

Routing Information Protocol

Problem och lösningar

TDTS09 Datornät och internetprotokoll

Grupp: DOIP26

Erik Eloff, Annica Lewin

eriel743@student.liu.se, annle867@student.liu.se

Linköpings universitet

22 februari 2008

Sammanfattning

Internet har en given plats i dagens tekniksamhälle. Men hur fungerar det egentligen?

Rapportens innehåll behandlar hur data i ett nätverk skickas till rätt enhet. Mekanismen som sköter detta kallas routing, och använder sig av ett routingprotokoll för att informera andra enheter om nätverkets tillgänglighet.

Vi har riktat in oss på att analysera routingprotokollet RIP (Routing Information Protocol). Rapporten tar upp hur RIP fungerar och vilka begränsningar protokollet lider av. Vi visar också lösningar på dessa problem.

Vår slutsats är att RIP, trots sina begränsningar, fortfarande är ett bra alternativ att använda som routingprotokoll.

Innehållsförteckning

1 Inledning	1
1.1 Syfte.....	1
1.2 Avgränsningar.....	1
1.3 Bakgrund.....	2
1.3.1 Routing.....	2
2 Routing Information Protocol (RIP)	3
2.1 Routingtabell.....	3
2.2 Distansvektoralgoritmen.....	4
2.3 Olika versioner av RIP.....	5
2.4 Exempel.....	7
2.4.1 Nätverkets struktur.....	7
2.4.2 Uppbyggnad av routingtabell.....	8
3 Problem och lösningar	10
3.1.1 Delad horisont.....	10
3.1.2 Triggade uppdateringar.....	11
3.1.3 Synkronisering av uppdateringsmeddelanden.....	12
4 Utvärdering	13
5 Diskussion och slutsatser	14
Referenslista	15

1 Inledning

I dagens samhälle tar väldigt många internet för givet. Men hur fungerar det egentligen? Hur hittar informationen fram till rätt dator? Denna rapport kommer att behandla ämnet routing, vilket är den mekanism som levererar data till rätt enhet.

I kursen ”TDTS09: Datornät och internetprotokoll” har vi fått i uppgift att skriva en rapport. Vi valde då ämnet RIP som är en typ av routingprotokoll.

1.1 Syfte

Rapportens syfte är att undersöka vad ett routingprotokoll är. Mer specifikt har vi analyserat protokollet RIP. Vi vill även veta hur routingprotokoll fungerar i praktiken, dvs. vilken information som skickas mellan routrarna i nätverket.

Vi har valt att arbeta efter följande frågeställningar:

- Vad är ett routingprotokoll och vilken uppgift har det?
- Hur fungerar RIP i praktiken som routingprotokoll?
- Hur ser RIP-paket ut och vad innehåller de?
- Vilka begränsningar har RIP?

1.2 Avgränsningar

Vi har begränsat oss till att behandla routingprotokollet RIP i rapporten. Det finns andra routingprotokoll som fungerar baserat på andra teorier, men dessa tar vi inte upp.

Eftersom vi valt att inrikta oss på enbart RIP så innebär det att vi bara behandlar ett autonomt system, dvs. ett system som sköter sig självt där alla routrar använder sig av samma protokoll.

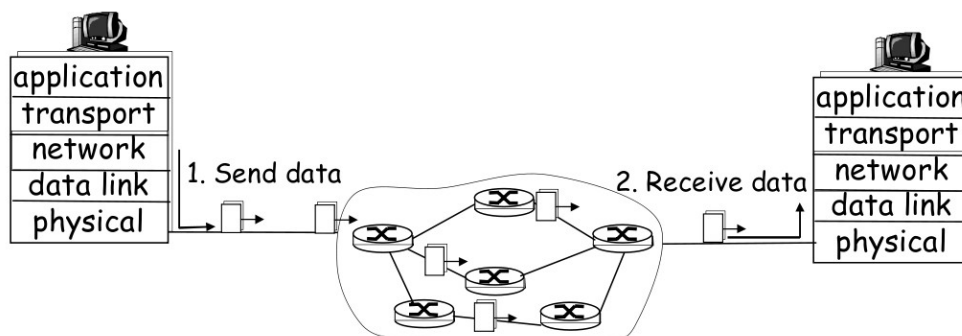
1.3 Bakgrund

1.3.1 Routing

En router är en nätverksenhet och är en väldigt viktig del i nätverket. Den har som uppgift att informationen som skickas av andra datorer kommer fram till rätt destination. [1]



I dagens nätverk (internet, företagsnätverk och hemnätverk) så används IP som nätverksprotokoll. Det innebär att varje enhet som behöver kommunicera med andra får en egen IP-adress. Kommunikationen sker genom att den data som skickas packas ner i så kallade paket. Paketerna innehåller all information som behövs för att routrarna ska veta vart paketen ska skickas. Det finns en begränsning på hur stora paketen kan vara, vilket innebär att en fil som skickas kanske delas upp i flera paket. Det finns inte någon garanti för att alla dessa paket transporteras samma väg genom nätverket för att nå sin destination [2].



Figur 2: Paketbaserat nätverk med flera routrar [9]

En router arbetar på IP-lagret där den avläser paketens destinations IP-adress. Routern ”vet” hur nätverket är uppbyggt, och kan då sända paketen vidare i rätt riktning; ett steg närmare sin destination. [3]

Hur vet då routern hur nätverket är uppbyggt? Med hjälp av ett så kallat routingprotokoll kan routrarna utbyta information om nätverket.

RIP är ett vanligt routingprotokoll. RIP utvecklades av företaget Xerox i början på 1980-talet. Det blev populärt då det följde med en av de tidigare versionerna av BSD UNIX 1982. [4]

2 Routing Information Protocol (RIP)

RIP står för Routing Information Protocol och är ett av de vanligaste routingprotokollen som används idag. [5]

Routingprotokollet har som uppgift att beräkna de bästa vägarna som paket kan skickas för att nå sina destinationer. För att kunna beräkna detta måste routern ha information om nätverkets uppbyggnad. [6]

RIP tillhör gruppen IGP, Interior Gateway Protokoll. Det innebär att RIP ska användas i autonoma system, dvs. ett system som sköter sig självt där alla routrar använder sig av samma protokoll. RIP var under många år det populäraste routingprotokollet [4] och används fortfarande i stor utsträckning [5]. Trots dess popularitet har det ett antal begränsningar som begränsar dess användande. [7]

Routingprotokoll kan använda sig av olika algoritmer för att beräkna den bästa vägen. Varje möjlig väg får ett värde uträknat som speglar kostnaden för vägen. Detta värde kallas för metrik. För att vägen ska kunna beräknas måste vi först definiera vad som ska räknas in i metriken. Routingprotokollet måste då ta hänsyn till omständigheter som till exempel antal routerhopp, bandbredd, trafik och administrativ kostnad. [4]

RIP använder sig enbart av antalet routerhopp [8], det vill säga antalet routrar som paketet måste passera på sin väg till destinationen. Dessa olika möjliga vägar måste sparas i routern, i en routingtabell eller forwarding table som den även kallas [3].

2.1 Routingtabell

Routingtabellen innehåller adresser till alla subnät inom systemet [4]. Subnät är ett sätt att dela upp ett nätverk i mindre delar. Till varje interface på routern ansluts ett subnät [1]. Alla enheter på samma subnät måste ha IP-adresser i samma intervall. [2]

I tabellen lagras den närliggande router som är närmast ett visst subnät och även kostnaden för subnätet. [4]

Tabellen används för att skicka paket vidare i rätt riktning. När ett IP-paket anländer till routern så läses destinationsadressen. Om adressen matchar ett nätverk i tabellen, så kommer paketet att skickas ett steg närmare målet. [4]

Om ett passande subnät inte hittas i tabellen så skickas ett felmeddelande tillbaka till sändaren som meddelar att nätverket inte är nåbart. [8]

Tabell 1 visar hur routingtabellen i Router 1 i Figur 5 kan se ut.

Tabell 1: Routingtabell A i ett stabilt nätverk

Subnät	Via	Metrik
10.0.1.0/24	10.0.1.1	1
10.0.2.0/24	10.0.4.2	2
10.0.3.0/24	10.0.4.2	3
10.0.4.0/24	10.0.4.1	1
10.0.5.0/24	10.0.4.2	2
10.0.6.0/24	10.0.6.2	3

2.2 Distansvektoralgoritmen

För att bygga upp routingtabellen behövs en metod som kan beräkna den bästa vägen till varje destination. Ekvationen som används för detta i RIP kallas Bellman-Ford. [4]

Algoritmen som RIP använder är en så kallad distansvektoralgoritm, som kortfattat innebär att varje router bara utbyter information med sina närmaste grannar. Enbart med den informationen kan varje enskild router göra bra vägval. [4]

Innan RIP har skickat någon information så innehåller routingtabellen bara information om de närmaste grannarna [4]. När RIP är aktiverat så kommer meddelanden att skickas till alla grannar varje halvminut [8]. Det som skickas är egentligen en kopia av routingtabellen. Om inget meddelande mottagits från en granne inom 3 minuter betraktas den grannen vara onåbar [8]. Detta innebär att metrikvärdet sätts till 16. [8]

Om information från två olika routrar beskriver samma subnät, så kommer bara den bästa av dessa två vägar att sparas, dvs. det vägval som enligt algoritmen har lägst metrik. Skulle informationen beskriva samma målnät och de har samma metrik för båda vägarna, så väljs den som annonserades först. [4]

Algoritmen klarar även av att det kan vara olika kostnad i olika riktningar [3], alltså att kostnaden från x till y inte är samma som från y till x.

RIP använder följande formel (Bellman-Ford) för att räkna ut lägsta kostnaden till alla tillgängliga subnät [9]:

$$d_x(y) := \text{lägsta kostnaden från } x \text{ till } y$$

$$v_1 \dots v_n := \text{grannroutrar}$$

$$c(x, v_n) := \text{kostnad från } x \text{ till granne } n$$

$$d_x(y) = \min \begin{pmatrix} c(x, v_1) + d_{v_1}(y) \\ c(x, v_2) + d_{v_2}(y) \\ \dots \\ c(x, v_n) + d_{v_n}(y) \end{pmatrix}$$

Kostnaden från x till y räknas ut på följande sätt:

För alla grannar v_1 till v_n summeras kostnaden från x till grannen med grannens kostnad till y. Detta resulterar i alla olika sätt som x kan vidarebefordra ett paket till y. Men eftersom bara det bästa alternativet, dvs. lägst kostnad, är intressant så är det bara den vägen som sparas i routingtabellen. Det som sparas i tabellen är alltså informationen att bästa vägen till subnät y går via den granne som ger det lägsta metrik-värdet.

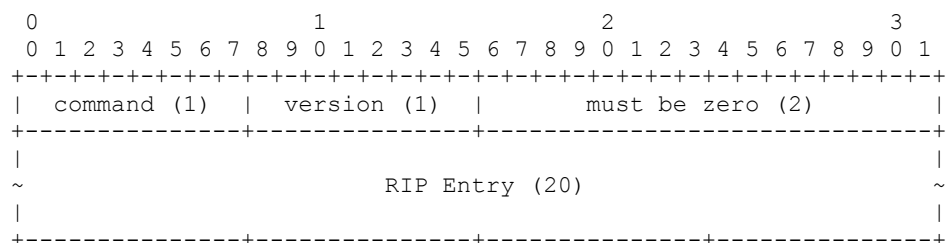
2.3 Olika versioner av RIP

Det finns två olika versioner av RIP. Den första kom i början av 1980-talet. Sedan dess har det gjorts en del tillägg i protokollet och 1994 kom RIP version 2. Protokollets meddelanden skickas i det format som visas i figur 3. Det som var en främsta anledningen till den nya versionen var behovet att kunna routa effektivt även över subnät. Detta har lösts genom att uppdateringsmeddelandet inte bara innehåller en IP-adress utan även slutmålets nätmask [10]. Nätmasken används för att skapa subnät i ett nätverk, vilket innebär att IP-adresser kan utnyttjas mer effektivt [9]. Säkerheten är även förbättrad i version 2. I version 2 är det nämligen möjligt att autentisera avsändaren av RIP-meddelandet. Om meddelandets lösenord är felaktigt kommer det omedelbart att förkastas. [10]

Version 2 har tre nya fält i sitt protokoll. ”Route tag” (Figur 4) kan användas för att visa vilken information som ursprungligen kommer från routrar i andra nätverk; de behöver inte nödvändigtvis använda sig av RIP. Då skickas normalt ett meddelande till den router som meddelat måladressen. [4]

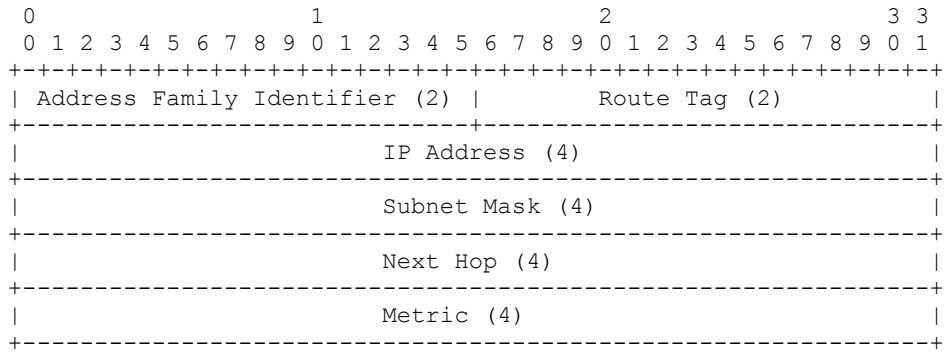
Ett annat fält som lagts till i protokollhuvudet är ”nästa hopp”. Om detta fält innehåller annat än nollor så kommer meddelandet att istället adresseras till fältet ”angiven router”. Funktionen är till för att i system med blandade routingprotokoll förhindra överflödiga hopp. [4]

Routrar som använder RIP skickar meddelanden via UDP på port 520 [10]. Det vill säga, de meddelanden som RIP skickar använder sig av transportlagret UDP, vilket innebär att RIP-protokollet är ett applikationslagerprotokoll. [3]



Figur 3: RIP-paket [10]

Figur 3 visar RIP-huvudet. Protokollet kan sedan innehålla information om upp till 25 olika nätverk i fältet *RIP Entry*. Varje nätverk beskrivs i formatat som figur 4 visar:



Figur 4: Route Entry [10]

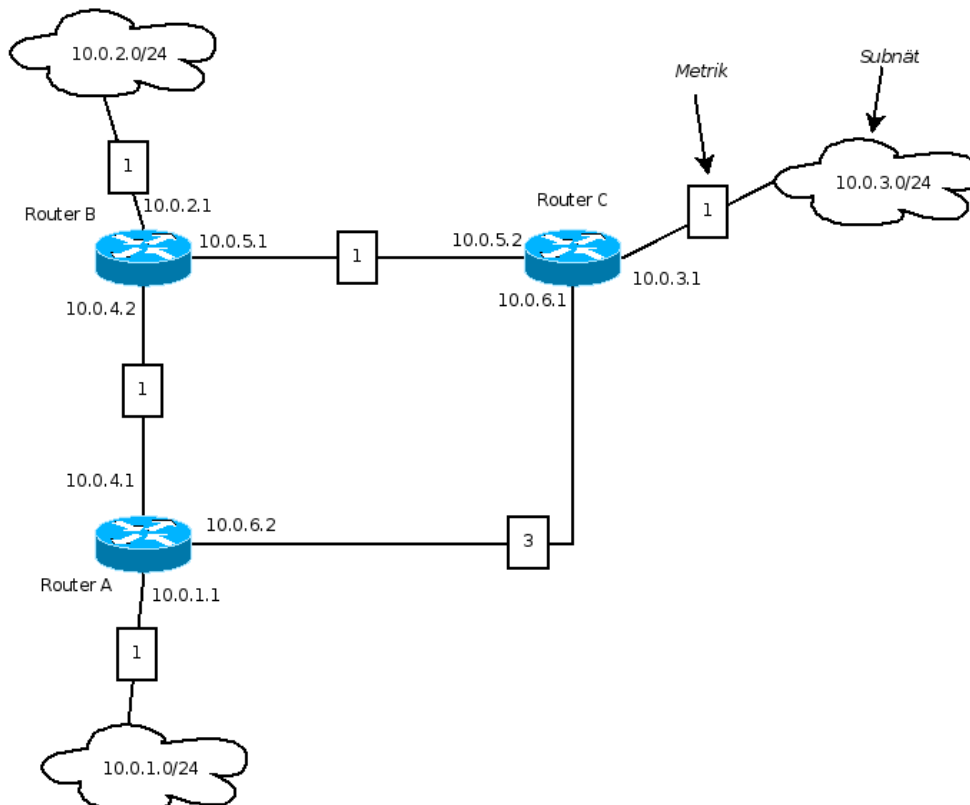
2.4 Exempel

I vår analys har vi valt att göra ett teoretiskt exempel på hur RIP fungerar och vilka meddelanden som skickas mellan routrarna i ett exempelnätverk.

2.4.1 Nätverkets struktur

Nätverk kan vara konstruerade på många olika sätt och att analysera routingprotokoll kan bli en komplicerad uppgift. Av den anledningen bygger den här rapporten i stor del på ett exempelnätverk enligt figur 5.

Nätverket består av tre routrar som alla är kopplade till varandra. Observera att förbindelsen mellan A och C har en högre kostnad.



Figur 5: Nätverket som exempen bygger på

2.4.2 Uppbyggnad av routingtabell

I detta exempel skall vi följa hur Router A:s routingtabell byggs upp med hjälp av RIP. Innan någon information skickats mellan routrarna så känner Router A enbart till de subnät som är direkt anslutna (Tabell 2).

Tabell 2: Routingtabell A innan RIP aktiverats

Subnät	Via	Metrik
10.0.1.0/24	10.0.1.1	1
10.0.4.0/24	10.0.4.1	1
10.0.6.0/24	10.0.6.2	3

Vid den första iterationen skickar alla routrar kopior av sina routingtabeller till varandra. Router A kommer nu att få information om de nätverk som är anslutna till B och C.

10.0.2.0/24 finns anslutet via B med kostnaden 1 + 1.

10.0.4.0/24 finns anslutet via B med kostnaden 1 + 1

10.0.5.0/24 finns anslutet via B med kostnaden 1 + 1.

10.0.3.0/24 finns anslutet via C med kostnaden 3 + 1.

10.0.5.0/24 finns anslutet via C med kostnaden 3 + 1.

10.0.6.0/24 finns anslutet via C med kostnaden 3 + 3

Router A kommer nu att veta om tre nya nätverk. Observera att 10.0.5.0/24 är tillgängligt via både B och C. Vägvalet via B ger lägst kostnad och är alltså det som sparas. Den uppdaterade routingtabellen i router A kommer att se ut så här efter första iterationen: (Tabell3, *Kursiverade* rader är uppdaterade)

Tabell 3: Routingtabell A då RIP skickat 1 meddelande till varje granne

Subnät	Via	Metrik
10.0.1.0/24	10.0.1.1	1
<i>10.0.2.0/24</i>	<i>10.0.4.2</i>	<i>2</i>
<i>10.0.3.0/24</i>	<i>10.0.6.1</i>	<i>4</i>
10.0.4.0/24	10.0.4.1	1
<i>10.0.5.0/24</i>	<i>10.0.4.2</i>	<i>2</i>
10.0.6.0/24	10.0.6.2	3

Även router B och C har uppdaterat sina tabeller; båda vet om varandras alla nätverk. Vid nästa iteration kommer dessa att skickas:

Alla subnät som meddelats tidigare skickas igen och dessutom ...

10.0.3.0/24 finns anslutet via B med kostnaden $1 + 2$

(eftersom B meddelar kostnaden 2 från B till C)

10.0.2.0/24 finns anslutet via C med kostnaden $3 + 2$

(eftersom C meddelar kostnaden 2 från C till B)

Eftersom 10.0.3.0/24 är tillgängligt via B med en lägre kostnad än vad som var känt tidigare, så kommer den vägen att uppdateras i tabellen.

Tabell 4: Routingtabell A efter andra iterationen

Subnät	Via	Metrik
10.0.1.0/24	10.0.1.1	1
10.0.2.0/24	10.0.4.2	2
<i>10.0.3.0/24</i>	<i>10.0.4.2</i>	3
10.0.4.0/24	10.0.4.1	1
10.0.5.0/24	10.0.4.2	2
10.0.6.0/24	10.0.6.2	3

3 Problem och lösningar

Att länkar i nätverk som använder routrar slutar fungera är inte ovanligt. När något sådant uppstår skulle paket kunna komma på villovägar och skapa loopar, istället för att fortsätta sin väg mot slutdestinationen. Den algoritm som RIP är uppbyggd av har i sig ingen lösning på detta problem. För att lösa detta problem med loopar och andra konflikter har RIP ett antal funktioner som ska försöka lösa dessa. [7]

För att hantera sådana situationer när nätverk eller länkar slutar att fungera ger routingtabellen ifrån sig metrikvärdet 16 för att visa att det inte längre är nåbart. [8] Detta värde ska representera oändligheten, det vill säga ett värde som talar om att nätverket är onåbart. Oändligheten är vald så att det överstiger största möjliga antal hopp mellan nätverk, men det ska samtidigt vara så litet som möjligt för att begränsa tiden att lösa konflikter. [7]

Då ett nätverk blir onåbart kommer grannarna att sätta metrikvärdet till 16 i tabellen efter ett tag. Efter det kommer de övriga routrarna i nätverket allt eftersom justera sina tabeller så att även de sätter nätet som onåbart. Sedan kommer grannarnas grannar att successivt öka metrikvärdet till 17, 18 och så vidare. Då oändligheten är satt till värdet 16 kommer dock även dessa grannar att sätta metriken till 16. Allt eftersom routrarna ger nätverket värdet 16 kommer det tillslut att av hela systemet att ha flaggats som onåbart. [7]

Det som i detta resonemang är viktigast att ta hänsyn till är hur lång tid det tar innan hela systemet har flaggat nätverket som onåbart. Detta problem kallas ”räkna till oändligheten” (count to infinity). Detta uppstår då de olika routrarna ska flagga nätverket onåbart. Då finns det nämligen en risk att de olika routrarna dras in i loopar där de lurar varandra då de försöker tala om att ett nätverk är onåbart. När de sänder felmeddelanden till varandra kommer metriken att ökas med ett steg i taget tills de når oändligheten eller en bättre väg föreslås av någon annan granne. Så efter ett antal varv blir systemet uppdaterat och det onåbara nätet tas bort ur tabellerna, men detta kan ta onödigt lång tid då looparna går om och om igen och alltså tvingas vänta på att oändligheten uppnås. [7]

Alltså vill man att ett så lågt värde som möjligt ska representera oändligheten så att looparna kan avslutas så snart som möjligt. Att räkna till oändligheten är dock en sista utväg för systemet. RIP har funktioner som påskyndar processen. RIP använder sig då av ”delad horisont med spärrad återgång” (Split horizon with poisoned reverse) och ”triggade uppdateringar” (Triggered updates) för att lösa problemet. Dessa metoder finns specificerade i RFC 1058 som beskriver den första versionen av RIP. [8]

3.1.1 Delad horisont

Problemet att räkna till oändligheten uppstår då olika routrar lyckas lura varandra då de ska flagga ett nätverk onåbart så att det uppstår loopar. Problemet kan förhindras genom att vara sparsam med de uppdateringsmeddelanden som sänds ut. Det är inte bra att en router berättar att den kan nå ett visst mål från de grannar den från början fått informationen ifrån. Funktionen ”delad horisont” [8] är till för att undvika problem orsakade av att vä-

ginformationen sänds tillbaka till den router från vilken den uppstod, det vill säga loopar. Det fungerar så att när routern ska sända uppdateringar till en granne om en viss väg utelämnar den den information som den fått av den grannen. Om routern har med sådan information i sina uppdateringsmeddelanden, men sätter den direkt till ”oändligheten” kallas det istället ”delad horisont med spärrad återgång”. [7]

”Delad horisont med spärrad åtgång” är vanligtvis mer pålitligt än ”delad horisont”. Anledningen är att två routrar har vägar som har hopp gemensamma så kommer en loop att genast att brytas om någon av dem flaggar ”oändligheten”. Vid endast ”delad horisont” skulle det krävas att time-out inträffade innan vägen kunde brytas. En nackdel med spärrad åtgång är dock att den ökar storleken på uppdateringsmeddelanden. Spärrad åtgång används dock fördelaktigt i nätverk där förändringar ofta uppstår. I stabila system vill man kunna minska onödig trafik genom att vid konfiguration kunna välja mellan de olika alternativen. Det ultimata skulle vara att vid förändringar i systemet använda ”spärrad återgång”, men att systemet sedan automatiskt återgår till endast ”delad horisont” när det åter igen blivit stabilt. [7]

3.1.2 Triggade uppdateringar

Att loopar mellan två routrar uppstår kan förhindras av ”delad horisont med spärrad återgång”, men problemet där tre routrar lurar varandra och skapar loopar återstår dock. Vid denna typ av problem är metriken fortfarande tvungen att nå värdet 16 innan loopen avbryts och nätverket flaggas att det inte är nåbart. [7]

För att påskynda upplösningen av sådana loopar används ”triggade uppdateringar”. Denna funktion innebär att då en router ändrar vägen till ett visst mål, så måste den direkt sända uppdateringsmeddelanden till alla, utan att invänta det meddelande som vanligtvis sänds var trettionde sekund [8]. Då en router får en uppdatering av information från en granne kommer den att räkna om metriken baserad på uppdateringen. Om resultatet av metriken då ändras, oavsett om den minskar eller ökar, måste routern direkt sända de uppdateringarna till sina grannar, som i sin tur kommer sända dessa vidare till sina så att informationen sprids över hela nätverket. [7]

Om systemet stod helt stilla medan den nya informationen spred sig skulle situationer där routrarna tvingades ”räkna till oändligheten” aldrig inträffa. Vägar till de länkar som inte längre kunde nås skulle då försvinna direkt, eftersom alla routrar omedelbart skulle nås av informationen att länken låg nere och då kunna eliminera den så att inga loopar skulle uppstå. Det är dock inte riktigt så verkligheten fungerar. ”Triggade uppdateringar” kan leda till problem genom att de kontinuerliga uppdateringarna kan nå routrarna samtidigt. Om routrarna då inte ännu nåts av informationen om de triggade uppdateringarna kommer de att fortsätta sända ut information om den länk som ligger nere. [7]

De felaktiga vägarna kan även återskapas om en router som först nås av den triggade uppdateringen sedan nås av det kontinuerliga meddelandet av en annan router som i sin tur inte nåts av de triggade uppdateringarna. Om de triggade uppdateringarna sker ögonblickligen minskar sannolikheten för att problem av denna typ ska uppstå. Om problemet ändå uppstår kommer det att lösas genom att routrarna återigen börjar räkna mot oändligheten. [7]

3.1.3 Synkronisering av uppdateringsmeddelanden

När det är många routrar i ett system brukar de vanligtvis synkronisera sig så att de sänder ut sina uppdateringsmeddelanden samtidigt [10]. Detta fenomen kan uppstå när belastningen på nätverket ökas. Denna synkronisering är inte önskvärd eftersom den kan leda till att kollisioner i nätverket uppstår, eftersom alla routrar skickar all trafik samtidigt [10]. Därför föreskriver RIP att tillverkare av routrar ska försöka förhindra detta genom att se till att routerns inbyggda timer inte påverkas av nätverkets belastning, eller att ge ett litet slumpat tal (0-5 sekunder) till timern varje gång den återställs till 30 sekunder. [10]

4 Utvärdering

RIP är konstruerat för att fungera inom ett autonomt system, som är ett nätverk som sköter sig själv internt [3]. Detta innebär en begränsning då det är avsett för mindre nätverk. Begränsningen beror på den distansvektor-algoritm som protokollet använder sig av. Max antal routerhopp som RIP kan hantera är 15 stycken, för att undvika loopar. Om kostnaden sätts till högre än 1 per länk kan det lätt bli problem med RIP. De 15 hoppen gäller en radie så RIP kan verka över flera hundra nätverk. [7]

Ett annat problem som RIP har svårt att hantera är när det uppstår loopar på vägen till en destination. Problemet för RIP blir då att om protokollet ska försöka lösa upp en stor loop, så kan det ta oacceptabelt lång tid och kräver alldeles för mycket bandbredd för att detta ska vara acceptabelt. [7]

Antalet hopp är den metrik som RIP använder för att jämföra vägval. Det är inte alltid antalet routerhopp som är det bästa att jämföra när routern ska bestämma sin väg till slut målet. Ibland kan det vara andra faktorer som tidsfördröjning, bandbredd, last eller tillförlitlighet som är bättre lämpade. RIP är inte avsett att utgå från dessa parametrar för att bygga upp sin routingtabell, eftersom det skulle leda till instabilitet hos nätverket. [4]

5 Diskussion och slutsatser

Routingprotokoll är väldigt viktigt för att nätverk ska fungera på ett bra sätt. Utan routingprotokoll så skulle man vara tvungen att manuellt konfigurera varje router i nätverket. I sådana fall kommer routrarna inte kunna ta hänsyn till förändringar i nätverket. Exempelvis om en länk går ner så kommer inte det meddelas till andra noder i nätverket, om inte ett routingprotokoll används.

Routingprotokollet RIP klarar av att dynamiskt hantera informationen om ett nätverks tillstånd. Uppdateringsmeddelanden skickas hela tiden för att informera närliggande enheter. De enheter som får meddelanden som visar på förändringar i nätverket skickar direkt vidare dessa förändringar till andra routrar.

RIP har en begränsning på maximalt 15 routerhopp. Det kan låta som väldigt lågt, men i verkligheten kan ett sådant nätverk vara väldigt stort. Begränsningen ligger i att avståndet från ena sidan till den andra i nätverket är max 15 steg. Det kan finnas många olika vägar ett nå dit. Hundratals routrar kan finnas inom räckhåll. Skulle nätverket vara större än så skulle det bli problem; RIP:s uppdateringsmeddelanden skulle då kräva en stor del av bandbredden.

Den Bellman-Ford-algoritm som RIP använder sig av har begränsningar. Det finns problem med att hantera loopar. Om det uppstår en loop så kan det ta lång tid innan den löses upp. Det kan även ta lång tid att markera otillgängliga nätverk som onåbara. I RFC1058 beskrivs metoder att kringgå detta [8]. RIP har implementerat dessa i form av *Delad horisont med spärrad återgång* och *Triggade uppdateringar*.

En annan begränsning är att metriken alltid utgörs av antalet routerhopp och det är inte alltid det bästa. Olika länkar kan ha väldigt olika prestanda; en fiberlänk är mycket snabbare än en modem-förbindelse. I sådana fall skulle det vara bättre att räkna in bandbredd och svarstid i metriken. RIP klarar inte av att hantera detta då det skulle leda till instabilitet i nätverket som RIP inte är anpassat för.

Numera finns det alternativa routingprotokoll som exempelvis OSPF, EIGRP och IS-IS är tekniskt bättre. Det som skiljer dessa protokoll är att RIP är lättare att implementera i ett nätverk då det kräver väldigt få inställningar. Vi tror att RIP kommer att fortsätta användas då det fungerar och används i många system [5].

Referenslista

- [1] Wikipedia, *Router*, <<http://en.wikipedia.org/wiki/Router>> [hämtat 9/2 2008]
- [2] Wikipeda, *Internet Protocol Suite*, <http://en.wikipedia.org/wiki/Internet_protocol_suite> [hämtat 9/2 2008]
- [3] Jeanna Matthews, *Computer networking: Internet protocols in action*. Wiley (2005)
- [4] Michael Seemann, *RIP - Routing Information Protocol, del 1*, Nätvärlden 1-97, E+T Förlag (1997)
- [5] Wikipedia, *Routing Information Protocol*, <http://en.wikipedia.org/wiki/Routing_Information_Protocol> [hämtat 9/2 2008]
- [6] Wikipedia, *Routing*, <<http://en.wikipedia.org/wiki/Routing>> [hämtat 9/2 2008]
- [7] Michael Seemann, *RIP - Routing Information Protocol, del 2*, Nätvärlden 2-97, E+T Förlag (1997)
- [8] *RFC1058: Routing Information Protocol*, IETF (1988)
- [9] James F. Kurose. & Keith W. Ross, *Computer networking: a top-down approach*. Addison-Wesley (2007)
- [10] *RFC2453: Routing Information Protocol version 2*, IETF (1998)