dette tidspunkt er den WebRTC-kompatibel og støttes av Chrome, Edge, Firefox og Opera. AV1s største ulempe er at den er ny, og komprimeringen foregår kun i programvaren, fremfor i en maskinvare. Dette fører til at det tar lang tid å komprimere en video til AV1 (MDN, 2021).

Selv om H.265 ikke er kompatibel med WebRTC, er det mulig å implementere det på egenhånd, slik som Antmedia har gjort. Antmedia er et teknologiselskap som selv har implementert H.265 kodeken slik at den er kompatibel med WebRTC, og selger dette som en ferdig strømmeløsning (Antmedia, u.å.). Denne koster penger og er derfor ikke relevant for dette prosjektet, men det viser at det er mulig å implementere.

I vår forskning fant vi også et annet alternativ, H.266. Dette er en nyutviklet etterkommer av H.265, som er i siste stadium av testing. Denne kodeken lover å være enda mer effektiv enn H.265, og kunne dermed vært et bedre alternativ (Bross *et al,* 2021). På bakgrunn av at H.266 er såpass ny, er den ikke støttet av WebRTC.

Dekning og hastighet

Basert på dekningskartet til Telenor er det utbygd solid 4G dekning i bebodde og nærliggende områder i hele Norge, bortsett fra på vidder og i mark hvor det sjeldent er store mengder folk. Etter å ha undersøkt både Telenor og Telia sine dekningskart i Innlandet fylket, er det minimum 4G dekning i alle store bygder og byer, og i stor grad langs riks og fylkesveier. I mange av de mest bebodde områdene i fylket er det også 4G+ dekning, som gir enda bedre hastighet og båndbredde. Det er viktig for prosjektet å skaffe et godt estimat på hvor lav båndbredden kan være i avsidesliggende områder i fylket, for å alltid kunne utnytte og tilpasse bitraten slik at den ikke overskrider båndbredden, da dette vil resultere i tap av informasjon. Mengden båndbredde man har tilgjengelig vil avgjøre hvor høy oppløsning og bildefrekvens man kan sende frem og tilbake. Det er derfor kritisk at ambulansene er utstyrt med sendere og mottakere av høy kvalitet, for å kunne utnytte mest mulig av den tilgjengelige båndbredden.

> Dekningen som er presentert er basert på teoretiske beregninger, og dekningen og hastigheten som oppleves kan variere fra det som indikeres på kartet. Dekningen kan variere blant annet som følge av antallet personer som ringer eller surfer samtidig i det aktuelle området (Telenor, 2021).

Problemet er at det er veldig vanskelig å regne på gjennomsnittet i et helt fylke, ettersom dette varierer enormt fra bebodde byer til små bygder og ute på landet. Basert på Nettfart sin nasjonale statistikk er opplastningshastigheten til Telenor 13,5 Mbps på 4G nettet, og 44 Mbps på 5G nettet. Dette er basert på målinger som er gjennomført de siste seks månedene (Nkom, 2021).

En gjennomsnittlig 1080p video med 30 bilder per sekund med SDR-kvalitet (Standard Dynamic Range) bruker ca 8 Mbps med data (Google, u.å.). Ved å bruke dette som et utgangspunkt kan vi gange opp med antall kameraer i ambulansen som tidligere nevnt i hensyn i forhold til prosjektet. De tre kameraene ender da totalt på 24 Mbps, og vi må i tillegg beregne mengden data som lydsporet og multimonitoren bruker. Maks bitraten til lydkodeken vi har i prototypen (Opus) er 510 Kbps, og dette vil bli brukt som utgangspunkt i denne oppgaven. Det var svært vanskelig å finne informasjon på hvor store datamengder som sendes fra en multimonitor, siden disse vanligvis ikke er koblet til andre enheter. Siden vi

ikke finner et svar på dette, velger vi å anta at det er de verdiene som beskrives i problembeskrivelse som skal sendes. Dette er oppgitt i tallformat og sendes ca en gang hvert sekund, som resulterer i en ubetydelig mengde sammenlignet med video-dataen. Totalt ender vi opp med en estimert datamengde på ca 25 Mbps.

Metode

Dybdeintervju

"Dybdeintervju er et dyptgående intervju med en person om gangen omkring en forhåndsdefinert problemstilling". "Dybdeintervjuer egner seg særlig på tematikk der man allerede vet ganske mye, men hvor man har behov for å trenge ned i materien på en del tema man ikke helt har forstått sammenhengene til" (Responsanalyse, u.å.).

Gjennom semesteret har vi hatt en god dialog med oppdragsgiver, Peder Andreas Stokke. Det meste vi har fått av informasjon, angående PreViS og hvordan ambulansen er satt opp, er gjennom personlig kommunikasjon. Vi har derfor valgt å sette opp dette som dybdeintervju, etter forslag fra veileder, Emil Bakke.

Kildekritikk

I løpet av forskningen på denne oppgaven har mengden sikre kilder vært et problem. Siden teknologiene er relativt nye, er det mye spredt informasjon og i hovedsak teknologiblogger og journalistiske artikler som kommer opp som resultater. På et senere tidspunkt i arbeidet med oppgaven skjønte vi at det var nødvendig å finne sikrere kilder for å kunne stole på informasjonen vi trengte. Vi fokuserte dermed mer på sikre kilder som offisiell dokumentasjon på de respektive teknologiene. Dette gjorde oss tryggere på at teorien vi bruker stemmer, og resulterte i bedre sammenligningsgrunnlag på de forskjellige kodekene.

Semi-strukturert intervju

Siden vi fastslo problemstillingen vår i starten av dette emnet, ønsket vi å samle inn informasjon for å belyse den. I den hensikt valgte vi å utføre et personlig intervju med en ambulansearbeider som tar del i PreViS prosjektet, og som har fått muligheten til å teste ut den nåværende prototypen. Dette er for å få syn på hva som er nyttig, og bruke vedkommendes synspunkter i den videre prosessen med å gjennomføre både IDG3101 Fordypningsprosjekt og IDG3920 Bacheloroppgave.

Før gjennomførelsen av intervjuet, utarbeidet vi en semistrukturert intervjuguide. Vi delte opp intervjuet i tre deler. Spørsmålene i intervjuguiden ble delt opp etter deltaker, PreViS prototypen og ambulanse. Dette gjorde vi slik at vi kunne skille spørsmålene og informasjonen fra hverandre (Andersen og Schwencke, 2020, s.134). Under gjennomførelsen av intervjuet brukte vi lydopptak som et verktøy. Dette ble klarert med informanten i forkant. Alle i gruppen var med på intervjuet, slik at vi kunne supplere hverandre eller komme med oppfølgingsspørsmål.

Etter intervjuet med ambulansearbeideren, satte vi i gang med transkribering av intervjuet. Ved hjelp av lydopptak som et verktøy var det enklere for oss å dokumentere svarene vi mottok grundigere. Transkribering av intervjuet og intervjuguide ligger vedlagt i mappen.

Risikoanalyse

Den største risikoen knyttet til dette prosjektet er signalstyrke. Et av de mest åpenbare problemene som er utenfor vår kontroll er hvorvidt terreng og miljø gjør at det er mulig å sende og motta data om ambulansen er på et isolert sted eller et område der det er dårlig dekning. Dette kan føre til tap av tilkobling og dermed risikere at hele systemet ikke fungerer. Som skrevet i teoretisk bakgrunn benyttes det topp moderne sendere og mottakere som skal sørge for at dette blir tatt hånd om, men noen steder i fylket er det så dårlig signalstyrke at dette ikke vil hjelpe. Dette er et problem som forhåpentligvis vil reduseres ettersom årene går, men som uten tvil vil påvirke kapasiteten til systemet foreløpig.

Vi har valgt å lage en risikomatrise for å analysere sannsynligheten og konsekvensen ved svekkelse av signalstyrke.

a. *Hva slags usikkerhet vil inntreffe, og når vil det skje?*

   Usikkerheten som vil inntreffe er tap eller svekkelse av signal. Dette kan skje når ambulansen befinner seg på et isolert sted eller et område der det er dårlig dekning.

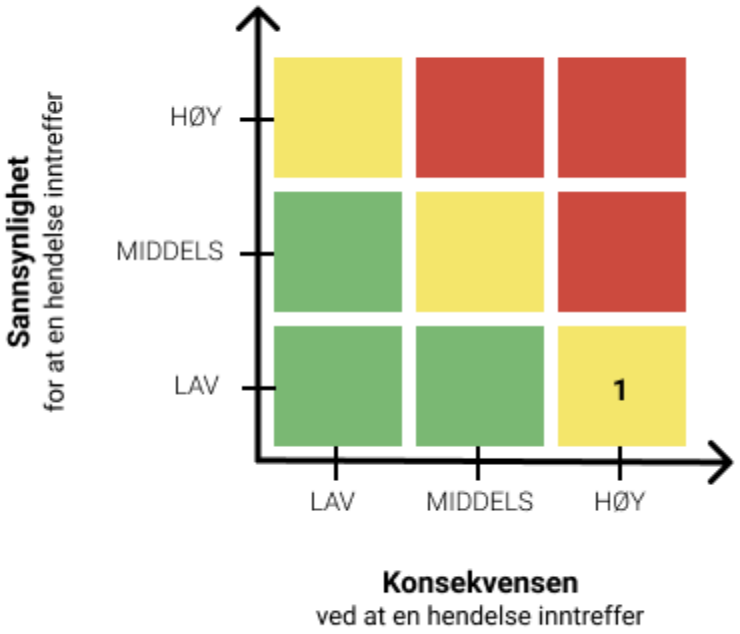
b. *Hvis en usikkerhet inntreffer, hva er effekten av den?*

Dersom usikkerheten inntreffer kan dette føre til store konsekvenser. Dette kan risikere at hele systemet ikke fungerer. Det kan føre til mangel på nødvendig støtte under behandling av pasienten.

c. *Hvordan vil vår respons på usikkerheten påvirke utfallet?*

Tap av signal er utenfor vår kontroll.

Med utgangspunkt i en matrise hvor grønt er lite kritisk, gult er middels kritisk og rødt er svært kritisk, er det ni mulige utfall når a) og b) plottes inn i en matrise hvor vertikal akse viser sannsynlighet for at hendelsen inntreffer, og horisontal akse viser konsekvens hvis hendelsen inntreffer (Vaagaasar og Skyttermoen, 2019, s. 159).



Figur 2: Risikomatrise

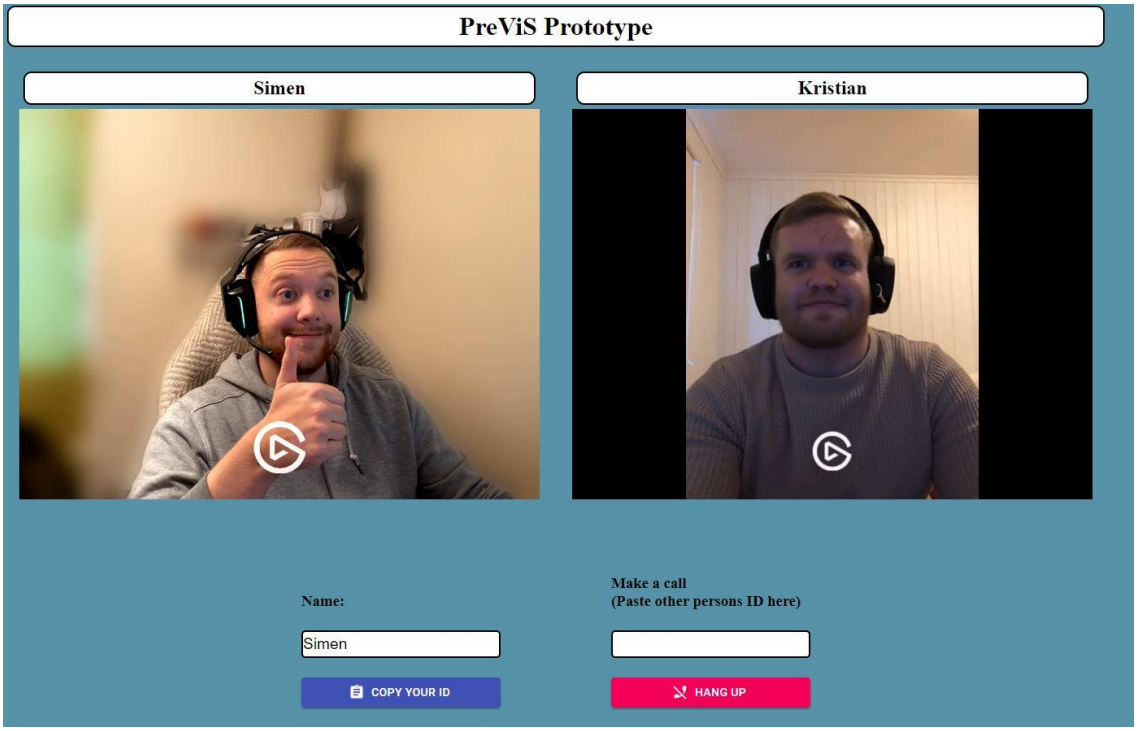
Som analysen viser er dette noe man ikke kan kontrollere, men det er en usikkerhet som er viktig å være klar over. Diagnostisering og behandling vil i nevnt tilfelle skje uten spesialist, og må dermed klare seg uten samhandlingssystemet.

Prototype

For å kunne sette oss dypere inn i hvordan en løsning av videooverføring ville sett ut, bestemte gruppen seg for å forsøke å utvikle vår egen prototype. Dette kan ikke anses som et forsøk på å lage en komplett løsning, men heller bidra til læring og være forberedende til bacheloroppgaven som vi skal gjennomføre til våren. Ved å selv benytte WebRTC får vi muligheten til å få en oversikt over hvordan teknologien fungerer, og tilegne oss en bedre forståelse enn det vi har fått gjennom den teoretiske bakgrunnen. Ambisjonen var derimot å lage prototypen som et testgrunnlag hvor vi kunne teste og sammenligne alle kodekene selv, og få et klart svar på hvilken kodek som passet best til vårt formål. Men etter mye innsiktsarbeid kom vi til et punkt hvor vi skjønte det ville ta for mye tid å implementere den nødvendige koden for å selv kunne endre hvilken kodek programmet bruker, siden dette er noe som egentlig styres internt av kode man ikke kan gå inn å endre, uten å bygge opp plattformen tilpasset til dette helt fra bunnen.

For å kunne vise frem prototypen vår på enklest mulig måte, valgte vi å bruke Netlify for å fremstille nettsiden, fordi vi er godt kjent med tjenesten fra tidligere prosjekter. Lenke til prototypen ligger vedlagt. https://previs-test.netlify.app/

Prototypen vår er bygget opp av en back end og en front end. Back enden oppretter en server, og en tilkobling mellom denne serveren og back enden ved hjelp av node moduler, som blant annet Socket.io. Dette gjør det mulig å koble peers sammen, slik at WebRTC-teknologien kan dele video og lyd mellom dem. På front enden bruker vi React som rammeverk for å bygge nettsiden med Javascript. Her kommer Socket.io sitt klient-bibliotek inn i bildet for å sende peer tilkoblingens data til front enden, slik at nettsiden kan vise video og lyd fra innkommende peer. Figur 3 viser prototypen under en funksjonstest.



Figur 3: Prototype under en funksjonstest.

Resultat

I teoretisk bakgrunn fant vi ut at WebRTC tilsynelatende er den beste teknologien for sanntidskommunikasjon gjennom nettlesere og apper. Vi fant også ut at Socket.io gjør det enkelt å koble sammen to, eller flere nettlesere, slik at WebRTC teknologien kan sende video, lyd og generisk data fritt mellom nettleserne.

Kodekene vi undersøkte i teorien har sine fordeler og ulemper. Vi samlet informasjonen i tabell 1 for bedre oversikt over kodekene, satt opp mot våre behov.

Navn	4K og HDR støtte	Nettleserkompatibilitet	WebRTC kompatibel	Kamerastøtte
H.264	Ja	Alle	Ja	Ja
H.265 (HEVC)	Ja	Safari	Nei	Ja
VP8	Ikke HDR, bare 4K	Alle	Ja	Nei
VP9	Ja	Alle, med unntak av Internet Explorer	Ja	Nei
AV1	Ja	Chrome, Edge, Firefox og Opera	Ja	Nei

Tabell 1: Sammenligning av videokodeker

Basert på teori som ble presentert i teoretisk bakgrunn ser vi at det er mange faktorer som spiller inn ved valg av videokodek. Det blir dermed vanskelig å snevre det inn til en enkelt kodek på dette tidspunktet. Som vist i Tabell 1 har alle kodekene støtte for videoer av høy kvalitet, noe oppdragsgiver presiserte som viktig. H.265 presterer ikke bra når det kommer til kompatibilitet med nettlesere. H.264 og VP8 derimot, har bredest kompatibilitet, med VP9 og AV1 tett bak med noe færre nettlesere. Det er kun H.265 som ikke er WebRTC kompatibel, som er problematisk basert på at det er en av de beste teknologiene. Vi vet at kameraene i ambulansen har støtte for H.264 og H.265, men fant ingen informasjon om støtte for VP8, VP9 eller AV1. De beste kandidatene med tanke på kvalitet og komprimerings-hastighet er AV1, VP9 og H.265.

Siden vi valgte og ha hovedfokus på videooverføringen og ikke lyd, har vi valgt å gå for den standardiserte lydkodeken Opus, som vi diskuterte i teoretisk bakgrunn.

Resultatet av sammenligningen viser at teknologien som bør benyttes for best mulig overføring av data er WebRTC for selve overføringen av data, og Socket.io for sammenkoblingen av enhetene. Vi ser også at det foreløpig er vanskelig å avgjøre hvilken kodek som best bevarer kvaliteten. AV1, H.265 og VP9 er alle gode kodeker som komprimerer hurtig og bevarer kvaliteten godt.

Drøfting

På grunn av oppbyggingen og kompleksiteten til teknologien i WebRTC, hadde vi som nevnt i prototype delen, ikke mulighet til å lage en løsning hvor vi kunne teste alle kodekene mot hverandre. Dette ville gitt oss et konkret svar på hvilken kodek som fungerer best til vår løsning. Dette anser vi som en svakhet i metoden, fordi vi har hatt begrenset med tid og ressurser. Gjennom sikre kilder har vi eliminert noen alternativer, og sett fordelene og ulempene med de som står igjen. For å kunne finne den best egnede kodeken til prosjektet, må vi utføre videre tester i del to, før vi kan komme frem til et endelig svar.

Som nevnt i teoridelen fant vi informasjon om nettleserkompatibilitet rundt kodekene, og vi tror at dette nødvendigvis ikke er like relevant som først tenkt. Løsningen skal brukes mellom ambulanse og sykehus, og vil derfor mest sannsynlig bruke en fast nettleser/enhet.

H.264 er for øyeblikket den eneste kodeken som tilfredsstiller alle kriteriene, og vi tror at denne foreløpig vil fungere best med nåværende kameraer og WebRTC-løsning, fordi det er usikkerhet knyttet til kameraenes støtte til VP9 og AV1. Kameraene støtter også H.265, men den blir utfordrende å implementere, da Safari er eneste nettleser som støtter denne kodeken.

Vi valgte og ikke ta med den nye H.266 standarden, fordi den ikke er støttet av WebRTC på nåværende tidspunkt. Problemet med nyutviklede kodeker som H.266, er at teknologien stadig kommer med nye og bedre alternativer ettersom vi utvikler oss teknologisk. Dilemmaet blir da om man tidlig skal forsøke å implementere den helt nyeste standarden for å kunne få best mulig resultat, på bekostning av stabilitet og pris. Det andre alternativet blir å holde seg til de etablerte og kjente standardene, som ikke vil gi like bra resultater, men være mer stabilt og bedre implementert i større grad enn annen teknologi. Som beskrevet i hensyn i forhold til prosjektet valgte vi å sette et utgangspunkt i den teknologien som allerede er installert i ambulansen, og H.266 er ikke støttet i kameraene som er installert.

Som skrevet i teknologi fant vi et alternativ for WebSockets som heter WebTransport. Nevnt kan være en erstatning for WebSockets i fremtiden, men siden det er rimelig nytt og under stadig utvikling er det fortsatt usikkerheter rundt WebTransport. Vi valgte derfor å ekskludere

WebTransport, men det kan være en mulig løsning i fremtiden, når teknologien blir videreutviklet.

Spørsmålet som vi fortsatt ikke helt har funnet svaret på er hvorvidt H.265 kodeken kan implementeres i WebRTC. Basert på valg av videokodek fra teoretisk bakgrunn er ikke H.265 kompatibelt med WebRTC. På den andre siden fant vi ut at Antmedia hadde klart å implementere dette, og vi vil ta med denne kodeken videre fordi den teoretisk sett er like effektiv som de to andre kodekene. Det kommer frem at det er veldig krevende å implementere dette på egenhånd, som kanskje gjør det til en urealistisk ambisjon i dette prosjektet og bacheloroppgaven. Vi kommer til å følge utviklingen tett på om dette blir offisielt implementert i perioden vi jobber med det. Siden teknologien er såpass ny og hele tiden under utvikling, virker det som om det kan ta tid før offisielle kilder får med seg slike oppdateringer, og gjør det vanskelig å finne et konkret svar på spørsmålet. Vi hadde kunne testet dette selv om vi hadde fått implementert funksjonaliteten til å selv endre hvilken kodek den bruker, men siden dette ikke var mulig i denne omgangen vil vi komme tilbake til dette spørsmålet i del to av prosjektet.

Vi kom frem til en estimert datamengde i dekning og hastighet, som vil være aktuell å bruke når vi skal utvikle og teste nye prototyper, i bachelorprosjektet.

Konklusjon

Selv om vi ikke har fått testet på et like godt grunnlag som vi ville, har vi fortsatt fått en bedre forståelse for alle de involverte teknologiene, og tilegnet oss relevant kunnskap vi kan ta med videre. Resultatene vi har fått er viktige veivisere vi kan ta med oss inn i bachelorprosjektet, som betyr at vi kan starte arbeidet på et høyere teknisk nivå enn om vi ikke hadde tilegnet oss denne kunnskapen i dette emnet. En av våre utvalgte kodeker kan hjelpe PreViS sitt samhandlingssystem med å oppnå god kvalitet, som er viktig for at systemet skal fungere godt. Vi kan dermed konkludere, basert på resultatene, at teknologien som bør benyttes blir WebRTC sammen med Socket.io. Vi kan også konkludere med at valg av kodek blir krevende å fastslå i skrivende stund, fordi AV1, H.265 og VP9 er alle gode kandidater.

Fremtidige forbedringer

Prototypen vår benytter seg foreløpig av WebRTC sin standard kodek, VP8. Fremtidige forbedringer inkluderer å endre kodeken til H.265, AV1 eller VP9, slik at løsningen bedre kan håndtere større filstørrelser. Dette vil som tidligere nevnt også la oss teste enda bedre hvilken kodek som gir best resultater i forhold til kravene. Ambisjonen er å kunne utvikle et godt og omfattende system, som kan gi vår oppdragsgiver mer innsikt på det respektive feltet, på veien mot en ferdig løsning. Dette vil også si at prototypen må gjennom store forbedringer med tanke på brukervennlighet, universell utforming og testing. Oppdragsgiver ønsket også en løsning på hvordan bruksflatene skal designes. Å gjøre løsningen sikrere er også et naturlig behov, med tanke på personvern, slik at ikke sensitiv data kommer på avveie. Under vår forskning fant vi eksempler på mange måter WebRTC dataen kan krypteres under bruk, som kan være et steg for å gjøre løsningen mindre utsatt for angrep. Det vil være aktuelt å se nærmere på lydkodekene i del to av prosjektet, for å sikre at også lyden blir godt bevart, ettersom verbal kommunikasjon er viktig for god samhandling.

Referanseliste

Andersen, E.S og Schwencke E. (2020) *Prosjektarbeid: En veiledning for studenter.* 5.utg. Bergen: Fagbokforlaget.

Bross, B. *Et al* (2021) *Overview of the Versatile Video Coding (VVC) Standard and Its Applications.* Tilgjengelig fra: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9503377 (Hentet: 6.desember 2021).

Christie, W., Hoholm, T., Mørk, B.E. (2018) *Innovasjon og samhandling i helsevesenet.* Tilgjengelig fra: https://www.idunn.no/pof/2018/01/innovasjon_og_samhandling_i_helsevesenet_ (Hentet: 1.oktober 2021).

Hancke, P. (2021) *Facetime finally faces WebRTC - implementation deep dive.* Tilgjengelig fra: https://webrtchacks.com/facetime-finally-faces-webrtc-implementation-deep-dive/ (Hentet: 3.desember 2021).

Jain, M. (2021) *Zoom Video Conferencing App: Features, Tech Stack, Monetization, Cost.* Tilgjengelig fra: https://www.konstantinfo.com/blog/zoom-video-conferencing-app/ (Hentet: 3.desember 2021).

Kilbride-Singh, K. (2021) Websockets vs Long Polling, *Ably,* 27.oktober. Tilgjengelig fra: https://ably.com/blog/websockets-vs-long-polling (Hentet: 5.desember 2021).

Levent-Levi, T. (2015) *Forget Whatsapp - Facebook Messages goes WebRTC - Big time.* Tilgjengelig fra: https://bloggeek.me/facebook-messages-webrtc/ (Hentet: 3.desember 2021).

Levent-Levi, T. (2021) *Tweaking WebRTC video quality: unpacking bitrate, resolution and frame rate.* Tilgjengelig fra: https://bloggeek.me/tweaking-webrtc-video-quality-unpacking-bitrate-resolution-and-frame-rates/ (Hentet: 6.desember 2021).

Lynne, A. *Dybdeintervjuer.* Tilgjengelig fra: https://responsanalyse.no/metoder/kvalitative-metoder/dybdeintervjuer/ (Hentet: 2.desember 2021).

Myren, S. (2020) *UHDTV.* Tilgjengelig fra: https://snl.no/UHDTV (Hentet: 5.desember 2021).

Nätt, T.H. og Rossen, E. (2021) *Kodek.* Tilgjengelig fra: https://snl.no/kodek (Hentet: 5.desember 2021).

Oftedahl, L. (2014) *Ny multimonitor i norske ambulanser.* Tilgjengelig fra: https://ambulanseforum.no/artikler/ny-multimonitor-i-norske-ambulanser (Hentet: 24.november 2021).

Posnick, J. (2021) *Experimenting with WebTransport.* Tilgjengelig fra: https://web.dev/webtransport/ (Hentet: 6.desember 2021).

Rossen, E. (2021) *Bitrate.* Tilgjengelig fra: https://snl.no/bitrate (Hentet: 24.november 2021).

Vaagaasar, A.L og Skyttermoen T. (2019) *Prosjektveilederen.* Oslo: Cappelen Damm.

Antmedia. *Ultra Low Latency & Adaptive WebRTC Streaming.* Tilgjengelig fra: https://antmedia.io/ (Hentet: 5.desember 2021).

Amazon Web Services. *High Dynamic Range (HDR) Video.* Tilgjengelig fra: https://aws.amazon.com/media/tech/what-high-dynamic-range-hdr-video/ (Hentet: 6.desember 2021)

Google. *Recommended upload encoding settings - Bitrate.* Tilgjengelig fra: https://support.google.com/youtube/answer/1722171?hl=en#zippy=%2Cbitrate (Hentet: 5.desember 2021).

Meld. St. 28 (2020-2021) (2021) *Vår felles digitale grunnmur.* Oslo: Det kongelige kommunal- og moderniseringsdepartementet. Tilgjengelig fra:

https://www.regjeringen.no/contentassets/e8441e5b035a4e18bbebf74737530c2f/no/pdfs/stm202020210028000dddpdfs.pdf (Hentet: 25.november 2021).

MDN (2021) *Signalling and video calling*. Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Signaling_and_video_calling (Hentet: 1.desember 2021).

MDN (2021) *HEVC (H.265)*. Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs (Hentet: 24.november 2021).

MDN (2021) *The Websocket API (Websockets)* Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API (Hentet; 6.desember 2021).

MDN (2021) *VP9*. Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs (Hentet: 24.november 2021).

MDN (2021) *Reduced frame rate.* Tilgjengelig fra: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs (Hentet: 24.november 2021).

Nkom (2021) *Nasjonal oversikt.* Tilgjengelig fra: https://nettfart.no/nb/data/internettstatistikk (Hentet: 3.desember 2021).

Nkom (2021) *Om 5G.* Tilgjengelig fra: https://www.nkom.no/frekvenser-og-elektronisk-utstyr/om-5g (Hentet: 6.desember 2021).

Opus Interactive Audio Codec. Tilgjengelig fra: https://opus-codec.org/ (Hentet: 6.desember 2021).

Socket.IO (2021) *Introduction.* Tilgjengelig fra: https://socket.io/docs/v4/ (Hentet: 24.november 2021).

Telenor (2021) *Dekning og hastighet.* Tilgjengelig fra: https://www.telenor.no/dekning/ (Hentet: 24.november 2021).

WebRTC. *Real-time communication for the web.* Tilgjengelig fra: https://webrtc.org/ (Hentet: 24.november 2021).

WebRTC (2020) *Signaling.* Tilgjengelig fra: https://webrtc.org/getting-started/peer-connections (Hentet: 3.desember 2021).

Lenke til prototype. Tilgjengelig fra: https://previs-test.netlify.app/

Teknisk utstyr

DS-2CD2545FWD-I(W)(S). Tilgjengelig fra: https://www.hikvision.com/en/products/IP-Products/Network-Cameras/Pro-Series-EasyIP-/DS-2CD2545FWD-I-W--S-/ (Hentet: 30.november 2021).

DS-2CD2787g2-izs. Tilgjengelig fra: https://www.hikvision.com/en/products/IP-Products/Network-Cameras/Pro-Series-EasyIP-/ds-2cd2786g2-izs/ (Hentet: 30.november 2021).

DS-2CD2043G0-1. Tilgjengelig fra: https://www.hikvision.com/en/products/IP-Products/Network-Cameras/Pro-Series-EasyIP-/DS-2CD2043G0-I/ (Hentet: 30.november 2021).

IBR1700 Series Ruggedized Router. Tilgjengelig fra: https://cradlepoint.com/product/endpoints/ibr1700/ (Hentet: 30.november 2021).

MC400 Modular Modem. Tilgjengelig fra: https://cradlepoint.com/product/accessories/mc400-modular-modem/ (Hentet: 30.november 2021).