

Toni Väisänen

TIEDONSIIRTOVERKOT AJONEUVOTEKNIKASSA

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Maaliskuu 2020

TIIVISTELMÄ

Toni Väisänen: Tiedonsiirtoverkot ajoneuvotekniikassa
Diplomityö
Tampereen yliopisto
Automaatiotekniikan diplomi-insinööri
Maaliskuu 2020

Ajoneuvojen lisävarusteet ja sähköjärjestelmät alkoivat kehittyä 80-luvulla. Kasvava määrä soveluksia toi eteen ongelmia johdotuksen määrän kanssa. Haasteisiin vastaamaan kehitettiin tiedonsiirtoverkko Controller Area Network (CAN). CAN-väylä on ollut käytössä tähän päivään saakka, mutta tällä hetkellä ajoneuvoteknologia on tilanteessa, jossa nykyisten verkkojen ominaisuudet eivät pysty vastaamaan tulevaisuuden teknologioiden, kuten autonomisen ajamisen, vaatimuksiin.

Tässä diplomityössä luodaan katsaus ajoneuvojen tiedonsiirtoverkkoihin, niiden nykytilaan ja tulevaisuuteen. Ensimmäiseksi työssä käsitellään ensimmäisen ajoneuvojen tiedonsiirtoväylän, CAN-väylän toimintaa ja rakennetta. CAN-väylä toimii edelleen 40 vuotta kehittämisensä jälkeen ajoneuvojen tiedonsiirrossa, minkä takia sen toimintaan on perehdytty syvemmin. Seuraavaksi työ käsittelee muita yleisimpiä käytössä olevia tiedonsiirtoverkkoja, sekä niiden käyttökohteita. Esittelyn jälkeen työ käsittelee nykyisten teknologioiden haasteita, sekä tulevaisuuden ajoneuvoteknologiaa, sekä sen väylille asettamia vaatimuksia.

Selvityksen perusteella nykyiset verkot ovat liian kapeita kaistanleveydeltään ja liian hitaita. Ajoneuvojen teknologian eteenpäin mennessä tarvitaan parannusta tiedonsiirtoverkkojen joka osa-alueella. Tällä hetkellä paras vaihtoehto on Automotive Ethernet, joka pystyy vastaamaan tulevaisuuden vaatimuksiin.

Avainsanat: CAN-väylä, tiedonsiirtoverkko, Automotive Ethernet, väylä, ajoneuvot

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Toni Väisänen: In-vehicle-networks in automotive technology

Master's thesis

Tampere University

Master of Science in automation technology

March 2020

Car accessories and electric systems were developing fast in the 80's. Rising amount of applications brought problems with the amount of cabling. Controller Area Network (CAN) were developed to meet these challenges. CAN-bus has been in use from the day since. But now car technology is in point where current in-car-networks does not meet the requirements of the future technologies. Such as autonomic driving.

A Survey of the current state and future of the automotive network is made in this master's thesis. First section is about CAN-bus and its structure and functions. CAN-bus is still in charge of the car networking after 40 year of its publishing. That is why it is studied more thoroughly in this thesis. Next is the introduction of the most common in-car-networks, and their applications. After the introduction this thesis deals with challenges of the current networks and future automotive applications and requirements that they set to the networks.

Study reveals that current networks are too slow and too narrow with the bandwidth. As the automotive technology has gone forward, so need to do the in-car-networks. Now solution is not to update the current networks but create new one. Best alternative that can meet the rose requirements is the networks that has been around for a long time, but not in automotive technology, Ethernet.

Key words: CAN-bus, data transfer network, Automotive Ethernet, bus, vehicle

The originality of this thesis has been checked using the Turnitin Originality Check service

ALKUSANAT

Diplomityön tekeminen ei ole ollut helppo prosessi. Se on aloitettu kahdesti, mutta toisella kerralla se vietiin maaliin. Uskon, että lopputyössä saamastani opista tulee olemaan hyötyä työssä sekä harrastuksien saralla. Tahdon kiittää Professori Kalevi Huhtalaa aiheen antamisesta ja rajauksesta. Suuret kiitokset myös Henrikille ja Hermannille korvaamattomasta avusta diplomityön loppuun saattamiseksi. Lisäksi haluan kiittää Herwannan hauiskääntöä, joka on siivittänyt menestykseen myös nostolavoilla ja sitä kautta luonut ystävyysuhteita, jotka tulevat kestämaan läpi elämän. Kiitoksissa ei sovi myöskään unohtaa Eläkeläisiä, jotka potkivat opinnoissa eteenpäin ensimmäisestä insinöörimatemaatiikasta saakka. Kiitos kuuluu myös perheelleni tukemisesta sekä Janinalle, joka uskoi diplomi-insinööriksi valmistumiseeni vaikka aina se ei siltä näyttänytkaan.

Tampereella, 11.3.2020

Toni Väisänen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	CAN-VÄYLÄ	3
2.1	Historia	3
2.2	Verkkojen rakenne	4
2.3	Fyysinen kerros	5
2.4	Elektroniset ohjausyksiköt	5
2.5	Siirtoyhteyskerros.....	6
2.5.1	Kehys	6
2.5.2	Bit stuffing –koodaus	6
2.5.3	Sanomakehys	7
2.5.4	Kyselykehys	8
2.5.5	Virhekehys ja ylikuormituskehys.....	9
2.6	Häiriön käsittely	10
2.7	Kilpavaraukset.....	11
2.8	CAN FD	11
2.9	Topologiat	12
3.	PROTOKOLLAT.....	17
3.1	Ajoneuvoissa käytetyt protokollat.....	17
3.2	Time-triggered ja event-triggered -verkot.....	17
3.3	Local Interconnect Network (LIN)	18
3.4	FlexRay	20
3.4.1	Kommunikaatiosykli.....	21
3.4.2	FlexRay viestikehys	22
3.5	Media Oriented System Transport (MOST).....	23
3.6	Siirtonopeus ja kaistanleveys	26
3.7	Protokollien käyttö ajoneuvoissa	28
4.	TIEDONSIIRTOVERKKOJEN TULEVAISUUS.....	31
4.1	Nykyisten teknologioiden ongelmat.....	31
4.2	Tulevaisuuden teknologiat	32
4.3	Fyysinen kerros tulevaisuudessa	35
4.3.1	Optinen kaapeli	35
4.3.2	Langattomat tekniikat	36
4.4	By-wire -teknologiat	37
4.5	Alijärjestelmien väylille asettamat vaatimukset.....	38
4.6	Tulevaisuuden väylä Automotive Ethernet	40
5.	POHDINTA JA YHTEENVETO	45
	LÄHTEET.....	48

KUVALUETTELO

Kuva 1.	<i>CAN-väylän OSI-malli. Muokattu lähteistä [5, 6].</i>	4
Kuva 2.	<i>Normaali sanomakehys.</i>	7
Kuva 3.	<i>Kyselykehys.</i>	9
Kuva 4.	<i>Virhekehys</i>	9
Kuva 5.	<i>Ylhäällä perinteinen CAN-viestikehys ja alhaalla CAN FD - viestikehys [14].</i>	12
Kuva 6.	<i>Väylätopologia</i>	13
Kuva 7.	<i>Tähtitopologia</i>	13
Kuva 8.	<i>Rengas- ja tuplarengastopologia</i>	14
Kuva 9.	<i>Puutopologia</i>	15
Kuva 10.	<i>Hybriditopologia</i>	15
Kuva 11.	<i>Lin-viestikehys</i>	19
Kuva 12.	<i>FlexRay-viestikehyksen otsikko.</i>	22
Kuva 13.	<i>FlexRay-viestikehyksen datakuormaosio.</i>	23
Kuva 14.	<i>FlexRay viestikehyksen trailer-osio.</i>	23
Kuva 15.	<i>MOST25-protokollan kehys.</i>	24
Kuva 16.	<i>MOST50 viestikehys</i>	25
Kuva 17.	<i>MOST150 viestikehys</i>	26
Kuva 18.	<i>Diagrammi protokollien siirtonopeuksista.</i>	26
Kuva 19.	<i>Tyypillinen ajoneuvon verkkoarkkitehtuuri. Muokattu lähteestä [21].</i>	28
Kuva 20.	<i>Autonomisen ajamisen tasot.</i>	33
Kuva 21.	<i>Automotive Ethernetin DoD-malli verrattuna perinteiseen OSI- malliin. Muokattu lähteestä [36].</i>	42
Kuva 22.	<i>Automotive Ethernet runkoverkko arkkitehtuuri. Muokattu lähteestä [27].</i>	43
Kuva 23.	<i>Keskitetty Automotive Ethernet arkkitehtuuri. Muokattu lähteestä [27].</i>	43
Kuva 24.	<i>Ethernet II paketti ja kehys. Muokattu lähteistä [36, 38].</i>	44

LYHENTEET JA MERKINNÄT

82526	Intelin valmistama ensimmäinen CAN-mikrokontrolleri
82C600	Philipsin ensimmäinen CAN-mikrokontrolleri
ACK	Acknowledge
ADAS	Advanced Driver Assistance Systems
AE	Automotive Ethernet
BasicCAN	CAN-protokollan halvempi versio
BRS	Bit Rate Switch
CAN FD	Controller Area Network with Flexible Data-Rate
CAN	Controller Area Network
CFI	Canonical Form Indicator
CRC	Cyclic Redundancy Check
DEI	Drop Eligible Indicator
DLC	Data Length Code
DoD	Department of Defence
ECU	Electronic Control Unit
EDL	Extended Data Length
EOF	End of Frame
ESI	Error State Indicator
FullCAN	CAN-protokollan normaali versio
IDE	Identifier Extension
ISO	International Organization of Standardization
ISO11898	Standardi, joka määrittelee CAN-väylän
kHz	Kilohertsi
LED	Light Emitting Diode
LIN	Local Interconnect Network
LKAS	Line Keeping Assist System
LLC	Logical Link Control
MAC	Medium Access Control
MDI	Media Dependent Interface
MOST	Media Oriented Systems Transport
OSI	Open Systems Interconnection
PCP	Priority Control Point
PLL	Phase-locked loop
PLS	Physical Signaling
PMA	Physical Medium Attachment
PMMA	Polymetyyliakrylaatti
POF	Polymer Optical Fiber
R0	Bittivaraus laajennuksia varten
R1	Bittivaraus laajennuksia varten
RTR	Remote Transmission Request
SAE	Society of Automotive Engineers
SOF	Start of Frame
SRR	Substitute Remote Request
TCI	Tag Control Information
TDMA	Time Division Multiple Access
TPID	Tag Protocol Identifier
UWB	Ultra-Wideband
V2X	Vehicle-to-Everything

VAN	Vehicle Area Network
VLAN	Virtual Local Area Network
VTC	Volcano Communications Technology
WDM	Wavelength-Division Multiplexing
WiFi	Wireless Fidelity
VID	VLAN Identifier
bit/s	Tiedonsiirtonopeus bittiä sekunnissa

1. JOHDANTO

Liikkuivissa sovelluksissa on nykypäivänä suuri määrä tekniikkaa, joka edesauttaa käyttäjää kuljettamaan ajoneuvoaan. Törmäyksen esto, adaptiivinen vakionopeudensäädin, automaattinen parkkeeraus, edellä mainitut varusteet vaativat huiman määrän anturointiteja ja johdotusta. Kaikkien saatavilla olevien järjestelmien toteuttaminen perinteisellä johdotustekniikalla olisi mahdotonta mahduttaa nykyaikaisiin kulkuneuvoihin. 80-luvulla järjestelmien kehittyminen aiheutti ongelmia painon ja kustannusten suhteen. Ongelman ratkaisuksi alettiin kehittää mobiilikäyttöön tarkoitettua tiedonsiirtomenetelmää, jolla anturoinnin ja johdotuksen määrää saataisiin vähennettyä. Tuloksena oli Boschin kehittämä Controller Area Network -väylä (CAN).

CAN-väylä on automaatiosovelluksissa käytettävä viestipohjainen tiedonsiirtotekniikka, jolla siirretään dataa eri ohjausyksiköiden välillä ilman erillistä isäntätietokonetta. Väylättekniikka oli ratkaisu edellisessä kappaleessa mainittuihin ongelmiin johdotusta vähentävän arkkitehtuurinsa ja kustannustehokkuutensa ansiosta. CAN-väylässä kaikki anturit sekä ohjainlaitteet yhdistyvät toisiinsa väyläkaapelin avulla. Aikaisemmin, esimerkiksi moottoriöljyn lämpötilatietoa tarvitsevat ohjainlaitteet tarvitsivat oman anturinsa, kun taas CAN-väylässä yksi moottoriöljyn lämpötila-anturi lähettää tilatietonsa väylään, se päätyy poimittavaksi niiden ohjainlaitteiden toimesta, jotka sitä tarvitsevat. Nykytilanteessa väylät ovat käytössä laajalti liikkuivissa sovelluksissa, mutta myös kiinteissä automaatiosovelluksissa esimerkiksi tehtaissa. Tuotantolaitoksissa käytettäville väylille onkin omat protokollansa. Tässä työssä perehdytään kuitenkin ajoneuvojen väyläprotokolliin, niiden nykytilaan ja tulevaisuuden näkymiin.

Nykytilanne ajoneuvojen tiedonsiirtoverkkojen kanssa on sama kuin 80-luvulla. Ajoneuvoissa olevan teknologian ja sovellusten määrä kasvattaa anturien sekä johdotusten määrää. Esimerkiksi nykyaikaisessa Volvo XC90 -ajoneuvossa on yli 100 ohjausyksikköä alijärjestelmissään [1]. Tämä on osasyynä sille, että ajoneuvon elektroniikasta on tullut kolmanneksi kallein ja painavin elementti ajoneuvossa [2]. Tulevaisuuden teknologiat lisäävät ajoneuvon tiedonsiirtoverkoissa liikkuvaa dataa joka näin ollen asettaa vaatimukset verkkojen ominaisuuksille. Nykyiset verkot eivät pysty vastaamaan niille annettuihin vaatimuksiin, joten ainoaksi mahdollisuudeksi jää uusien tiedonsiirtoväylien tutkiminen.

Työn tavoitteena on ollut perehtyä ajoneuvojen väyläteknologiaan. Työtä varten on kerätty tietoa eri aikakausilta ajoneuvojen tiedonsiirtoväylien kehityksen ajalta lähtöpisteestä 80-luvulta tähän päivään. Työn aihe on rajattu käsittelemään yleisimpiä ajoneuvoikäytössä olevia väyläprotokollia, ja todennäköisintä tulevaisuuden teknologiaa Automotive Ethernetiä (AE).

Työssä esitellään perinteinen CAN-väylä ja sen toimintaperiaatteet. Tämän jälkeen työssä esitellään nykyhetken käytetyimmät verkkoprotokollat, ja perehdytään niiden teoriaan. Lisäksi perehdytään uusien teknologioiden ja sovellusten väylille asettamiin vaatimuksiin ja haasteisiin sekä käydään läpi sitä, mihin suuntaan väylien kehitys on menossa. Lisäksi työssä käydään läpi tulevaisuuden teknologioita ja niiden aiheuttamia haasteita ja väylille asettamia vaatimuksia. Haasteiden ja vaatimusten valossa tutkitaan myös tulevaisuuden tekniikkaa, joka pystyy vastaamaan näihin haasteisiin. Tulevaisuuden haasteisiin verrataan myös nykyisten ratkaisujen riittävyttä ja sitä kuinka valideja ne tulevaisuudessa ovat.

Työn toisessa kappaleessa käsitellään perinteistä ajoneuvon tietoliikenteessä käytettyä CAN-väylää. Kolmannessa kappaleessa esitellään yleisimmät tällä hetkellä käytössä olevat ajoneuvojen tiedonsiirtoverkot ja niiden käyttökohteita. Neljännessä kappaleessa tutkitaan tiedonsiirtoverkkojen haasteita, luodaan katsaus tulevaisuuden haasteisiin sekä tarkastellaan vaihtoehtoja tulevaisuuden teknologiaksi.

2. CAN-VÄYLÄ

2.1 Historia

1980 alkuvaiheissa Boschilla alettiin miettiä väyläteknologian käyttöä henkilöautoissa. Alkusysäyksen uuden teknologian kehittämiseksi antoi uusien toiminnallisuuden mahdollistaminen, ei niinkään johdotuksen vähentäminen. Sen hetkistä tiedonsiirtoteknologioista yksikään ei soveltunut ajoneuvokäyttöön, joten 1983 alettiin kehittää uutta sarjaväylää. Jo aikaisessa vaiheessa Mercedes Benz oli mukana väylän kehittämisessä [3].

CAN- väylä esiteltiin ensimmäisen kerran 25 vuotta sitten vuonna 1986 Robert Boschin toimesta. Ensijulkaisu tapahtui Society of Automotive Engineers -kongressissa (SAE). CAN-väylä perustui kilpavarausmenetelmään, jossa suurimman prioriteetin viesti saa varata väylän. Väylässä ei ollut ollenkaan isäntäyksikköä. CAN-väylän kehittäjiksi voidaan lukea Boschin insinöörit Wolfgang Borst, Wolfgang Botzenhard, Otto Karl, Helmut Schelling sekä Jan Unruh [3].

Vuonna 1987 Intel esitteli ensimmäisen CAN-mikrokontrollerin, malliltaan 82526. Hie- man tämän jälkeen myös Philips esitteli oman kontrollerinsa, 82C200. Edellä mainitut mikrokontrollerit erosivat toisistaan viestien käsittelyn osalta. Tästä syystä Intelin kontrolleria käytettiin enemmän FullCAN-protokollan yhteydessä, kun taas Philipsin kontrolleria käytettiin BasicCAN-protokollan kanssa [3].

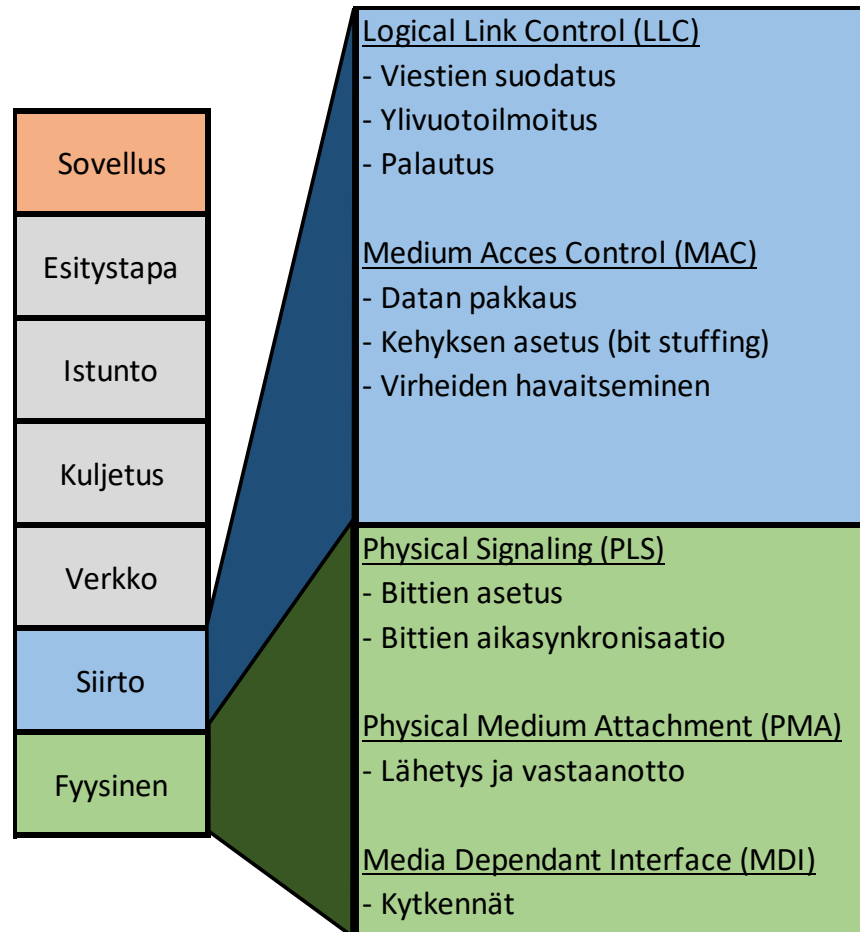
Boschin CAN 2.0 -protokollaa esitettiin kansainvälisesti standardoitavaksi 1990-luvun alussa. Poliittisten erimielisyyksien takia julkaistiin vuonna 1993 ISO11898 standardi yhteistyössä ranskalaisten ajoneuvovalmistajien Vehicle Area Network -väylän kanssa (VAN) [3, 4].

Vaikka CAN-väylä oli kehitetty palvelemaan ajoneuvoteknologiaa, myös muiden alojen valmistajat hyödynsivät väylää tuotteissaan. Esimerkiksi Kone, käytti CAN-väylää hisseissään, sekä ruotsalainen Kvaser hyödynsi CAN-väylää tekstiilikoneiden valmistuksessa. Ajoneuvoissa CAN-väylää käytettiin ensimmäisen kerran vuonna 1987 BMW 850i:ssä, jossa CAN-väylän avulla saatiin diagnostiikkatietoa moottorilta. Mercedes Benz esitteli kuitenkin vuonna 1991 ensimmäisen ajoneuvon, korimallin w140 S-sarjalaisen, jonka koko sähköjärjestelmä pohjautui CAN-väylään [3].

Vaikka CAN-väylä on kehitetty ajoneuvokäyttöön, sitä on kuitenkin hyödynnetty monilla muilla tekniikan alueilla, kuten junissa, lentokoneissa, laivoissa, teollisuuslaitosten automatisoinnissa sekä sairaalatekniikassa. CAN-väylä on vielä tänäkin päivänä laajalti käytössä tiedonsiirtoväylänä. Kehitys on kuitenkin menossa siihen suuntaan, missä CAN-väylän kapasiteetti ei enää kohtaa sovellusten luomia vaatimuksia.

2.2 Verkojen rakenne

Ajoneuvon tietoliikenneverkon Open Systems Interconnection -malli (OSI) koostuu seitsemästä eri kerroksesta. Näistä kolme on määritelty CAN-väylän standardissa ISO11898. CAN-väylän OSI-malli on esitetty kuvassa 1 [5].



Kuva 1. CAN-väylän OSI-malli. Muokattu lähteistä [5, 6].

CAN-väylän standardi siis määrittelee ainoastaan fyysisen- ja siirtokerroksen. Muut kerrokset määrittelee järjestelmän suunnittelija. Siirtokerroksessa hallitaan viestien käsittely. Siirtokerros jaetaan kahteen alikerrokseen Logical Link Control -kerrokseen (LLC), sekä Medium Access Control -kerrokseen (MAC). LLC- kerroksessa tapahtuu viestien suodatus, eli isoimman prioriteetin viesti saa luvan käyttää väylää. Lisäksi myös ylivuotoilmoitus ja järjestelmän palautuminen on LLC-kerroksen vastuulla [5, 6].

MAC-alikerroksessa tapahtuu viestikehyksen asettaminen. Kyseisessä kerroksessa tapahtuu myös mahdolliset virheiden havaitsemiset ja niiden käsittely, datan pakkaus sekä viestien bit stuffing.

Fyysinen kerros koostuu fyysisestä johdotuksesta sekä komponenteista. Fyysinen kerros myös määrittää sen, kuinka viestit todellisuudessa lähetetään. Bittien ajastus sekä synkronointi tapahtuu fyysisellä tasolla [5, 6].

2.3 Fyysinen kerros

Fyysinen CAN-väylä koostuu topologiaaltaan väyläkaapelista ja sen päätepisteistä. Tätä topologiaa kutsutaan väylätopologiaksi. Väyläkaapeli on kytketty jokaiseen asemaan ja sen molempiin päihin on kytketty päätevastukset. CAN-väylillä on olemassa useita erilaisia fyysisiä tasoja, mutta niistä tunnetuimmat ovat ISO-standardin mukaiset ISO-11898-2, joka on saanut nimen high-speed CAN, ja ISO-11898-3, jolla on vastaavasti nimitys low-speed CAN [7]. Standardeja on näiden lisäksi useita, mutta tässä työssä paneudutaan kyseisiin standardeihin niiden yleisyyden vuoksi. Jotkin eri fyysisen tason väylät voivat tietyissä olosuhteissa toimia toistensa kanssa ristiin, mutta nyrkkisääntönä eri tasojen väylät eivät pysty operoimaan keskenään [7].

Väylille on määritelty enimmäispituudet riippuen väylän nopeudesta. Enimmäispituudet määräytyvät siirtotien viiveen mukaan, joka aiheutuu väylässä kulkevan sähkömagneettisen aallon kulkunopeudesta. Mitä matalanopeuksisempi väylä, sitä pidempi siirtotie sallitaan.

Taulukko 1. Väylän nopeus suhteessa maksimipituuteen. Muokattu lähteestä [7].

Nopeus (kbit/s)	Pituus (m)
1000	40
500	100
250	200
125	500
10	6000

Väyliä välillä käytetään yleensä parikaapelia, jonka resistanssi on n.120 ohmia tai ainakin 108-132 ohmia [7]. Nopeissa CAN-väylissä kaapelin tulee olla suojattu, mutta hitaassa CAN-väylässä tätä vaatimusta ei ole. Kyseinen kaapeli on määritelty ISO-11898 standardissa, mutta parikaapelin tilalle on kehitetty yksinkertaista kaapelia, kuten optista kuitua. Optista kuitua käytetäänkin mediajärjestelmien datan siirtoon kehitetyssä MOST-protokollassa.

2.4 Elektroniset ohjausyksiköt

Ajoneuvon tietoliikennejärjestelmä koostuu elektronisista ohjausyksiköistä, jotka väylä liittävät toisiinsa. Elektroniset ohjausyksiköt eli Electronic Control Units (ECU) hoitavat jokainen omaa tehtäväänsä ajoneuvon järjestelmässä. Tyypillisimpänä esimerkkinä

ECU:sta on esimerkiksi moottorinohjausyksikkö, jonka tehtävänä on antaa moottorille tietyt parametrit, joiden pohjalta moottori toimii. Moottorinohjausyksikköön on myös yhdistetty kaapelilla kaikki tarvittavat anturit, joiden antaman datan perusteella se voi ohjata toimilaitteita. Toimilaitteet toimivat yleensä jonkun energianlähteen avulla, esimerkiksi sähkövirralla, hydraulikalla tai pneumatiikalla [8].

ECU:ja voi olla ajoneuvoissa useita kymmeniä. Se, kuinka kompleksinen ECU on, riippuu sen käyttöpaikasta. Yksinkertaisempaan tehtävään tarkoitettu ECU sisältää yhden mikrokontrollerin, mutta monimutkaisempien tehtävien hallintaan tarkoitettujen ECU:t sisältävät useita mikroprosessoreja [8].

Mikäli autoissa on käytössä vain yksi tietoliikenneverkko, kuten esimerkiksi CAN, ajoneuvon kaikki ECU:t ovat yhteydessä toisiinsa. Nykyteknologian mukaisesti ajoneuvoissa on kuitenkin useampi verkko, jotka toimivat omissa alijärjestelmässään [8].

2.5 Siirtoyhteyskerros

CAN- väylä on Broadcast-tyyppinen verkko, jossa asemille ei ole omia osoitteita lainkaan. Jokaisen aseman on mahdollista lähettää sekä vastaanottaa viestejä vuorollaan. Viestin kehysrakenteen taas määrittää sen, mille asemalle viesti on osoitettu. Viesteissä on tunnisteet, joiden mukaan asemat poimivat viestivirrasta itselleen tarpeelliset sanomat, ja lähettävät eteenpäin [9, 10].

2.5.1 Kehys

Väylässä kulkee protokollan määrittelemänä neljä kehystä. Kehykset ovat sanoma-, kysely-, virhe- ja ylikuormituskehys. Kehyksen pääosat ovat tunnistekenttä, joka kertoo sanoman tunnistenumeron, Tietokentän pituuden määrittelevä Data Length Code -kenttä (DLC) sekä databittien kenttä, jossa varsinainen sanoma asemalle on kirjattu. CAN-väylän viestin koko on melko pieni, maksimipituudeltaan 94 bittiä [10, 11].

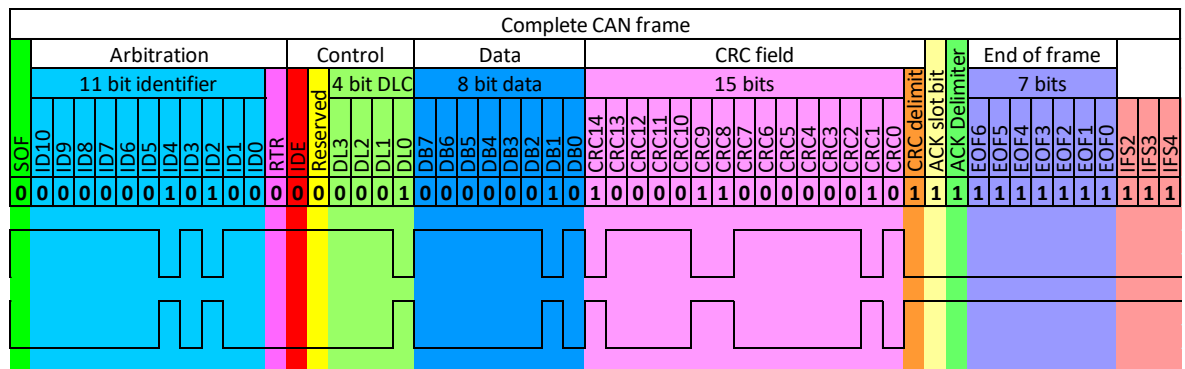
2.5.2 Bit stuffing –koodaus

Bit stuffing on yksi keino CAN-väylän virheenhallinnassa. Ideana on lisätä yksi ylimääräinen erimerkkinen bitti viiden samanlaisen peräkkäisen bitin perään. Tällä tavoin ehkäistään normaalin kehysten sekoittaminen virhekehykseen. Virhekehys ei noudata bit stuffingia joten siinä voi esiintyä kuusi samanmerkkistä bittiä. Lähettävä asema lisää stuff bitit viestikehykseen ja vastaanottava asema poistaa bitit viestistä ennen sen välittämistä valittuun sovellukseen. Stuff bittien takia on otettava huomioon, että kehykset voivat olla suurempia [12].

2.5.3 Sanomakehys

Sanomakehys on ainoa kehys, joka on tarkoitettu datan siirtämiseen. Sanomakehystä on kahta erilaista, 11 bitin tunnisteella varustettu normaalikehys, sekä 29 bitin tunnisteella varustettu pidennetty sanomakehys [11]. Kehysmallin valintaan vaikuttaa se, kuinka monta eri viestiä asemat tarvitsevat. Lyhemmällä tunnisteella varustettu viestikehys pysyy luomaan 2048 erilaista tunnistetta. Pidennetty tunniste taas mahdollistaa yli 500 miljoonan eri tunnisteen käytön [10]. Kuvassa 2 on esitetty standardimallisen sanomakehysen rakenne. Kehys voidaan jakaa viiteen eri osioon, jotka koostuvat yksittäisistä biteistä. Osiot ovat nimeltään arbitraatiokenttä (Arbitration), ohjauskenttä (Control), datakenttä (Data), sekä syklinen tarkistussummakenttä (Cyclic Redundancy Code, CRC field).

Standardikehys



Kuva 2. Normaali sanomakehys.

Kehyksen osat:

- SOF, Start of Frame, dominantti bitti, joka merkkää sanoman alun.
- 11 bittinen tunnistekenttä, jonka mukaan asema vastaanottaa sille tarkoitetun sanoman.
- RTR, Remote Transmission Request, bitti on dominantti, mikäli kehys on datakehys ja tarkoitettu jonkin aseman luettavaksi. Mikäli kyseinen bitti on resessiivinen, kehys on kyselykehys.
- IDE, Identifier Extension, dominantti bitti kertoo kehysen olevan normaali CAN-sanomakehys ja resessiivinen bitti merkkää kehysen olevan jatkettu kehys.
- Reserved bit, varattu bitti tulevaisuuden standardimuutoksia varten.
- DLC, Data Length Code, 4 bittiä, jotka kertovat datakentän suuruuden.
- DATA, varsinainen sanomakehys, joka voi sisältää laajimmillaan 64 bittiä.
- CRC, Cyclic Redundancy Check, tarkistussumma, jonka viestin lähettävä asema luo lähetettävistä biteistä. Vastaanottava asema tekee saman ja vertaa tulosta lähetettyyn tarkistussummaan. Tämä tehdään virheidensuodatuksen takia.
- CRC delimiter, resessiivinen bitti, joka päättää tarkistussumman.

- ACK, Acknowledge, kuittauskenttä. Koostuu kahdesta bitistä, joista ensimmäiseen vastaanottava asema kirjoittaa dominantin bitin onnistuneen vastaanoton merkiksi, mikäli tarkistussumma CRC on ollut oikein. Toinen bitti, ACK delimiter, on aina resessiivinen ja päättää kuittauskentän.
- EOF, End of Frame, seitsemän peräkkäistä resessiivistä bittiä päättävät sanoman.
- IFS, Interframe space, resessiiviset bitit sanoman lopussa ovat odotusaikaa sanomien välillä, jonka ohjain tarvitsee siirtääkseen sanoman oikealle paikalleen sanomapuskurissa [10].

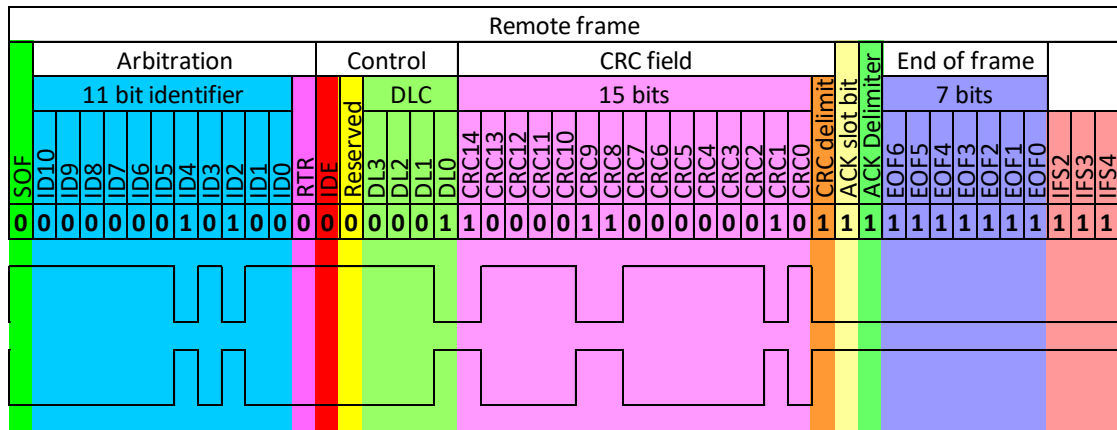
Jatkettu kehys

Jatkettu sanomakehys eroaa standardi sanomakehyksestä käytännössä siten, että sen tunnistekenttä on 11 bitin sijaan 29 bittinen. Kenttä on kuitenkin jaettu kahteen osaan siten, että 11 bittinen tunnistekenttä on alkuperäisellä paikallaan. Sitä seuraa SSR kenttä ja IDE kenttä. SRR (Substitute Remote Request) bitti asetetaan aina resessiiviseksi, jotta kilpa-vaaraustilanteessa standardimittainen kehys priorisoidaan tärkeämmäksi, mikäli molemmilla on samanlainen 11 bitin tunnistekenttä. Seuraavana tulee IDE-osio, joka on merkattu resessiiviseksi jatkettun tunnistekentän merkiksi [10, 11].

2.5.4 Kyselykehys

Kyselykehyyksen tarkoitus on pyytää sanomaa halutulta asemalta. Kehys toimii siten, että asema, joka tarvitsee sanoman, lähettää kyselykehyyksen tietyllä tunnistekentällä. Asema jolle kysely on tarkoitettu, vastaanottaa kyselykehyyksen ja lähettää tilalle samalla tunnistekentällä olevan sanomakehyyksen.

Kuvassa 3. on esitelty kyselykehys. Kyselykehys eroaa sanomakehyksestä kahdella tavalla. Ensimmäinen on RTR, Remote Transmission Request, kenttä on asetettu resessiiviseksi bitiksi merkkamaan kehyyksen olevan kyselykehys. Toinen eroavaisuus on se, että kyseisestä kehyyksestä puuttuu kokonaan datakenttä [10].

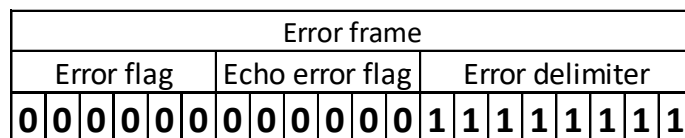


Kuva 3. Kyselykehys

Kyselykehys on kuitenkin käytännössä vähän käytetty ominaisuus, mutta niillä pystytään implementoimaan kysely-vastaus tyyppistä väyläliikennettä.

2.5.5 Virhekehys ja ylikuormituskehys

Asema lähettää virhekehysten, kun se on havainnut virheen vastaanottamassaan viestikehyksessä, tai se on vastaanottanut virhekehysten. Vastaanotettuaan virhekehysten lähettävä asema automaattisesti yrittää lähettää uutta sanomaa. Jokaisella asemalla on lascurit, jotka pitävät kirjaa lähetetyistä virhekehyksistä. Lähetetyistä virhekehyksistä pidetään kirjaa, jotta virhettä lähettävä asema ei tuhoa koko väylää ylikuormittamalla sitä toistuvilla virhekehyksillä. Virhekehys koostuu virhelipusta ja virheen päättökentästä kuten kuvassa 4 on esitetty. Virhelippu on 6-12 dominanttia tai resessiivistä bittiä. Mikäli virhekehysten lähettävä asema, joka on huomannut virheen väylässä, se lähettää kuusi dominanttia bittiä sisältävän virhelipun. Mikäli asema vastaanottaa aktiivisen virhelipun se lähettää passiivisen virhelipun, eli kuusi resessiivistä bittiä alkuperäisen aktiivisen virhelipun perässä. Loppuosa, eli virhekehysten päättävä kenttä koostuu kahdeksasta resessiivisestä bitistä. Virhekehys on ainoa kehys, johon ei kuuden peräkkäisen bitin jälkeen bit stuffing –säännön mukaisesti lisätä vastakkaistilaisia bittejä. Kun vastaanottava asema ei saa vastaanotettua oikeanlaista data- tai kyselykehystä, asema keskeyttää vastaanoton ja lähettää välittömästi väylään virhekehysten [10, 11].



Kuva 4. Virhekehys

Ylikuormituskehys on käytännössä samanlainen kuin virhekehys. Eron, milloin kehystä kutsutaan virhekehykseksi ja milloin ylikuormituskehykseksi, tekee se, milloin kehys lähetetään. Asema luo ylikuormituskehysten siinä tilanteessa, kun asema on liian kiireinen

aikaisempien viestien käsittelyjen kanssa, joten se lähettää virhekehyksen saadakseen lisää aikaa viestien väliin. Ylikuormituskehyksiä käyttäviä kontrollereja on nykypäivänä enää vain harvoja [10].

2.6 Häiriön käsittely

CAN-väylä ei ole tiedonsiirtoverkkona virheetön vaan myös siinä ilmenee häiriöitä. Häiriöitä yritetään estää ja sellaisen sattuessa ne yritetään käsitellä siten, että muu järjestelmä pystyisi jatkamaan toimintaansa. CAN-väylän virheentarkistus koostuu bit stuffingin lisäksi kolmesta eri tekniikasta. Mikäli yksikin näistä tarkastuksista epäonnistuu, lähettää kyseinen asema väylään virhekehyksen.

Ensimmäinen kohta viestin virheen tarkastuksessa on kehyksen CRC osion tarkastussumma. Asema laskee viestin CRC luvun, ja vertaa sitä sitten kehyksessä olevaan CRC tarkastussummaan. Mikäli luvut pitävät paikkansa, virhekehystä ei lähetetä [9].

Klassisessa CAN-protokollassa käytetään 15 bittistä polynomia CRC:n laskemiseen. Lasku toimii Hammingin koodin mukaisesti kaavalla 1.

$$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + x^0 \quad (1)$$

laskentakaava toimii tavalla, jolloin viisi sattumanvaraista bittivirhettä on mahdollista huomata [9].

Kehyksessä seuraavana virheen tarkastuksessa on ACK osio. Lähettävä asema asettaa ACK osion bitin resessiiviseksi ja lähettää kehyksen väylään. Jokainen asema, joka vastaanottaa kehyksen kirjoittaa ACK osion bitin dominantiksi. Jokainen asema tekee tämän huolimatta siitä, onko asema kiinnostunut kyseisestä kehyksestä vai ei. Mikäli lähettävä asema vastaanottaa kehyksen, jonka ACK bitti on resessiivinen, lähettää se ACK error – kehyksen. Käytännössä tämä virheen tarkistus ottaa huomioon sen, että saako mikään asema vastaanotettua kehystä [12].

Virheen tarkastuksen keinona käytetään myös kehyksen aina resessiivisten bittien tarkastelua. SOF, EOF, AKC delimiter ja CRC delimiter osiot ovat aina resessiivisiä bittejä. Mikäli asema huomaa jonkin kyseisistä biteistä olevan dominantti, lähettää se väylään virhekehyksen [10].

Virhekehyksen lähettää myös bit stuffing –säännön rikkominen. Mikäli asema vastaanottaa viestin, joka sisältää yli viisi peräkkäistä samassa tilassa olevaa bittiä, tulkitaan kehys virheelliseksi [10].

2.7 Kilpavaraukset

Koska mikä tahansa asema voi lähettää väylään viestiä silloin kun väylä on vapaa, tulee väistämättä tilanteita, että kaksi väylää yrittää lähettää viestiä samanaikaisesti. Tilannetta kutsutaan kilpavaraukseksi. Kilpavaraustilanteet ratkaistaan tarkastelemalla viestin tunnistenumeroa, jonka mukaan vuorot jaetaan. Tunnistenumero tarkastellaan bitti bitiltä. Kahden viestikehyksen bittejä verrataan toisiinsa bitti kerrallaan. Pääsyn väylään voittaa se viestikehyks, jonka verrattavat bitit ovat dominantteja (0) [13].

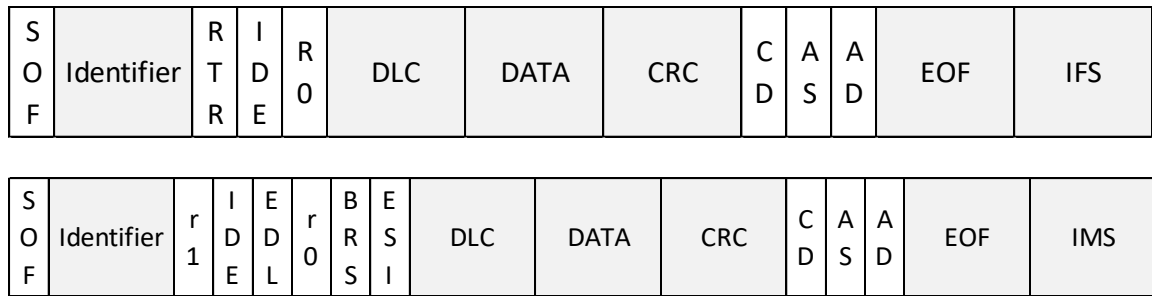
2.8 CAN FD

Kun ensimmäisen kerran CAN-väylän rajat tulivat vastaan, täytyi uusia sovelluksia ja lakeja varten luoda CAN-väylälle laajennus. 80-luvulla kehitetty perinteinen CAN-väylä ei enää pysty vastamaan sovellusten asettamiin vaatimuksiin. Näitä vaatimuksia täyttämään kehitettiin CAN with Flexible Data-Rate (CAN FD). CAN FD ratkaisi ongelmat lisäämällä kaistanleveyttä minimaalisella kustannuksella [14].

Klassisen CAN-väylän kaistanleveydeksi on määritelty ISO 11898 standardin mukaan 1 Mbit/s. Vaativissa sovelluksissa tämä aiheuttaa rajoitteita tietoliikenteelle. Kapea kaistanleveys on aiheuttanut sen, että ajoneuvojen suunnittelijat alkoivat käyttää päällekkäisiä CAN-verkkoja. CAN väylä oli samassa pisteessä, jota ratkaisemaan se alun perin oli kehitetty, ylimääräisen johdotuksen määrää oli vähennettävä [14].

CAN FD:n ratkaisuna olemassa olevaan ongelmaan oli kasvattaa 1 Mbit/s siirtonopeutta suuremmaksi, joka mahdollistaa yli 8 Mbit viestien lähettämisen. Suosituimmat käytössä olevat kaistanleveydet ovat 2 Mbit/s ja 5 Mbit/s, mutta jotkin valmistajat tavoittelevat jopa 8 Mbit/s kaistanleveyttä. CAN FD vastaa näihin vaateisiin ja mahdollistaa lisääntyneen datasiirron mahdollistamalla 64 bittiä yhteen viestiin [14].

Kuvassa 5 ja 6 on kuvattu perinteinen CAN-viestikehys sekä CAN FD viestikehys. Ensimmäinen eroavaisuus huomataan identifier-kentän jälkeen, jossa CAN FD viestikehyksessä on r1 kenttä. R1 kenttä on bittivaraus tulevaisuuden protokollien mahdollisille standardimuutoksille. Perinteisen CAN kehyksen r0 kenttää käytetään CAN FD:ssä uudelleensynkronointiin ennen mahdollista siirtonopeuden muutosta. Seuraava perinteisestä CAN viestistä eroava kenttä on Extended Data Length (EDL). Kyseinen bitti on dominantti, joka toimii indikaattorina laajennetun datakehyksen käytöstä. Seuraavana on bit rate switch -bitti (BRS). BRS-bitin tarkoituksena on tiedottaa data osiossa käytettävästä bittinopeudesta. Dominantti bitti tarkoittaa data kentässä käytettävän samaa nopeutta kuin tunnistekentässä. Resessiivinen bitti taas tarkoittaa bittinopeuden olevan korkeampi data kentässä. Error State Indicator -kenttä (ESI) kertoo dominantilla bitillä kehyksen lähettävän aseman olevan virhetilassa. Resessiivinen bitti kertoo taas aseman toimivan oikein. Loppuosa kehyksestä koostuu samoista osioista kuin perinteisen CAN-viestin viestikehys [14].



Kuva 5. Ylhäällä perinteinen CAN-viestikehys ja alhaalla CAN FD -viestikehys [14].

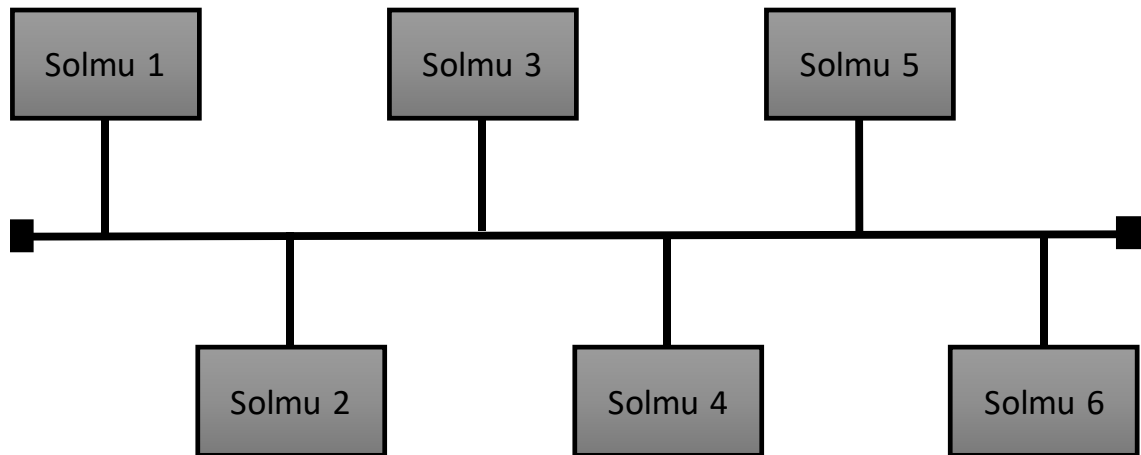
CAN FD tarjoaa paljon etuja muihin verkkoprotokolliin verrattuna. CAN FD on helppo ottaa käyttöön vanhan CAN protokollan rinnalle, sillä sen fyysiset komponentit eivät eroa vanhasta CAN väylästä, joten muutoksen tekeminen ei aiheuta suuria hankintoja ja kustannuksia. CAN FD:n fyysinen kerros on itseasiassa sama kuin CAN high -protokolla, jota käytetään normaalin CAN väylän yhteydessä. Ainut ero komponenteissa on se, että CAN FD:n käyttöä varten tarvitaan uuden protokollan kontrolleri, joka soveltuu nopeammalle tiedonsiirrolle [14].

2.9 Topologiat

Topologialla tarkoitetaan sitä, kuinka verkossa olevat laitteet on aseteltu ja kuinka ne on kytketty toisiinsa. Edellä on esitelty eri tapoja liittää laitteet toisiinsa, ja käsitelty asioita, jotka vaikuttavat oikeanlaisen topologian valintaan. Topologiat voidaan jakaa kahteen eri osaan, fyysiseen topologiaan ja loogiseen topologiaan. Fyysinen topologia tarkoittaa laitteiden ja solmujen kytkentää ja fyysistä asetelua. Looginen topologia taas tarkoittaa sitä missä järjestyksessä data kulkee kyseisessä verkossa. Saman verkon looginen ja fyysinen topologia eivät tarvitse olla samanlaiset, vaan ne voivat poiketa toisistaan. Topologiat voidaan jaotella viiteen erilaiseen perusmalliin,

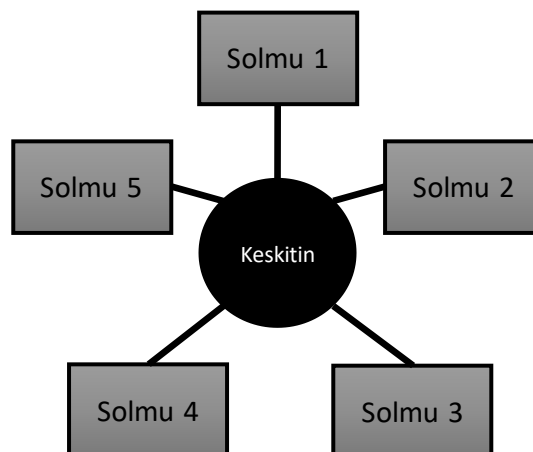
- Väylä
- Tähti
- Rengas
- Puu
- Hybriditopologiaan [15].

Topologian valintaan vaikuttaa useat asiat. Käytettävän kaapelin pituus ja millaista kaapelia aiotaan käyttää. Kaapelin hinnat vaihtelevat sen pituuden ja laadun mukaan, joten käytettävissä oleva budjetti vaikuttaa myös topologian valintaan. Mitä monimutkaisempi topologia on, sitä kalliimpaa sen asennuksesta tulee. Väylän fyysinen sijainti vaikuttaa myös topologian valintaan. Esimerkiksi väylämallinen topologia myötäilee ajoneuvon johtosarjaa. Myös ajoneuvon tiedonsiirtoverkon skaalautuvuusvaatimus vaikuttaa topologian valintaan. Esimerkiksi tähtitopologiaan on paljon helpompi lisätä solmuja, verrattuna esimerkiksi rengastopologiaan [15].



Kuva 6. Väylätopologia

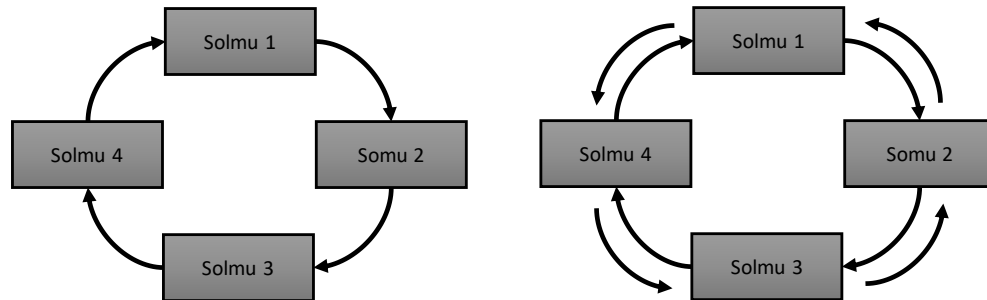
Kuvassa 6 on esitetty perinteinen väylätopologia. Väylätopologian ytimenä on runkokaapeli, johon solmut kytketään. Pääväyläkaapelin molemmat päät ovat terminoitu, eli niihin on kytketty vastukset, jotka päättävät väylän. Väylän päättäviä vastuksia kutsutaan terminaattoreiksi. Väylätopologian etuna on johdotuksen yksinkertaisuus ja kustannustehokkuus, sillä kaapeloinnin määrä säilyy minimissään. Väylätopologian huonona puolena on se, että se perustuu kilpavaraustekniikkaan. Tekniikan periaatteena on se, että vain yksi solmu kerrallaan voi kirjoittaa väylään. Väylätopologia on yksisuuntainen väylä, eli viestit kulkevat vain yhteen suuntaan kerrallaan. Tilanteessa, jossa moni yksikkö haluaa kirjoittaa väylään, aiheutuu viesteillä yhteentörmäys ja väylä ruuhkautuu. Huonona puolena väylätopologiassa on myös sen vikaherkkyys. Mikäli pääkaapeli vioittuu, koko verkko ajautuu alas [15].



Kuva 7. Tähtitopologia

Tähtitopologiassa verkon jokainen solmu yhdistyy keskellä sijaitsevaan keskittimeen tai niin kutsuttuun keskussolmuun kuten kuvassa 7 on esitetty. Näin ollen jokainen solmu on yhteydessä toisiinsa keskittimen välityksellä. Tähtitopologia mahdollistaa muotonsa ansiosta pidemmät välimatkat solmujen välille, verrattuna esimerkiksi väylätopologiaan.

Tähtitopologia on myös luotettavampi, sillä yhden haaran vioittuessa, muu verkko voi jatkaa toimintaansa välittämättä siitä. Rakenteensa ansiosta verkon käyttäjät voivat helposti vikatilanteessa paikallistaa vialliset komponentit. Tähtitopologia tarjoaa rakenteellaan myös helpomman muokattavuuden. Lisättäessä uutta yksikköä verkkoon ei koko verkkoa tarvitse pysäyttää kytkentää varten. Tämä ei onnistu esimerkiksi kuvassa 8 esitellyn rengastopologian kanssa. Rakenteen tärkein komponentti on keskussolmu eli keskittin. Keskittimen vikaantuminen lopettaa liikenteen solmujen välillä ja aiheuttaa verkon kaatumisen [15].

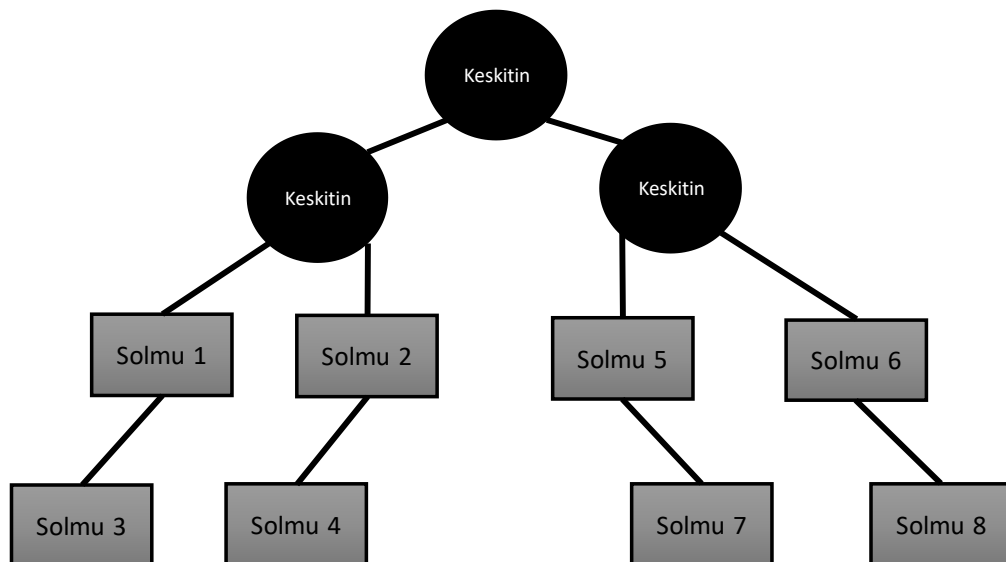


Kuva 8. Rengas- ja tuplarengastopologia

Rengastopologiassa solmut ovat kiinnitetty kuvan 8 mukaisesti toisiinsa renkaanmuotoon. Tällöin jokainen solmu on kiinnittynyt kahteen muuhun solmuun. Yksi solmuista on ensimmäinen solmu, joka tulee olemaan verkon konfiguraattori ja valvoo muita solmuja. Perinteinen rengastopologia on yksisuuntainen topologia, eli viestit kulkevat ainoastaan yhteen suuntaan. Rengastopologiasta on kuitenkin mahdollista tehdä kaksisuuntainen kytkemällä se kuvan 8 mukaisesti tuplarengastopologiaksi. Tällöin johdotuksen määrä tuplaantuu koska tarvitaan kytkennät myötäpäivään sekä vastapäivään [15].

Rengastopologiassa on sama ongelma kuin väylätologiassakin. Samaan aikaan lähetetyt kehykset törmäävät ja väylä ruuhkaantuu. Rengastopologiassa viestien törmäyksiä vähennetään tekniikalla, jossa ainoastaan yksi solmu kerrallaan saa luvan lähettää viestejä verkkoon. Tuplarengastopologia tarjoaa ratkaisun viestien törmäyksen välttämiseen. Lisäksi tuplarengas on varmempi fyysisien vikojen varalta. Se esimerkiksi kestää kahden solmun välisen kaapelin rikkoontumisen, ilman merkittävää vaikutusta verkon toimintaan [15].

Edun varjopuolena on kuitenkin se, että yhdenkin solmun vikaantuminen saa koko verkon kaatumaan. Lisäksi normaalissa rengastopologiassa myös kaapelin vikaantuminen saa verkon kaatumaan. Rengastopologian huonoja puolia on myös sen heikko skaalautuvuus. Mikäli verkkoon halutaan lisätä komponentteja, joudutaan koko verkko sulkemaan kytkemisen ajaksi [15].

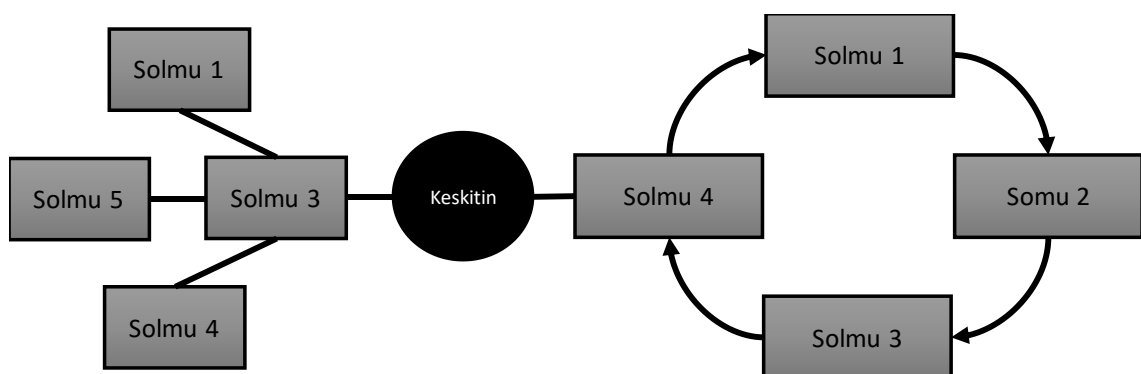


Kuva 9. Puutopologia

Kuvassa 9 on esimerkki puutopologiasta. Puutopologiassa on yksi juurisolmu, joka on yhdistettynä jokaiseen verkon muihin solmuihin. Hierarkia on vanhempi-lapsi tyyppinen, jossa solmujen välillä on yksi molemminpuolinen yhteys. Jotta topologiaa voidaan kutsua puutopologiaksi, solmuja täytyy olla vähintään kolmessa tasossa [15].

Puutopologian tarjoamia etuja on hyvä skaalautuvuus sekä helppo vianetsintä. Puutopologiaan on helppo lisätä uusia yksikköjä, sitä mukaa kun verkko kasvaa. Vianetsinnässä on helppoa käydä läpi jokainen oksa yksitellen [15].

Huonoja puolia puutopologiassa on järjestelmän toiminnan nojautuminen yhteen solmuun. Mikäli juurisolmu vikaantuu, koko verkon toiminta keskeytyy. Puutopologia vaatii myös huomattavasti enemmän johdotusta kuin muut topologiat. Lisäksi layout voi kasvaa melko monimutkaiseksi, mikäli verkkoon lisätään paljon laitteita [15].



Kuva 10. Hybriditopologia

Hybriditopologia on kahden topologian yhdistelmä. Kuvassa 10 on esitetty esimerkkinä tähti- ja rengastopologian yhdistelmä. Hybriditopologioita käytetään yleensä niissä pai-

koissa, missä yhden topologian ominaisuudet, esimerkiksi skaalattavuuden ja vikaherkyyden takia eivät kohtaa. Hybriditopologian luo verkoille joustavuutta. Hybridiverkkojen vaarana on kuitenkin muodostua liian kompleksiksi. Tämä voi aiheuttaa hankaluuksia vian paikantamiseen, mikäli topologioita on sekoitettu liiaksi keskenään [15].

Kaikkia edellä mainittuja topologioita löytyy nykyajoneuvojen tiedonsiirtoverkoista. Yleisin tapaus on kuitenkin se, että suuri osa verkoista on kytketty toisiinsa, jolloin ne muodostavat suuren hybridiverkon. Tällöin jokaisella protokollalla on optimaalinen topologiansa omaa toimintaansa varten ja tällöin erilliset verkot pystyvät olemaan yhteydessä toistensa kanssa.

3. PROTOKOLLAT

3.1 Ajoneuvoissa käytetyt protokollat

Erilaiset vaatimukset kaistanleveydessä, tehokkuudessa, kustannuksissa ja luotettavuudessa ovat saaneet aikaan laajan kirjon hieman toisistaan eroavia tiedonsiirtoverkkoja ajoneuvokäytössä. Tarpeen eri teknologioille luo myös itse alijärjestelmä, jonka dataliikennettä tiedonsiirtoverkko tukee. Tietoliikennejärjestelmä voi esimerkiksi toimia järjestelmissä syntyvien tapahtumien ajamana, tai järjestelmän vaatiman reaaliaikaisuuden mukaan. Tällöin järjestelmät jaetaan tapahtumaliipaisu ja aikaliipaisuverkkoihin.

Tässä diplomityössä paneudutaan tarkemmin yleisimpiin ajoneuvoissa käytettyihin tiedonsiirtoprotokolliin. Työssä käsitellyt protokollat ovat CAN-, CAN FD-, LIN-, MOST- ja FlexRay-väylä. Jokaisella edellä mainitulla väylällä on oma tehtävänsä ajoneuvon tiedonsiirrossa. Väylillä on ominaisuutensa, minkä takia ne soveltuvat juuri kyseisen alijärjestelmän tiedonsiirtoväyläksi.

Ajoneuvo systeeminä voidaan jakaa seuraaviin alijärjestelmiin, voimalinja, alusta, kori ja media/diagnostiikka. Jokaisella alijärjestelmällä on erilaiset vaatimukset tiedonsiirtoväylän suhteen ja siksi niille on jokaiselle kehitetty omansa.

3.2 Time-triggered ja event-triggered -verkot

Tietoliikennejärjestelmät ajoneuvotekniikassa voidaan jakaa kahteen ryhmään. Event-triggered ja Time-triggered -järjestelmä. Tässä työssä käytetään sanaa englannin kielen sanoja *event-triggered* ja *time-triggered*, koska niille ei ole sopivia suomenkielisiä termejä.

Se, että järjestelmä on event-triggered, tarkoittaa sitä, että viestejä siirretään ainoastaan, kun väylän järjestelmässä tapahtuu toimintoja. Esimerkiksi ajoneuvon sähköikkunat. Usein ikkunoita ohjataan LIN-väylällä, joka on event-triggered -tyyppiä. Kun käyttäjä haluaa ikkunan laskeutuvan ja painaa kytkintä, lähtee kytkimeltä signaali ikkunamoduulille, joka lähettää viestin eteenpäin ikkunan moottorille laskea ikkuna. Event-triggered -tyyppisillä väylillä täytyy olla menetelmä, jolla annetaan väylän käyttöoikeus sille ohjausyksikölle, joka väylää kulloinkin haluaa käyttää. Tällä tavalla estetään samanaikainen lähetys ja viestien törmäykset. CAN-väylä käyttää tekniikkaa, jossa jokaiselle viestille luodaan prioriteetti arvo. Jos jollain hetkellä kaksi ohjausyksikköä haluaa lähettää viestin samaan aikaan, paremmalla prioriteetilla varustettu ohjausyksikkö saa luvan lähettää viestin väylään ensin. Koska väylään lähetetään ainoastaan tärkeät viestit silloin kun jokin toiminto tapahtuu, event-triggered väylät ovat taloudellisia kaistan kapasiteetin käytön suhteen [16].

Time triggered -järjestelmässä viestit lähetetään väylään ennalta määrätyn aikataulun mukaisesti. Tämä sykli toistaa itseään niin kauan kuin väylä on toiminnassa. Tästä puhutaan myös nimellä TMDA, joka tulee sanoista Time Division Multiple Access. Suomenkielissä käytetään termiä aikajakokanavointi. Time-triggered tyyppistä väylää käytetään sellaisissa sovelluksissa, jotka vaativat datansiirrolta reaaliaikaisuutta. Esimerkkinä tällaisesta sovelluksesta on esimerkiksi polttomoottorin moottorinohjaus, jonka täytyy koko ajan lukea moottorin antureilta tilatietoa, jotta moottori kävisi oikealla seossuhteella joka tilanteessa. Time-triggered väylän etuna on viallisten ohjausyksiköiden nopea tunnistaminen. On helppo havaita, kun jokin verkon ohjausyksikkö ei lähetä viesti vuorollaan. Huonoja puolia time-triggered -tyypin verkoissa on niiden huono joustavuus. Jos verkoon halutaan lisätä uusi solmu, joudutaan aikasykli luomaan uudestaan. Lisäksi jokainen solmu päivitetään vastaamaan uutta aikataulua. Lisäksi time-triggered verkot ovat huonoja toimimaan ajastamattomien viestien kanssa. Kun väylään lähetetään ajastamaton viesti, siihen lähetetään vastaus vasta kun sille on aikataulussa oma vuoronsa, vaikka vastaus tarvittaisiin järjestelmän toiminnan kannalta heti [16].

3.3 Local Interconnect Network (LIN)

Kaikki sovellukset mobiileissa laitteissa eivät kuitenkaan aina tarvitse massiivisia määriä dataa tai kaistaa sen kuljetukseen. Esimerkiksi ajoneuvojen sähköikkunoiden ohjaus toimii hyvinkin pienellä datamäärällä, johon jopa perinteinen CAN-väylä on ylimitoitettu. Tällaisia sovelluksia varten on kehitetty Local Interconnect Network, eli LIN-väylä. Lin väylä tarjoaa kustannustehokkaan tiedonsiirtoverkon sellaisia sovelluksia varten, jotka eivät tarvitse suurta kaistanleveyttä tai monipuolisuutta. Liikuvissa laitteissa LIN-väylää käytetään usein korin sähköjen toteutuksessa, kuten sähköistuimet, valot tai ikkunannostimet. LIN-väylän protokolla perustuu Volvon tytäryhtiön Volcano Communications Technology (VTC) kehittämään Volcano-Lite teknologiaan. Vuonna 1999 ensimmäinen LIN-protokolla julkaistiin vaihtoehtona CAN-väylälle. Protokollaa kuitenkin päivitettiin 2000-luvulla kahdesti. Marraskuussa 2002 julkaistiin LIN1.3, joka sisälsi muutoksia lähinnä väylän fyysisessä kerroksessa. Viimeisin versio julkaistiin vuonna 2003, joka kantaa nimeä LIN 2.0. Uudessa versiossa oli mukana muun muassa diagnostiikkaominaisuus [17].

LIN-väylä on topologialtaan isäntä ja renki -tyyppinen (master/slave -verkko). Tämä tarkoittaa sitä, että väylässä on yksi isäntäyksikkö, joka ohjaa muita väylässä olevia yksiköitä eli niin kutsuttuja renkiyksiköitä. Isäntäyksiköt voivat toimia myös renkinä, mutta renkit voivat toimia vain ja ainoastaan renkeinä. Kaikesta viestinnästä verkossa vastaa isäntäyksikkö, joka lähettää viestikehykset verkkoon [17, 18].

Ennen LIN-verkon käynnistämistä, renkiyksiköt tulee konfiguroida joko julkaisemaan dataa verkkoon, tai lukemaan dataa viestin vastausosion. Isäntäyksikön lähettämä otsikko, voidaan jakaa kolmeen osioon: tauko-, synkronointi- sekä tunnistekenttään. ren-

kiyksiköiden vastaanottaessa otsikkoviestin, ne tarkastavat tunnisteiden, joka määrittää tuleeko kyseisen aseman julkaista vai vastaanottaa dataa. Mikäli tunniste määrittää aseman dataa julkaisevaksi asemaksi, se kirjoittaa vastausosion datakenttään kahdeksanbittisen viestin. Mikäli asema on tunnisteiden mukaan dataa vastaanottava, se lukee vastausosion datakentän ja välittää käskyt toimilaitteelle sen mukaan. Normaalitilanteessa isäntä ja renki -tyyppisessä verkossa isäntäyksikkö lähettää tunnisteiden verkkoon ja vain yksi renkiyksikkö vastaa datalla [18].

Break	Sync	Identifier	Data	Checksum
Header			Response	

Kuva 11. Lin-viestikehys

Kuvassa 11 on LIN-verkon viestikehys, joka jakautuu kahteen osaan, otsikkoon (header) ja vastaukseen (response). Otsikkoon kirjoittaa ainoastaan isäntäyksikkö ja vastaukseen renkiyksikkö. Otsikko osio koostuu kolmesta kentästä, tauosta (break), synkronoinnista (sync) sekä tunnisteesta (identifier). Vastaus-osio pitää sisällään kaksi kenttää, datakuorma sekä tarkistussumma (checksum) [18].

Jokainen kehys alkaa tauolla (break) ,joka koostuu 13:sta dominantista bitistä ja yhdestä resessiivisestä bitistä, joka toimii rajoittajana. Tämän kentän tehtävänä on ilmoittaa solmuja saapuvasta viestistä. Tauko-kenttä toimii kuten SOF kenttä perinteisessä CAN-viestissä [18].

Tauko-kenttää seuraa synkronointikenttä, jonka tehtävänä on synkronoida renkiolosmut yhteneväiseksi väylän siirtonopeuden kanssa. Synkronointikenttä on määritelty heksadesimaaliluvuksi x55 [18].

Otsikon viimeinen kenttä on tunniste. Jokaisella isäntäsolmun verkkoon lähettämällä viestillä on oma tunnisteensa. Se määrittää kuuluuko tietyn renkiolosmun vastata viestiin tai vaihtoehtoisesti lukea viestin vastausosion datakuorma. Jokainen renkiolosmu lukee jokaisen lähetetyn tunnisteiden ja laskee pariteetin, jonka perusteella määäräytyy, onko kyseinen solmu lähettäjä vai lukija tälle kyseiselle tunnisteelle. LIN-väylässä on kokonaisuudessaan 64 eri tunnistetta. Ensimmäiset 0-59 bittiä käytetään signaalien kuljettamiseen, bitit 60 ja 61 ovat tarkoitettu diagnostiikka datan siirtämiseen, bitti 62 on varattu käyttäjähajautusten laajennusten varalle ja bitti 63 on varattu tulevaisuuden protokollaparannuksille [18].

Väylään tunnistekenttä lähetetään yhtenä suojattuna tunnistetavuna, jossa alemmat kuusi bittiä sisältävät raakatunnisteiden ja ylempät kaksi bittiä ovat pariteettibittejä. Pariteetti lasketaan siten, että tietyistä bittiryhmistä lasketaan resessiivisten bittien määrä. Pariteettibitit lasketaan LIN-väylässä kaavan 2 mukaisesti.

$$P(0) = ID(0) \oplus ID(1) \oplus ID(2) \oplus ID(4)$$

$$P(1) = ID(1) \oplus ID(3) \oplus ID(4) \oplus ID(5) \quad (2)$$

Viestikehyksessä seuraavana on renkiasemien vastausosio. Tässä osiossa lähetetään varsinainen kehyksessä liikkuva data. Kun isäntäyksikkö on lähettänyt otsikon, oikealla tunnisteella varustettu renkiyksikkö vastaa kirjoittamalla viestinsä dataosioon. Dataosio sisältää yhteensä yhdestä kahdeksaan tavua datakuormaa [18].

Datakuormaa seuraa vastauksen ja koko viestikehyksen viimeinen osio, tarkastussumma. Tarkastussumma luodaan toisella LIN-väylän kahdesta tarkastussumma-algoritmista. Klassinen tarkastussumma lasketaan summaamalla datakuormat tavut yhteen. Parannettu tarkastussumma taas lasketaan summaamalla datakuorman tavut sekä suojattu tunniste [18].

3.4 FlexRay

FlexRay -väylä on kehitetty vastamaan nykypäivän väylien tarpeisiin. Myös FlexRay väylä on kehitetty kasvavien kaistanleveysvaatimusten ja ennen kaikkea reaaliaikavaatimusten myötä. FlexRay -väylää ei ole kuitenkaan tehty korvaamaan CAN- ja LIN-väylää, vaan ratkaisemaan nykyisiä korkean tason sovellusten aiheuttamia ongelmia. Vanhat väyläteknikat ovat sopivia suorittamaan alemman tason sovelluksissa, joissa niiden kapasiteetti riittää.

FlexRay-väylän tavoitteena on tarjota korkeaa tehokkuutta vaativissa ympäristöissä. Sen fyysinen kerros koostuu asemista, jotka ovat yhdistetty väylään kierteitetyllä parikaapelilla. FlexRay tukee yksi- ja kaksikanavaista konfiguraatiota. Parikaapelin vastakkaissignaalin tarkoituksena on vähentää ulkoisen kohinan vaikutusta, jolloin ei ole tarvetta kalliiseen kohinansuojaukseen. Kaksikanavaisen kytkennän on tarkoitus tarjota parempaa vikasetokykyä, sekä lisätä kaistanleveyttä. Myös FlexRay-väylä tarvitsee terminaattorit väylän päätepisteisiin. Terminaattoreina toimii vastus, joka on kytketty parikaapelin väliin. FlexRay-väylän kaapelin impedanssi on 80-110 ohmia. Terminaattorit päätetään vastamaan tätä arvoa. Liikaa tai liian vähän terminaattoreita voi aiheuttaa verkoille vahinkoa [19].

FlexRay tukee monia verkkotopologioita. Ajoneuvon sähköjärjestelmän layout sekä väylän käyttöaste määrittelee sen, minkälainen topologia palvelee parhaiten ajoneuvon järjestelmiä. Oikeanlaisen topologian valinta säästää materiaaliskustannuksia sekä parantaa luotettavuutta ja tehokkuutta. Yleensä FlexRay-väylän kanssa käytetään väylä topologiaa, sillä se muistuttaa ajoneuvojen omaa johdotusta, mutta se toimii myös muunlaisilla topologioilla [19].

FlexRay protokolla on toiminnaltaan huomattavasti tehokkaampi verrattuna CAN-protokollaan. CAN-väylä on tyypiltään event-triggered protokolla, kun taas FlexRay, toimii time-triggered-protokollana.

Time-triggered protokollissa on otettu viestien törmäykset huomioon eri tavalla kuin event-triggered protokollissa. Väylätopologiaa käytettäessä vain yksi solmu voi kirjoittaa väylään kerrallaan. Mikäli kaksi solmua yrittää kirjoittaa väylään samalla hetkellä, on kyseessä kilpavaraus ja data korruptoituu. Näitä tilanteita varten CAN-väylä käyttää viestikehyksessään tunnisteosiota, jolla määritetään lähetetyn viestin prioriteetti-arvo verrattuna väylässä liikkuviin muihin viesteihin. Tämä tapa ei kuitenkaan takaa sitä, että viestit saapuvat perille tarkoin määritellyissä aikaikkunoissa. Tästä syystä FlexRay käyttääkin monien solmujen viestien hallitsemiseen TDMA -järjestelmää. Jokainen solmu FlexRay-verkossa on synkronoitu samaan aikaan. Tällöin jokainen solmu kirjoittaa väylään omalla vuorollaan. TDMA:n ansiosta FlexRay väylän on mahdollista taata varma ja paikkansa-pitävä datan kuljetus [19].

3.4.1 Kommunikaatiosykli

TDMA- tyyppisten verkkojen toiminta perustuu kommunikaatiosykliin, joka toistuu verkoin toiminnan aikana useasti peräkkäin. Kommunikaatiosykli jakautuu neljään osaan, jotka ovat staattinen segmentti, dynaaminen segmentti, symboli-ikkuna sekä verkon tauko-aika. Yhden syklin mitta on yleensä noin 1-5 millisekuntia ja tyypillisesti kommunikaatiosyklin pienin yksikkö on makrotick, joka on yleensä noin 1 mikrosekunnin mittainen [19].

Staattinen segmentti on osio, joka on varattu viestikehykselle. Jokainen solmu lähettää viestikehyksensä oman aikaikkunansa aikana. Segmentti jaetaan lisäksi vielä osioihin, joista jokaisessa on ennalta varattu paikka tapahtumapohjaiselle datalle. Kun tulee kunkin osion oma aika syklissä, ohjausyksiköllä on mahdollisuus lähettää dataa omassa osiossaan. Mikäli ohjausyksikkö ei lähetä dataa vuorossa olevalla kierroksella, on sen odotettava seuraavaa sykliä. Vaikka yksikkö ei jollain kierroksella lähettäisikään tietoa, ei muutkaan verkossa olevat yksiköt voi lähettää dataa kyseiselle yksikölle varatussa osiossa. Tästä syystä järjestelmässä tiedetään tarkkaan kuinka vanhaa mikäkin yksikköjen lähettämä data on [19].

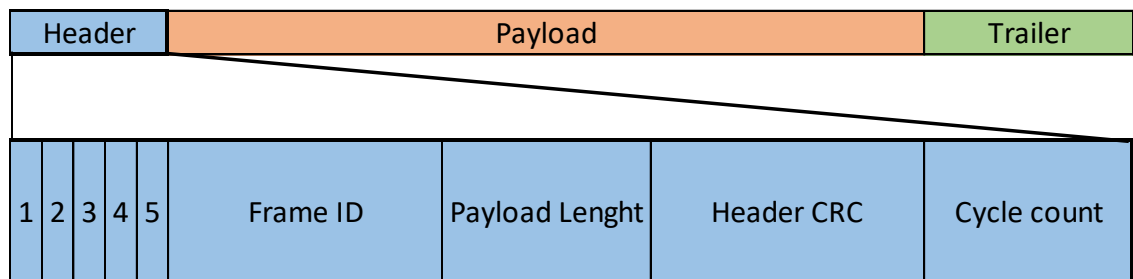
Dynaaminen segmentti koostuu muuttuvamittaisista aikaikkunoista, minisloteista. Nämä ovat tarkoitettu satunnaiseen tiedonsiirtoon. Minislottit järjestyvät lähetettävän datan prioriteetin mukaan. Korkeimman prioriteetin viestit lähtevät luonnollisesti ensin, ja mikäli tarpeeksi alhaisella prioriteetilla oleva viesti ei mahdu mukaan, se jää odottamaan seuraavaa kommunikaatiosykliä [19].

Seuraavat kaksi segmenttiä ovat symboli-ikkuna sekä verkon joutoaika. Symboli-ikkunaa käytetään pääosin huoltoon ja erikoissykliin tunnistamiseen. Suurin osa korkean tason

sovelluksista ei ole symboli-ikkunan kanssa vuorovaikutuksessa. Verkon joutoaika on ohjausyksiköiden ennalta asettama aikaikkuna, jonka aikana ohjausyksiköt voivat tehdä muutoksia säätöihin edellisen syklin lähettämän datan perusteella [19].

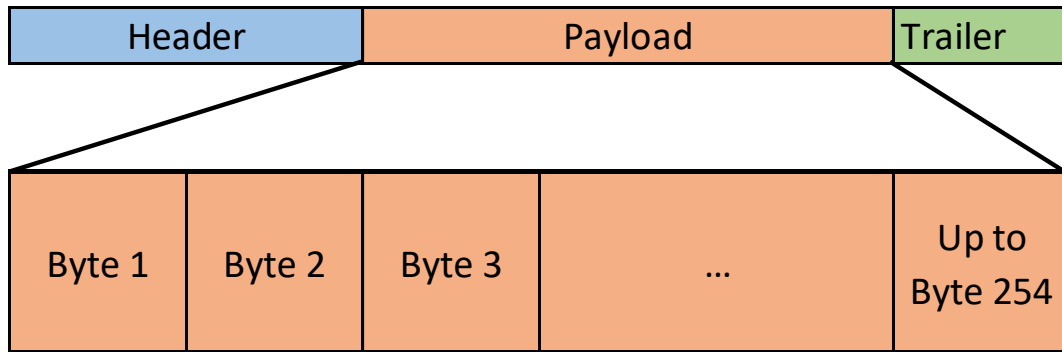
3.4.2 FlexRay viestikehys

Jokainen osio staattisessa ja dynaamisessa segmentissä sisältää FlexRay -viestikehysen. Viestikehykset jakautuvat kolmeen eri osioon: otsikkoon (header), datakuorma (payload), sekä loppuosa (trailer).



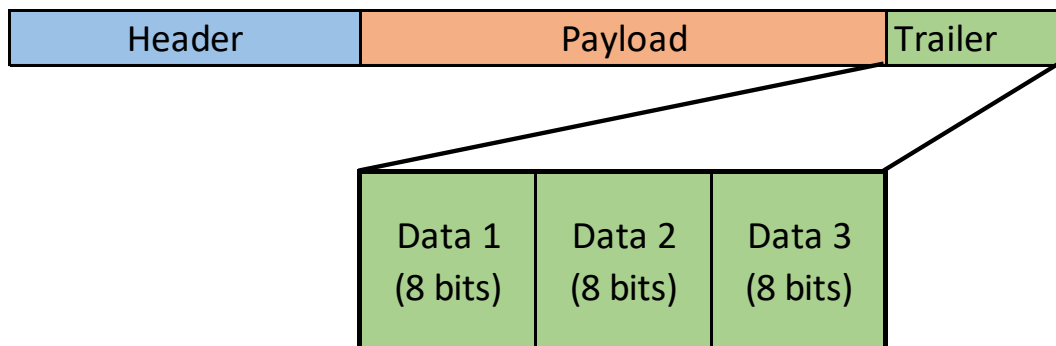
Kuva 12. FlexRay-viestikehysen otsikko.

FlexRayn viestikehysen otsikko-osio koostuu kuvassa 12 esitetyistä osioista. Ensimmäiset viisi bittiä ilmaisevat kehysen tyypin. Seuraavana oleva Frame ID, kooltaan 11 bittiä, määrittelee aikaosion, jossa kyseinen kehys tulee lähettää. Lisäksi Frame ID:a käytetään myös priorisoimaan tapahtumapohjaisia viestejä. Payload Length (7 bittiä) ilmaisee datakuormassa lähetettävien sanojen määrän. Header CRC (11 bittiä) on virheiden havaitsemista varten. Viimeisenä on cycle count -osio (6 bittiä) joka laskee suoritettujen kommunikationsykliden määrän [19].



Kuva 13. FlexRay-viestikehyksen datakuormaosio

Datakuorma osiossa sijaitsee itse lähetettävä data, kuten kuvassa 13 on esitetty. Datakuorma osion koko on yhteensä 254 tavua [19].



Kuva 14. FlexRay viestikehyksen trailer-osio

Kuvassa 15 on esitetty FlexRay viestikehyksen trailer-osio. Trailer-osiossa on kolme tavua, jotka toimivat virheentarkastusta varten kuin CRC [19].

3.5 Media Oriented System Transport (MOST)

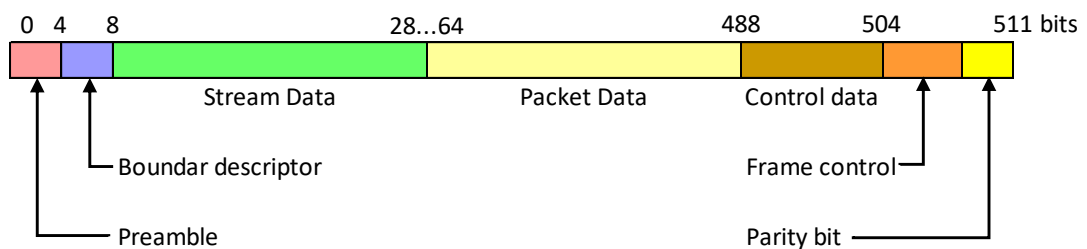
Autojen mediajärjestelmät ovat kehittyneet takavuosien tavallisista 1din autoradioista ja fyysisistä musiikkitalenteista laajoihin informaatio ja mediajärjestelmiin, jotka ovat jatkuvassa vuorovaikutuksessa käyttäjän kanssa. Nykyajoneuvojen niin kutsutut infotainment-järjestelmät sisältävät monesti enemmän ominaisuuksia kuin monella on AV-järjestelmissä kotonaan. Tätä kaikkea media- ja infodatan virtaa on suunniteltu ohjaamaan MOST-väylä, joka tulee sanoista Media Oriented Systems Transport. MOST-väylä on kehitetty vuonna 1998 ja ensimmäinen MOST-väylää käyttävä auto julkaistiin vuonna 2001. Tänä päivänä MOST-väylää käyttää jo yli 100 ajoneuvomallia.

MOST-väyläprotokollaa on saatavana kolmena eri versiona. MOST25 tekniikkaa käytetään usein eurooppalaisilla ja korealaisilla markkinoilla. Japanilaiset ja amerikkalaiset markkinat taas luottavat MOST50 protokollaan. Uusin MOST150 protokolla on esitelty vuonna 2007 ja sitä käyttivät ensimmäisenä Audi ja Daimler [20].

Protokollien nimet tulevat väylien kaistanleveydestä. MOST25 tarjoaa 25 Mbit/s kaistanleveyden ja kehyksen, jonka koko on 512 bittiä. MOST25 käyttää fyysisessä kerroksessaan optista kaapelia. MOST50 tarjoaa siirtonopeudeksi nimen mukaan 50 Mbit/s ja 1024 bittisen kehyksen. Molemmat ovat taajuudeltaan 44,1 kHz. MOST50-protokollaa varten suunniteltiin myös sähköinen kaapelointi. MOST150 pystyy 150 Mbit/s nopeuksiin ja toimii myös optisilla sekä sähköisillä kaapeleilla. Fyysisen kerroksen väylänä käytetään polymeerisiä optisia kaapeleita (Polymer Optical Fiber, POF), joissa johdin on valmistettu polymetyyliakrylaatista (PMMA), eli puhekielellä akryylista. Kaapeli koostuu yhden millimetrin paksuisesta akryyliytimestä ja eristeestä. Lähettimenä toimii väriltään punainen LED ja signaalin vastaanottimena silikoninen fotodiodi. Optinen fyysinen kerros pystyy ainoastaan noin 25 Mbit/s nopeuksiin, jonka takia sitä käytetään lähinnä vain MOST25-protokollan kanssa. Optisella siirtotekniikalla toteutetut suuremmat kaistanleveydet eivät ole vielä sillä tasolla, että niitä voitaisiin luotettavasti hyödyntää [20].

Perinteinen MOST-verkon topologia on malliltaan rengas, joka lähettää dataa laitteelta toiselle. Verkon topologiana käytetään usein myös tähtitopologiaa. Tämä mahdollistaa solmujen helpon lisäämisen ja poistamisen verkkoon ilman että loppu verkko häiriintyy. Mikäli verkko halutaan pitää mahdollisimman toimintavarmana mahdollisten virheiden sattuessa, tuplarengastopologia on hyvä vaihtoehto. Tuparengastopologiassa solmut ovat liitetty toisiinsa kahdella optisella kaapelilla ja jokaisessa solmussa on kaksi optisen signaalin vastaanotinta. Mikäli virhe kahden solmun välillä ilmenee, rengas voidaan sulkea [20].

Jokaisella kolmella eri MOST-protokollalla on erilaiset viestikehykset. MOST25 kehys koostuu 512 bitistä eli 64 tavusta. 60 tavua käytetään stream-datan tai pakettidatan kuljettamiseen. Kaksi tavua kuljettaa yhden osan hallintaviestistä, joka koostuu kokonaisuudessaan 32 tavusta. Hallintaviesti lähetetään 16 kehyksen mukana, joista se kootaan yhdeksi lohkoksi. Ensimmäinen ja viimeinen tavu sisältää hallintatiedon kehykselle [20].



Kuva 15. MOST25-protokollan kehys

Kuvassa 15 on esitetty MOST25 protokollan kehys. Ensimmäisenä MOST25-protokollan kehyksessä on johdanto-osio, eli preamble. Sitä käytetään renkisolmujen synkronointiin bittivirtaa varten. Väylässä on vaihelukittu-silmukka -ohjauspiiri (Phase-locked loop, PLL), jonka tehtävänä on pitää tulotaajuuden vaihe vakiona. Renkisolmut käyttävät PLL:ia synkronoidakseen itsensä verkkoon. Isäntäsolmu luo johdannon värähtelytaajuu-

den perusteella tai käyttöliittymältä tulevan käyttäjän määrittelemän tulosignaalin perusteella. Kuljettuaan kaikkien solmujen läpi takaisin isäntäsolmulle, bittivirtaan on muodostunut vaihe-ero, jonka aiheuttaa jokaisessa solmussa oleva etenemisviive. Isäntäsolmu synkronoi itsensä seuraavaa kehystä varten PLL:in avulla. Tämän jälkeen luo viestin uudelleen ja kompensoi vaihe-eron [20].

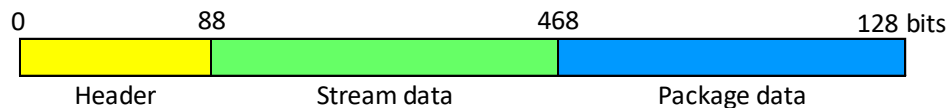
Striimausdatakanava ja pakettidatakanava jakavat kokonaisuudessaan kehyksessä saatavilla olevat 60 tavua. Näiden kahden kanavan kaistanleveydet adaptoidaan vastaamaan toistensa vaatimuksia rajakuvaajan (Boundary Descriptor) avulla. Rajat molempien kanavien välille voidaan vaihtaa quadletin, eli neljän tavun askelissa. Striimauskanavan kaistanleveys voi vaihdella välillä 24 – 60 tavua ja pakettidatakanava 0 – 36 tavua. Rajakuvaajan arvo itsessään voi vaihdella kuudesta 15 tavuun. Kaistanleveys kanaville laskeaan kaavalla 3 [20]:

$$\text{pakettidata kaistanleveys} = \text{kokonais kaistanleveys} - (\text{rajakuvaaja} * 4)$$

$$\text{striimauksen kaistanleveys} = (\text{rajakuvaaja} * 4)$$

$$\text{pakettidatan kaistanleveys} + \text{striimauksen kaistanleveys} = 60 \quad (3)$$

Kehyksen lopussa viimeinen tavu on kehyksen hallintaa varten ja viimeinen bitti, eli pariteettibitti, on virheen havaitsemista varten.



Kuva 16. MOST50 viestikehys

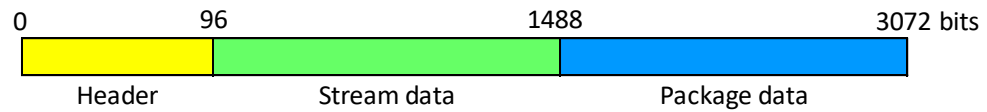
Kuvassa 16 on esitetty MOST50-protokollan viestikehys. Koska protokolla käyttää siirtonopeutena 50mbps sen kaistanleveys on kaksinkertainen verrattuna MOST25-protokollaan. Näin ollen kehyksen mitta kasvaa 1024 bittiin eli 128 tavuun. MOST50-viestikehys eroaa huomattavasti MOST25-kehyksestä. Kehys jakautuu kolmeen osaan, otsikkoon, striimausdataosioon ja pakettidataosioon, muistuttaen hieman jopa LIN-viestikehystä. Otsikko sisältää toiminnan kannalta kaiken olennaisen hallintadatan. Otsikko on kooltaan 11 tavua. Hallintadata vie otsikosta 4 tavua. Lisäksi otsikko sisältää rajakuvaajan sekä System lock flag:n. Striimausdatan osuus kehyksestä voi olla nolasta 117 tavua ja pakettidatan nolasta 116 tavuun. Kaistanleveydet molemmille kanaville voidaan laskea kaavan 4 avulla [20].

$$\text{pakettidata kaistanleveys} = \text{kokonais kaistanleveys} - (\text{rajakuvaaja} * 4) - 1$$

$$\text{striimauksen kaistanleveys} = (\text{rajakuvaaja} * 4) + 1$$

$$\text{pakettidatan kaistanleveys} + \text{striimauksen kaistanleveys} = 117 \quad (4)$$

Kuvasta 18 huomataan että MOST150-protokollan kehys on rakenteeltaan samanlainen kuin MOST50. Nopeamman siirtonopeuden takia kehyksen koko on huomattavasti pienempi, jopa 3072 bittiä [20].



Kuva 17. MOST150 viestikehys

Erona MOST50 protokollaan uusimman protokollan otsikkokenttä sisältää myös preamble-bitit sekä shutdown flag -bitit. Myös MOST150-kehyksessä pystytään muokkaamaan kehyksen kaistanleveyttä. Striimaukskanavan koko voi olla nolasta tavusta 372 tavuun, kuin myös pakettidatakanavan. Kehyksen suuruus mahdollistaa myös molempien kanavien lähettää täydellä kaistanleveydellä. Kaava 5 esittää kuinka kaistanleveydet on mahdollista laskea molemmille kanaville [20].

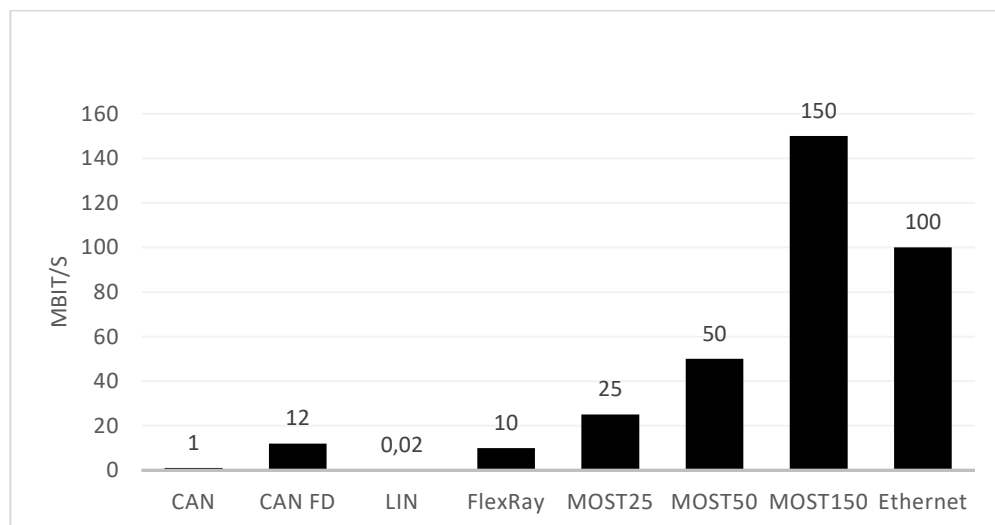
$$\text{pakettidata kaistanleveys} = \text{kokonais kaistanleveys} - (\text{rajakuvaaja} * 4)$$

$$\text{striimauksen kaistanleveys} = (\text{rajakuvaaja} * 4)$$

$$\text{pakettidatan kaistanleveys} + \text{striimauksen kaistanleveys} = 372 \quad (5)$$

3.6 Siirtonopeus ja kaistanleveys

SAE on kehittänyt luokituksen ajoneuvojen tiedonsiirtoprotokollille. Luokittelu perustuu datan siirtonopeuteen sekä toimintoihin, joita verkon avulla suoritetaan. Kuvassa 18 on esitetty tässä työssä tutkittujen protokollien siirtonopeudet.

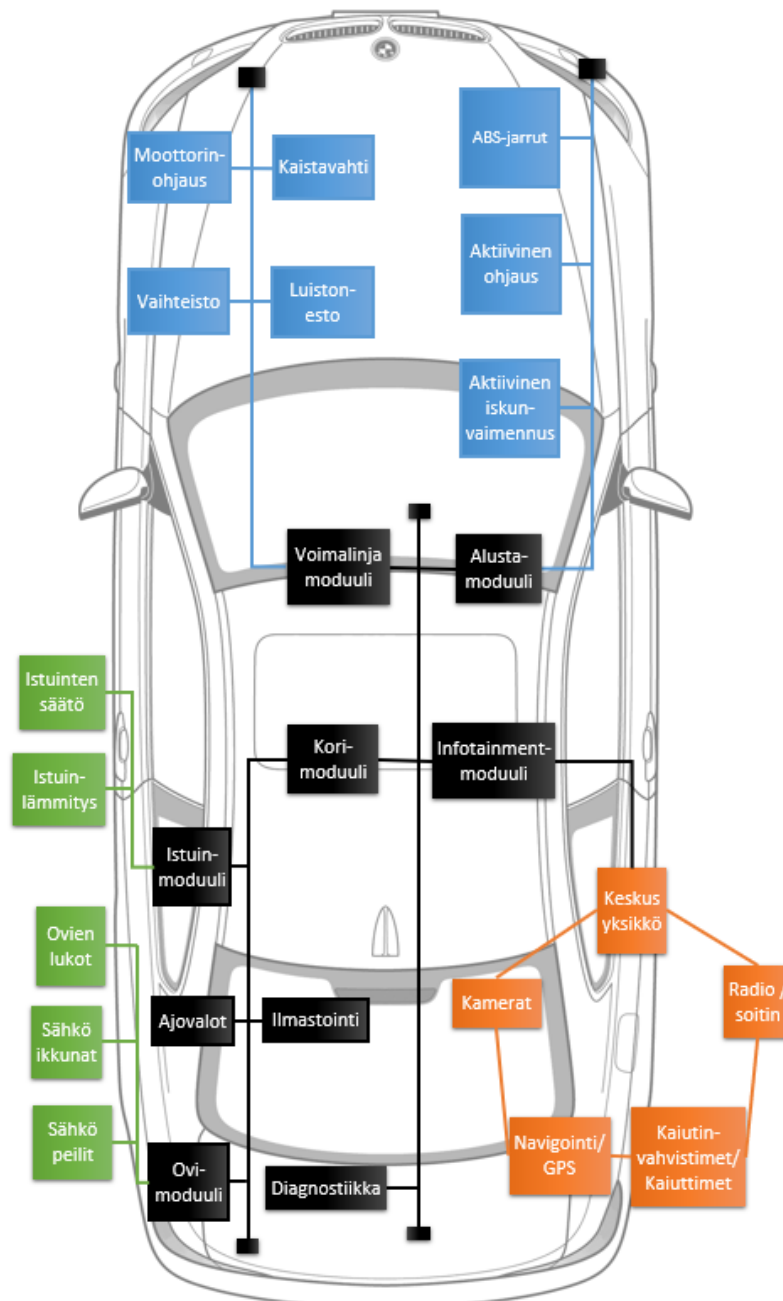


Kuva 18. Diagrammi protokollien siirtonopeuksista

Luokka A sisältää verkkoprotokollat, joiden siirtonopeus on alle 10 Kbit/s. Tämän luokan väylät ovat suunniteltu siirtämään yksinkertaista kontrollidataa, joka yleensä liittyy ajoneuvon korin sovelluksiin. Nämä verkot ovat myös yleensä alhaisempaa kustannustasoa kuten LIN-väylä. Luokan B protokollien on tarkoitus toimia ohjausyksiköiden välisessä tiedonsiirrossa, ettei samaa toimintoa tarvitsisi anturoida useaan kertaan. Tämän luokan väylien siirtonopeudet asettuvat väylille 10 Kbit/s – 125Kbit/s. Tällaisia on esimerkiksi CAN low -protokolla. Luokan C väylien siirtonopeudet ovat välillä 125 Kbit/s – 1Mbit/s. Luokan D väylät taas pystyvät 1Mbit/s siirtonopeuteen. Luokkaa D ei ole virallisesti määritelty, mutta yleisenä oletuksen on ollut, että yli yhden 1 Mbit/s siirtonopeuden omaavat väylät kuuluvat siihen. Molempien luokan väyliä käytetään, kun järjestelmän vaatimuksena on nopea reaaliaikainen yhteys. CAN high-protokolla lukeutuu C-luokan väyläksi. Kyseisiä väyliä käytetään ajoneuvojen alustan toimintojen ohjauksessa. D-luokan väyliä käytetään multimediadatan siirtoon ja korkean turvallisuusvaatimuksen, korkean vian-sietokyvyn omaavien applikaatioiden toteutuksessa [16].

3.7 Protokollien käyttö ajoneuvoissa

Ajoneuvo on monimutkainen järjestelmä, joka voidaan jakaa useaan eri alijärjestelmään, jotka toimivat itsenäisesti, hoitaen oman osuutensa kokonaisen järjestelmän toiminnassa. Kokonaisjärjestelmä koostuu esimerkiksi alustasta, korista, moottorista, voimansiirrosta turvajärjestelmistä ja media- sekä infojärjestelmästä. Jokaisella järjestelmällä on omat ohjausyksikkönsä eri toimintoja varten.



Kuva 19. Tyypillinen ajoneuvon verkkoarkkitehtuuri. Muokattu lähteestä [21].

Kuvassa 19 on esimerkki tyypillisestä ajoneuvon verkkoarkkitehtuurista, mitä tehtäviä eri verkkoprotokollilla nykyteknologian mukaisella ajoneuvolla on. Mustalla kuvatut väylät

sekä moduulit ovat perinteistä CAN-väylää. Alustamoduuliin ja voimalinjamoduuliin yhdistyvät siniset väylät ovat FlexRay-väylää. Vihreä istuinmoduuliin ja ovimoduuliin kiinnittyvä väylä on LIN-väylää ja infotainmentmoduuliin kiinnittyvä rengastopologinen oranssi väylä on MOST-väylä [21].

Alusta koostuu komponenteista, joiden tehtävänä on mahdollistaa autolla ajaminen ja pitää auto tiessä. Alusta tukee myös voimalinjaa ja tekee osansa moottorin tuottaman tehon liike-energiaksi muuttamiseen. Alusta on myös muutakin kuin mekaanisia komponentteja. Anturoinnit ja sähkökomponentit ovat osa modernien autojen alustaa. Näistä esimerkkeinä ajettavan tien mukaan automaattisesti säätyvä iskunvaimennus tai jarrujen huollontarvetta tarkastelevat anturoinnit [1, 22, 23]. Edellä mainittuja ominaisuuksia varten parhaiten vastaava väylä, on CAN. CAN on moneen taipuva väylätekniikka, jolla voidaan toteuttaa sekä nopeutta vaativia verkkoja (CAN high) taikka viivettä sietävämpiä järjestelmiä (CAN low). Lisäksi CAN ei tarjoa kovinkaan paljoa kaistanleveyttä, mutta tarpeeksi edellä mainituille sovelluksille.

Voimalinjan tehtävänä on siirtää moottorin tuottama teho vaihteiston ja renkaiden välityksellä tiehen. Voimalinjaa hallitaan useiden ohjausyksiköiden ja anturien kautta. Esimerkiksi päästöjä mitataan ja seurataan antureilla, jotta moottorin toiminnasta saadaan mahdollisimman puhdasta ja taloudellista. Antureilta saadun tiedon mukaan ohjausyksiköt säätelevät ja ohjaavat moottorin ja vaihteiston toimintaa. Moderneissa ajoneuvoissa voimalinjan käyttäytymistä on mahdollista muuttaa ajon aikana esimerkiksi taloudellisempaan tai tehokkaampaan suuntaan. Kyseiset toimenpiteet vaativat yleensä tapahtumia, joiden tulee tapahtua mikrosekunneissa. FlexRay toimii reaaliaikaisena suurella kaistanleveydellä varustettuna väylätekniologiana voimalinjan tiedonsiirrossa [1, 23].

Ajoneuvon kori muodostaa myös oman järjestelmänsä. Korin järjestelmä kattaa yleensä mukavuuspainotteiset sovellukset, kuten sähköistuimet, sähköikkunat, valot, sähköiset luukut ja niin edelleen. Kyseiset sovellukset eivät vaadi niin nopeita toimenpiteitä, kuin esimerkiksi voimalinjassa toimivat apulaitteet. Reaaliaikavaatimukset ovat millisekunnin luokkaa. Myös liikkuva datakuorma on pienempi. Kyseisissä sovellutuksissa toimivat anturit ja toimilaitteet pärjäävät pienemmällä kaistanleveydellä. LIN- väylä on kustannustehokas ja kaistanleveydeltään pieni väylä vastaamaan optimoidusti korin sovellusten tarpeisiin [1, 22].

Modernien ajoneuvojen yksi tunnusomaisimmista piirteistä on niiden viihdejärjestelmät, eli niin kutsutut infotainment-järjestelmät. Infotainment-järjestelmä on käyttöliittymä auton järjestelmien ja käyttäjän välillä. Järjestelmällä ohjataan auton medialaitteita, sekä luetaan auton järjestelmien tarjoamaa infodataa. Mediajärjestelmiin pystytään myös liittymään käyttäjien langattomilla laitteilla, joita voidaan käyttää auton oman medialaitteiston jatkeena. Kehittyneimpien järjestelmien avulla pystytään jopa lukemaan vikakoodeja ja diagnosoida vikaa etänä. Infotainment-järjestelmät vaativat suuren kaistanleveyden, josta esimerkkinä peruutuskameroiden lähettämän videodata, mutta eivät ole niin vaativia

viiveen suhteen vaan riittävä viive on millisekuntien tarkkuudella. Infotainment datansiirtoa varten on kehitetty väyläprotokolla MOST [1, 22].

Kuten edellä mainituista esimerkeistä huomataan, eri alijärjestelmillä on hyvin erilaiset tarpeet tiedonsiirron suhteen. Siinä missä ajoneuvon voimalinja tarvitsee reaaliaikaista ja suurella kaistanleveydellä varustettua FlexRay-tiedonsiirtoväylää mahdollistamaan toimivuuden, ajoneuvon korin applikaatiot pärjäävät pienemmällä siirtonopeudella ja tapahtumaliipaisutyypisellä LIN-tiedonsiirtoverkolla. Jokainen verkko siis on suunniteltu tarkoituksen mukaan.

4. TIEDONSIIRTOVERKKOJEN TULEVAISUUS

4.1 Nykyisten teknologioiden ongelmat

Nykypäivän ajoneuvojen automaatio luo koko ajan kasvavan määrän dataa, jota siirretään ajoneuvossa itsensä ja sen oheislaitteiden toiminnan varmistamiseksi. Tämä johtaa siihen, että lähitulevaisuudessa väylien suhteen on saavutettava korkeampia tasoja datan siirto-kyvyn näkökulmasta. Nykyinen suuntaus on ollut se, että sen sijaan että olisi lähdetty kehittämään uutta tiedonsiirtoteknologiaa, on laajennettu nykyisiä sekä käytetty nykyisiä verkkoja päällekkäin [8]. Tällä hetkellä ajoneuvossa voi olla jopa viisi eri tiedonsiirtoverkkoa hoitamassa omia tehtäviään.

Useaa verkkoa käyttävästä sovelluksesta on hyvänä esimerkkinä esimerkiksi kaista-assistentti. Kaista-assistentin pääkomponentti on taustapeiliin tai muualle ajoneuvon etuosaan sijoitettu eteenpäin suunnattu kamera, joka tarkkailee auton kulkemista menosuuntaan. Ohjelma lukee kameran välittämästä kuvasta kaistaviivoja, ja pitää huolen, että auto pysyy niiden välissä. Mikäli auto ajautuu liian lähelle reunaviivaa, järjestelmä antaa äänimerkin ja haptisen palautteen ohjauspyörän kautta. Mikäli kuljettaja ei näidenkään jälkeen tee muutosta ohjaukseen, järjestelmä kääntää hienoisesti ohjauspyörää siihen liitettyllä moottorilla ohjatakseen auton takaisin reunaviivojen väliin [24]. Kyseinen järjestelmä käyttää vähintään kahta, jopa kolmea tiedonsiirtojärjestelmää. Ohjauspyörän liikkeen toteuttavat anturit ja moottorit ovat liitettynä CAN-väylään. Kamera taas täytyy olla yhdistettynä esimerkiksi MOST-väylään, jotta videokuva tiestä saadaan välitettyä järjestelmän ohjausmoduulille. Lisäksi äänimerkit ja haptiset signaalien toimilaitteet voivat olla liitettynä LIN-väylään.

Useiden tietoverkkojen päällekkäisyyksien takia, sekä kasvavan toimilaitemäärien takia johdotusten määrä kasvaa myös, mikä aiheuttaa turhan suurta painonlisäystä ajoneuvon kokonaispainoon. Materiaalin kulutusta lisää myös päällekkäisten verkkojen väyläkaapeleiden suojaus. Osa verkoissa ilmenevistä ongelmista johtuu verkoista itsestään. Johdotusten lisäys aiheuttaa materiaalikuluja mutta myös kasvava paino saa ajoneuvon polttoaineen kulutuksen kasvamaan [8].

Kuten aiemmassa kappaleessa tuli ilmi, CAN-väylän yksi ongelma on sen rajoittunut kaistanleveys. Sen kaistanleveydellä 125 kbit/s – 1Mbit/s pystytään lähettämään vain sovellusten toiminnan kannalta välttämättömimmät viestit. Lisäksi väylän kilpavaraus-toimintaperiaatteen takia siihen ei voida lisätä määräänsä enempää solmuja [8].

CAN-väylän yksi riittämätön ominaisuus on sen taipumus ruuhkautumiseen. Solmujen määrä lisää suorassa suhteessa väylässä kulkevien viestien määrää. Viestien määrän kasvua lisää myös se, että CAN-viesti on kapasiteetiltaan lyhyt, maksimissaan 130 bittiä.

Tällä viestin pituudella 1 Mbit/s siirtonopeus on riittävä, mutta mikäli kyseisten viestien tulisi päästä perille täysin ilman törmäyksiä, ei maksimimittaisia viestejä voida käyttää [8].

Perinteinen CAN-väylä ei pysty vastaamaan sovellusten reaaliaikaisuuden tarpeisiin. Kun lähetetään paljon pitkiä viestejä, törmäyksiä tapahtuu ja viestin siirto kestää pidempään. CAN- väylän viive voi olla jopa 10ms, jolloin reaaliaikaisuus ei toteudu. Tulevaisuuden tavoitteessa autonomisessa ajamisessa tämä on auttamatta liian hidasta [8].

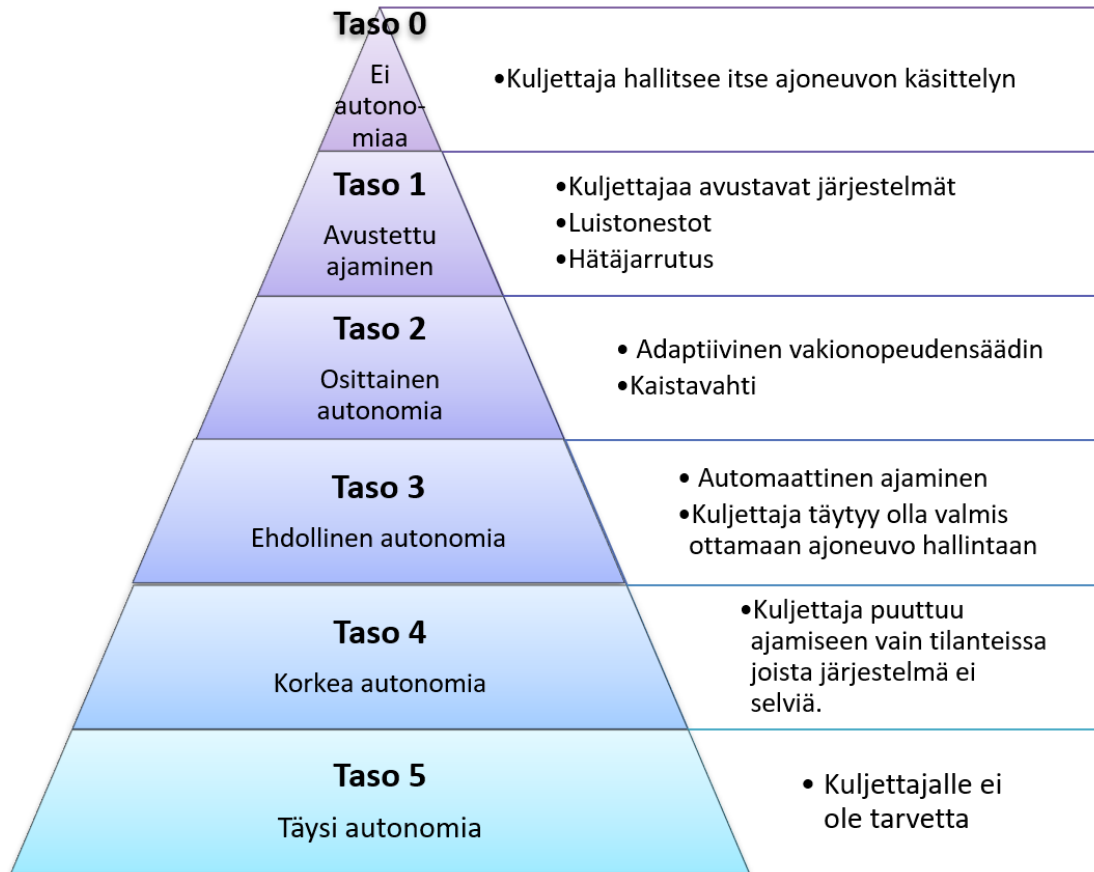
4.2 Tulevaisuuden teknologiat

Yksi suurimmista väylille haasteita asettavista sovelluksista on autonomiset ajoneuvot. Ajoneuvoteollisuus on kovaa vauhtia menossa kohti itsestään ajavia ajoneuvoja. Nykyajoneuvot sisältävät jo monia autonomisen ajamisen esiasteita olevia järjestelmiä:

- Kehittyneet ajonvakautusjärjestelmät (Advanced Driver Assistance Systems, ADAS)
- Liikennemerkkien tunnistus -järjestelmä
- Kaistavahti sekä kaista-assistentti (Lane Keeping Assist System, LKAS)
- Parkkeerausassistentti

Edellä mainitut sovellukset käyttävät reaaliaikaista videodataa. Dataa kuljettamaan tarvitaan väylä, joka pystyy turvallisesti ja virhevarmasti siirtämään datan reaaliajassa kameralta ohjasyksikölle. Mitä edemmäksi mennään, sitä vaativimmaksi nämä järjestelmät kehittyvät, ja lopulta yhdistyvät luoden autonomisesti ajettavan järjestelmän.

Autonomiset autot ovat tulevaisuuden ajoneuvoteknologian seuraava suuri tavoite, ja se asettaa omat tavoitteensa myös väyläteknologialle. Autonomisen ajamisen voi jakaa kuvan 20 mukaisesti viiteen eri autonomisen ajamisen tasoon.



Kuva 20. *Autonomisen ajamisen tasot.*

Tasolla 0 ajoneuvossa ei ole ollenkaan autonomiia. Kaikki auton hallintaan liittyvät toimenpiteet ovat kuljettajan vastuulla. Tämä ei kuitenkaan koske automaattisia turvallisuussovelluksia. Ensimmäistä tasoa kutsutaan avustetuksi ajamiseksi. Ensimmäisellä tasolla ajoneuvoissa on ajotiloja, joiden aikana järjestelmä auttaa kuljettajaa pitämään auton hallinnassaan ja hetkellisesti jopa kuljettaa autoa itsestään. Tästä esimerkkinä Volkswagenin ensimmäisenä lanseeraama parkkeeraus assistentti. Toisen tason autonomisuudessa auto ajaa jo itsestään. Kuljettaja voi antaa ajoneuvon kuljettamisen järjestelmien haltuun tietyissä tilanteissa. Kuljettajan täytyy kuitenkin olla valppaana ja ottaa ajoneuvon hallinta takaisin, mikäli tilanne niin vaatii. Teslan Autopilot-ajotila on hyvä esimerkki toisen tason autonomiasta. Kolmannen tason autonomiassa ajoneuvo ottaa täyden vastuun ajamisesta tietyissä tilanteissa. Kuljettajan täytyy olla silti paikallaan ja ottaa ajoneuvo takaisin hallintaan tilanteen niin vaatiessa. Ajoneuvo ajaa kuitenkin itsestään ja elävä kuljettaja on vain varotoimena. Neljäs autonomian taso on hyvin samanlainen kuin kolmas mutta siinä ajoneuvo osaa ajaa lähes autonomisesti. Se pyytää kuljettajaa ottamaan hallinnan vain erittäin vaikeissa tilanteissa, joista se ei itse selviä. Viidennen, eli viimeisen tason autonomiassa puhutaan ajoneuvon täysin itsenäisestä ajamisesta. Tässä tilassa ajoneuvossa ei tarvitse olla kuljettajaa, joka itse osaisi ajaa. Ajoneuvo suorittaa ajamiseen liittyvät tehtävät itse [25, 26].

Autonomisen ajamisen kehittyminen aiheuttaa vaatimuksia myös väyläteknologialle. Autonomiset autot ovat sisäisen tiedonsiirtoverkkonsa lisäksi yhdistetty pilveen, josta ajoneuvot vaihtavat informaatiota keskenään. Verkkojen täytyy olla luotettavampia ja tarjottava reaaliaikaisuutta. Verkkotopologia tulee olemaan enemmän nykyaikaisen Ethernet-tiedonsiirtoverkon kaltainen. Hierarkkinen-topologia, jonka perustana on nopea ja laaja-kaistainen Automotive Ethernet.

Suurta roolia tulee näyttämään myös tiedonsiirron nopeus. Monikameraiset järjestelmät ovat varustettu 4k tai 8k kuvantarkkuudella. Tarkka kuva tulee siirtyä tiedonsiirtojärjestelmän kautta reaaliajassa, jotta järjestelmän toiminnallisuus säilyy turvallisena. Esimerkkinä tiedonsiirron määrästä on tason viisi autonominen auto. Yhteydessä olevat autot (connected cars) ovat koko ajan vuorovaikutuksessa ympäristön kanssa. Ajoneuvojen yhteyttä kuvataan myös termillä Vehicle-to-everything (V2X). Niiden anturit ja kamerat keräävät dataa ympäröivästä maailmasta, ja myös autosta itsestään. Tämän informaation perusteella ajoneuvojen tekoäly tekee päätökset ajoneuvon ohjaamisesta [27, 28].

Ajoneuvot tulevat olemaan myös yhteydessä keskenään. Esimerkkinä kaistan vaihtaminen. Reittivalinnan mukaan ajoneuvon täytyy vaihtaa kaistaa vasemmalle, ja laittaa vilkun päälle. Vasemman puoleisella kaistalla onkin jo ajoneuvo, jolloin kaistaa vaihtava ajoneuvo ei voi siirtyä siihen varauksetta. Mikäli molemmat ajoneuvot ovat V2X-ajoneuvoja, kaistalla oleva ajoneuvo saa tiedon viereisen ajoneuvon kaistanvaihtoaikeistaan, ja voi mahdollisuuksien mukaan mukauttaa ajoaan siten, että kaistaa vaihtava ajoneuvo pysyy tekemään kaistanvaihdon. Lisäksi ajoneuvot voivat viestiä toisilleen esimerkiksi sääolosuhteista. Jotta edellä kuvatun kaltainen kommunikaatio olisi turvallista, sen tulee olla reaaliaikaista. Nykyiset langattomat matkapuhelinverkot voivat siirtää 30-40ms viiveellä, mikä on liian hidaskokoinen reaaliaikainen. Tulevaisuuden 5G -verkot pyrkivät 1 ms viiveeseen, joka riittäisi autonomisten ajoneuvojen tarpeisiin [28].

Tulevaisuuden tavoitteena on myös ajoneuvoissa käytettyjen ohjausyksiköiden vähentäminen. Nykypäivän autoissa voi olla yli 70 eri ohjausyksikköä kontrolloimassa sovelusta, jota varten se on asennettu [29]. Ongelman aiheuttaa se, että eri tiedonsiirtoteknologiat käyttävät antureita ja komponentteja, jotka ovat tarkoitettu ainoastaan tietylle teknologialle. Ajoneuvotekniikassa on prosesseja, joiden dataa tarvitaan kahden eri verkon applikaatioissa toiminnan kannalta. Tämä aiheuttaa sen, että samaa asiaa tarvitaan mittaamaan kaksi eri anturia, jotta saadaan data kahdelle eri tiedonsiirtoverkolle. Tämän ongelman on aiheuttanut ajoneuvojen järjestelmien heterogeenisuus. Lisäksi tarvitaan yhden anturin tilatiedon välittäminen kahdelle eri verkolle vähentäisi helpottaisi verkkojen suunnittelua. Lisäksi ratkaisu vähentäisi johdotuksen sekä käytettyjen antureiden kokonaisuutta. Myös pienempi materiaalmäärä keventää ajoneuvon lopullista painoa. Ohjausyksikköjen vähentämisen tarpeen luo myös niiden kustannukset. Ajoneuvon sähköjärjestelmä, on auton kolmanneksi kallein alijärjestelmä voimansiirron ja alustan jälkeen [30]. Suunta ajoneuvojen tiedonsiirtoverkkojen kehityksessä on tähän mennessä ollut

pääsääntöisesti se, että ei ole lähetty kehittämään uusia ratkaisuja, vaan on laajennettu ja kehitetty vanhoja tekniikoita. Esimerkkinä CAN ja CAN FD.

4.3 Fyysinen kerros tulevaisuudessa

Anturien määrä kasvaa ja jokainen anturi on liitettävä väylään omalla johdollaan. Kaapelointien määrä kasvaa ja lisää ajoneuvon painoa. Järjestelmien monimutkaisuus ajaa saman ongelman äärelle, jonka takia CAN-väylä alun perin kehitettiin. Painavaa kaapelointia on liikaa. Autovalmistajat yrittävät pienentää ohjausyksikköjen määrää, jopa sadasta alle kahteenkymmeneen.

Tulevaisuus aiheuttaa myös haasteita verkkojen fyysiselle kerrokselle. Perinteinen kuparijohdin ei tule olemaan pitkäikäinen ratkaisu. Kuparijohtimen hintataso, sekä paino nousevat päällimmäisiksi ongelmiksi sen käytössä. Kasvava järjestelmien kompleksisuus ja uusien ominaisuuksien lisääminen järjestelmään kasvattaa ajoneuvon kokonaispainoa. Lisäksi perinteisen kuparijohtimen ongelmana on sähkömagneettisen säteilyn aiheuttamat häiriöt. Häiriöitä varten johtimet on eristettävä asianmukaisella tavalla, joka lisää painoa entisestään. Häiriöt voivat aiheuttaa järjestelmässä myös vakavia vahinkoja aiheuttamalla ongelmia järjestelmien toiminnassa. Ongelmat voivat aiheuttaa kolarin ja vaarantaa jopa ajoneuvon kuljettajan ja matkustajan hengen [8].

Ympäristönäkökulma aiheuttaa myös tavoitteita kuparijohtimista luopumiselle. Kupari on luonnonvara, joka ei tule riittämään ikuisuuksiin. Elektroniikan lisääntyessä kysyntä kuparille kasvaa koko ajan.

4.3.1 Optinen kaapeli

Painon vähentämiseen on olemassa monia keinoja, mutta aina ei ole mahdollista vähentää yhteyksiä laitteistojen välillä. Tällöin joudutaan miettimään vaihtoehtoista kevyempiä vaihtoehtoja yhteyksien mahdollistamiseksi. Optinen datakaapeli on hyvä vaihtoehto perinteiselle kuparijohdinkaapelille. Sen korkea siirtokapasiteetti sekä keveys ovat huomattavia etuja. Lisäksi useampia signaaleja varten ei tarvita lisää kaapeleita, vaan yhdessä kaapelissa voidaan lähettää useaa signaalia samanaikaisesti käyttämällä aallonpituuskanavointia (wavelength-division multiplexing, WDM). Aallonpituuskanavoinnin käytössä tulee ottaa huomioon, että eri signaalien on käytettävä eri aallonpituuksia, etteivät signaalit kumoa toisiaan. Ajoneuvokäytössä kaapeleiden pituudet eivät kasva kovin suuriksi, joten monisignaalistien optisten kaapelien käyttö onnistuu ongelmitta [8].

Huonoina puolina optisessa kaapelissa on sen optisen signaalin lähetysyksikön vikaherkkyys. Mikäli lähetysyksikkö ja sen herkät optiset komponentit vikaantuvat, ei signaali kulje väylässä. Myös kaapelin liiallinen taivuttaminen voi saada optisen kuidun halkeamaan, jolloin signaalin eteneminen estyy myös. Optista kaapelia käytettäessä sen päitä

ei tarvitse terminoida, kuten kuparijohdinväylässä välttääkseen signaalin heijastumista [8].

Optisen kaapelin käytössä ei tarvitse huolehtia sähkömagneettisten kenttien aiheuttamista häiriöistä, koska kaapelissa ei ole metallia, eikä siellä kulje sähköä. Sähkömagneettisten kenttien aiheuttamat häiriöt tulevat olemaan ongelma tulevaisuudessa, mikäli autonomisesti ajavien autojen tiedonsiirto toteutetaan kuparijohtimilla.

Kuparijohdinten määrä vaihtelee ajoneuvovalmistajien välillä paljonkin, mutta keskimäärin johdotusten yhteispaino on n. 60-100kg ja pituudeltaan johtosarjat ovat noin kolme kilometriä. Saman pituinen määrä optista silica-valmisteista kaapelia painaa noin 33kg. Tämä tarkoittaa vähintään 50% painonkevennystä johdotusten osalta. Optisen kaapelin materiaalin vaihtaminen polymeeriin keventää kaapelointia vielä entisestään. Polymeeristä valmistetut kaapelit ovat myös paitsi kevyempiä, myös halvempia valmistaa. Lisäksi ne kestävät enemmän taivutusta kuin silica-kaapelit. Huonona puolena polymeerikaapelissa on sen taipumukseen aiheuttaa signaalin vaimenemista ja vääristymiä. Siksi kaapelien pituus tulee pitää lyhyinä, alle 200 metrisinä [8].

4.3.2 Langattomat tekniikat

Todellinen tulevaisuuden ratkaisu johdotuksen vähentämiseen on jättää ne pois kokonaan. Langattomat teknologiat ovat varteenotettavia teknologiaa ajoneuvon tiedonsiirrossa. Langattomuus antureiden ja ohjauksyksiköiden välillä toisi huomattavia säästökuuja materiaalikustannuksissa ja keventäisi ajoneuvon kokonaispainoa merkittävästi.

Wireless Fidelity (WiFi) on hyvin paljon käytössä jokapäiväisessä elämässä paikallisverkkojen yhteydessä. WiFi onkin korvannut suuren osan kaapeloinneista tavallisissa kotitalouksissa ja julkisissa tiloissa. WiFi on ollut mukana jo myös ajoneuvotekniikassa, jossa sen avulla on yhdistetty käyttäjän omia laitteita ajoneuvon järjestelmään kuten esimerkiksi älypuhelimia. WiFi:n maksimisiirtonopeus on jopa 600Mbit/s, jolloin se tarjoaa suuria määriä kaistaa datan siirtoa varten [22]. WiFi onkin varmasti yksi tulevaisuuden datansiirtokeino ajoneuvojen mediajärjestelmissä. WiFi toimiikin ehkä parhaiten yhdistäessä ajoneuvot keskenään, jolloin ne voivat jakaa tietoa esimerkiksi liikenteen sujuvuudesta ja sitä kautta tehdä valintoja navigoinnin suhteen.

Toinen langaton arkipäivästä tuttu teknologia fyysisten kaapeleiden korvaajaksi on Bluetooth. Se on kehitetty palvelemaan lyhyen matkan langattomaan viestintään, missä sen maksimisiirtonopeus on 3Mbit/s. Tällä hetkellä Bluetoothia käytetään ajoneuvoissa lähinnä audion siirtämiseen puhelimesta ajoneuvon infotainment-järjestelmän kautta ajoneuvon kaiuttimiin. Tällä Bluetooth 2.1 tekniikalla pystytään ajoneuvon audiojärjestelmää käyttämään kaiutinjärjestelmänä sekä handsfree-systeeminä. Uudempi Bluetooth LE (4.1) tekniikka kuluttaa vähempi energiaa ja on nopeampi sekä luotettavampi. Aiempi

Bluetooth 2.1 kykenee ainoastaan point-to-point yhteyksiin, kun taas uudempi 4.0 teknologia tukee mesh-topologiaa, jolloin sitä voidaan käyttää laajemmissa järjestelmissä [22]. Kyseisten ominaisuuksien valossa se on potentiaalinen tekniikka ajoneuvon alijärjestelmien kuten esimerkiksi alustan tai voimalinjan tiedonsiirtoverkon ohjausyksikköjen välille. Bluetooth 4.0 on lupaava tekniikka korvaamaan fyysisen johdotuksen komponenttien välillä ajoneuvojen tiedonsiirtoverkoissa [22].

Muita langattomia teknologioita fyysisen johdotusten korvaajaksi on Ultra-Wideband (UWB) ja ZigBee. UWB on kehitetty lyhyen kantaman langattomaan tiedonsiirtoon kuten Bluetooth. UWB teknologia perustuu siihen, että data lähetetään lyhyinä pulsseina laajalla kaistanleveydellä. UWB pystyy lähettämään dataa 480Mbit/s nopeudella kolmen metrin kantaman päähän ja 110Mbit/s nopeudella 10 metriin asti. Ongelmana tekniikalla on satunnaisin välein lähetetyt viestit. Viestejä lähetetään satunnaisella väliajalla, jotta pulssit eivät alkaisi resonoida läheisten rakenteiden kanssa. Ongelmana tekniikassa on nanosekuntien tarkkuudella lähetettyjen viestien vastaanotto [22].

ZigBee on suunniteltu kustannustehokkaaksi matalatehoiseksi langattomaksi yhteydeksi. Sen maksimilähetytnopeus on jopa 250 Mbit/s. ZigBee taipuu topologialtaan moneksi. Se tukee point-to-point -, tähti- ja mesh-topologioita. ZigBee-teknologiaa on tähän mennessä käytetty ainoastaan älytalojen yhteydessä [22].

Vanhalle kuparijohtimelle on olemassa montakin vaihtoehtoa, jotka tulevat ajan kuluessa syrjäyttämään vanhat johdinkaapelit. Todennäköisin vaihtoehto seuraavaksi väyläkaapeliksi on optinen kuitu. Sen halvempi kustannus ja keveyden tuomat edut verrattuna kuparijohtimeen ovat merkittäviä. Langattomat tekniikat ovat eittämättä tulossa, mutta niiden osalta toimintavarmuus ja turvallisuus ei ole vielä sillä tasolla, että niitä voitaisiin käyttää tuotantoajoneuvoissa.

4.4 By-wire -teknologiat

Ajoneuvojen teknologia on menossa suuntaan, joka pyrkii muuttamaan perinteisiä mekaanisia mekanisme ja toimimaan sähköisesti tai ainakin sähkö-mekaanisesti. Tätä kutsutaan drive-by-wire -teknologiaksi. Drive-by-wire -teknologiat voidaan jakaa neljään eri aliryhmään [1].

- Throttle-by-wire
- Brake-by-wire
- Steer-by-wire
- Shift-by-wire.

Throttle-by-wire -teknologia poistaa kaasutuksessa kaasupolkimen ja kaasuläpän välisen kaapelin. Kaasupolkimen asento mitataan asentoanturilla, jolta viedään viesti väylää pitkin kaasuläppää säätävälle moottorille. Tekniikan etuna on painon kevennys sekä tarkempi kaasun vaste [31].

Brake-by-wire -teknologian tarkoituksena on poistaa ajoneuvoista perinteinen hydraulinen jarrupiiri. Sähköiset anturit ja toimilaitteet korvaavat mekaaniset jarrusylinterit ja hydrauliliikkapiirit. Etuina kyseisellä tekniikalla on nopeampi vasteaika. Luopuminen mekaanisista toimilaitteista vähentää melua sekä eliminoi järjestelmän värähtelyä. Sähköiset toimilaitteet myös vievät tilaa vähemmän ja ovat kevyempiä kuin mekaaniset. Sähköisen järjestelmän myötä jarrujärjestelmä on helpompi liittää osaksi esimerkiksi adaptiivista vakionopeudensäädintä tai ajonvakautusjärjestelmää [31].

Steer-by-wire -teknologian tehtävänä on korvata ajoneuvon ohjauksessa mekaanisesta akselista ohjauspyörän ja renkaiden raidetankojen välillä. Perinteisesti ohjauspyörä yhdistyy ohjausakselin välityksellä ohjausvaihteeseen ja ohjaustehostimeen. Steer-by-wire -teknologiassa ohjauspyörä kiinnitetään ohjausakselin sijasta asentoanturiin, joka lukee ohjauspyörän liikkeitä. Tieto ohjauspyörän asennosta välitetään tiedonsiirtoväylän kautta ohjaustoimilaitteelle. Fyysistä yhteyttä ohjauspyörän ja renkaiden välillä ei ole lainkaan [31].

Shift-by-wire -teknologia poistaa mekanismit kuljettajan vaihdekepin ja vaihdelaatikon välillä. Mekaaninen vaihdekeppi korvataan sähköisellä kytkinvivulla, joka lähettää tiedonsiirtoverkon välityksellä tilatiedon vaihdelaatikon ohjausmoduulille. Ohjausmoduuli käskee vaihteistoa muuttamaan halutun välityssuhteen. Mekanismin vähentyessä painonsäästöedut korostuvat, sekä vaihtojen viive pienenee [31].

By-wire -teknologia vaatii väyliltä pääasiallisesti reaaliaikaisuutta ja viiveen minimoimista. Kun rattia käännetään, täytyy renkaiden kääntyä myös täysin viiveettä. NykYTEknologia esimerkiksi FlexRayn tai Automotive Ethernet -tiedonsiirtoverkkojen avulla mahdollistaa tarvittavan reaaliaikaisuuden. By-wire -teknologian kehitystä on jarruttanut huoli turvallisuudesta. Väylien viansietokyvyssä on vielä kehitettävää. By-wire -soveluksissa järjestelmän vikaantuminen voi aiheuttaa matkustajien loukkaantumisen [31].

4.5 Alijärjestelmien väylille asettamat vaatimukset

Edellä mainitut ja tulevaisuuden ajoneuvojen kehittyneet sovellukset vaativat väyliltä lisää ominaisuuksia kuudella eri osa-alueella: kaistanleveydessä, viansietokyvyssä, joustavuudessa, deterministisyydessä, turvallisuudessa sekä kustannustehokkuudessa [1, 32]. Vaatimukseen vaikuttaa yleensä varusteiden ja alajärjestelmien määrä ajoneuvoissa, mutta yleensä edellä mainitut vaatimukset tulevat vastaan jokaisessa tuotannossa olevassa nykyajoneuvossa. Taulukossa 2 on esitetty ajoneuvojen alijärjestelmien päävaatimukset sen mukaan kuinka kriittisiä ne ovat ajoneuvon alijärjestelmille.

Taulukko 2. Ajoneuvon alijärjestelmien perusvaatimukset. Muokattu lähteestä [1].

Järjestelmä	Viansieto	Deterministisyys	Kaistanleveys	Joustavuus	Turvallisuus	Reaaliaikaisuus
Alusta	Korkea	Korkea	Keskitaso	Matala	Matala	Keskitaso
Voimalinja	Keskitaso	Korkea	Korkea	Matala	Matala	Korkea
Kori	Matala	Keskitaso	Keskitaso	Korkea	Matala	Matala
By-wire	Korkea	Korkea	Keskitaso	Matala	Matala	Korkea
Infotainment	Matala	Keskitaso	Korkea	Korkea	Matala	Keskitaso
Diagnostiikka	Matala	Keskitaso	Matala	Korkea	Korkea	Matala

Järjestelmän kaistanleveyden vaatimusten suhteen joudutaan yleensä tasapainoilemaan myös kustannustehokkuuden kanssa. Yleensä kustannukset kaistanleveyden kasvaessa kasvaa myös. Järjestelmät, jotka auttavat ajoneuvoa tarjoamaan paremmat ajo-ominaisuudet, tarvitsevat entistä enemmän kaistanleveyttä tiedonsiirtojärjestelmiltä. Eri alijärjestelmillä on kuitenkin erilaiset vaatimukset kaistanleveyden ja verkon nopeuden suhteen. Esimerkiksi alustan kaistanleveyden vaatimus verkon osalta ei vaadi paljoa leveyttä, kun taas sen tulee toimia reaaliajassa. Reaaliaikavaatimus tarkoittaa mikrosekunnin aikasyklejä. Toisaalta kamerakuvaan pohjautuvat ajonvakautusjärjestelmät vaativat kaistaa yli 100Mbit/s. Väistämätöntä on, että järjestelmien kehittyminen vaatii järjestelmältä lisää kaistanleveyttä.

Viansietokyky on kriittinen vaatimus väylien toimintakuvauksessa. Tulevat järjestelmät, kuten steer-by-wire ja brake-by-wire eivät pidä sisällään vikaturvallista tilaa ollenkaan, kuten esimerkiksi lukkiutumattomat jarrut (ABS). Mikäli ABS-jarrujen järjestelmä menee vikatilaan, eikä jaksottainen jarrutustapa toimi, käyttäjällä on kuitenkin käytettävissään perinteiset lukkiutuvat jarrut. Steer-by-wire -ohjaus, on täysin sähköinen, eikä siinä ole ollenkaan mekaanista yhteyttä ohjauspyörän ja renkaiden välillä. Ohjauspyörän akselissa on asentoanturi, joka lukee ohjauspyörän asennon. Asentoanturi välittää väylää pitkin tiedon ohjausvaihteelle, joka kääntää renkaat haluttuun kulmaan. Yhteyden katketessa anturin ja ohjausvaihteen välillä käyttäjällä ei ole mitään mahdollisuutta saada renkaiden käännettyä. Vikatilanteita varten järjestelmissä on kuitenkin mekaaninen ohjaus varalla, mutta itse väylä on kestävämpiä vikoille. Brake-by-wire toimii samalla periaatteella, eli mekaaniset laitteistot poistuvat sähköisten tieltä. Jarrujärjestelmässä se käytännössä tarkoittaa hydraulisen jarrupiirin korvautumista väylällä ja sähköisillä jarrukomponenteilla. Edellä mainitut teknologiat vaativat tiedonsiirtoverkoilta suurta viansietokykyä.

Verkkojen joustavuus ja skaalautuvuus on yksi tiedonsiirtoverkoille asetettu vaatimus. Verkoista halutaan mahdollisimman joustavia, jolloin niihin tulee pystyä lisäämään ja poistamaan komponentteja ilman että verkko joudutaan sammuttamaan. Tähän ominaisuuteen voidaan vaikuttaa verkkotopologian valinnalla. Myös verkon kyky käsitellä vaihtelevaa määrää viestejä on osa verkon joustavuutta.

Nykyisten tiedonsiirtojärjestelmät ovat melko heterogeenisiä ja hyvin kompleksisia. Järjestelmien kompleksisuuden vähentäminen on keino lisätä järjestelmien skaalautuvuutta. Tiedonsiirtoverkkojen välille on jouduttu kehittämään järjestelmiä, että ne voisivat kommunikoida keskenään. Järjestelmien homogenisoiminen yksinkertaistaa järjestelmiä.

Ollakseen deterministinen, väylän tarvitsee pystyä toimimaan täsmällisesti. Viestien täytyy liikkua täydellisesti ajallaan, ja oikeaan osoitteeseen. Monet turvallisuuteen liittyvät sovellukset ovat toteutettu ennalta määrätyillä ajoituksilla. Esimerkiksi onnettomuustilanteissa ajoneuvojen turvatyynyjen tulee toimia täsmälleen tietyllä hetkellä. Tällöin järjestelmän täytyy toimia ennalta määritetyllä aikataulun mukaisesti ja viestien tulee väylää ruuhkauttamatta löytää oikeaan osoitteeseen.

Auton toimintojen nojaaminen verkkojen varaan luo verkoille turvallisuusvaatimuksen. Tulevaisuuden verkkojen vaatimuksena on se, että kun ajoneuvon toimintoja ohjaaviin verkkoihin päästään käsiksi fyysisellä yhteydellä sekä langattomasti on varmistuttava siitä, että ulkopuolisilla tahoilla ei ole mahdollisuutta päästä tekemään muutoksia verkkoon.

Reaaliaikaisuus on ja tulee olemaan iso vaatimus ajoneuvon tiedonsiirtoväylille. Tällä hetkellä reaaliaikaisuutta vaaditaan eritoten ajoneuvojen moottorinohjauksilta. Nykyaikaisten ajoneuvojen moottorinohjaus mahdollistaa niiden erittäin hyvän taloudellisuuden. Pystyäkseen siihen moottorinohjaukset tarvitsevat reaaliaikaista takaisinkytkentätietoa antureilta, jotka mittavat moottorin toimintaa. Mikäli tietoa ei saada reaaliajassa, voi vääränlainen data väärässä kohdassa moottorin toimintaa aiheuttaa jopa moottorivaurion. Esimerkiksi kone voi väärällä ilmamäärämittarin datalla toimia aina kovilla kierroksilla liian laihalla seoksella, joka aiheuttaa nakutusta. Reaaliaikaisuus on tulevaisuuden autonomisissa ajoneuvoissa kriittinen ominaisuus. Turvavarusteiden toiminta autonomisesti ajavissa autoissa perustuu ympäristön havainnointiin ja havaintoihin reagointiin reaaliajassa. Kyseisissä sovelluksissa reaaliaikaisuudesta poikkeaminen saattaa matkustajat ja muut liikenteessä olijat hengenvaaraan.

4.6 Tulevaisuuden väylä Automotive Ethernet

Automotive Ethernet on ollut markkinoilla jo useita vuosia, vaikka sitä ei vielä ole otettu laajalti käyttöön ajoneuvoteollisuudessa. Ethernetilla on jo yli 30 vuoden mittainen historia muilta verkkoteknologialta hyödyntäviltä teollisuudenaloilta. Nykytilanteessa ajoneuvon tiedonsiirtoverkko koostuu monesta eri verkosta. Kuten kuvassa 19 on esitetty,

käytössä olevia verkkoja ovat FlexRay, LIN, CAN sekä MOST. Verkot tulevat kuitenkin jäämään ominaisuuksiltaan vajaiksi ajoneuvojen tekniikan kehittyttyä. Päätaavoite autonominen ajaminen vaatii väyliltä enemmän.

Alkuperäinen Ethernet sai alkunsa vuonna 1973 Xeroxin Palo Alton tutkimuskeskuksessa Bob Metcalfen toimesta. Ethernet oli hänen kehittämänsä verkko, jolla saatiin siirrettyä dataa tietokoneelta toiselle, sekä lähetettyä sitä laser tulostimille [33]. Ethernet saapui ajoneuvotekniikkaan aluksi kahteen applikaatioon korvaamaan vanhempia tekniikoita. Ensimmäinen kohdeapplikaatio oli päivitykset ja diagnostiikka[32]. Kun päivitetään esimerkiksi auton infotainment-järjestelmän navigaattorin karttoja, tarvitaan suuri määrä kaistanleveyttä siirtämään gigabittien kokoisia tiedostoja. Vanhan CAN-väylän kapasiteetilla kyseisten päivitysten siirtäminen kestäisi liian pitkään. Perinteisimmistä väylistä MOST, tarjoaa riittävän suuren kaistanleveyden tähän tarkoitukseen, mutta muut väylän ominaisuudet tekevät käytöstä haastavaa [34]. Rengastopologiansa ansiosta MOST-väylään liittyminen vaatii väylän alasajon, mikä ei järjestelmän toiminnan kannalta ole järkevää. Myös ajoneuvojen infotainment-järjestelmiin MOST-väylän tilalla on alettu käyttämään Ethernet-väylää. Mediadatan siirtoon Ethernet on alun perin suunniteltukin [34]. Ethernet on valmis ja matalakustanteinen teknologia, joka tarjoaa suurimman määrän kaistanleveyttä, mikä tällä hetkellä on tarjolla [35]. Automotive Ethernet on tämän hetken paras ehdokas uudeksi tiedonsiirtoverkon rungoksi. Se tullaan ottamaan käyttöön kuten CAN-väyläkin aikoinaan. Ensiksi sitä käytetään ajoneuvon diagnostiikkaan sekä yksittäisiin pienempiin alijärjestelmiin, kuten esimerkiksi peruutuskameroihin. Sen jälkeen Ethernetillä tullaan toteuttamaan suurempia järjestelmäkokonaisuuksia. Esimerkiksi ajoneuvon moottorinohjaus. Viimeisempänä sukupolvena Ethernet tulee syrjäyttämään CAN-väylän tiedonsiirtoverkkojen runkona.

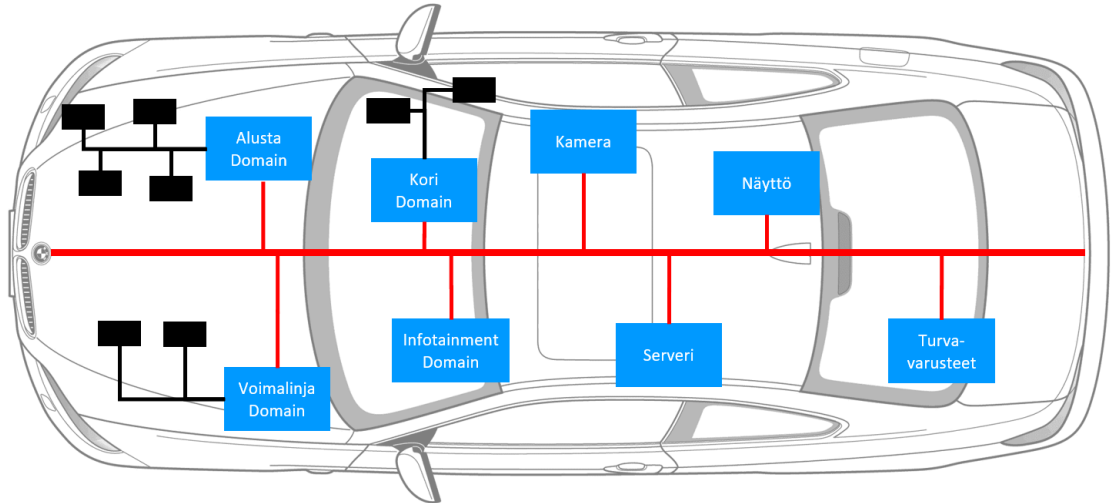
Automotive Ethernet -tiedonsiirtoverkko voidaan kuvailla Department of Defense -mallin avulla (DoD). DoD on tiivistetty versio OSI-mallista. Automotive Ethernetin DoD-malli koostuu neljästä kerroksesta, verkko-, internet-, kuljetus- ja sovelluskerroksista. Kerrokset ovat esitetty kuvassa 20.

OSI-malli	Kerrokset	DoD-malli
Sovellus	7	Sovellus
Esitystapa	6	
Istunto	5	
Kuljetus	4	Kuljetus
Verkko	3	Internet
Siirto	2	Verkko
Fyysinen	1	

Kuva 21. Automotive Ethernetin DoD-malli verrattuna perinteiseen OSI-malliin. Muokattu lähteestä [36].

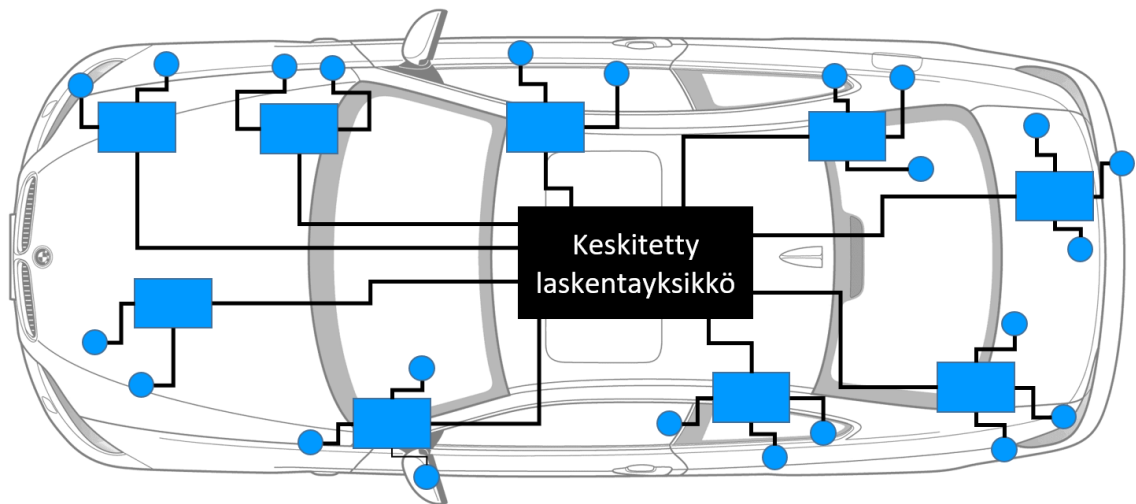
Ensimmäisenä kerroksena on verkko. Se pitää sisällään lähetyserroksen ja fyysisen kerroksen. Automotive Ethernetillä on useita eri fyysisiä kerroksia, joita käytetään tarpeen mukaan. Yleisimpiä ovat 100 Base TX, 100 Base T1 ja 1000 Base T1. Kyseiset kerrokset ovat suunniteltuja ajoneuvokäyttöön. Ne eroavat toisistaan johdotuksen ja sen pituuden osalta, mutta myös siirtokapasiteetiltaan. 100 Base TX on käyttäjä neljää suojattua johdinta, jotka voivat olla maksimissaan 100m pituudeltaan. 100 Base T1 käyttää taas suojaamatonta parikaapelia, jonka maksimipituus voi olla 15m. Molemmat ovat siirtonopeudeltaan 100Mbit/s. 1000 Base T1 on taas siirtonopeudeltaan 1000 Mbit/s ja se käyttää maksimissaan 15m pituista suojaamatonta parikaapelia [36].

Automotive Ethernet tulee muuttamaan ajoneuvojen johtosarjoja huomattavasti. Nykyiset heterogeeniset hybriditopologiat tulevat vaihtumaan yksinkertaisempaan malliin. Uusien verkkoarkkitehtuurien suunnittelussa otetaan huomioon autonomisen ajamisen vaatimukset. Vanhat verkkotopologiat ovat suunniteltu vastaamaan tilannetta, jossa ajoneuvossa on aina fyysinen kuljettaja. Kuvan 22 mukainen domain-pohjainen arkkitehtuuri, jossa ajoneuvon alijärjestelmien laskenta on jaettu erillisiin domain-moduuleihin yhden yhdyskäytävän sijaan. Tällöin laskentaan käytettävä data ei varaa runkoväylän kaistanleveyttä turhaan.



Kuva 22. Automotive Ethernet runkoverkko arkkitehtuuri. Muokattu lähteestä [27].

Toinen tulevaisuuden verkkoarkkitehtuuri on kuvassa 23 esitetty keskitetty arkkitehtuuri. Keskitetty arkkitehtuuri koostuu keskelle sijoitetusta laskentayksiköstä, sekä jokaiselle alueelle sijoitetusta aluekontrollerista. Keskitetyn arkkitehtuurin tarkoituksena on kaikki laskenta yhteen yksikköön ja sijoittaa muut komponentit mahdollisimman lähelle niiden fyysistä sijaintia. Keskitetty arkkitehtuuri mahdollistaa kaiken datan analysoimisen yhdessä paikassa. Lisäksi se on joustava ja siihen on helppo lisätä uusia komponentteja.



Kuva 23. Keskitetty Automotive Ethernet arkkitehtuuri. Muokattu lähteestä [27].

Automotive Ethernet toimii monilla erilaisilla kehysformaateilla. Tässä työssä esitellään Ethernet II formaatin paketti ja kehys, sillä sitä käytetään tyypillisesti Automotive Ethernetin yhteydessä. Kuvassa 24 on esitetty malli paketista ja kehuksesta.

Packet							
Preamble	SFD	Frame					
		Header			Data		CRC
		Destination MAC	Source MAC	VLAN Tag	Type	Data	
7	1	6	6	4	2	42-1500	4 (bytes)

Kuva 24. Ethernet II paketti ja kehys. Muokattu lähteistä [36, 38].

Ethernet kehyyksen ensimmäinen osio on otsikko (Header), joka koostuu MAC-osoitteista ja Virtual Local Area Network (VLAN) tagista. MAC osoitteista ensimmäisenä on kehyyksen kohdeosoite, jonka tehtävä on määrittää viestin vastaanottaja. Seuraavana otsikossa on lähettävän yksikön MAC-osoite. Ethernet kehyyksellä on voi olla ainoastaan yksi lähetysosoite, mutta useampia vastaanottajia. Viimeisenä otsikossa on VLAN tag. VLAN tag on osana kehystä riippuen siitä, onko kyseessä tavallinen MAC kehys vai tagged MAC kehys. Tagged MAC -kehystä käytetään silloin kun viestejä halutaan priorisoida. VLAN tag koostuu Tag Protocol Identifier (TPID) biteistä sekä Tag Control Information (TCI) biteistä. TPID sisältää tiedon siitä mitä tyyppiä kehys on. TCI koostuu osioista Priority Control Point (PCP), Drop Eligible Indicator (DEI), Canonical Form Indicator (CFI) sekä VLAN Identifier (VID). VID sekä PCP -osioita käytetään ajoneuvotekniikassa viestien priorisoimiseen. Tällä tavoin voidaan väylän ruuhkatilanteissa varmistaa tärkeiden viestien saapumisen perille. Type field -kenttä osoittaa onko kehys normaali vai tagged-kehys. Bitin arvo 0x8100 kertoo kehyyksen olevan tagged-kehys. Data-kentässä lähetetään varsinainen data. Normaalin kehyyksen datakentän minimimitta on 46 tavua ja tagged-kehyyksen 42 tavua. Molemmat voivat sisältää enintään 1500 tavua. Viimeisenä kenttänä on CRC tarkastussumma virheentarkistusta varten. Ethernet pakettiin kuuluu vielä kehyyksen edellä olevat preamble sekä start-of-frame-delimiter -kentät. Niiden tehtävänä on toimia signaalina lähetyksen aloittamiselle [36, 38].

Automotive Ethernet on edellä esitettyjen ominaisuuksiensa takia sopiva vaihtoehto ajoneuvojen tiedonsiirtoverkkojen pohjaksi CAN-pohjaisten verkkojen tilalle. Ethernet pystyy vastaamaan kaistanleveyden ja viiveen osalta tulevien sovellusten vaateisiin.

5. POHDINTA JA YHTEENVETO

Nykyinen teknologia ajoneuvojen tiedonsiirtoverkoissa on saavuttamassa rajansa. Autonominen ajaminen sekä kuljettajaa avustavat sovellukset asettavat kaistanleveydelle ja reaaliaikaisuudelle sellaiset vaatimukset, että CAN-runkoinen tiedonsiirtoverkko ei siihen pysty vastaamaan. CAN-väylän rinnalle on myöhemmin kehitetty paremmin vaatimuksiin vastaavia väyliä, jotka on esitelty työssä aikaisemmin. Nämäkin väylät käyvät tulevaisuuden sovelluksille riittämättömiksi. Ratkaisu löytyy paikallisverkoista. Ethernet verkosta on tuotettu ajoneuvokäyttöön sopiva Automotive Ethernet.

Yksi tulevaisuuden tavoitteista on keventää ja halventaa ajoneuvojen elektroniikkaa. Nyky päivän ajoneuvojen johtosarjat ovat kolmanneksi painavin komponentti ajoneuvossa. Johtosarjaa painavampia ovat ainoastaan alusta sekä moottori. Jo pelkästään väylien arkkitehtuuria muuttamalla voidaan saada huomattavia säästöjä painon ja materiaalikulutuksen suhteen. Automotive Ethernet ajoneuvon johtosarjan runkona mahdollistaa kevyemmän verkkoarkkitehtuurin. CAN-runkoiset arkkitehtuurit ovat tulleet tiensä päähän jo muutamia vuosia sitten kaistaleveyden puutteen vuoksi. Perinteisen hybriditopologian pullonkaulana on keskusportti, jonka kautta kaikki ohjausyksiköt ja moduulit ovat yhteydessä toisiinsa. Keskusportti toimii tiedonsiirtokeskuksena, joka mahdollistaa kommunikaation eri moduuleiden välillä. Arkkitehtuurien seuraava askel tulee olemaan Ethernetin mahdollistama domain-pohjainen arkkitehtuuri, jossa useampi domain-yksikkö ovat kytkeyty Ethernet-runkoverkolla toisiinsa. Kyseinen arkkitehtuuri mahdollistaa tarvittavien ohjausyksiköiden vähentämisen, koska jäljellä eri ohjausyksiköiden ja antureiden toiminnot ja laskenta tapahtuu domain yksiköissä. Tällöin domain-moduuleissa tehtävä kommunikaatio ei tapahdu runkoväylässä, mikä parantaa väylän tehokkuutta [27].

Kevennystä haetaan myös fyysisten kaapelointien muutoksella. Tällä hetkellä Automotive Ethernetiä käytetään vielä perinteisellä kuparijohtimisella parikaapelilla, mutta tilalle on kehitteillä optinen kaapeli. Optinen kaapeli on kevyempää ja halvempaa valmistaa. Lisäksi materiaalin ympäristövaikutukset ovat suotuisimmat. Mitä pidemmälle tulevaisuuteen katsotaan näyttää siltä, että johdotukset antureiden ja ohjausyksikköjen välillä tulevat katoamaan kokonaan. Langaton tekniikka, kuten Bluetooth tai koko auton kattava langaton WiFi, ovat ratkaisu painonkevennykselle. Tekniikka ei ole kuitenkaan vielä niin kehittynyttä, että se vastaisi vaatimuksiin viansietokyvyn ja turvallisuuden osalta. Vaatimuksena toimiselle on yhteyden toimiminen kaikissa mahdollisissa tilanteissa katkeamatta. Lisäksi tietoturva täytyy olla sillä tasolla, ettei kukaan ulkopuolinen pysty tunkeutumaan järjestelmään.

Nykyhetken ajoneuvojen tiedonsiirtoverkot koostuvat useista eri protokollista, jotka toimivat keskenään sulautetusti. On ehdotettu, että koko ajoneuvon tiedonsiirtoverkko tulisi olla yhtä ja samaa teknologiaa. Päinvastoin kuin nykytilanteessa, jossa CAN, MOST,

FlexRay ja LIN toimivat yhdessä omilla osa-alueillaan. Ratkaisu helpottaisi järjestelmien joustavuutta ja skaalautuvuutta, mikäli kaikkialla ajoneuvoa tekniikka toimisi yhteen keskenään. Idea ei ole toteutuskelpoinen ainakaan lähitulevaisuudessa. Automotive Ethernet vielä uutena tekniikkana pitää hintaansa sen verran korkealla, että ei ole kannattavaa käyttää yliampuvaa tekniikka sellaisissa paikoissa, missä pärjätään halvemalla tekniikalla, kuten LIN-väylällä. Tästä syystä vanhat ja halvemmat teknologiat eivät vielä ole poistumassa vaan niillä on paikkansa ajoneuvon tiedonsiirtojärjestelmässä siellä missä ne pysyvät vastaamaan ominaisuuksiltaan vaatimuksiin.

Taulukossa kolme on eritelty tämän hetken väyläteknologioiden avainominaisuuksia. Ethernet pystyy 1 Gbit/s siirtonopeudellaan vastaamaan kasvavaan tarpeeseen kaistanleveyden suhteen. Automotive Ethernet kaapelin pituus on rajoitettu 15 metriin kytkimien välillä viiveen minimoimiseksi. Siinä Ethernet ei pärjää vanhemmille tekniikoille, mutta kytkimien takia 15 metrin raja ei koidu ongelmaksi. Kustannuksiltaan Ethernet on vielä hieman kalliimpi tarkastellessa etenkin nopeimpia verkkoja. Automotive Ethernetin yleistyessä sen kustannukset kompensoituvat domain-arkkitehtuurin mahdollistamalla vähemmällä johdotuksella. Kaapelointien osalta MOST-väylä pystyy hyödyntämään optista kaapelia, mutta tulevaisuudessa myös Ethernet tulee hyödyntämään optisia kaapeleita.

Taulukko 3. Väyliä ominaisuudet.

	CAN	FlexRay	MOST	LIN	Ethernet
Kaistanleveys (Mbit/s)	1-10 (CAN FD)	20	150	0,02	1000
Maksimipituus (m)	40	24	1280	40	15
Viestintä	Multi-master	Multi-master	Paketti /Stream	Master/Slave	IP
Hintataso	Matala	Matala	Korkea	Matala	Korkea
Johdotus	Suojaamaton parikaapeli	Suojaamaton parikaapeli	Optinen kaapeli	Yksinkertainen kaapeli	Suojaamaton parikaapeli
Alijärjestelmät	Kaikki	Alusta /Voimansiirto	Infotainment	Kori	Infotainment /Runkoväylä

Suurin tulevaisuuden tavoite ajoneuvoteollisuudessa on autonominen ajaminen. Autonomisuus asettaa myös vaatimuksensa ajoneuvon tiedonsiirtoväylille. Autonominen ajaminen perustuu suuresti ympäristön havainnointiin, mikä tarkoittaa videokuvan analysoimista. Väylille tämä tarkoittaa suurta määrää videodataa reaaliajassa kameroilta laskentamoduulille. Autonomisten ajoneuvojen matkustajien henki riippuu tiedonsiirtoväyliä toiminnasta. Edellä mainitut toiminnot nojautuvat juuri kaistanleveyteen sekä reaaliai-

kaisuuteen. Kameroiden lähettämä data vaatii yli 10 Gbit kaistanleveyttä, johon ei nykyinen Automotive Ethernet teknologia pysty vielä vastaamaan, mutta kehitteillä on useiden gigabittien nopeudella varustettuja protokollia. Tällä hetkellä ajoneuvojen autonomia on suurimmaksi osaksi tasolla kaksi. Tuoreimmissa ajoneuvoissa on tiloja, jossa ajoneuvot hoitavat ajotehtäviä, kuten esimerkiksi parkkeerauksen, itsenäisesti kuljettajan istuessa ajajan paikalla. Jotkin autovalmistajat ovat jo julkaisseet ajoneuvoja, joissa on saavutettu autonomisen ajamisen kolmas taso. Kolmannella ajoneuvot ajavat kuljettajan valvonnassa itsekseen. Edellä mainituilla tasoilla olevat ajoneuvot käyttävät jo nyt Ethernet pohjaista tiedonsiirtoverkkoa. Lopulta ajoneuvojen tiedonsiirtoverkot tulevat käyttämään kauttaaltaan Automotive Ethernet -verkkoa, kun tavoitteena on autonomisesti ajava auto.

Tällä hetkellä Automotive Ethernet on ottamassa enenevässä määrin alaa ajoneuvojen tiedonsiirtoverkoissa. Vanhat teknologiat eivät ole poistuneet käytöstä vaan niitä käytetään sovelluksissa, joissa nykyhetkellä Automotive Ethernetin ominaisuudet menisivät hukkaan. Ajoneuvoteknologian tulevaisuuden päätavoite ja suurin haaste on autonominen ajaminen. Tämä ajaa ajoneuvojen tiedonsiirtoverkkoja vääjäämättä suuntaan, jossa nykyiset teknologiat eivät vastaa sovellusten tarpeisiin.

LÄHTEET

- [1] T. Nolte, H. Hansson, L. Lo Bello, Automotive communications - past, current and future, IEEE, pp. 8-992.
- [2] S. Carlson, 10G+ Automotive Ethernet Electrical PHYs, IEEE 802.3 Working Group, pp. 31.
- [3] History of CAN technology, web page. Available (accessed 3.10.2018): <https://www.can-cia.org/can-knowledge/can/can-history/>.
- [4] K. Matheus, T. Königseder, Automotive Ethernet, Cambridge University Press, 2014, 238 p.
- [5] P. Richards, A CAN Physical Layer Discussion, Microchip Technology Inc, 2002, pp. 12.
- [6] R. Bosch, CAN Specification, Bosch GmbH, 1991, 72 p.
- [7] CAN Physical Layers, web page. Available (accessed 25.11.2019): <https://www.kvaser.com/about-can/the-can-protocol/can-physical-layers/>.
- [8] D. Kraus, E. Leitgeb, T. Plank, M. Loschnigg, Replacement of the Controller Area Network (CAN) protocol for future automotive bus system solutions by substitution via optical networks, IEEE, 2016, 8 p.
- [9] CAN data link layers in some detail, web page. Available (accessed 25.11.2019): <https://www.can-cia.org/can-knowledge/can/can-data-link-layers/>.
- [10] S. Corrigan, Introduction to the Controller Area Network (CAN), Texas Instruments Incorporated, 2015, 17 p.
- [11] CAN Data Frame Messages, web page. Available (accessed 25.11.2019): <https://www.kvaser.com/about-can/the-can-protocol/can-messages-13/>.
- [12] CAN Bus Error Handling, web page. Available (accessed 16.1.2020): <https://www.kvaser.com/about-can/the-can-protocol/can-error-handling/>.
- [13] J. Alanen, CAN - ajoneuvojen ja koneiden sisäinen paikallisväylä, VTT, 2000, 16 p.
- [14] Understanding CAN with Flexible Data-Rate (CAN FD), web page. Available (accessed 26.11.2019): <http://www.ni.com/fi-fi/innovations/white-papers/14/understanding-can-with-flexible-data-rate--can-fd-.html>.
- [15] What is Network Topology? Best Guide to Types & Diagrams, web page. Available (accessed 2.12.2019): <https://www.dnsstuff.com/what-is-network-topology>.

- [16] N. Navet, F. Simonot-Lion, A Review of Embedded Automotive Protocols, Industrial Information Technology Series, 2008, 4.1-4.31 p.
- [17] Introduction to the LIN bus, web page. Available (accessed 13.11.2019): <https://www.kvaser.com/about-can/can-standards/linbus/>.
- [18] Introduction to the Local Interconnect Network (LIN) Bus, web page. Available (accessed 15.11.2019): <https://www.ni.com/fi-fi/innovations/white-papers/09/introduction-to-the-local-interconnect-network--lin--bus.html>.
- [19] FlexRay Automotive Communication Bus Overview, web page. Available (accessed 17.1.2020): <https://www.ni.com/fi-fi/innovations/white-papers/06/flexray-automotive-communication-bus-overview.html>.
- [20] A. Grzemba, MOST The Automotive Multimedia Network, Franzis Verlag GmbH, 2011, 32 p.
- [21] S. Shanker, Enhancing Automotive Embedded Systems with FPGAs, Nanyang Technological University, 2016, 250 p.
- [22] J. Huang, M. Zhao, Y. Zhou, C. Xing, In-Vehicle Networking: Protocols Challenges and Solutions, IEEE Network, 2019, pp. 92-98.
- [23] K. Van Cleave, A Survey of Automotive Ethernet Technologies and Protocols, Washington University, 2019, 13 p.
- [24] P. Nowak How does my car's lane-keep assist technology work? web page. Available (accessed 12.12.2019): <https://www.theglobeandmail.com/drive/culture/article-how-does-my-cars-lane-keep-assist-technology-work/>.
- [25] Self-driving cars — The human side, in: 2017 IEEE 14th International Scientific Conference on Informatics, 2017, pp. 383-387.
- [26] Autonomous Driving – five steps to the self-driving car, web page. Available (accessed 20.1.2020): <https://www.bmw.com/en/automotive-life/autonomous-driving.html>.
- [27] D. Yang, K. Jiang, D. Zhao, C. Yu, Z. Cao, S. Xie, Z. Xiao, X. Jiao, S. Wang, K. Zhang, Intelligent and connected vehicles: Current status and future perspectives, google Springer, 2018, 1446-1471 p.
- [28] 6 Key Connectivity Requirements of Autonomous Driving, web page. Available (accessed 20.1.2020): <https://spectrum.ieee.org/transportation/advanced-cars/6-key-connectivity-requirements-of-autonomous-driving>.
- [29] H. Lim, K. Weckemann, D. Herrscher, Performance study of an in-car switched ethernet network without prioritization, Springer, 2011, 165-175 p.
- [30] Automotive Ethernet: An Overview, ixia.com, 2014, Available: https://support.ixia.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper_1.pdf.

- [31] P. Parsania, K. Saradava, Drive-By-Wire Systems In Automobiles, 2015, 5 p.
- [32] Z. Levi, Automotive Ethernet: The Future of In-Car Networking? Electronic Design, 2018, <https://www.electronicdesign.com/automotive/automotive-ethernet-future-car-networking>.
- [33] C.E. Spurgeon, Ethernet: The Definitive Guide, O'Reilly Media, Inc., 2000, 459 p.
- [34] H. Lim, L. Völker, D. Herrscher, Challenges in a future IP/Ethernet-based in-car network for real-time applications, 2011, pp. 7-12.
- [35] N. Navet, F. Simonot-Lion, In-vehicle communication networks - a historical perspective and review, in: R. Zurawski (ed.), Industrial Communication Technology Handbook, CRC Press Taylor & Francis, 2013, pp. 38.
- [36] S. Gugenhan, Automotive Ethernet - Beginners guide, Star Cooperation GmbH, 2019, 11 p.
- [37] Introduction to Automotive Ethernet - Ethernet frame, web page. Available (accessed 27.1.2020): <https://elearning.vector.com/mod/page/view.php?id=157>.