Mittuniversitetet
MID SWEDEN UNIVERSITY

DT123G Nätverksanalys

Laboration II

Presentation av data

Lennart Franked*Nayeb Maleki†

28 september 2015

Innehåll

Mål

I denna laboration ska vi nu kolla på hur vi på bästa sätt kan presentera den det data vi samlat in. Till er hjälp för detta kommer ni att arbeta med olika verktyg som visualiserar datat i form utav exempelvis grafer.

Syfte

Efter att du har avklarat *L2 – Presentation av data* har du uppfyllt följande delmål:

- kunna på ett korrekt sätt tolka utförda mätningar,

*E-post: lennart.franked@miun.se.
†E-post: nayeb.maleki@miun.se.

- presentera dina mätningar på ett överskådligt sätt, samt
- välja korrekt presentationsteknik beroende på vad som skall presenteras.
- diskutera presentationsteknik med dina studiekollegor, kunna försvara ditt val av presentationsteknik för dina utförda mätningar.

Läsanvisningar

Innan ni påbörjar laboration två skall ni ha läst [1, kap 8], därefter [3, kap 9.2]. Ni ska också ha bekantat er med [2] för att generera grafer med hjälp av python, samt MRTGs dokumentation [4] som ni senare kommer ha till hjälp för att sätta upp MRTG i moment 2.2.

1 Introduktion

Vi har nu kollat på olika verktyg för att samla in vårt data. Problemet vi har är att det är en ansenlig mängd data vi har (rådata) och data in denna form är oöverskådlig och ytterst olämplig att använda för att redovisa och presentera sina resultat för andra. Vi kommer därför i denna laboration att kolla på ett antal olika sätt att presentera sitt data på och vilken presentationsmetod är lämplig till vilket ändåmål.

En stor del av nätverksanalys är att vi känner vårt eget nätverk. Detta innebär att vi måste veta hur vårt nätverk ser ut under ordinarie förhållanden. För att få denna bild skapar vi en så kallad **baseline** över vårt nätverk genom att samla in data under en lång period av vanlig användning. Vi kommer därför att börja med att samla in data för att ta fram en baseline, och därefter med denna rådata presentera olika delar av vår nätverksanvändning.

2 Genomförande

Nedan följer de tre deluppgifter där vardera uppgift innehåller övningsuppgifter ni skall utföra för att lära er behärska olika presentationsmetoder.

Varje uppgift innehåller ett antal frågor som skall besvaras. I uppgift 2.2 samt 2.4 skall ni besvara samtliga uppgifter. I uppgift 2.3 ska ni enbart utföra en av uppgifterna. Vilken uppgift ni skall göra bestäms utifrån ert efternamn. Se tabell 1 på nästa sida.

Se avsnitt 3 för information om hur ni skall redovisa era resultat.

2.1 Baseline

För att plocka fram en Baseline måste vi först samla in data över vårt nätverksanvändande. För att få en så korrekt baseline som möjligt behöver vi samla

Tabell 1: Uppgiftsindelning

Efternamn börjar med:	Uppgifter som skall utföras
A - I	1
J - R	2
S - Ö	3

in data över en lång period. Detta medför förstås problem, då det rör sig om väldigt stora mängder data. Vi måste därför begränsa vår datainsamling till att enbart spara paketheaders. Detta är också det vanligaste sättet att utföra datainsamlingar på, då mängden data blir näst-intill ohanterlig om även payload skall sparas. Vid felsökning behöver vi oftast inte heller ha tillgång till Payloaden om det inte är något specifikt vi måste felsöka.

Det finns inget enkelt sätt att enbart spara headers, utan för att utföra detta måste vi själva begränsa hur många bytes per paket som vi skall spara. För att veta detta måste vi också veta vilka protokoll som finns på vårt nätverk. Börja därför med att i Wireshark starta en kortare insamling av data på cirka 15 min. Därefter i statistiken plocka fram vilka applikationslagersprotokoll som körs på just ert nätverk. Avslutningsvis kolla upp header-storleken på dessa och på så sätt kan ni räkna ut hur många bytes ni måste läsa in för varje paket för att vara garanterade att ni fångar in samtliga headers. När ni hittat detta värde så starta igång en insamling på er dator och låt den vara igång i 8 timmar[1]. Vilket verktyg ni väljer att använda för att samla in denna data får ni själva bestämma. Det viktiga är att ni kan spara ner resultatet i en fil som ni senare kan öppna i bland annat packet-tracer eller läsa in i ett pythonprogram.

2.2 Presentation med hjälp utav Wireshark

I Wireshark finns det redan inbyggt ett antal verktyg för en god presentation utav det insamlade datat. Majoriteten av dessa verktyg går att hitta under "Statistics". För detaljer kring hur dessa fungerar se Wiresharks officiella dokumentation.

1. Plocka fram en tabell som visar vilka externa servrar som din dator haft mest kommunikation med.
2. Plocka fram en graf som tydligt visar när på dygnet som aktiviteten mot dessa servrar var som högst.
3. Skapa en graf som visar omsändningar, dubbla ACKs och tappade segment. Se till så att varje fel representeras i grafen i olika stilar så att man kan se informationen även om bilden skrivs ut i svartvitt.
4. Skapa en valfri graf som visar något du anser vara viktigt att få en tydlig Baseline över.

[1] En seriös baseline bör sträcka sig över någon vecka eller till och med månad

2.3 Presentation med terminalverktyg

I den här deluppgiften skall vi utveckla våra kunskaper om python. Ni ska nu läsa in den pcap-fil ni samlade in i början av labben och därefter med hjälp utav dpkt och ytterligare paket som matlibplot skapa diagram. Diagrammet kan vara ett stapel- eller stolpdiagram, linjediagram eller cirkeldiagram.

1. Skapa ett valfritt cirkeldiagram.
2. Skapa ett valfritt stolpdiagram.
3. Skapa ett valfritt linjediagram.

2.4 Presentation med hjälp utav SNMP och MRTG

Avslutningsvis ska vi kolla på ett exempel på hur vi kan generera överskådliga grafer av data insamlat med SNMP genom att använda oss utav programmet MRTG. MRTG är ett simpelt verktyg som utför SNMP-polls och lagrar datat i en vanlig textfil. Från denna fil genereras sedan en graf med hjälp utav verktyget rrdtool som man kan komma åt via en hemsida som körs på en lokal http-server.

Gå in på MRTGs hemsida [4] och ladda hem den senaste versionen av programmet, alternativt kolla i er distributions pakethanterare om MRTG finns att hitta där. För detaljerad information kring installations- och konfigurationsprocessen se dokumentationen på [4].

Efter installationen är det två skript du måste köra innan du kan börja använda MRTG. Det första skriptet är `cfgmaker(1)` som du kan använda för att generera en grundläggande konfigurationsfil. Det andra skriptet är `indexmaker(1)` som används för att generera HTML-filerna som presenterar dina övervakningar.

Efter att du genererat din *MRTG.cfg*, kan du öppna denna med valfri textredigerare. Du borde nu ha en grundkonfiguration där du kan hitta information om vilken katalog den skall skapa graferna och placera HTML-filerna i. `Cfgmaker` kommer även att hämta den systeminformationen du fyllde i då du konfigurerade din SNMP-agent i en tidigare laboration. Om du valde att ange `-ifref` vid körningen av `cfgmaker(1)` kommer den även att skapa ett färdigt konfigurationsexempel som kommer att läsa in och utgående trafik på den angivna datorns nätverkskort och presentera detta.

Om allt ser ut att vara i sin ordning kan du gå vidare och köra skriptet `indexmaker(1)` för att generera HTML-sidorna. Kom ihåg att placera dem i samma katalog som MRTG placerar graferna i.

Öppna därefter HTML-filen med din webbläsare för att bekräfta att allt fungerar.

Det är nu dags för dig att börja skapa egna övervakningar. Se *Configuration Reference* i MRTGs dokumentation [4] för information om hur du skapar egna övervakningar. Målet här är att skapa en baseline över din egen dators användning, på så sätt kan du få tydliga indikationer på ifall du börjar få något

hårdvarufel, vilket på servrar är ytterst viktigt att få information om innan det händer.

1. Skapa en mrtg-övervakning som hämtar hur mycket primärminne som används.
2. Skapa en övervakning som visar nätverksanvändningen på din dator.
3. Skapa en egen valfri övervakning som du anser vara viktig att få fram en baseline på.

3 Examination

Dina svar på de uppgifter du utfört ska skickas in som PDF i inlämningslådan för L2. När du lämnat in din uppgift och blivit godkänd på denna får du tillgång till forumet för L2 där du ska skriva ett inlägg som innehåller följande:

1. Hur många bytes behövde du läsa in för varje paket för att få samtliga headers och minimalt med payload? Förklara hur du kom fram till detta värde.
2. I 2.2. Hur väl tyckte du att graferna redovisade den data du ville presentera? Ge ett exempel på ett alternativt format som skulle vara lämpligt att presentera datat i? Tydligt visa ditt resonemang.
3. I 2.2. Vad valde du att skapa en graf över i Wireshark? Motivera ditt resonemang. Lägg ut en bild på din graf med tillhörande kommentar om vad den visar.
4. Beskriv vad du valt att skapa ett diagram över i 2.3, motivera varför du valde just denna typ av data för denna typ av diagram och förklara därefter hur ditt program fungerar.
5. I 2.4. Vad valde du att övervaka med hjälp av SNMP? Motivera väl och lägg ut en bild som visar den graf MRTG skapade med tillhörande kommentar om vad den visar.

Efter att du publicerat ditt inlägg ska du nu kommentera på två av dina studentkollegors inlägg. Följande gäller vid val av inlägg att kommentera på.

- De inlägg du kommenterar på måste behandla de uppgifter du inte utfört.
- I den mån det går, inläggen ska inte varit kommenterade på tidigare. Om det inte finns några okommenterade inlägg, avvakta någon dag innan ni skriver en kommentar, eller 'efterlys' ett inlägg.
- 'Lås upp inlägget' genom att först skriva ditt namn innan du börjar kommentera (på så vis undviker vi att flera kommenterar på samma inlägg).

Referenser

[1] Ulf Lamping, Richard Sharpe och Ed Warnicke. *Wireshark User's Guide*. Accessed: 2015-06-30. URL: https://www.wireshark.org/docs/wsug_html_chunked/.

[2] *Matplotlib*. Accessed: 2015-09-28. URL: http://matplotlib.org/.

[3] Mani. Subramanian, Timothy A. Gonsalves och N. Usha Rani. *Network management : principles and practice*. Noida, India: Dorling Kindersley, 2011. ISBN: 978-81-317-3404-9.

[4] "Tobi Oetiker's MRTG - The Multi Router Traffic Grapher". 2013. URL: http://oss.oetiker.ch/mrtg/.