

Åbo Akademi

# Bilens inbyggda CAN-system och framtida alternativ

Fakulteten för naturvetenskaper och teknik

Benjamin Österholm, 36208  
Handledare: Jerker Björkqvist  
Februari – Maj 2015

## Referat

Denna kandidatavhandling kommer att ta upp bilens inbyggda CAN-system och vad som kan tänkas hända med det i framtiden. I första delen berättas om CAN-bussens historia och hur den hamnade i sin nu väldigt starka position inom bilindustrin. Efter detta tas delar av hur protokollet fungerar upp med bl.a. synkroniseringen, datalänksskiktet och till sist mera konkreta detaljer om det fysiska skiktet. Den andra delen av avhandlingen berättar kort om varför man kunde tänkas byta ut CAN och tar upp några möjliga alternativ till CAN som ett par standarder av ethernet och en CAN-hybrid delvist bestående av trådlösa noder.

Sökord: CAN, bilens inbyggda system, ethernet, IEEE 802.1 AVB, TTEthernet, ViCAN

## Innehållsförteckning

Referat .....	1
1 Inledning .....	3
2 Controller Area Network .....	4
2.1 CAN:s historia.....	4
2.2 Så fungerar CAN.....	5
2.2.1 CAN synkronisering och NBT.....	6
2.3 Datalänksskiktet.....	8
2.4 Det fysiska skiktet .....	9
3 Alternativ till CAN .....	11
3.1 Allmänt om Ethernet .....	11
3.1.1 IEEE 802.1 AVB och TTEthernet .....	12
3.2 ViCAN .....	15
4 Avslutning .....	17
Källförteckning.....	18

## 1 Inledning

CAN eller som det på engelska står för *Controller Area Network* är ett nätverksprotokoll som har förekommit i många inbyggda system sedan det utvecklades. Eftersom det ursprungligen utvecklades för bilar är det naturligtvis så att det har sin starkaste ställning i bilar också men med de allt mer utrustade bilarna måste även CAN-protokollet ses över. Även om CAN nog har uppdaterats med åren så finns det en övre gräns för med vilken hastighet CAN-bussen kan överföra data i nätverket. Den ligger på 1 Mbit/s och är då väldigt låg jämfört med ethernet-tekniken som år 2017 förutsägs nå en hastighet på 400 Gbit/s. De största orsakerna till att bilens nätverksbuss borde uppdateras är alltså att det utvecklas många hastighetskrävande system till moderna bilar som använder sig av t.ex. HD-kameror (högupplösningskameror), ultraljudssensorer, radar m.m. En annan orsak till varför man kunde uppdatera CAN-bussen är att alla dessa nya system som sätts in i bilen börjar ta mera och mera utrymme. När CAN utvecklades behövdes inte så många kablar men under de senaste åren har antalet ökat. Därför har koncept som grundar sig på den ursprungliga CAN-bussen, men delvis är trådlösa, också simulerats under det senaste året.

## 2 Controller Area Network

På CiA:s (CAN in Automation) [1] hemsida under fliken *CAN history* får man veta att CAN-protokollet blev presenterat av Robert Bosch GmbH år 1986 men att själva konceptet påbörjades redan tre år tidigare av Uwe Kiencke. Orsaken till detta var att ingenjörer vid Bosch hade undersökt existerande seriella bussystem och märkt att inga av de dåvarande nätverksprotokollen uppfyllde kraven för att kunna fungera effektivt i en bil. CiA [1] nämner också att eftersom det nya systemet skulle användas i bilar var ingenjörer från Mercedes-Benz tidigt involverade i projektet liksom Intel som var en potentiell leverantör för halvledare. Andra medverkande parter var en konsulterande professor vid namn Dr. Wolfhard Lawrenz från University of Applied Science i Braunschweig-Wolfenbüttel, Tyskland. Dr. Lawrenz var också den person som gav det nya systemet namnet *Controller Area Network* d.v.s. CAN.

### 2.1 CAN:s historia

I februari år 1986 framförde Robert Bosch, grundaren av företaget Bosch, själv det nya systemet vid kongressen SAE (Society of Automotive Engineers). Det nya nätverksprotokollet var ett så kallat multi-master-protokoll som beroende på prioritet ger de olika meddelandena tillgång till bussen utan fördröjning. I och med det nya systemet kom också en stor reducering av kabeluppsättningen vilket egentligen bara var en biprodukt av utvecklingen [1]. Här började alltså utvecklingen som har lett till det system som används i dagens moderna bilar. Även om det ursprungligen var menat för personbilar så användes CAN-system tidigt också i andra sammanhang som t.ex. i finska Kones hissar. Efter SAE-kongressen gick det ännu ett år innan Intel och Philips Semiconductors släppte de två första CAN-chipsen, 82526 respektive 82C20. Dessa två skilde sig ändå lite från varandra i och med detaljer som bland annat CPU-användning och gräns på inkommande meddelanden [1].

Åtta år efter att CAN offentliggjorts kom den första bilmodellen från Mercedes-Benz, med den nya CAN-bussen installerad, ut på marknaden och samma år grundades CiA som var en grupp avsedd för att marknadsföra CAN för både internationella användare och tillverkare. Efter att Mercedes-Benz haft endast ett system i sina mer högklassiga bilmodeller plus en andra styrenhet, förutom motorerna, så introducerades också en till CAN-buss som kopplades ihop med den gamla genom en gateway. Senare följde också andra bilmärken som BMW, Fiat, Renault, Saab, Volkswagen och Volvo i MB:s spår och tog i bruk två CAN-bussar i sina bilar. Idag är det vanligt att så mycket som tre CAN-bussar för olika moduler i bilen är ihopkopplade genom en gateway [1].

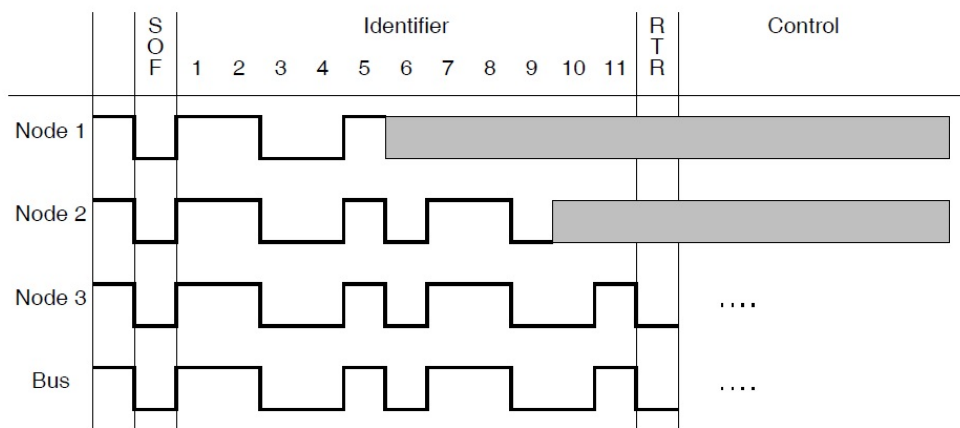
Som tidigare nämnts används CAN i många andra sammanhang än bara inom bilindustrin. Förutom i bilar och så gott som alla andra transportmedel används CAN också i automation av produktionslinjer, hissar, rulltrappor, jordbruksmaskiner, medicinal utrustning och mycket annat [2].

## 2.2 Så fungerar CAN

På CiA:s webbsida [1] förklaras mera ingående vad CAN-bussens specifikationer är. I en bil är det ett nätverk som specifikt binder ihop bilens olika moduler, ursprungligen de olika ställdonen (eng. *actuator*), sensorerna och styrenheterna i realtid. CAN kallas också ibland synkront. Detta påstående är dock inexact eftersom de meddelanden som sänds kommer från noder vars klockor inte nödvändigtvis är synkrona på grund av faktorer som temperatur, fukt o.s.v. [2]. CAN-protokollet är egentligen en asynkron seriebuss med en kodning kallad NRZ (eng. *Non-Return to Zero*) som är väldigt tålig och snabb [3]. Orsaken till att CAN borde vara synkront är att när de olika noderna i nätverket samtidigt sänder meddelanden måste bussen kunna klargöra vilket meddelande som har företräde. Ordningen klargörs genom att jämföra en sträng med bitar från början av meddelandet, kallade ID-bitar som också är den sändande nodens ID [2]. När då flera meddelanden sänds på samma gång jämförs varje bit enskilt med varandra

och i CAN är alltid en logisk 0:a dominant. När en 1:a och en 0:a ställs mot varandra kommer noden som ger 1:an att sluta sända för att den har lägre prioritet [2] [4]. Noden med lägre prioritet kommer därefter att ställa sig i kö tills den får möjlighet att sända igen.

I Johansson m.fl. [5] finns ett exempel på ett CAN-nätverk med 11-bitars ID-meddelanden (eng. *identifier*) där *node 1* = 11001101010, *node 2* = 11001011011 och *node 3* = 11001011001. I figur 1 nedan ser man hur protokollet jämför de tre meddelandena. Från i det här exemplet ID-bit nummer 6 visar det sig att *node 1* har lägre prioritet än de båda andra noderna. Den slutar genast sända och fortsätter i ett mottagande läge indikerat som grått för att ge de högre prioriterade



Figur 1 från Johansson m.fl. [5] visar hur CAN-protokollet jämför och utesluter lägre prioriterade noder.

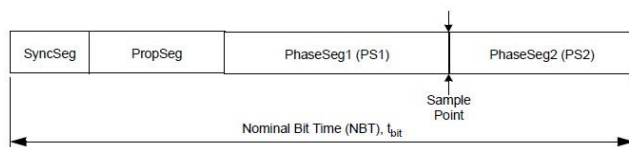
noderna förtur. Vid ID-bit nummer 10 faller också *node 2* bort och lämnar den högst prioriterade *node 3* kvar för att den ska få sända sitt meddelande. Förutsättningen för att detta ska fungera är att varje nod ha ett unikt ID [5].

### 2.2.1 CAN synkronisering och NBT

Richards [3] skriver att för att över huvud taget lyckas med denna sorts jämförelse måste alltså meddelandena synkroniseras kontinuerligt. Detta görs med två olika metoder. Först och främst hård synkronisering som sker när bussen är överksam och specifikt när den första recessiva 1:an skapar en s.k. kant då den går till en dominant 0:a. Detta är en SOF (eng. *Start-of-Frame*), vilket betyder att ett

meddelande börjar. En hård synkronisering kan endast ske en gång per meddelande. Bitsynkronisering, som är den andra typen av synkronisering, ske på samma bit som en hård synkronisering. Eftersom meddelanden lätt kan komma i otakt på grund av frekvensdrift i nodernas klockor behövs också bitsynkronisering [3].

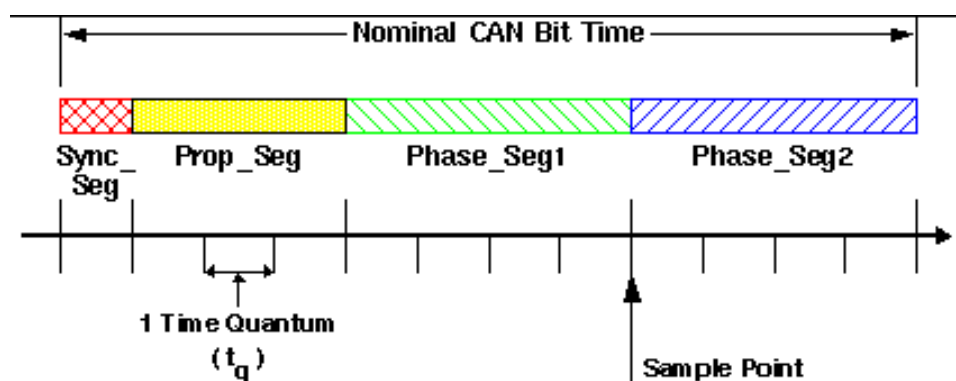
Richards [3] förklarar också att synkroniseringen sker med hjälp av den så kallade NBT:n (eng. *Nominal Bit Time*). Detta är den tid det tar att sända en bit och den består av fyra segment: synkroniseringssegmentet (eng. *synchronization*



*segment*), spridningssegmentet (eng. *propagation segment*) och de två fassegmenten (eng. *phase segment 1 and 2*) som

Figur 2 visar de fyra segment som en NBT består av [3].

visas i figur 2. NBT:n som också syns i figur 2 är den sammanlagda tiden före de fyra segmenten och varje segment består dessutom av ett antal så kallad tidskvanta eller TQ (eng. *Time Quanta*). Figur 3 är ett exempel på hur många tidskvanta de olika segmenten kan bestå av. Synkroniseringssegmentet är det som används för att synkronisera noderna eftersom meddelandebitarnas kanter förväntas inträffa inom detta segment som dessutom är fast fixerat till 1TQ.

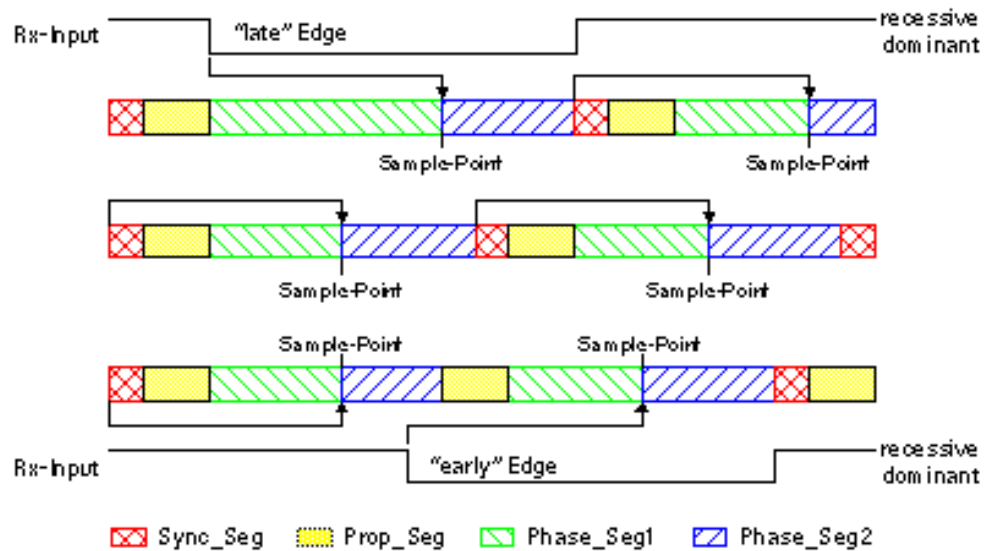


Figur 3 är exempel på de olika segmentens tidskvanta [6].

Spridningssegmentet som kommer efter synkroniseringssegmentet är till för att utjämna den fysiska fördröjning som kan finnas mellan de sändande noderna. Detta segment har en ställbar tidskvanta på 1 – 8TQ och programmeras skilt enligt CAN-bussens längd och nodernas hårdvara. De två fassegmenten 1 och 2 som har



tidskvanta 1 – 8TQ förlängs respektive förkortas under bitsynkroniseringens gång för att alla noder skall synkroniseras [6]. Samplingspunkten (eng. *sample point*) som ses i både figur 2 och 3 ligger mellan de två fasssegmenten i NBT:n. Det är här som den logiska nivån läses av d.v.s. om biten är en 1:a eller 0:a [3] [6]. Figur 4 visar tre fall där fasssegmenten efter behov justeras när en bits kant kommer för sent eller för tidigt. I mitten av bilden finns det fall där ingen synkronisering behövs.



Figur 4 visar hur fasssegmenten förlängs och förkortas efter behov [6].

## 2.3 Datalänksskiktet

Voss [2] skriver att datalänksskiktet är den del av CAN-systemet som för ihop data med själva protokollet och slutligen till applikationsskiktet som i sin tur hanterar operativsystemet eller applikationen som finns i en CAN-enhet. Datalänksskiktet står för utbyten av nodernas meddelanden, felhanteringen och egenskaper som t.ex. nodernas prioritet för tillgång till bussen, datakonsistens i systemet, felhantering och automatisk återsändning av förlorade meddelanden [2].

När man talar om meddelanden i CAN-bussen hänvisar man till dem som ramar. De olika typerna av ramar är datameddelanden (eng. *Data Frame*) som sänder

data, meddelandebegäran (eng. *Remote Frame*) som begär meddelanden från andra noder, felmeddelanden (eng. *Error Frame*) som signalerar ett fel i bussen och kömeddelanden (eng. *Overload Frame*) som begär en fördröjning mellan två datameddelanden eller meddelandebegäranden [2] [7]. Prioriteten för ramarna är som nämndes på sida 5 beroende på ID:n. Den ram med lägst ID får vid första möjliga tillfälle tillgång till bussen [2].

Felhanteringen i en CAN-buss är känd för att vara väldigt effektiv och pålitlig. Enligt en matematisk uträkning kommer en CAN-buss att missa endast ett fel på 1000 år om den ger en bit fel per 0,7 s, har en baudhastighet på 500 kbit/s, är igång 8 timmar/dag och 365 dagar/år [2].

## 2.4 Det fysiska skiktet

Kvasers [7] skriver på sin webbsida att det fysiska skiktet är bl.a. den konkreta elektroniken i CAN-bussen d.v.s. kopplingarna och kablarna mellan noderna. Skiktet definierar också de elektriska nivåerna och signalscheman som bör finnas i bussen. I avsnittet 2.2 beskrivs att bitarna representerar en logisk 1:a eller 0:a och dessa motsvarar en viss elektrisk nivå beroende på vilket av följande fysiska standardskikt som är implementerat i bussen [7].

**SAE J2411** är en enkelkabelns standard med den normala hastigheten 33 kbit/s (83,3 kbit/s när man felsöker). Upp till 32 noder kan finnas i nätverket och det används huvudsakligen i bilar i de mindre krävande komfortområdena [7] [8].

**ISO 11898-2** standarden är en av de vanligaste typerna av CAN-bussen. Den är en partvinnad kabelbuss med den maximala hastigheten på 1 Mbit/s också kallad höghastighets CAN [7] [8].

**ISO 11898-3** standarden är också en partvinnad kabelbuss kallad låghastighets CAN, men med den skillnaden att den endast har en hastighet på 125 kbit/s men är däremot väldigt feltålig. Den kan fortsätta sända fastän en kabel kapas eller kortsluts. Precis som SAE J2411 kan detta nätverk ha upp till 32 noder [7] [8].

**ISO 11992** är en punkt-till-punkt buss som också har den låga hastigheten 125 kbit/s. Den används huvudsakligen i lastbilar som kommunikation mellan fordonet och dess släp [7] [8].

Med en maxhastighet på 1 Mbit/s tillåter CAN-bussen kablar så långa som 40 meter. Kabelns längd är begränsad av ljusets hastighet. På Kvasers webbsida finns en skämtsam kommentar om att åtgära problemet:

*“A proposal to increase the speed of light has been considered but was turned down because of its inter-galactic consequences.”*

Det kan också tilläggas att desto längre kablarna blir, desto lägre blir också den maximala hastigheten i dem vilket följande exempel visar [2] [7]:

- 100 meter har en maximal hastighet på 500 kbit/s
- 200 meter har en maximal hastighet på 250 kbit/s
- 500 meter har en maximal hastighet på 125 kbit/s
- 6 kilometer har en maximal hastighet på 10 kbit/s

De kablar som vanligen används i CAN-nätverket är den partvinnade kabeln och enkelkabeln som används i SAE J2411. Dessa är två av få kablar som inte faller på kriteriet att impedansen måste ligga mellan 108 och 132 ohm. Den optimala impedansen är 120 ohm. Vad gäller kopplingarna som används för felsökning och uppdatering av CAN-systemet finns det inga egentliga standarder. De vanligaste är [7]:

- 9-pin DSUB (vanliga i bilar och rekommenderad av CiA)
- 5-pin Mini-C och Micro-C
- 6-pin Deutch (används i hydrauliska system på tyngre fordon)

### 3 Alternativ till CAN

Även om CAN är ett väldigt passande nätverk för bilindustrin så har det med nyare bilar kommit allt fler system som t.ex. adaptiva farthållare, automatisk parkering och mörkerseende kameror som reagerar på fotgängare eller djur. Dessa nya system använder sig av nyare teknik som ultraljudssensorer, radar, LIDAR (eng. *Laser Illuminated Detection And Ranging*), HD-kameror och infraröddetektorer. För att få dem att samverka måste de kunna kommunicera med sina respektive ECU:n (eng. *Electronic Control Unit*), vilka i sin tur binds ihop med ett inbyggt system. Detta medför att det krävs ett snabbare nätverk för att bilens inbyggda system skall kunna hantera all information som sänds mellan de olika noderna. Det finns redan nu bilmodeller som använder sig delvist av Ethernet för bandbreddskrävande applikationer som t.ex. multimedia, diagnostik och snabba uppdateringsenheter [9].

Det finns flera olika system som skulle kunna användas i framtidens bilindustri och i detta avsnitt beskrivs IEEE 802.1 AVB (eng. *Audio/Video Bridging*) och TTEthernet (eng. *Time-triggered Ethernet*) [10] [11]. ViCAN (eng. *Vehicular wireless CAN*) är ett koncept som med tanke på sin snabbhet inte skulle förbättra det nuvarande inbyggda systemet. Däremot är det intressant eftersom det delvis använder sig av trådlösa noder jämsmed det befintliga CAN-systemet vilket skulle minska på antalet kablar. ViCAN tas upp för att även om CAN i tiden minskade på antalet kablar så finns det fortfarande orsak att ytterligare reducera på dem [12].

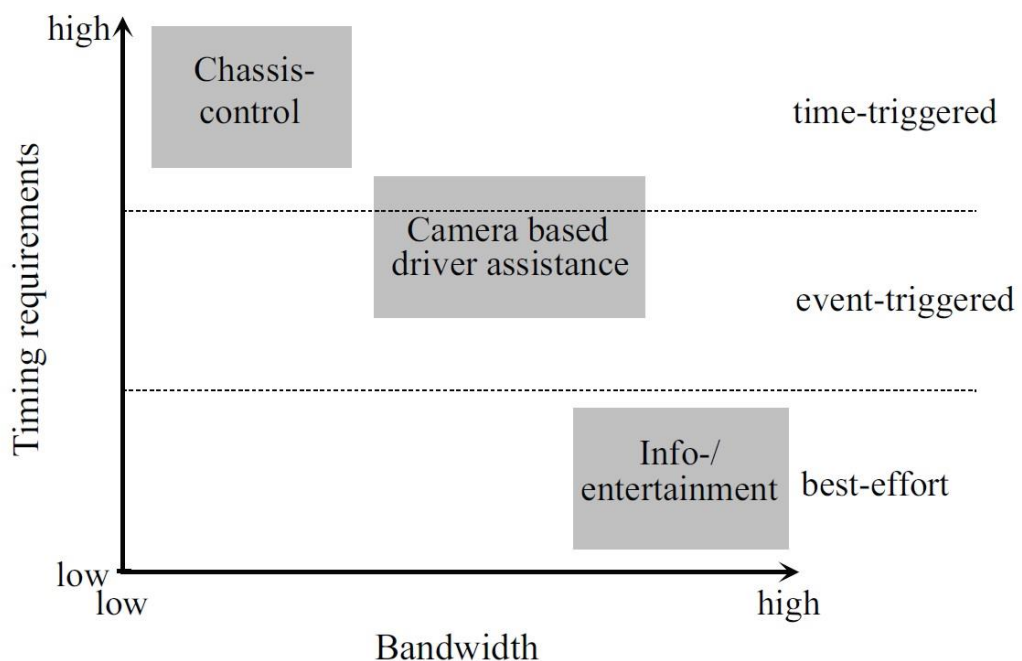
#### 3.1 Allmänt om Ethernet

I jämförelse med CAN så har ethernetbaserade system fortsättningsvis några nackdelar. Än så länge finns det inga lösningar som är lika tillförlitliga som CAN-bussen med att upptäcka fel. Ethernet behöver mera resurser, kablar och hade förr ett högt pris i jämförelse med CAN-kablar. Priset har sänkts mycket under de senaste åren och därför nu gjort ethernet till en möjlig lösning. I och med en lösning baserad på ethernet följer också mer etablerade komponenter av hög

kvalitet som leder till en betydligt högre systemprestanda med en hastighet på upp till 100 Mbit/s. Det kan konstateras att i motsats till CAN har ethernet inte någon fysisk längdbegränsning. Även om detta inte är det viktigaste i t.ex. en bil så är det ändå en fördel [13].

### 3.1.1 IEEE 802.1 AVB och TTEthernet

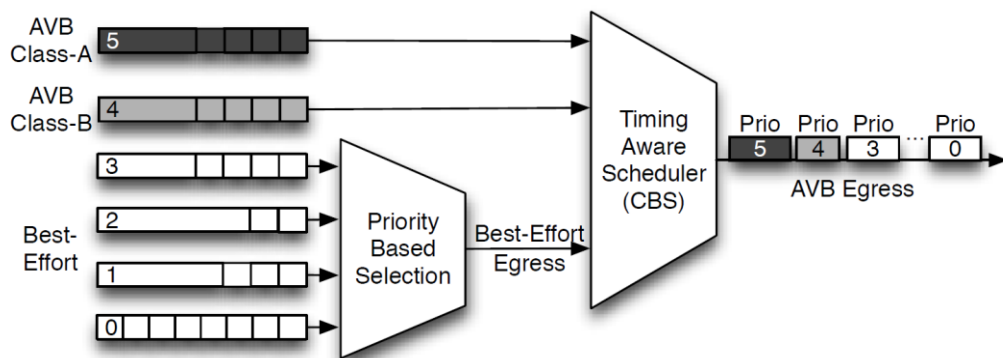
IEEE 802.1 AVB standarden är som förkortningen antyder en inriktning för att stöda större överföringshastighet för ljud och bild i en bil vars inbyggda system måste kunna sända stora datamängder från kameror eller sensorer. Standarden är specificerad av gruppen *Time-Sensitive Networking Task Group* med hjälp av det befintliga 802-nätverket och är baserad på ett tidssynkroniseringsprotokoll som gör att två noder i ett nätverk med sju noders mellanrum kan synkroniseras och ha en latens på högst 2 eller 50 ms beroende på vilken klass meddelandet har. Detta förklaras närmare på nästa sida [10]. TTEthernet eller *Time-triggered Ethernet* som nämndes ovan är också en möjlig kandidat för framtidens bilar med tanke på realtidskravet som ställs av sådana funktioner som behöver en hård deadline för systemets reaktion. I figur 5 visas en graf över vilka tidskrav som krävs



Figur 5 visar tre grupper i bilen och deras tids- och bandbreddskrav [11].

av några funktionsgrupper i en bil. *Chassis-control* är funktioner som t.ex. bilens antisladdsystem, automatiska bromsar (tillsvidare endast på konceptnivå) och krockkudden. *Camera based driver assistance* är t.ex. inbyggda backningskameror och *info-/entertainment* är den med tanke på tidskrav minst krävande gruppen som består av information som bilen ger åt chauffören och de underhållningssystem som kan finnas. Denna grupp kräver ändå en bandbredd som automatiskt kommer med ethernet. TTEthernet är utvecklat av TTEch och blev standardiserat av SAE år 2011 [10] [11].

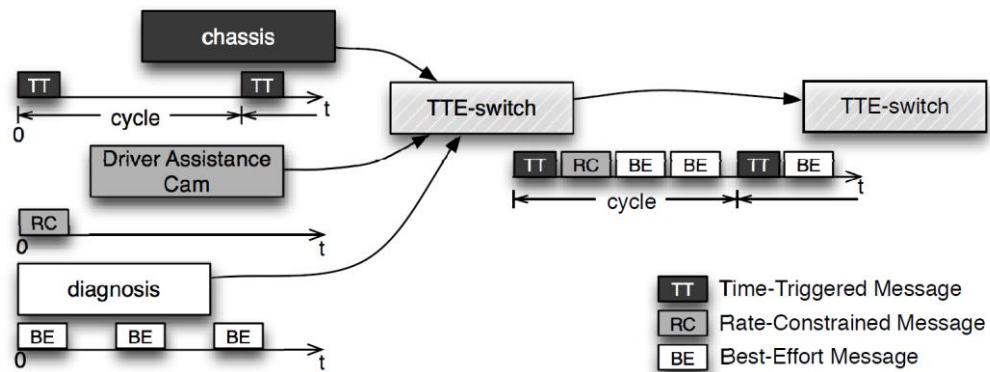
De båda standarderna fungerar både som realtidssystem och har algoritmer som väljer företräde för meddelanden med högre prioritet, vilket är nödvändigt i fordon. IEEE 802.1 AVB specificerar sina köer och regler för tidskänsliga applikationer i standarden IEEE 802.1Qav. De olika ramarna definieras av två huvudklasser, AVB klass-A som har högst prioritet och klass-B. Resterande ramar väljs inbördes i en algoritm enligt prioritet (lägre än A och B) som efter bästa förmåga (eng. *Best-Effort*) sänds vidare till en CBS (eng. *Credit Based Shaper*) som håller reda på synkroniseringen och vilka ramar som först får tillgång till bussen. Figur 6 är en beskrivande bild på detta [10].



Figur 6 visar vilka algoritmer som ingår i IEEE 802.1Qav och vidare i IEEE 802.1 AVB [10].

I TTEthernet definieras tre klasser för ramar som sänds. Den första är för tidsutlösande (eng. *TT: Time-Triggered*) ramar från t.ex. *Chassis-control* som nämns på sida 12. Detta kallas TDMA (eng. *Time-Division-Multiple-Access*) vilket betyder att ramar från denna grupp får ett förinställt schema med angivna

tidpunkter eller luckor med jämna mellanrum, där de har tid att sända. Den andra klassen är frekvensbegränsad (eng. *RC: Rate-Constrained*) och fungerar på ett liknande sätt. Det skapas sändningsluckor i en jämn takt med den skillnaden att de ramar som hör till klassen inte behöver bussen lika ofta och därför har längre tid



Figur 7 visar exempel på vilken ordning ramar får tillgång till TTEthernets buss [10].

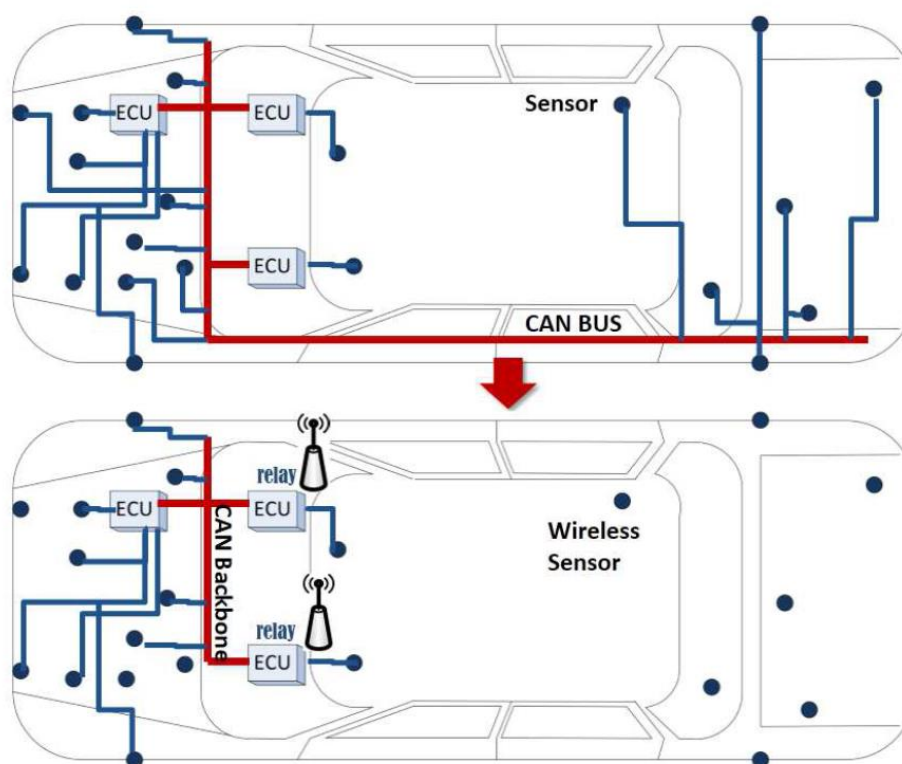
mellan sina sändningsluckor. Den sista klassen är resterande ramar som får sända efter bästa förmåga (eng. *BE: Best-Effort*). I figur 7 ses ett exempel på hur dataväxlarna väljer ramar med hjälp av TTEthernets protokoll [10] [11].

I [10] finns de två ethernetstandarderna simulerade i en prestandajämförelse. Det ställs samma krav som i en biltillämpning varvid det visar sig att båda klarar av realtidskraven och att de båda har unika för- och nackdelar. Där visar resultaten att precis som inriktningen säger klarar AVB-protokollet bättre av ramar med bild- och ljudströmmar jämfört med TTEthernet, men dess prestanda försämras däremot snabbare vid ökande ramstorlekar. TTEthernet klarar sig bäst när det gäller determinism och precision och lämpar sig därför bättre att transportera data med kritisk information som måste prioriteras [10]. I [14] har TTEthernet blivit implementerat vid sidan om CAN-bussen i en riktig Volkswagen Golf av årsmodell 2014 tillsammans med högteknologiska HD-kameror och 3D-laserskannrar och också där visat att detta koncept är fullt möjligt. Under ett par år framöver kommer prototypen att användas för flera tester och i forskning för att undersöka vad som måste ändras och vilka befintliga komponenter som bör designas om [14].

### 3.2 ViCAN

Laifenfield och Filosof [12] har i sin rapport tagit fram ViCAN eller *Vehicular wireless CAN* som är ett koncept för framtidens bilar med målet att minska på totala antalet kablar. I den typiska moderna bilen finns det uppskattningsvis 3,2 km kabel som sammanlagt väger mer än 23 kg. En trådlös lösning skulle åtgärda detta. Med en sådan lösning kommer nya utmaningar emot såsom dataöverföringens tillförlitlighet, latens, strömförbrukning och kostnader. Dessa utmaningar är en orsak till att endast några få trådlösa tillämpningar har stadgat sig i den moderna bilen. Ett par exempel är system för att övervaka däcktryck och fjärrstyrt centrallås [12].

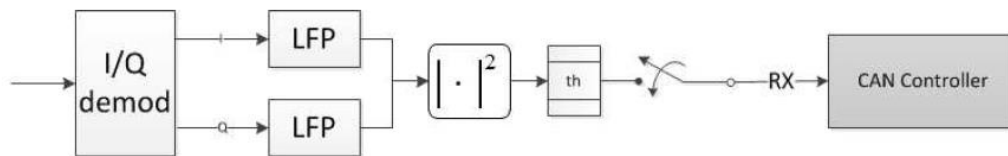
ViCAN-konceptet går ut på att i området närmast motorn behålls den normala uppsättningen av en CAN-buss, men alla sensorer bakom föraren ersätts med trådlösa noder. Ett exempel på skillnaden mellan CAN och ViCAN visas i figur 8 [12]. En fördel med ViCAN som koncept är att det kan använda sig av CAN-



Figur 8 visar hur en normal CAN-buss skulle ändras om man ersatte den med ViCAN [12].



kontrollern MCP2515 och bygger därför också delvist på den normala CAN-bussens fysiska skikt. Istället för att skilja på bitarna i ramarna så använder sig ViCAN och dess trådlösa sändare av OOK (eng. *ON-OFF Keying*). Detta innebär att när sändaren är "på" (*ON*) sänds en dominant bit respektive recessiv bit när den är "av" (*OFF*). Med "av" menas att den sänder med hälften mindre styrka jämfört med "på". Tack vare detta kan också sändaren i sitt recessiva stadium känna av ifall det är mycket annan trafik. I andra ändan finns en mottagare som tar emot



Figur 9 illustrerar ett förenklat blockdiagram över en mottagare i ViCAN [12].

sändarens signaler och för dem vidare på den vanliga CAN-bussen [12]. Laifenfield och Filosof [12] har illustrerat ett förenklat blockdiagram som ses i figur 9 och visar vilka delar en sådan mottagare innehåller. I/Q-demodulatorblocket tar in OOK-signalen och delar den till "på" och "av". Notera att sändaren har ett motsvarande modulatorblock som sänder en signal till mottagarens block. Därefter följer två LFP (eng. *Low-Pass filter*) som är förinställda på att märka om den ursprungliga signalen bestod av "på" eller "av" och därefter bestämma om en 0:a eller 1:a sänds vidare till CAN-kontrollern [12].

I sin rapport kommer Laifenfield och Filosof [12] genom analyser och simuleringar fram till att ViCAN kan tillämpas i en bil. Störningar från andra trådlösa applikationer och säkerhetshot gentemot systemet går att minimera genom ett noggrant val av frekvensbanden. Man kan också använda sig av metoder som frekvenshopp. ViCAN måste ännu vidareutvecklas innan man kan överväga att helt introducera det till bilindustrins marknad. Med en del optimering och fortsatta undersökningar om systemets tålighet har ViCAN goda förutsättningar i framtiden [12].

## 4 Avslutning

I sin text skriver Voss [13] att CAN-bussens position i bilen är ohotad för tillfället. Med en förväntad försäljning på 2 miljarder chips år 2015, varav 80 % för bilindustrin torde kunna konstateras att systemet inte byts ut inom den närmaste framtiden [15]. Även om de alternativ som tagits upp i denna avhandling befinner sig i tidiga utvecklingsstadier så har de tillsammans med liknande system ändå stor möjlighet att överta CAN-bussens roll i framtiden. Som påpekats måste en utveckling ske för att bilens inbyggda system skall klara av de krav som kommer att ställas i framtiden. Själv kan jag bara spekulera, men det redan etablerade ethernet kan potentiellt modifieras så att det helt och fullt kan uppfylla dessa krav.

En annan synvinkel på problemet är att ethernet tillsammans med den trådlösa CAN-hybridversionen skulle kunna ge en lösning, där den trådlösa delen sköter trafiken för icke krävande noder och ethernet skulle ta hand om den mer bandbreddskrävande delen. Det verkar sannolikt att det är just mot sådana hybrid som bilens inbyggda system utvecklas mot, att flera olika nätverk och protokoll tar hand om bilens olika funktioner beroende på vilket system som passar vilken funktion bäst. Redan nu finns det som nämns i avsnitt 3 bilar vars nätverk delvis består av ethernet för att ta hand om t.ex. bilens uppdateringsenhet. För att uppnå konkurrenskraftiga hybridssystem måste mera forskning, flera simuleringar och prototyper göras.

## Källförteckning

- [1] CAN in Automation, "CAN history," CiA, [Online]. Available: <http://www.can-cia.de/index.php?id=systemdesign-can-history#>. [Använd 31 3 2015].
- [2] W. Voss, *A Comprehensible Guide to Controller Area Network*, Copperhill Media, 2010.
- [3] P. Richards, "Microchip," 2001. [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/00754.pdf>. [Använd 31 3 2015].
- [4] I. R. Davis, A. Burns, R. J. Bril och J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, nr 3, pp. 239-272, 2007.
- [5] K. H. Johansson, M. Törngren och L. Nielsen, "Vehicle applications of Controller Area Network," i *Handbook of Networked and Embedded Control Systems*, Birkhäuser Basel, 2005, pp. 741-765.
- [6] F. Hartwich och A. Bassemir, "The Configuration of the CAN Bit Timing," i *6th International CAN Conference*, Turin, 1999.
- [7] Kvaser, "CAN Protocol Tutorial," Kvaser, [Online]. Available: <http://www.kvaser.com/can-protocol-tutorial/>. [Använd 31 3 2015].
- [8] CAN in Automation, "CAN physical layer," CiA, [Online]. Available: <http://www.can-cia.org/index.php?id=systemdesign-can-physicallayer>. [Använd 31 3 2015].

- [9] H.-T. Lim, B. Krebs, L. Völker och P. Zahrer, "Performance evaluation of the inter-domain communication in a switched Ethernet based in-car network," i *Local Computer Networks (LCN), 2011 IEEE 36th Conference*, Bonn, 2011.
- [10] T. Steinbach, H.-T. Lim, F. Korf, T. C. Schmidt, D. Herrscher och A. Wolisz, "Tomorrow's In-Car Interconnect? A Competitive Evaluation of IEEE 802.1 AVB and Time-Triggered Ethernet (AS6802)," i *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, Quebec City, QC, 2012.
- [11] T. Steinbach, F. Korf och T. C. Schmidt, "Real-time Ethernet for automotive applications: A solution for future in-car networks," i *Consumer Electronics - Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, Berlin, 2011.
- [12] M. Laifenfeld och T. Filosof, "Wireless controller area network for in-vehicle communication," i *Electrical & Electronics Engineers in Israel (IEEEI), 2014 IEEE 28th Convention*, Eilat, 2014.
- [13] W. Voss, "The Future of CAN / CANopen and the Industrial Ethernet Challenge," 2011. [Online]. Available: <http://www.esd-electronics-usa.com/shared/library/outsidearticles/thefutureofcan.pdf>. [Använd 31 3 2015].
- [14] T. Steinbach, K. Müller, F. Korf och R. Röllig, "Demo: Real-time Ethernet in-car backbones: First insights into an automotive prototype," i *Vehicular Networking Conference (VNC), 2014 IEEE*, Paderborn, 2014.
- [15] T. Carlsson, "The future of CAN and CANopen," HMS Industrial Networks, 17 2011. [Online]. Available: <http://www.anybus.com/readnews.asp?NID=135>. [Använd 31 3 2015].