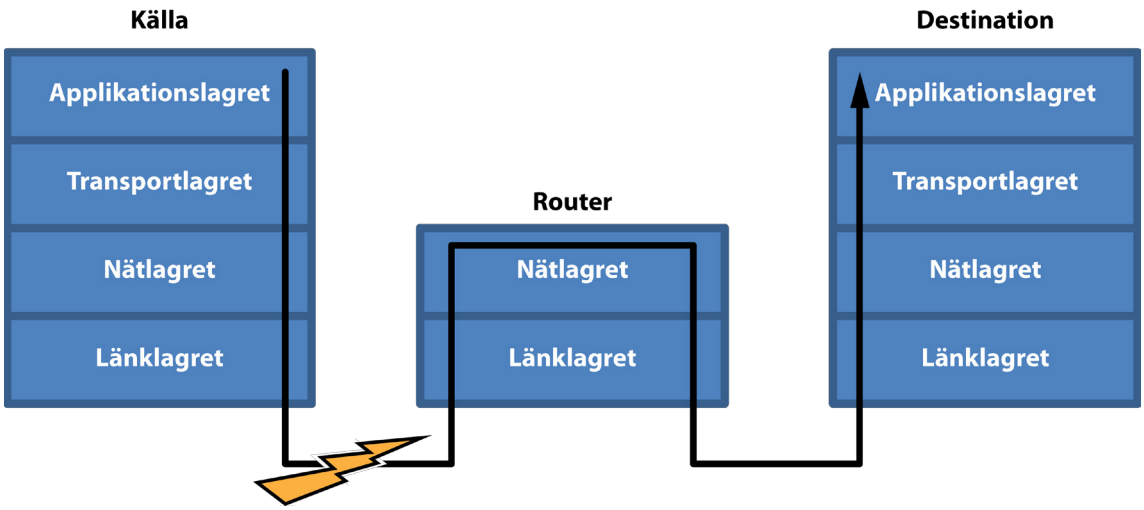
FOI

Effekter av elektromagnetiska störningar på TCP/IP-protokoll

JIMMI GRÖNKVIST, ANDERS HANSSON, KIA WIKLUNDH



FOI-R--4554--SE
ISSN 1650-1942 December 2017

FOI-R--4554--SE

Jimmi Grönkvist, Anders Hansson, Kia Wiklundh

Effekter av elektromagnetiska störningar på TCP/IP-protokoll

Titel	Effekter av elektromagnetiska störningar på TCP/IP-protokoll
Title	Effects of Electromagnetic interference on the TCP/IP protocols
Rapportnr/Report no	FOI-R--4554--SE
Månad/Month	December
Utgivningsår/Year	2017
Antal sidor/Pages	32
ISSN	1650-1942
Kund/Customer	FMV
Forskningsområde	4. Informationssäkerhet och kommunikation
FoT-område	Ledning och MSI
Projektnr/Project no	E720642
Godkänd av/Approved by	Christian Jönsson
Ansvarig avdelning	Ledningssystem
Exportkontroll	Innehållet är granskat och omfattar ingen information som är underställd exportkontrollagstiftningen.

Sammanfattning

Elektromagnetiska störningar på radiokanalen kan orsaka en prestandanedsättning i kommunikationssystem. Idag införs IP-baserad kommunikation på bred front i samhället och även i Försvarsmaktens radionät. IP är dock ursprungligen utvecklat för trådbundna nät där mottagningsförhållandena är annorlunda jämfört med vad som gäller för ett radiosystem. Eftersom TCP/IP protokollen inte är anpassade för de kanalvariationer som orsakas av elektromagnetiska störningar, är det viktigt att analysera vilka problem som kan uppstå. Störningar på det fysiska lagret fortplantas uppåt till högre lager i TCP/IP-stacken, där tjänster och applikationer påverkas. I rapporten undersöks konsekvenser av elektromagnetiska störningar på högre lager och vi beskriver hur protokollen påverkas av radiokanalens egenskaper. Av kompatibilitetsskäl bör inte TCP/IP-protokollen modifieras utan en omfattande standardiseringsprocess. Därför är målsättningen här att ge rekommendationer avseende länk- och applikationsdesign för att få effektivare kommunikationssystem. En slutsats från arbetet är att feldetektering på länklagret är viktigt eftersom oupptäckta bitfel i paketen kan påverka trafiken i nätet negativt.

Nyckelord: TCP, IP, elektromagnetiska störningar

Summary

Electromagnetic interference on the radio channel cause a performance reduction in the communication systems. Today, IP-based communications are introduced on a broad front in society and also in the Swedish Defence Forces' radio network. However, IP is originally developed for wired networks where the receiver conditions are different from those of a radio system. Because the TCP/IP protocols are not adapted to the channel variations caused by electromagnetic interference, it is important to analyse which problems that may occur. Disruptions from the physical layer are propagated up to higher layers in the TCP/IP stack, where services and applications are affected. The report examines the consequences of electromagnetic interference on higher layers, and describes how the protocols are affected by the radio channel's properties. For compatibility reasons, the TCP/IP protocol should not be modified without a comprehensive standardization process. Therefore, the goal here is to provide recommendations for link and application design for more efficient communication systems. One conclusion from the work is that error detection on the link layer is important, because undetected bit errors in the packets can adversely affect traffic in the network.

Keywords: TCP, IP, electromagnetic interference

Innehållsförteckning

1 Inledning 7

1.1 Översikt av TCP/IP 8

2 TCP/IP 9

3 Länklagret 12

4 Nätlagret 15

4.1 Sammanfattning 17

5 Transportlagret 19

5.1 Transmisson Control Protocol 19

5.1.1 Sammanfattning 21

5.2 User Datagram Protocol 21

5.2.1 Sammanfattning 22

6 TCP/IP-säkerhetsprotokoll: IPsec 23

6.1 Första fasen – nyckelutbyte 24

6.2 Andra fasen – dataöverföring 24

7 Robust Header Compression 26

8 Övriga TCP/IP-protokoll 28

9 Slutsatser 30

9.1 Länkdesign 30

9.2 Applikationsdesign 30

10 Referenser 32

1 Inledning

Elektromagnetiska störningar på radiokanalen kan skapa en prestandanedsättning hos kommunikationssystem. Fenomenen är välkända och för en radiomottagare finns flera typiska konsekvenser. Följande punktlista ger några exempel.

- Avbrott på förbindelsen
- Minskad radioräckvidd
- Missade anrop/meddelanden
- Ökad tidsfördröjning hos sända datameddelanden
- Positionsfel (GPS-mottagare)
- Ökad känslighet mot elektronisk attack (aktiv störning)
- Minskat upptäcktsavstånd hos varnarsystem och sensorer

Rent konkret beror dessa konsekvenser på att det uppstår bitfel i en digital mottagare till följd av störningarna, vilket gör att mottagaren tolkar delar av meddelandet felaktigt.

Begreppen IP och TCP/IP används ofta i en vid mening som benämning för den arkitektur för datakommunikation som ligger till grund för Internet. Vi följer detta språkbruk i rapporten. Begreppen är egentligen namn på två centrala protokoll som används för en mycket stor del av Internets data-trafik: *Internet Protocol (IP)* och *Transmission Control Protocol (TCP)*. Båda protokollen beskrivs närmare i kapitel 4 och 5. Idag införs IP på bred front i samhället och även i Försvarsmaktens radionät. Den främsta anledningen är att IP är standardiserat och underlättar och i vissa fall är en förutsättning för att kunna skicka information mellan olika system och radionät. IP skapar förutsättningar för så kallad sömlöshet mellan system och nät. IP är ursprungligen utvecklat för trådbundna nät, där störningar inte är lika vanliga, har en annan karaktär och där mottagningsförhållandena inte förändras nämnvärt. Eftersom de utvecklade protokollen inte är anpassade för en varierande kanal och elektromagnetiska störningar uppstår problem som är viktiga att genomskåda. Konsekvenserna är oförutsägbara och störningarna fortplantar sig från det fysiska lagret (radiokanalen) uppåt i till högre lager och får slutligen effekter på en viss tjänst eller applikation.

I rapporten undersöks hur TCP/IP-protokollen påverkas av radiobärare och deras egenskaper. Syftet är att översiktligt beskriva hur elektromagnetiska störningar påverkar högre lager i en radionod då IP-trafik skickas.

1.1 Översikt av TCP/IP

TCP/IP [1] är en standardiserad arkitektur med en rad ingående protokoll för hur datakommunikationen över datanät ska gå till. TCP/IP-protokollen har utvecklats och använts under lång tid. Många av de nuvarande protokollen standardiserades redan i början av 80-talet, baserat på arbeten som gjordes under 60- och 70-talen. Den stora spridningen kom dock med WWW och Internet i början på 90-talet och TCP/IP-protokollen kom att användas för att förmedla all trafik på Internet.

TCP/IP-protokollen var inte de enda standardprotokoll för nätverkstrafik som utvecklades. Andra exempel är OSI-modellens protokoll ATM, eller X.25(L3) etc., men TCP/IP är det protokoll som blivit det mest använda och är den dominerande standarden för datakommunikation idag.

TCP/IP utvecklades redan från början specifikt för att sammankoppla flera olika typer av trådbundna datanät. Detta åstadkoms genom att så kallade *gateways* översatte specifik information mellan näten (dessa kallades senare routrar).

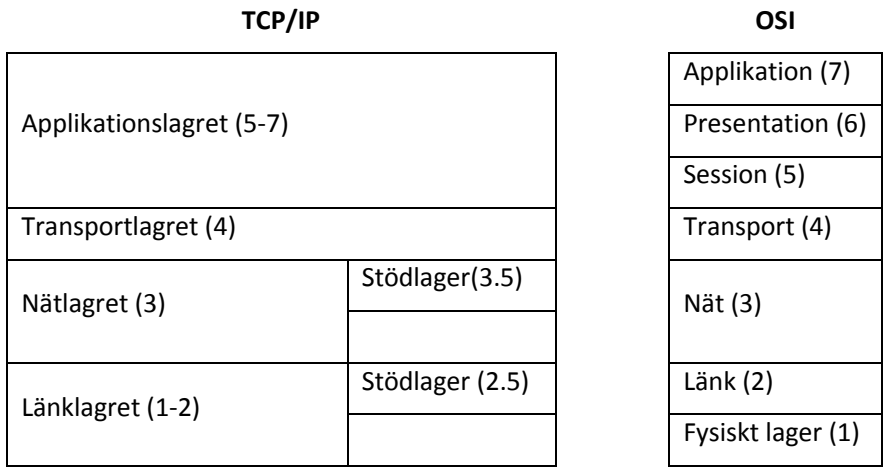
Allt fler av de militära systemen utformas för att hantera IP-trafik. Utvecklingen leder dessutom till ökad användning av IP även i de delar av systemet där IP tidigare normalt inte använts. Till och med HF-system som HF2000 och Marlin kan numera hantera IP även om huvuddelen av trafiken baseras på andra format. Även civil telefoni har blivit allt mer TCP/IP-baserad. På samma sätt som att tal inom 4G på sikt kommer att bli helt IP-baserat, kommer även VoIP att införas i de militära taktiska systemen. Försvarsmaktens nya taktiska radiosystem, Ra570, blir till viss del VoIP-baserat och om Buret Krypto införts hade taltrafiken i princip blivit helt VoIP-baserad.

I kapitel 2 ges en översikt av TCP/IP och kapitel 3 beskriver länklagret i TCP/IP-stacken. Sedan följer en beskrivning av konsekvenser av radiomediets egenskaper på TCP/IP-stackens nätlager och transportlager i kapitel 4-5. Kapitel 6 beskriver hur säkerhetsprotokollet IPsec påverkas och kapitel 7 tar upp komprimeringsprotokollet ROHC. I kapitel 8 behandlas ytterligare protokoll vars funktionalitet kan påverkas av överföring via radio. Sammanfattande slutsatser ges i kapitel 9.

2 TCP/IP

TCP/IP består av en mängd protokoll som tillsammans möjliggör all kommunikation mellan applikationer (datorprogram) på Internet. TCP/IP-protokollen klassificeras i fyra lager vilka har sitt ursprung i OSI-modellens sju lager, se [1]. OSI-modellen *Open Systems Interconnection standard* [2] är en lite mer detaljerad uppdelning utifrån protokollens funktion, men eftersom det i praktiken är svårt att separera funktionalitet på den nivån så grupperas några av OSI-modellens lager samman till ett lager i TCP/IP-modellen. Nedan beskrivs kort varje lager i TCP/IP-stacken tillsammans med exempel på de protokoll som finns i de olika lagren. Några av de protokoll vars funktion kan påverkas av radiosystemens egenskaper beskrivs sedan mer detaljerat längre fram i rapporten.

Applikationslagret (OSI-lager 5-7): Kan i princip vara vilket Internet-kompatibelt protokoll som helst, exempelvis HTTP, DNS, etc. I jämförelse med OSI-modellen innehåller TCP/IP-stackens applikationslager även funktionalitet från sessions- och presentationslagret. Vi analyserar inga av dessa protokoll, men föreslår istället allmänna designregler för detta lager.



Figur 1 Schematisk bild av TCP/IP-stacken till vänster och OSI-modellen till höger.

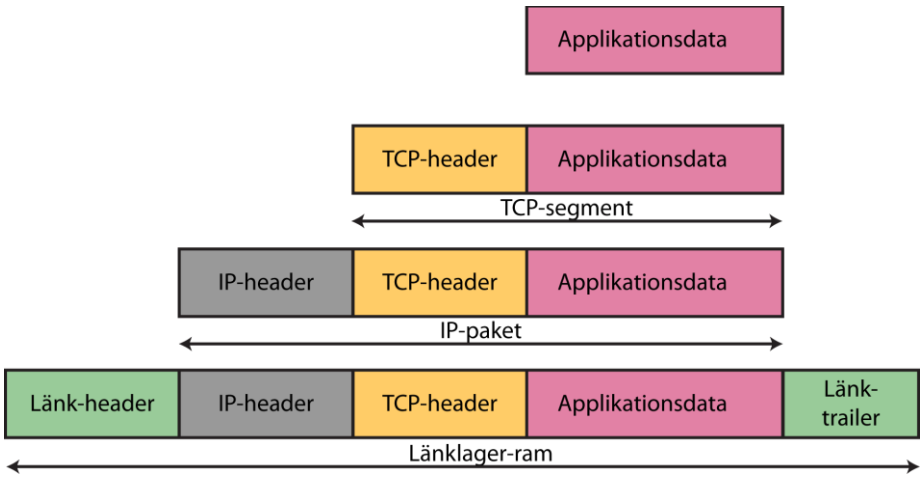
Transportlagret (OSI-lager 4): Innehåller protokoll för att identifiera och binda samman specifika applikationer på slutanvändarsystemen (hosts), det vill säga det ser till att användardata som nått destinationen också når rätt applikation hos mottagaren. I det här lagret kan förbindelsen göras mer tillförlitlig genom detektering av fel och återsändning av tappade paket. Använda protokoll här är i första hand TCP (Transmission Control Protocol) och UDP (User Datagram Protocol) som båda beskrivs mer nedan.

Nätverkslagret (OSI-lager 3): Innehåller protokoll som kan hantera multihoppkommunikation över olika typer av bärartekniker, till exempel olika radiotekniker och fast fiber. Nätlagret sköter adressering av noder och metoder för att hitta de olika noderna. IP (Internet Protocol) är det protokoll som krävs för att vara del av Internet. Det finns i två versioner: IPv4 och IPv6, där det sistnämnda idag införs mer och mer. IP avgör vilken väg paket skickas i nätet, men garanterar varken att paket kommer fram till mottagaren eller i vilken ordning de kommer fram. IP beskrivs mer senare.

Utöver IP förekommer andra protokoll i ett stödlager (lager 3.5 i Figur 1) till IP såsom ICMP, IGMP och IPsec. Även routingprotokoll, det vill säga de protokoll som håller rätt på var noder är i näten och vilken väg paketen ska skickas, kan läggas i detta stödlager.

Länklagret (OSI-lager 1-2): Innehåller protokoll som behövs för överföring över en enskild länk med funktioner som medium access control (MAC). Detta är en något förenklad beskrivning eftersom vidareförmedling även kan finnas inom ett transmissionsmedium, exempelvis en switch. Även länklagret kan ha ett stödlager (lager 2.5 i Figur 1), som bland annat används till att översätta IP-adresser till Lager 2-adresser (ARP).

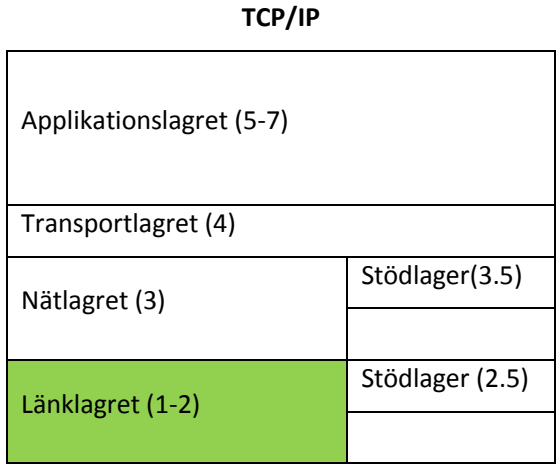
Interaktion mellan lager: När en applikation i en nod sänder data till en annan applikation i en annan nod skickas denna ner genom lagren i stacken. Lagrens protokoll kan då lägga till information i paketen, se Figur 2. Denna overhead ökar den faktiska mängden data som ska överföras på länkarna, men möjliggör TCP/IP-stackens funktionalitet. I mellanliggande noder skickas paketen enbart upp till nätlagret, eftersom information om rutten till destinationen hanteras i detta lager.



Figur 2 Exempel på headrar från protokoll på olika lager.

3 Länklagret

IP-protokollen designades ursprungligen för att kunna användas tillsammans med olika typer av bärartekniker, som realiseras på länklagret, se Figur 3. Även om en mycket stor andel av moderna transmissionsmedia karaktäriseras av hög kapacitet och få bitfel ska man komma ihåg att IP-protokollen har utvecklats under lång tid och att begreppet hög datatakt har förändrats efterhand. När till exempel modemuppkopplingar började användas överfördes IP-trafik över länkar med jämförelsevis låga datatakter med tanke på vad som anses vara normalt idag. Modern applikationsdesign förutsätter dock ofta höga tillgängliga datatakter.



Figur 3 Schematisk bild av TCP/IP-stacken, med länklagret markerat.

I förbindelser mellan mobila enheter kan det vara svårt att uppnå höga datatakter. En mobil radiobaserad förbindelse skiljer sig från en fast länk på flera sätt, inte bara vad gäller datatakt.

- Mobila länkar kan tillfälligt försvinna, gå ner, för att sedan gå upp igen. En fast länk går normalt bara ner när ena sidan stängs av. Snabba tillfälliga länkavbrott orsakar variationer i näten som IP-protokollen normalt inte är designade för att hantera.
- Mobila länkar kan ha en snabbare variaration i datatakt än fasta länkar, beroende på vilken länkteknik som används. Länkar med förmåga att anpassa datatakten påverkar fördröjningar på paket och orsakar variationer i hur belastat nätet är. Variationer i paketfördröjningar kan

påverka vissa applikationer och även protokoll som använder paketfördröjningar för att mäta nätegenskaper.

- För mobila länkar finns en betydligt större risk för både detekterade och oupptäckta paketfel än för fasta länkar. Korta tillfälliga länkavbrott medför intermittenta paketfel, vilket ger skurar av paketfel lite då och då. I fasta länkar detekteras nästan alla paketfel och tappade paket orsakas normalt på grund av andra skäl än länkproblem. Paketförluster kan hanteras med omsändningar på länken, vilket också medför variationer i fördröjningen.
- I radionät kan kollisioner inträffa på grund av samtidiga sändningar från flera användare. Att transmissionsmediet delas mellan flera användare förekommer även i fasta nät, exempelvis med Ethernet-standarden, men i praktiken är det mer komplicerat att upptäcka och hantera kollisioner i radiobaserade nät. Kollisioner kan medföra varierande fördröjningar och paketförluster.
- Fragmentering innebär att datapaket avsiktligt delas upp i mindre delar och överförs i flera sändningar över länken. Beroende på hur omsändningar sker, kan en förlust av ett fragment leda till en förlust av hela paketet. Över radionäts-länkar med låg kapacitet och länkavbrott måste fragmenten ha en förhållandevis liten storlek, vilket leder till fler fragment per paket och därmed en högre risk att något av fragmenten tappas.
- När en länk bryts, kan gamla rutter brytas. En konsekvens av detta är att routingprotokollet tvingas sätta upp nya rutter som ersätter de gamla. Under den tiden som en ny rutt sätts upp kan då en stor mängd paket försvinna. Omrouting förekommer även i fasta nät, men eftersom mobila länkar är mer föränderliga och har lägre kapacitet kan det ta längre tid att upptäcka och reparera.

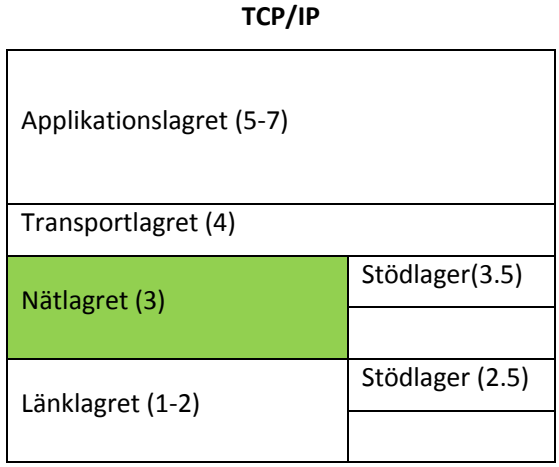
I TCP/IP krävs att vissa egenskaper uppfylls av länklagret. Bland annat måste länklagret hantera paketering av data, det vill säga att länklagret tar emot hela IP-paket från nätlagret och att motsvarande hela paket återlämnas till nätlagret på mottagarsidan. Det innebär att länklagret inte enbart hanterar en ström av data utan det identifierar även vilka bitar som hör ihop i paket. För radiosystem som inte hanterar detta behövs ofta extraprotokoll på lager 2, såsom PPP [3].

En viktig länkparameter är *maximum transmission unit* (MTU). MTU bestäms utgående från den maximala datamängd som länklagret kan överföra i ett paket från nätlagret. MTU kan vara fast för ett givet gränssnitt eller i en protokoll-standard. En hög MTU ger bra effektivitet i nätet, eftersom varje paket kan bära mycket användardata i förhållande till fast protokoll-overhead. En större MTU innebär också att färre paket hanteras för samma mängd användardata. En nack-del med en stor MTU är att stora paket som skickas över en långsam länk tar

längre tid än ett mindre paket, vilket ökar fördröjningen av efterföljande paket. Stora paket är också känsligare för kommunikationsfel. Om felkorrigeringen inte klarar att rätta bitfelen i paketet så måste det sändas om, vilket sänker nätkapaciteten.

4 Nätlagret

Protokollet *IP*, *Internet Protocol*, är i stort sett det enda förkommande på nätlagret i TCP/IP. I stort sett alla protokoll som beskrivs i den här rapporten kommunicerar via IP-paket. IP är en ”best-effort” och sessionslös paketförmedlingstjänst. Med ”best-effort” menas att inga garantier ges för att ett IP-paket når sin destination. Om paket-köerna blir för långa, exempelvis på grund av en överlastad router eller kabelbrott, så kastas paket. Om behov av tillförlitlighet finns så måste omsändningar av paket hanteras i högre lager. Sessionslös innebär att IP inte behåller någon information om de paket den förmedlar; alla paket hanteras oberoende av andra paket även om de ingår i samma session. Detta innebär till exempel att paket kan komma fram i fel ordning. All nödvändig information för paketförmedlingen finns i varje paket. De skickas oberoende av varandra i den meningen att själva förmedlingen av ett enskilt paket inte påverkas av vad som händer med andra paket som skickas mellan samma källa och destination.



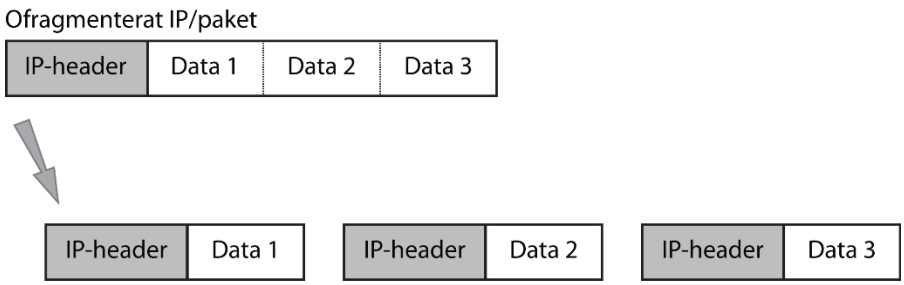
Figur 4 Schematisk bild av TCP/IP-stacken, med nätlagret markerat.

IP-protokollet finns i två versioner: IPv4 och IPv6. Versionen IPv6 är nyare och en uppenbar skillnad är ett betydligt större adress-fält i IPv6, 128 bitar jämfört med 32 bitar i IPv4. Därmed är den extra information som IP lägger till i varje paket (”header” på engelska) dubbelt så stor (40 bytes mot 20 bytes i IPv4).

En IP-header innehåller förutom källans och destinationens adresser en del annan information. För IPv4 finns bland annat fragmenteringsinformation, vilket används om det ursprungliga IP-paketet är för stort för att sändas helt över en

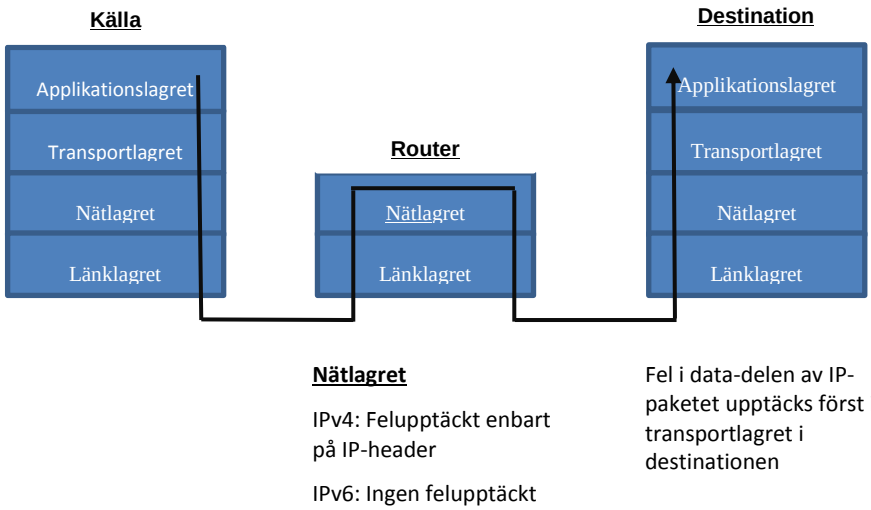
länk. Varje länk har ett MTU-värde (Maximum Transmission Unit) som avgör hur stora paket som kan hanteras. Ett IP-paket som är för stort delas ett upp i flera mindre paket (fragment) som vart och ett är tillräckligt litet för att hanteras av länken. Dock sätts inte IP-paketet ihop igen förrän i destinationen och om minst ett av fragmenten tappas kommer alla fragmenten kastas bort i destinationen.

Den overhead som orsakas av IPv4 ökar alltså väsentligt vid fragmentering eftersom varje fragment får en full IP-header (20 B), se Figur 5. I IPv6 görs ingen fragmentering på vägen; i stället bestämmer källan ett MTU-värde för rutten och fragmenterar IP-paketet redan före första hoppet. Dock kräver IPv6 ett MTU på minst 1280 B. Det här innebär att länklösningar bör vara utformade för minst så stora MTU-värden och fragmentering i mindre delar måste utföras på lägre lager istället, vilket kan leda till varierande fördröjningar.



Figur 5 Illustration av fragmentering.

IP har en begränsad förmåga att identifiera fel i paket. För IPv4 finns en checksumma som är till för att detektera fel i IP-headern. Den checksumman skyddar exempelvis mot att paket levereras till fel nod. Det finns inte någon kontroll av data-delen av IP-paketet. IPv6 har varken en checksumma för header- eller data-delen av IP-paketet. Odetekterade fel på länklagret kan därmed orsaka onödiga belastningar i nätet (övre lager kan ha skydd mot detta, men då har redan paketet nått destinationen), se Figur 6. Där källa och destination visas tillsammans med en router.

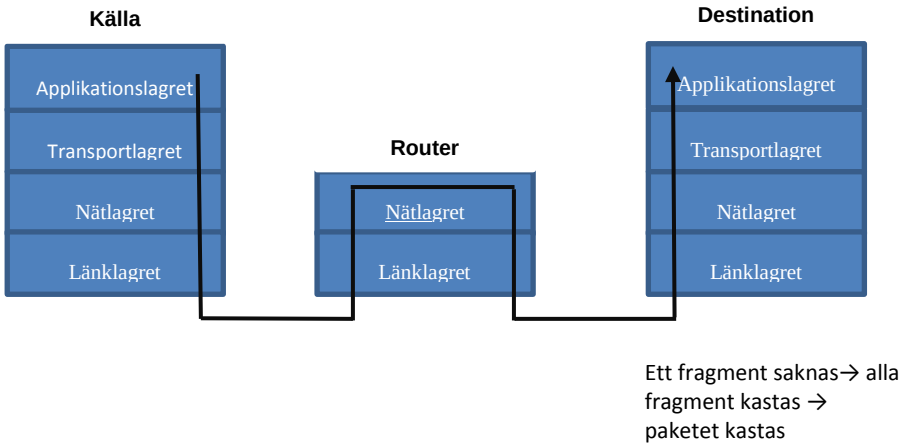


Figur 6 Illustration av effekter för nätlagret.

IP-protokollet i sig är inte känsligt för problem på radiolänken i den meningen att förlorade eller fördröjda paket påverkar hur protokollet utför sina uppgifter. Å andra sidan finns inga mekanismer för att hantera fel i lägre lager, vilket potentiellt kan orsaka spridningar till övriga nätet, det vill säga överföringsproblem i en radioförbindelse kan påverka andra noder längre bort i nätet. För kommunikation med små paket kan IP dessutom generera en märkbar overhead.

4.1 Sammanfattning

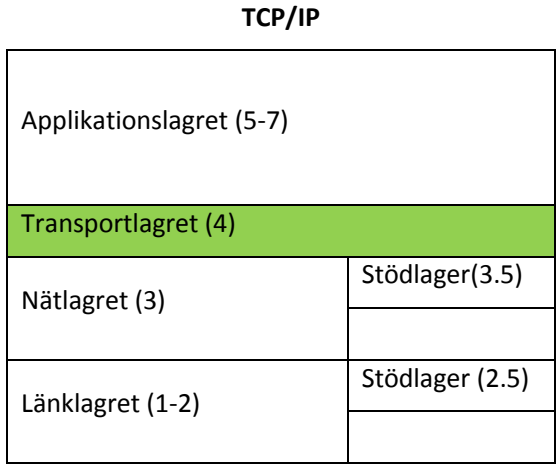
IP ger inga garantier för att ett IP-paket når sin destination och ansvarar inte för att paketen kommer fram i rätt ordning. IP-paketen delas upp i fragment och om minst ett fragment saknas, kastas alla fragment som tillhörde samma paket i destinationen. IP har även en mycket begränsad förmåga att upptäcka fel; IPv4 kan upptäcka fel i headern, medan IPv6 inte har någon förmåga alls att upptäcka fel. Tappade paket hanteras av transportlagret, se Figur 7.



Figur 7 Illustration av effekter för transportlagret.

5 Transportlagret

I det här kapitlet beskriver vi de två vanligaste protokollen på transportlagret.



Figur 8 Schematisk bild av TCP/IP-stacken, med transportlagret markerat.

5.1 Transmisson Control Protocol

Transmisson Control Protocol (TCP) är ett transportlagerprotokoll som ser till att paket överförs från källa till destination, mellan två applikationer, på ett kontrollerat sätt, se Figur 8. TCP återsänder paket som har slängts innan de nått destinationen och anpassar datatakten efter kapacitet och belastning längs rutten. En stor andel av all trafik på Internet är TCP-trafik, men protokollet hanterar dock enbart punkt-till-punkt-trafik.

Initialt skapar protokollet en uppkoppling mellan applikationerna; först därefter kan information överföras. Avslutningsvis tas uppkopplingen ner när inget mer ska överföras, vilket kan ske ofta om en applikation genererar data sporadiskt.

Med hjälp av IP-adressen kommer paketen fram till rätt destination i nätet. För att paketen sedan ska tas om hand av rätt applikation i destinationen, används dessutom *portar* (numrerade 0–65 535), en slags tilläggsadress i form av ett heltal. TCP delar upp data i lagom stora segment (så ingen ytterligare fragmentering av paket ska behöva ske) och lägger till en header med information om bland annat portnummer (för att identifiera applikationer) och

sekvensnummer för att kunna detektera tappade paket och fel ankomstordning för paketen. Utöver detta finns också en 16-bitars checksumma för att detektera fel i paketet. Felaktiga paket slängs bort och sänds om tillsammans med andra tappade paket. För att kommunicera vilka omsändningar som behövs, så har TCP dessutom en bekräftelsemekanism som uppdaterar ett ACK-nummer i ett fält i TCP-headern som anger nästa förväntade byte data och att all data fram till denna byte har emottagits korrekt.

Denna form av ACK gör det möjligt att skicka många paket utan separata bekräftelser per paket, och tappade ACK:ar kan hanteras utan extra återsändningar. TCP använder sig dessutom av ett så kallat ”sliding window” för att kontrollera hur många paket som maximalt får vara obekräftade. Dessa två mekanismer används även för att styra hur snabbt protokollet skickar data eftersom IP-routrar kastar paket när de blir överlastade. Genom att reglera fönsterstorleken så kan alltså TCP dynamiskt anpassa sändningstakten efter trafikbelastningen över rutten. Förenklat är principen att om packet tappas (missade ACK) så minskas fönsterstorleken, medan lyckade sändningar leder till en långsam ökning av fönsterstorleken. En tillfällig trafikstockning längs rutten kan därmed lösas upp.

Över en radioförbindelse mellan mobila noder inträffar dock paketförluster mer eller mindre sporadiskt på grund av störningar och kanalvariationer, med betydligt snabbare dynamik än på grund av ökade trafikbelastningar. Om TCP ofta sänker takten för paketsändningarna (med syftet att reglera trafikbelastningen), men paketförlusterna beror på tillfälliga variationer i radioförbindelsen, så underutnyttjas kapaciteten längs rutten.

Hur länge en bekräftelse får dröja styrs av TCP-parametern *retransmissions timeout* (RTO). Efter den tid som bestäms av RTO återsänds paketen. Om flera succesiva omsändningar görs ökas dessutom RTO-värdet. Kopplat till RTO finns parametern *roundtrip time* (RTT), tiden mellan en utsändning och mottagen bekräftelse på sändningen. RTO uppdateras kontinuerligt och skattas baserat på RTT. Kraftigt ökade fördröjningar, till exempel på grund av omroutning till rutter med lägre kapacitet, kan därför leda till onödiga återsändningar i näten. För att minska den risken är RTO-värdet även baserat på en skattning av variansen av RTT. Dock kan variansskattningar ge höga värden i många fall, speciellt om länklagret har en egen återsändningsmekanism. Höga värden på RTO gör att det tar lång tid att reagera på paketförluster.

Ett annat exempel på lågt kapacitetsutnyttjande är när TCP används över enkelriktade länkar [4]. TCP kräver dubbelriktad kommunikation för att fungera, vilket inte är möjligt för till exempel radiotysta noder. I tillämpningar med radiotysta noder förekommer dock ofta broadcast (en till alla) eller multicast (en till flera) och då används ett annat transportprotokoll, UDP.

Konsekvenserna av att använda TCP över radio är kända sedan länge och det finns många förbättringsförslag, men förändringar i TCP-standarden är en långsam process och drivkrafter saknas för att kommersiella produkter ska anpassas till radio.

En föreslagen metod är Explicit Congestion Notification (ECN) [5]. Det är ett tillägg till TCP där mellanliggande routrar kan lägga till information om överlast till datapaketen. TCP kan då reducera paket-takten innan förluster sker. TCP behöver därmed inte heller sänka takten lika mycket efter detekterade paketförluster. Detta hjälper inte fullt ut, eftersom alla routrar kanske inte använder ECN och vid faktisk överlast kommer routrar ändå att kasta paket.

För radiosystem väljer man ofta UDP som transportprotokoll istället för TCP, särskilt för system med höga paketförluster och bitfel över länkarna. Dock skickas TCP-trafik i stor utsträckning över 2.5G- och 3G-näten, så anpassning för att hantera vissa radioegenskaper är standardiserat för TCP [6].

5.1.1 Sammanfattning

Transportlagret kan upptäcka paketfel genom att det upptäcker att ett segment saknas eller att checksumman visar på fel. TCP återsänder paket som har slängts innan de nått destinationen och anpassar datatakten efter kapacitet och belastning. Effekter av paketfel och varierande datatakt blir för applikationslagret:

1. Upptäckt paketfel → Paket slängs → Omsändning av TCP → Fördröjning av information
2. Varierande datatakt på länklagret → Routrar kan bli överbelastade → Paket kastas → Paket sänds om av TCP → Fördröjning av information

5.2 User Datagram Protocol

User Datagram Protocol (UDP) är ett transportlagerprotokoll som kan användas som alternativ till TCP [7]. Det är i högre grad än TCP utformat för att ge en låg overhead. I princip innehåller UDP-paket enbart portnummer och checksumma för att detektera fel i paketet. Denna checksumma detekterar fel i UDP-header och data, samt också i delar av IP-headern. Därmed går det att kontrollera att paketet har nått rätt nod och att transportprotokollet stämmer (varje transportprotokoll har en unik uppsättning portnummer). Felaktiga IP-paket kastas bort, vilket även här innebär onödigt resursutnyttjande eftersom paket med fel kastas först vid destinationen. I IPv4 är denna checksumma valbar och används inte alltid. I IPv6 har den dock blivit obligatorisk även för UDP.

Till skillnad från TCP kan UDP användas för både unicast-trafik och multicast-trafik. I militära applikationer är gruppkommunikation vanligt och där används i stort sett enbart UDP som transportprotokoll.

Tack vare UDPs minimalistiska utformning påverkas protokollets funktion ganska lite av radiomediets egenskaper, men å andra sidan hanteras bara detektion av felaktiga paket. I UDP kontrolleras inte paketordning, paketstorlekar eller paketsändningstakt och tappade paket återsänds inte. Så alla sådana mekanismer måste hanteras i applikationerna.

5.2.1 Sammanfattning

Transportlagret kan upptäcka paketfel genom att checksumman visar på fel (valbar i IPv4, men obligatorisk i IPv6). UDP återsänder inte paket och anpassar inte datatakten efter kapacitet och belastning. Effekter av paketfel blir för applikationslagret.

1. Upptäckt paketfel → Paket slängs → Inga omsändning → Saknade paket
2. Varierande datatakt på länklagret → Routrar kan bli överbelastade → Paket kastas → Ingen omsändning → Saknade paket

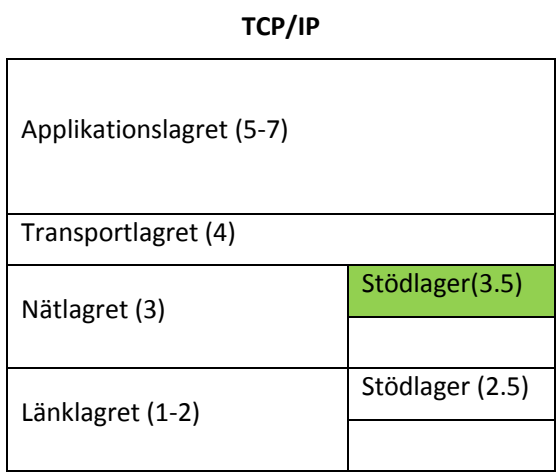
6 TCP/IP-säkerhetsprotokoll: IPsec

Det finns flera olika protokoll som kan användas till att skydda TCP/IP-baserade tjänster. Olika protokoll har olika egenskaper och skyddar olika saker. Alla protokollen utnyttjar kryptering och har mekanismer i alla lager som tidigare diskuterats. I den här rapporten kommer vi i huvudsak fokusera på IPsec som har sin funktionalitet på stödlagret (3.5), se Figur 7.

IPsec är en samling standarder som ger autenticitet, integritet, konfidentialitet, och tillgänglighetskontroll (access control) på nätlagret för både IPv4 och IPv6. I protokollet ingår också metoder att utbyta nycklar mellan två kommunicerande enheter. De kommunicerande enheterna kan vara hosts eller gateways som bildar en gräns mellan skyddade och oskyddade delar av ett nät. IPsec kan stödja multicasttrafik, men unicast-trafik är mycket vanligare.

IPsec kan hantera olika paket oberoende av varandra baserat på de policies som satts upp av administratörerna av systemet. Det innebär att vissa paket till vissa adresser kan krypteras på ett sätt, andra paket kan skickas helt oändrade (bypass) och ytterligare paket kan helt spärras från att sändas baserats på de regler som satts upp för systemet.

IPsecs användning kan delas i två faser: (1) en första fas, där nycklar med mera utbytes mellan källa och destination och en så kallad *Security Association* (SA) sätts upp, och (2) en andra fas, där data utbytes med två olika protokoll *Authentication Header* (AH) eller *Encasulation Security Payload* (ESP).



Figur 7: Schematisk bild av TCP/IP-stacken, med nätlagrets stödlager markerat.

6.1 Första fasen – nyckelutbyte

Första steget i en IPsec uppkoppling är att förhandla fram SAer, som i praktiken är parametrar som beskriver hur data ska sändas från en nod till en annan, såsom algoritmer, nycklar etc. Detta görs vanligtvis (på Internet) med ett protokoll som heter Internet Key Exchange (IKEv2) [8]. Med hjälp av IKE kan två noder skapa en gemensam symmetrisk nyckel över Internet. IKE är baserat på Diffie-Hellman. Protokollet är dock bara användbart för punkt-till-punkt-trafik. För multicast används andra protokoll för nyckelutbyte som är mer komplexa om man ska kunna hantera att noder tillkommer och försvinner eller enklare (fasta förkonfigurerade nycklar).

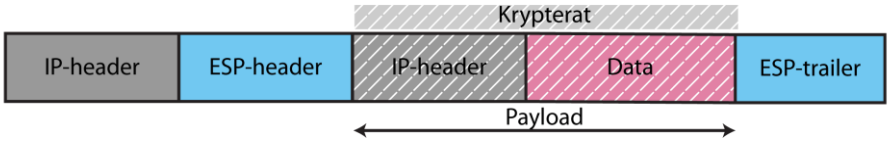
IKE-meddelanden sänds över UDP, vilket innebär att UDPs feldetekteringsskydd kan användas för att upptäcka fel i meddelanden. För att hantera tappade paket används sekvensnummer och tidsbaserade återsändningar i IKE. Återsändningstiden ökar för varje paketförlust så protokollet kan leda till stora fördröjningar om några av de första meddelandena tappas.

IKE är förmodligen inte speciellt använt i militära nät, däremot är militära krypton ofta baserade på den andra fasen av IPsec som beskrivs nedan.

6.2 Andra fasen – dataöverföring

Den andra fasen i IPsec dataöverföringen inträffar efter att SA-förhandlingen är avklarad. Det finns två olika protokoll för hur data hanteras: *Authentication Header* (AH) och *Encapsulating Security Payload* (ESP). AH används bland annat för integritet och autentisering av ett meddelande, dvs. att meddelandet är från rätt avsändare och inte ändrat på vägen. AH krypterar dock inte den sända datan. Det gör däremot ESP (utöver integritet och autentisering) vilket innebär att enbart rätt mottagare kan läsa meddelandet.

Båda protokollen har dessutom två moder: *tunnel-mode* och *transport-mode*. ESP är det klart mest använda av dessa. I ESP tunnel-mode kapslas hela IP-paketet (med headrar) in i ett nytt IP paket. Det inkapslade paket kommer vara krypterat och förutom den nya yttre (okrypterade) IP-headern finns nu också ett ESP-meddelande, som är delat i en header för det krypterade paketet och en *trailer* efter paketet, se Figur 9. ESP-headrar innehåller information om vilket SA paketet är kopplat till samt ett sekvensnummer. För vissa kryptoalgoritmer kan dessutom ESP-headern innehålla en separat initialiseringsvektor (IV).



Figur 9 Inkapslat krypterat paket.

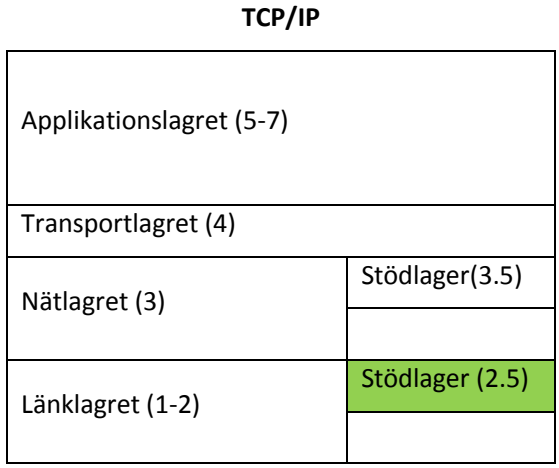
Tillsammans med ESP-trailer finns bland annat en *Integrity Check Value* (ICV), som används för att kontrollera om paketet har ändrats på vägen, vilket också skyddar mot oupptäckta fel, som leder till att paketet kastas bort. ICV är enbart beräknad utgående från ESP-meddelandet och payload, och omfattar inte yttre IP-header, men i praktiken spelar detta inte så stor roll. Användandet av ICV är dock inte obligatoriskt.

ESP-krypterade paket med ICV innehåller tillräckligt med information för att kunna hantera både paket i felaktig ordning, tappade paket och felaktiga paket. Dock sker detta till en rätt stor overheadkostnad i alla fall för små paket. ESP i tunnel mode kräver minst 30 B plus ICV, oftast 12 B, utöver den vanliga overheaden av IP. Exempelvis innebär detta att ett krypterat VoIP-paket (som normalt skickas som IP/UDP/RTP med sammanlagd headerstorlek 40B) får minst 82 B overhead per paket, ibland mer.

Hur mycket extra overhead detta leder till kan inses med följande exempel: MELP är en talkodare som används i många militära radiosystem. MELP generar ett 54 bitars talpaket var 22.5 ms. Den totala datatakten blir då 2400 bit/s. Om dessa talpaket i stället skulle kapslas in krypterade VoIP-paket som beskrivits ovan, får vi istället 710 bitar var 22.5 ms, vilket totalt ger över 32 kbit/s. Detta är mer än 13 gånger högre än den faktiska informationsmängden.

IPsec har inte några större problem med radiomediet eftersom paketförluster bör kunna hanteras väl. Eventuellt kan det dock finnas problem med första fasen, själva nyckelutbytet, se avsnitt 6.1. Dock kan overheaden potentiellt bli väldigt stor för små paket, som exemplifierats ovan. Något som kan hanteras med hjälp av komprimering av headrar, se kapitel 7 .

7 Robust Header Compression



Figur 8: Schematisk bild av TCP/IP-stacken, med nätlagrets stödlager markerat.

Robust Header Compression (ROHC) är ett standardiserat protokoll för att komprimera IP, UDP, UDP-Lite, RTP och TCP headrar på IP-paket [9]. Komprimeringen görs normalt över enskilda hopp, det vill säga ROHC kan ses som funktionellt på lager 2.5, som ett stödlager under IP, se Figur 8. Men protokollet används även i tunnlar mellan gateways som använder exempelvis IPsec.

IP-header-kompression används i första hand för paket med lite payload, som tal, då headrar blir en stor del av total sänd data.

Idén är att utnyttja att det finns mycket redundant information både inom ett IP-pakets olika headrar och framför allt sammantaget i flera IP-pakets headrar. Inom ett paket finns exempelvis längdinformation, både i IP-och UDP-headern och mellan paket finns exempelvis IP-adresser i varje paket.

Redundant information sänds enbart i de första paketen i en ström. Efterföljande paket innehåller enbart information som varierar från paket till paket. Dessutom skickas sådan information komprimerad om möjligt, exempelvis genom att skicka information om förändringar.

Under visa omständigheter kan ROHC komprimera en IPv4/UDP/RTP 40-bytes-header ner till 1 B. Detta kräver dock en back-kanal med relativt snabb återkoppling och inte speciellt mycket fel på kanalen. Dessutom kräver det att UDP har sin checksumma avstängd. Även under mindre ideala förhållanden kan header-informationen reduceras ner till några få bytes.

ROHC har en egen checksumma för att upptäcka fel i överföringen. Dock består den bara av några få bitar; hur många bitar beror på vilket typ av ROHC-paket som sänds, 3, 7 eller 8 bitar. Därför är risken för oupptäckta fel ganska stor vilket kan orsaka felfortplantning även till andra paket. I [9] rekommenderas att länken även bör ha egen felrättning och detektering, men specifika krav på länken ges ej. Det uppges dock att med en bitfelshalt på 10^{-5} uppstår lika många fel genom felfortplantning som det ursprungligen fanns.

ROHC antar att lägre lager ger information om paketlängd. Ursprungligen antogs också att paket inte ändrar ordning eller dupliceras, men uppdaterade kompressionsprofiler i RoHCv2 [10] tillåter att några få paket ändrar ordning, dock till kostnaden att färre konsekutiva paketförluster kan hanteras.

RoHCv1 kan normalt hantera upp till 14 konsekutivt tappade paket vid normalinställningar, men vid något mindre kompression kan betydligt fler hanteras. Sannolikheten för detta beror på länkens egenskaper samt pakettakten. Många snabbt skickade paket i en ström gör ROHC mer känsligt för skurfel på länken.

Header-kompression kan vara ett effektivt redskap för att minska den overhead som IP-protokollen orsakar och ROHC är gjort för att vara robust mot problem som uppstår i trådlösa länkar, genom att exempelvis kunna hantera många tappade paket. För att det praktiskt ska ge fördelar krävs att länken ger relativt få oupptäckta fel.

8 Övriga TCP/IP-protokoll

Ovan beskrivna protokoll är bara några av alla protokoll som används i IP-nät. Den absoluta majoriteten av datatrafiken över Internet kommer använda TCP eller UDP och all datatrafik använder IP. Utöver tidigare beskrivna protokoll, finns en lång rad stödprotokoll som är nödvändiga för att få näten att fungera och vars funktionalitet kan påverkas av överföring via radio.

Nedan är några sådana exempel:

Internet Control Message Protocol (ICMP) är protokoll för felmeddelanden, routinghjälp och diagnostik i IP-nät. Det ses som nödvändigt i alla IP-implementationer. Protokollet utnyttjar en checksumma för att detektera fel, men effekten av många tappade eller fördröjda paket är inte helt klar. ICMPv6 är viktigare för IPv6 än vad ICMPv4 är för IPv4.

Address Resolution Protocol (ARP) är ett protokoll som översätter IP-adresser till de adresser som används lokalt på länklagret. ARP är generiskt nog att hantera översättning mellan olika typer av länk och nätprotokoll, men används nästan uteslutande för att översätta mellan Ethernet/Wi-Fi och IP. ARP i sig har dock inget skydd mot fel utan förlitar sig på länkprotokollet. Däremot återsänds förfrågningar som ingen svarar på vilket gör att vissa paketförluster kan hanteras. Eventuella problem med ARP är dock förmodligen enbart intressant att studera för WiFi-nät.

Utöver detta finns olika typer av routing-protokoll (det vill säga protokoll som bestämmer vilka vägar IP-paket ska ta i näten. Specifikt kan nämnas *Open Shortest Path First* (OSPF) [11], vilket är ett av de mest använda protokollen på Internet. OSPF är dock i första hand utvecklad för rätt oföränderliga nät och kan ta lång tid på sig för att upptäcka att länkar gått ner.

OSPF hittar punkt-till-punkt-rutter och för att sända multicast-trafik krävs en annan typ av routing-protokoll. Vanligast är Protocol Independent Multicast - Sparse Mode (PIM-SM) [12].

Protocol Independent Multicast (PIM) är ett familj multicast-routing protokoll, det vill säga det används för att avgöra hur gruppkommunikationspaket ska vidaresändas. PIM finns i flera varianter med olika egenskaper, men ett gemensamt drag är att PIM inte själv detekterar topologin utan tar den från andra routing-protokoll, såsom OSPF. Detta innebär att problem hos unicast-routingprotoll kan spridas till multicastroutingen.

Internet Group Message Protocol (IGMP) är ett protokoll som används av hosts för att ta tala om för routrar vilka multicastgrupper de är intresserade av. Protokollet används i IPv4. I IPv6 skickas denna information som del av ICMPv6. IGMP använder checksummor för att detektera fel.

Real-Time Transport Protocol (RTP) [13] är ett protokoll för att överföra bland annat tal och video över IP-nät. RTP skickas vanligtvis över UDP och används i kombination med RTP Control Protocol (RTCP), där det sistnämnda används för statistik och QoS-mätningar. RTP är det överlägset mest använda protokollet för VoIP på Internet.

I varje RTP-paket finns tidsstämpel och sekvensnummer för att kunna hjälpa applikationer att spela upp multimedia-data korrekt. Detta möjliggör också detektering av tappade paket och paket i fel ordning. Däremot har RTP ingen mekanism för återsändning utan det får applikationer göra vid behov.

9 Slutsatser

Vi har berört olika IP-protokolls känslighet för problem på radiokanalen vid trådlös kommunikation. Eftersom protokollen i sig normalt inte kan modifieras, ger vi här några slutsatser och kommentarer för hur detta påverkar länkdesign och applikationsdesign.

9.1 Länkdesign

Det finns många egenskaper som kan orsaka problem för olika protokoll, men oupptäckta fel på länken är förmodligen det som är mest allvarligt. Många protokoll kan hantera paketförluster eftersom det är en vanligt förekommande företeelse vid transmissioner över Internet. Däremot kan specifika protokoll som TCP bli ineffektiva av paketförluster. Någon form av feldetektering är därför en väldigt viktig funktion för radiosystemet om IP-trafik ska fungera väl.

Långa uppkopplingstider är sällan ett problem för de protokoll vi diskuterat här. Specifika applikationer kan dock fungera dåligt om de inte designats för detta. Dessutom kan TCP bli ineffektivt av variationer i fördröjning.

Hur fragmentering av paket utförs i radiosystem kan ha påverkan på IP-protokollens prestanda. Om kortvariga (starka) interferenser är vanliga kan enskilda fragment tappas. Om dessa inte återsänds (eller kan återskapas) på länklagret riskerar hela IP-paketen att tappas. Detta kan leda till stor dataförlust även hos system med förhållandevis bra felrättning.

En lösning för att hantera förluster är återsändning av fragment (eller hela paket). Detta kan dock skapa variationer i fördröjning vilket exempelvis TCP har problem med, därför bör återsändningen vara snabb. Alternativt kan stark felrättning användas för att undvika att fragment förloras och låta högre lager ta hand om eventuella återsändningar.

Vi har inte analyserat applikationers känslighet för radioförbindelser. Många applikationer har inte utformats för att användas i dessa miljöer, till exempel för att de kräver för hög datatakt. Att använda IP-protokoll löser inte dessa problem.

9.2 Applikationsdesign

För att applikationer ska fungera när delar av IP-nätet är radiobaserat är det relevant att applikationsdesignen tar hänsyn till detta. Antalet olika applikationer är mycket större än antalet protokoll på nätlagret och transportlagret och i slutänden har applikationsdesignen en stor påverkan på kommunikationssystemet.

Valet av transportprotokoll (i praktiken UDP eller TCP) är en del av vad vi kan kalla applikationsdesign. TCP är ett vanligt val när det är viktigt att all data kommer fram. För strömmande applikationer, där enstaka förlorade paket inte måste eller hinner omsändas, passar UDP bättre. Att använda UDP även för applikationer där all data måste komma fram innebär att någon form av återsändningsmekanism måste hanteras av applikationen. En sådan återsändningsmekanism leder inte nödvändigtvis till en bättre lösning än vad TCP ger. För applikationer som använder UDP måste dessutom paket som kommer i fel ordning kunna hanteras.

Oavsett vilket transportprotokoll som används måste låga datatakter, varierande fördröjning och varierande kapacitet kunna hanteras.

10 Referenser

[1] K. Fall W. and R. Stevens, *TCP/IP Illustrated, Volume 1, second edition, the protocol*, Addison Wesley, 2012.

[2] *Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*, ISO/IEC 7498-1:1994.

[3] W. Simpson, Editor, *The point-to-point protocol (PPP)*, Request for Comments 1661, juli 1994.

[4] H. Balakrishnan, m.fl., *TCP performance implications of network path asymmetry*, Request for Comments 3449, december 2002.

[5] K. Ramakrishnan, m.fl., *The addition of explicit congestion notification (ECN) to IP*, Request for Comments 3168, september 2001.

[6] H. Inamura, m.fl., *TCP over second (2.5G) and third (3G) generation wireless networks*, Request for Comments 3481, februari 2003.

[7] J. Postel, *User datagram protocol*, Request for Comments 768, augusti 1980.

[8] C. Kaufman, m.fl., *Internet key exchange protocol version 2 (IKEv2)*, Request for Comments 7296, oktober 2014.

[9] C. Bormann Editor, *Robust header compression (ROHC): framework and four profiles: RTP, UDP, ESP, and uncompressed*, Request for Comments 3095, juli 2001.

[10] G. Pelletier och K. Sandlund, *Robust header compression version 2 (ROHCv2): profiles for RTP, UDP, IP, ESP and UDP-Lite*, Request for Comments 5225, april 2008.

[11] J. T. Moy, *OSPF Anatomy of an Internet Routing Protocol*, Addison Wesley, 1998.

[12] B. Fenner, m.fl., *Protocol independent multicast - sparse mode (PIM-SM): protocol specification (revised)*, Request for Comments 7761, mars 2016.

[13] H. Schulzrinne, m.fl., RTP: *A transport protocol for real-time applications*, Request for Comments 3550, juli 2003.

FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

FOI
Totalförsvarets forskningsinstitut
164 90 Stockholm

Tel: 08-55 50 30 00
Fax: 08-55 50 31 00

www.foi.se