

# Bilaga

## Innehållsförteckning

1	Allmänt .....	3
2	Deltagare övningen viking 11 .....	3
3	Dataunderlag krigsdagbok .....	3
3.1	Diagram avseende antalet händelser vid olika dagar.....	3
3.2	Matematiskt samband mellan olika dagar .....	5
4	Dataunderlag kapacitetsanvändning .....	7
4.1	Insamling och bearbetning av data .....	7
5	Förslag på genomförande av ytterligare mätningar. ....	13
6	Routerkonfiguration NBG 11 .....	14
6.1	Köhantering och märkning av paket.....	14
6.2	Interface .....	15
6.3	Konfiguration insamling mätdata .....	16
7	Python script .....	16
7.1	Anvisning.....	16
7.2	Script.....	17
8	Källförteckning .....	20

## **Förteckning bilder, figurer, diagram och tabeller**

Figurer, tabeller och foton av Anders Gradh om inte annat anges.

Diagram 1 Krigsdagbokens händelser per timme under torsdag till lördag.....	4
Diagram 2 Krigsdagbokens händelser per timme under söndag till tisdag.....	4
Diagram 3 Krigsdagbokens händelser per timme under tisdag till onsdag första veckan och onsdag andra veckan .....	5
Diagram 4 Data från blått nät. Blå färg är all travesterad data och lila färg är 95:e percentilen.....	9
Diagram 5 Överförd data på rött nät från stab till högre chef.....	9
Diagram 6 95 percentilen av trafiken i nätverket till och från staben. Observera att RÖTT, BLÅTT och SVART visas.....	11
Diagram 7 95-percentilen av röd trafik till och från staben.....	12
Diagram 8 95-percentilen av blå trafik till och från staben.....	12
Tabell 1 Sammanställning deltagare vid övningen VIKING 11 från planeringsdokument februari 2011.....	3
Tabell 2 Korrelationskoefficienten mellan olika dagar under övningen.....	5
Tabell 3 antalet händelser fredag och lördag .....	6
Tabell 4 JCSS lägger till utläsningens inställningar i csv första rader, ovanstående är ett exempel från rött nät.....	8
Figur 1 Flödesschema på arbetsstegen vid framtagning av resultat av mätningarna.....	7
Figur 2 Överförd data på blått nät från stab till ISTAR .....	10

## 1 Allmänt

Denna bilaga skall ses som ett komplement till huvuddokumentet. Syftet är att möjliggöra för läsaren att fördjupa sig och även för att stödja upprepbarheten av studien.

## 2 Deltagare övningen viking 11

Nedanstående tabell skall inte ses som det faktiska deltagandet. Detta är ett utdrag från planeringen för övningen. Deltagandet från respektive förband bör dock ha legat runt nedanstående värden. Observera att staben var fullbemannad och underställda förband bestod av stabs- och driftpersonal. Behov klienter och telefoner finns enbart för de förband som hade behov av stöd med att upprätta dessa. Observera att detta enbart är deltagarna för NBG11s del i övningen viking 11, totalt i övningen var det runt 2500 deltagare.

Förband	Deltagare	SVART	BLÅTT	RÖTT	CN	Övr.
(F)HQ	120				HQ2	
HQ Coy	10				HQ2	
CBn	12				HQ1	
EAW	27+5				EAW	
LOG	17				HQ1	
ISTAR	17+3	4	2	15	HQ1	
Eng	3	4	10	2	NSE	Totalt 15 pers.
MP	2				NSE	
LAP	2				NSE	
CIMIC	3				NSE	
GEO SE	3				NSE	
NSE	1				NSE	

Tabell 1 Sammanställning deltagare vid övningen VIKING 11 från planeringsdokument februari 2011.

## 3 Dataunderlag krigsdagbok

### 3.1 Diagram avseende antalet händelser vid olika dagar

De dagar som sticker ut är tisdag och onsdag första veckan vilket kan ses i Diagram 3, där det i princip inte är några händelser. Torsdag till tisdag är rätt så lika (Diagram 1 och Diagram 2) förutom att söndagens händelser går ner till noll. Onsdagen vecka 2 går också antalet händelser ner till noll efter klockan 16 (Diagram 3) då övningen avslutades.

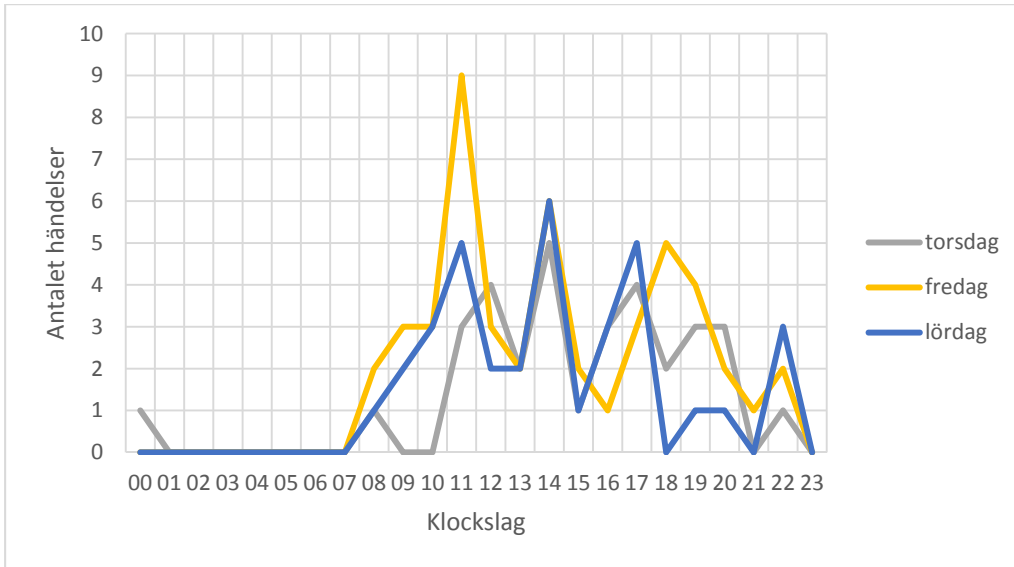


Diagram 1 Krigsdagbokens händelser per timme under torsdag till lördag

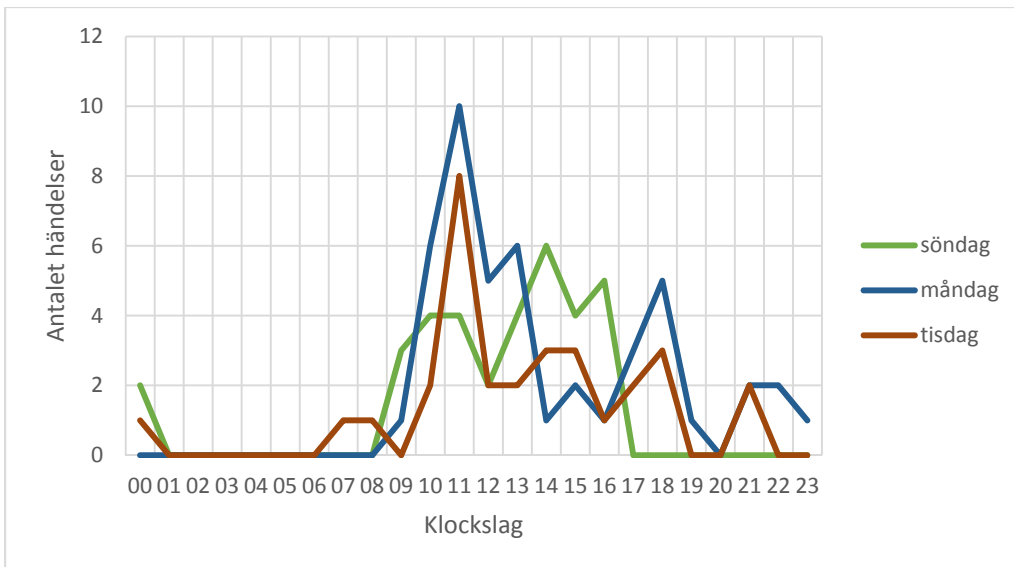


Diagram 2 Krigsdagbokens händelser per timme under söndag till tisdag

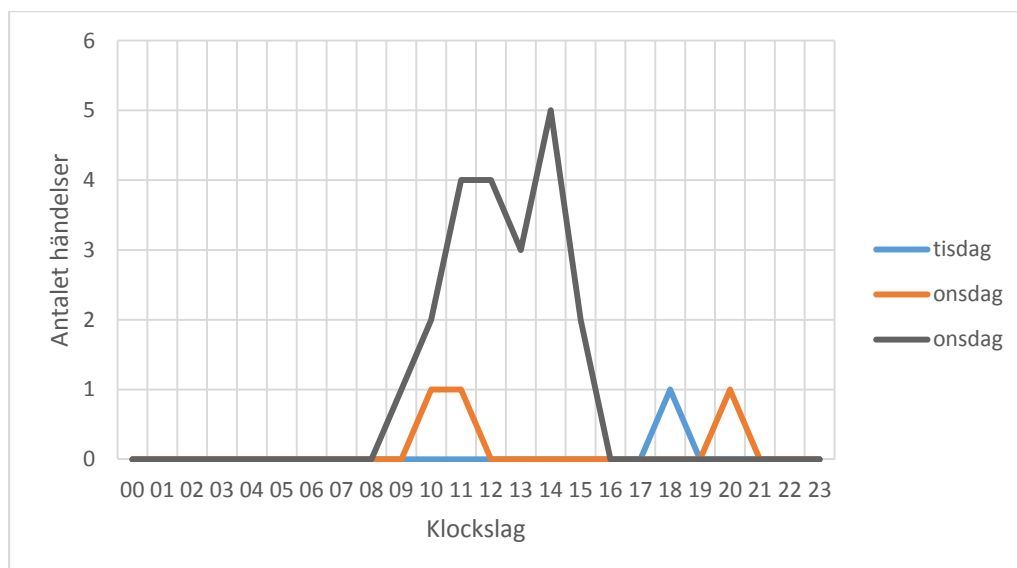


Diagram 3 Krigsdagbokens händelser per timme under tisdag till onsdag första veckan och onsdag andra veckan

### 3.2 Matematiskt samband mellan olika dagar

För att statistiskt påvisa likheter i fördelning av händelser över dygnet mellan olika dagar har korrelationskoefficienten (r) räknats ut mellan dagarna. Korrelationskoefficienten påvisar hur väl värden mellan två variabler samverkar linjärt. Variablerna i det här fallet är händelser i krigsdagboken under två olika dagar. Ett högt värde på korrelationskoefficienten skulle innebära att ett högt samband mellan dagarna, t.ex. ett högt värde på tisdagen motsvaras av ett högt värde på onsdagen.

För att räkna ut korrelationskoefficienten har Excels dataanalysfunktion korrelation använts med följande resultat.

	tis	ons	tor	fre	lör	sön	mån	tis	ons
tis	1,00								
ons	-0,08	1,00							
tor	0,08	0,15	1,00						
fre	0,28	0,45	0,66	1,00					
lör	-0,17	0,32	0,72	0,72	1,00				
sön	-0,15	0,24	0,43	0,49	0,65	1,00			
mån	0,25	0,50	0,37	0,73	0,52	0,45	1,00		
tis	0,20	0,44	0,45	0,80	0,57	0,54	0,82	1,00	
ons	-0,12	0,28	0,55	0,67	0,65	0,76	0,63	0,67	1,00

Tabell 2 Korrelationskoefficienten mellan olika dagar under övningen

Korrelationskoefficienten räknas fram genom att kovariansen mellan två variabler divideras med variablernas standardavvikelse.

Korrelationskoefficient r räknas fram enligt formeln:

$$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right) \left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}}$$

(Körner & Wahlgren, 2005, p. 76)

Exempelvis blir korrelationskoefficienten mellan fredag (x) och lördag (y) (se data i Tabell 3).

$$r = \frac{(140) - \frac{48 \times 35}{24}}{\sqrt{\left(216 - \frac{(48)^2}{24}\right) \left(\sum y^2 - \frac{(35)^2}{24}\right)}} = 0,724$$

Slutsatser avseende korrelationen mellan dagarna bör dras med viss försiktighet då en fördubbling av korrelationsvärdet inte innebär en fördubbling av sambandet (Körner & Wahlgren, 2005, p. 77). Korrelationsvärdena bör dock kunna användas för att se tendenser.

Observera att många tal par mellan de två funktionerna består av värdet 0, se till exempel mellan klockan 00 och 07 på fredag och lördag i tabell 2. Detta innebär att korrelationskoefficienten felaktigt får för höga värden. Trots ovanstående begränsningar i uträkningarna går det dock att återfinna genomslag av den faktiska verksamheten i uträkningarna.

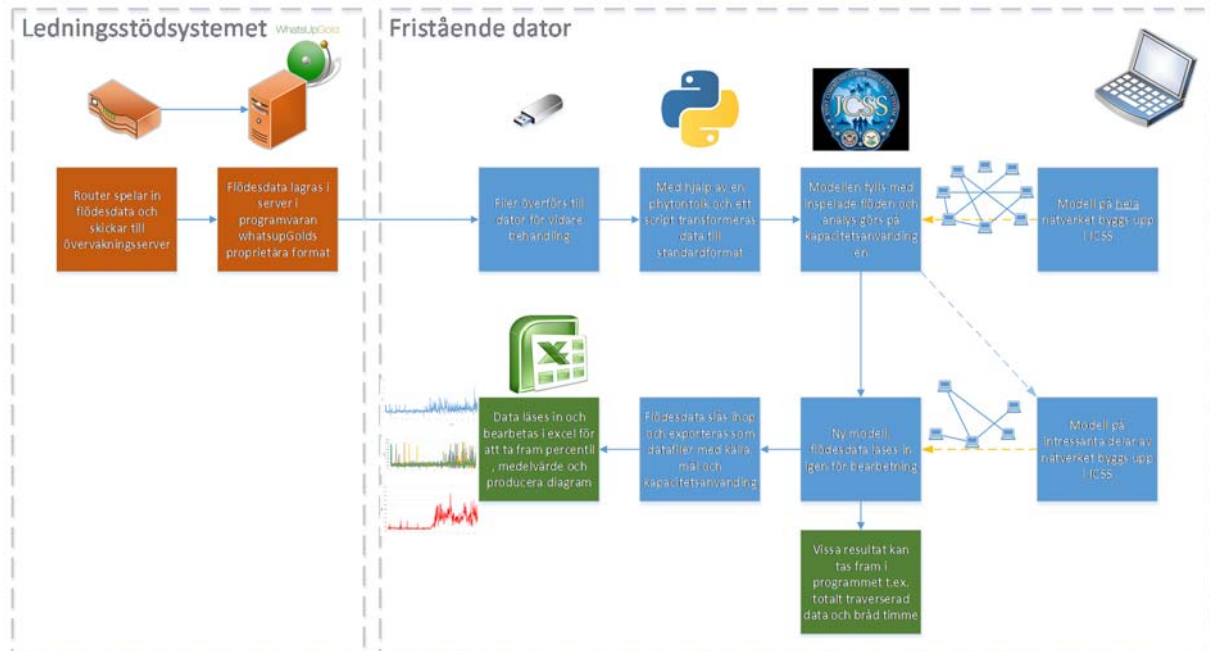
Den deltagande studien visar på att torsdag första veckan till tisdag andra veckan är intressanta dagar att fördjupa sig i korrelationsvärdena mellan dagarna i den perioden stödjer detta. Observera att söndag har generellt låga korrelationsvärden och det torde vara ett utslag av inga inskrivna händelser efter klockan 17 p.g.a. återhämtning. Söndagen har högst korrelation med onsdagen andra veckan, troligtvis på grund av att inget är inskrivet efter klockan 16 på onsdagen.

Klockslag	Lördag (x)	Söndag (y)
00	0	2
01	0	0
02	0	0
03	0	0
04	0	0
05	0	0
06	0	0
07	0	0
08	1	0
09	2	3
10	3	4
11	5	4
12	2	2
13	2	4
14	6	6
15	1	4
16	3	5
17	5	0
18	0	0
19	1	0
20	1	0
21	0	0
22	3	0
23	0	0
Tot	35	34

**Tabell 3** antalet händelser fredag och lördag

## 4 Dataunderlag kapacitetsanvändning

### 4.1 Insamling och bearbetning av data



Figur 1 Flödesschema på arbetsstegen vid framtagning av resultat av mätningarna

Bearbetningen av dataunderlaget har skett i många steg. Insamling i nätverket har skett med hjälp av driftledningens personalen. Routerarna konfigureras till att samla in flödesinformation som skickas till driftledningens övervakningsserver. Övervakningsservrarna spar ner data i databasfiler. Servrarna har sedan tömts manuellt vilket har resulterat i en eller flera filer per övervakningsserver, mycket beroende på hur mycket trafik som gått igenom respektive router. Datafilerna har varit i ett proprietär format avsett för programmet WhatsUpGold från Ipswitch. WhatsUpGold används i systemet för nätverksövervakning. Totalt har data samlats i 12 filer med en total storlek på 1,44GB.

Databasfilerna har sedan överförts till en dator för att transformera dem till ett standardformat. Genom att använda en applikation transformeras databasfilerna till standardformatet netflow med filändelse tr2. Totalt genererar det 65 filer med en total storlek på 1,13 GB.

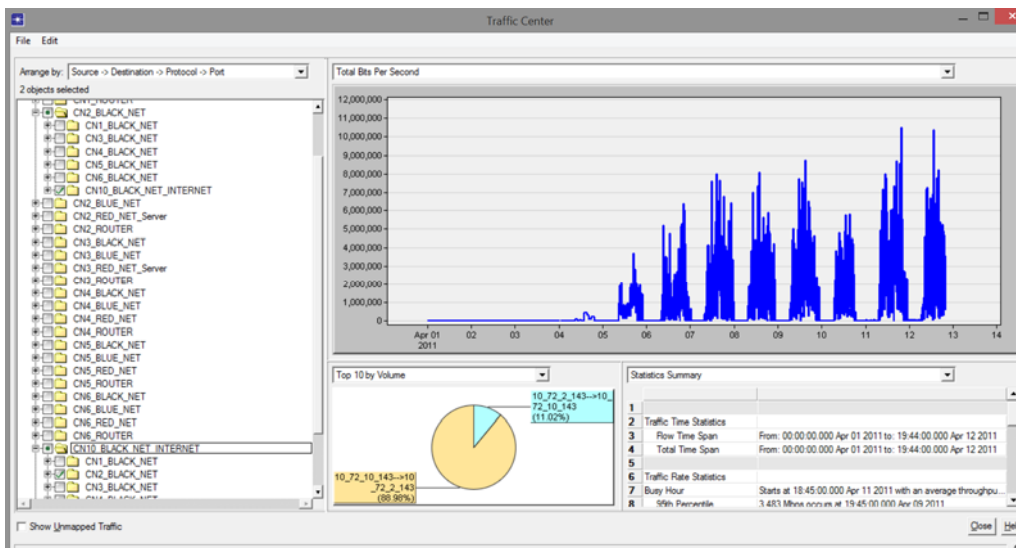
Utifrån hur nätverket var uppkopplat under övningen byggdes en modell upp i JCSS programvaran. Modellen var till viss del förenklad då viss nätverksutrustning förenklades. Till exempel fick en switch representera alla switchar på en nod. Detta kan ha betydelse om du vill titta på fördröjningar och liknande men påverkar inte denna studie. Alla routrar i nätverket och terminalutrustning som klienter och telefoner fanns dock med. Viktigt var att ip-adresseringen var rätt i modellen. Om inte flödets IP adresser för destination och källa finns med läses inte den informationen in i modellen.

Nätverkstrafiken analyserades sedan i modellen genom att utifrån IP adressplanen på systemet leta efter för studien relevanta nätverksutrustningar och klienter. Det var också viktigt att identifiera det som inte var relevant men hade mycket trafik. Exempel på en relevant nätverksutrustning är krypton för de sekretessklassade nätverken vilket är fokalpunkten för nodens trafik på ett nät. Exempel på icke relevant utrustning är servern för namnuppslag (Domain Name System Server). Klienter skickar förfrågningar dit kontinuerligt men kan betecknas som maskin till maskin trafik och inte relevant för denna studie.

Utifrån analysen av den mer fullständiga modellens trafik togs en ny modell fram där enbart relevant utrustning fanns med. Det innebar att inte lika mycket flödesdata lästes in i modellen och det gjorde modellen mer lättarbetat. I programmet valdes därefter relevant flödesinformation ut och vissa resultat kunde utläsas direkt (Se Bild 1). Till exempel bråd timme och fördelning av trafiken. För att kunna manipulera data behövdes den dock exporteras till en fil med kommaseparerade värden (Comma Separated Value CSV) vilket Excel kan läsa. Flödesdata lästes ut med start 2011-04-07 och sex dagar framåt med ett värde på kapacitetsanvändningen varje minut.

Calendar start time: 00:00:00.000 Apr 07 2011		
Duration: 6.000000		
Duration Units: days		
Step Size: 1.000000		
Step Units: minutes		
Number of Steps: 8640		
Traffic Units: Kbits/s		

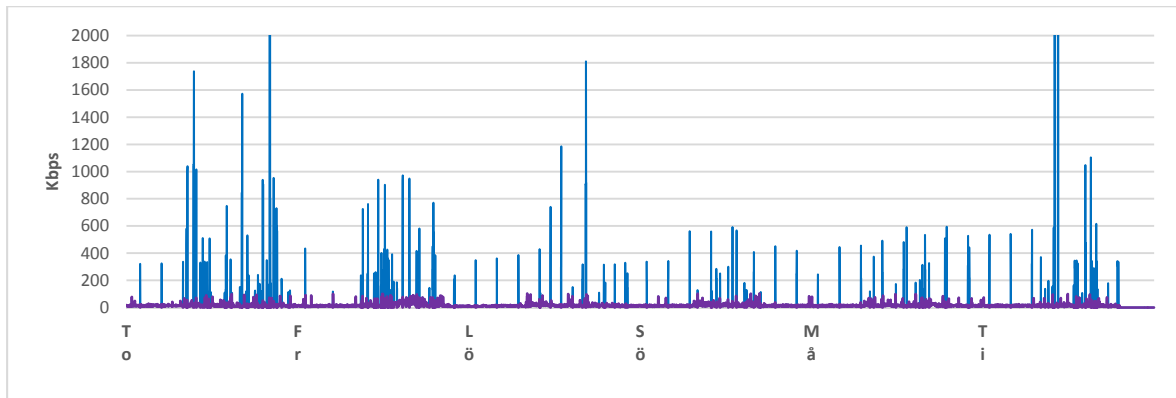
**Tabell 4** JCSS lägger till utläsningens inställningar i csv första rader, ovanstående är ett exempel från rött nät



**Bild 1** Skärmbild på internettrafiken till och från staben. Observera valda flöden till vänster, bandbreddsanvändningen till höger och resultat i nederkant.

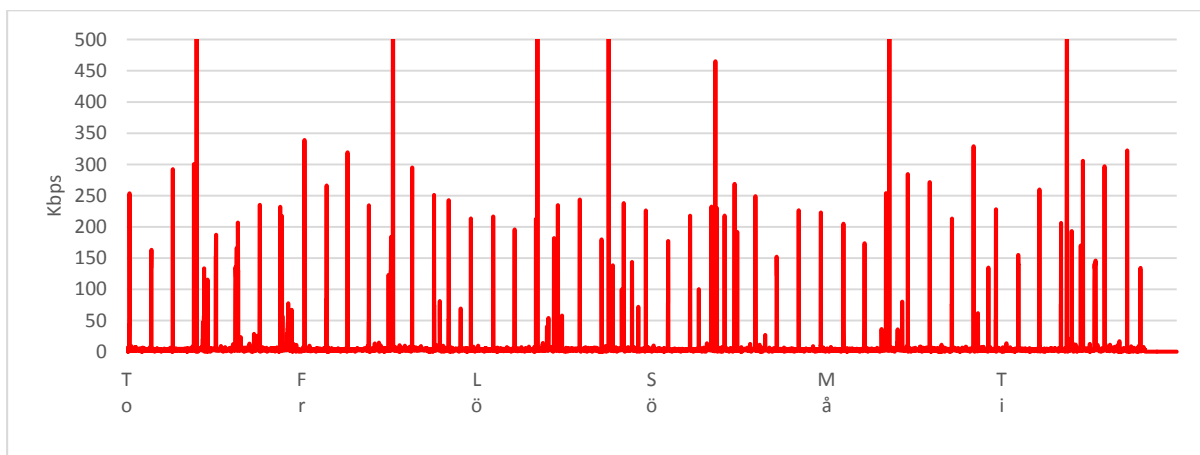
CSV filen lästes in i Excel för att kunna bearbetas. Alla flöden summerades för varje tiominutersvärde, det kan vara många simultana flöden. Utifrån detta

räknades 95:e percentilen fram. De värden under 95:e percentilen lämnades orörda och de över minskades till maxvärdet i 95:e percentilen. Ett alternativ hade varit att sätta dessa till noll men det visade sig att det försvårade analys av diagram senare. Se Diagram 4 för skillnaden mellan all data och enbart den 95:e percentilen.



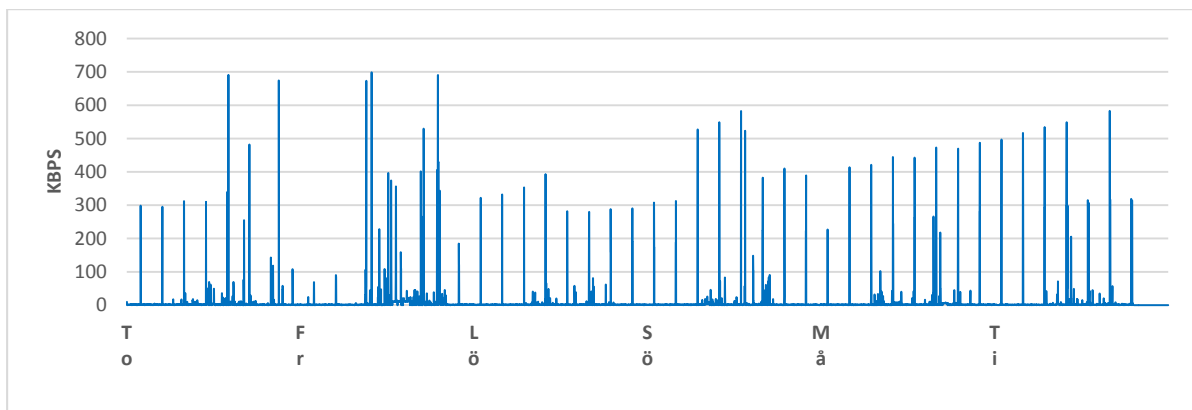
**Diagram 4 Data från blått nät. Blå färg är all travesterad data och lila färg är 95:e percentilen**

Data för torsdag till tisdag lyftes sedan in i en matris med en dag per rad. Därefter kunde medelvärde räknas ut för varje 10 min period på dygnet. Slutligen skapades diagram för både medelvärdet och enskilda dagar.



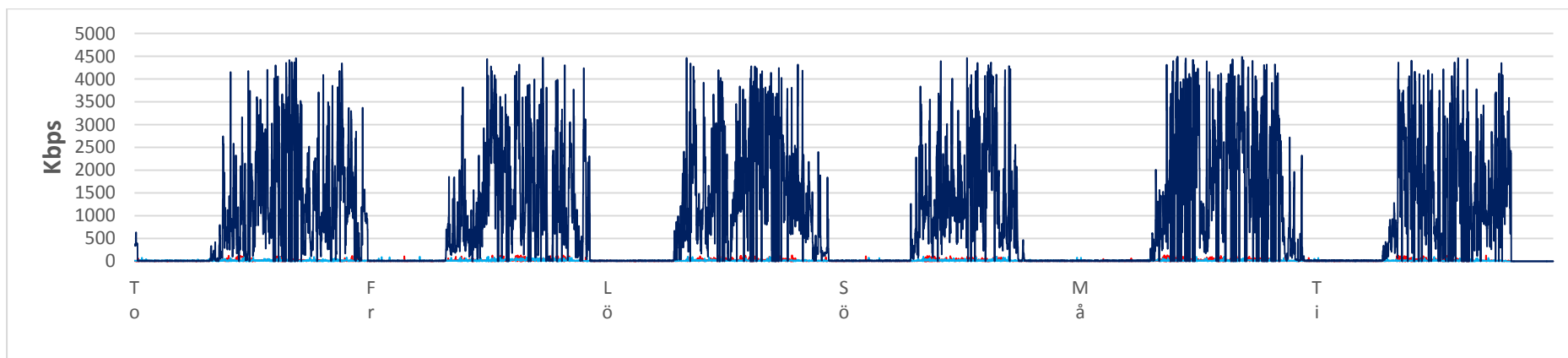
**Diagram 5 Överförd data på rött nät från stab till högre chef**

Det återfinns ett antal periodiska toppar avseende kapacitetsanvändningen, både på blått och rött nät. På blått nät återfinns periodiska toppar mellan kärnnod 1 (ISTAR) och kärnnod 2 (Staben). På rött nät återfinns denna trafik mellan kärnnod 2 (stab) och kärnnod 10 (högre chef) (se Diagram 5 för trafiken åt ena hållet). Periodiciteten är var överföring var tredje timme.



**Figur 2 Överförd data på blått nät från stab till ISTAR**

Systemarkitekten Roger Lindholm vid BASALT AB har tittat på delar av statistiken och enligt honom går det inte att med säkerhet fastställa vilken trafik det är. Dock bedömer han att det troligtvis är SITAWARE trafik. Det är dock oklart varför kapacitetsanvändningen ökar över tiden. Andra möjliga tjänster kan vara domän (Windows Active Directory-AD) eller antivirushantering (McAfee EPO) i nätet, det är dock mindre troligt. Eftersom trafiken är periodisk anser han att mail, chat, portalsurf och DNS är uteslutet, då den trafiken torde vara mer stokastisk (Lindholm, 2014).



**Diagram 6 95 percentilen av trafiken i nätverket till och från staben. Observera att RÖTT, BLÅTT och SVART visas.**

Internettrafiken är så dominerande att trafiken i det två andra nätverken knappt syns i Diagram 6. Kapacitetsanvändningen är tydlig under dagtid där den fluktuerar mellan noll och i korta perioder upp till över 4 Mbps. Det går att se spår av mönster av minskningar mitt på dagen och runt middagstid.

I Diagram 7 går att se likheter mellan dagarna i det röda nätverket och söndagens nedgång återfinns också här som i internettrafiken. I det blåa nätverket, Diagram 8, är det svårare att se likheter i trafik. Det går dock att ana höjning under dagtid men inte lika tydligt som i rött nät.

För att kunna jämföra arbetsbelastningen i staben med dess kapacitetsanvändning har jag medelvärdesbildat 95-percentilen för att få fram medelsanvändning över dygnet. Detta för att kapacitetsdata skall vara jämförbar med data i krigsdagboken.

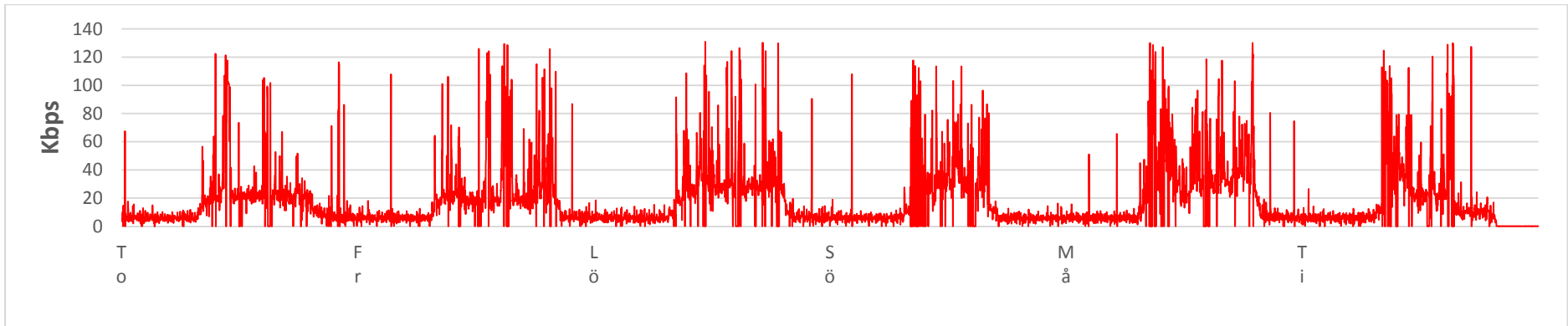


Diagram 7 95-percentilen av röd trafik till och från staben.

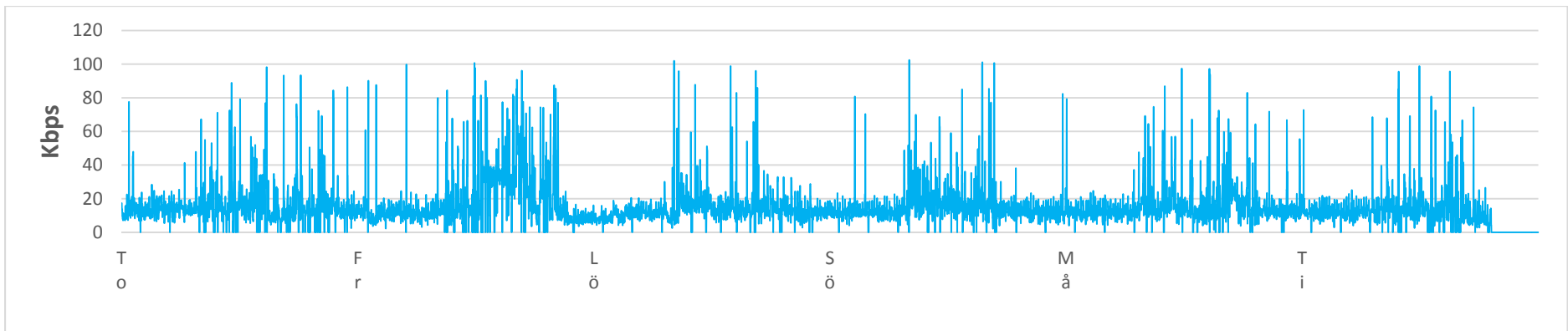


Diagram 8 95-percentilen av blå trafik till och från staben

## 5 Förslag på genomförande av ytterligare mätningar.

Samla in namnen på all driftpersonal för att i ett senare skede kunna kontakta individer om kompletterande information behövs. Se till att analysera nätverksdata i så nära anslutning till insamlingstillfället som möjligt. Detta för att kunna ställa kontrollfrågor till driftpersonal när de fortfarande har genomförandet i färskt minne.

Genomför en enkät med användarna där information samlas in från den enskilde för att kunna jämföra med genomförda mätningar. Information kan vara hur mycket användaren uppfattar att de har använt nätverket. Hur fördelar sig internet användning mellan tjänste- och icke tjänstetraffic? Hur upplever användaren belastningen i staben under de dagar övningen pågått?

Försök komplettera med en deltagande studie, om inte uppsatsskrivaren deltar själv kanske det finns en deltagare med kompetens för att kunna dokumentera viktiga skeenden.

Förslag på underlag att samla in vid liknande arbeten.

Vad	Syfte
Styrdokument och eventuell dygnsrytm	Ger spårbarhet på hur verksamheten har bedrivits.
Deltagarlistor gärna med kontaktuppgifter	Det dyker alltid upp frågor efter hand
Telefonkataloger	För att kunna få fram vem som ringer vem.
Loggar från olika servrar; Telefoniserver, proxyserver för internet	Telefoniservrens logg behövs för att kunna göra en relevant bedömning på vem som har ringt vem.
Tekniska systembeskrivningar	Viktigt för att kunna verifiera hur nätet var designat.
Konfigurationer på nätverksutrustning switchar och routrar	Ger möjlighet att utreda hur nätverket har uppträtt samt detaljer avseende nätverksutrustningens bearbetning av nätverkstrafiken.

## 6 Routerkonfiguration NBG 11

Observera att detta ej är hela routerkonfigurationen utan ett utdrag kopplat mot QoS och insamling mätdata.

### 6.1 Köhantering och märkning av paket

```
class-map match-all QUEUE_CRYPTO
  match dscp af13
class-map match-any SET_MANAGEMENT
  match protocol telnet
  match protocol ssh
  match protocol snmp
  match protocol ntp
  match protocol http
  match protocol icmp
class-map match-all QUEUE_MANAGEMENT
  match dscp af11
class-map match-all SET_CRYPTO
  match access-group name ACL_CRYPTO
class-map match-all QUEUE_SIP
  match dscp af41
class-map match-any SET_ROUTING_DNS
  match protocol ospf
  match protocol rip
  match protocol dns
class-map match-all SET_VOICE
  match protocol rtp
class-map match-all QUEUE_ROUTING_DNS
  match dscp cs6
class-map match-any SET_SIP
  match access-group name ACL_VTC
  match protocol sip
  match access-group name ACL_TVS
  match protocol ldap
class-map match-all QUEUE_VOICE
  match dscp ef
!
!
policy-map SET_DSCP
  class SET_ROUTING_DNS
    set dscp cs6
  class SET_VOICE
    set dscp ef
  class SET_SIP
    set dscp af41
  class SET_MANAGEMENT
```

```
set dscp af11
class SET_CRYPTO
set dscp af13
policy-map QUEUE1
class QUEUE_VOICE
priority percent 25
class QUEUE_SIP
bandwidth percent 5
class QUEUE_MANAGEMENT
bandwidth percent 5
class QUEUE_CRYPTO
bandwidth percent 40
```

## 6.2 Interface

```
!
interface Serial0/0/0
description Transmission (max 8 Mbit/s)
ip unnumbered Loopback0
ip pim sparse-mode
clock rate 2016000
clock rate 2016000
dce-terminal-timing-enable
service-policy output QUEUE1
!
interface Serial0/0/1
description Transmission (max 8 Mbit/s)
ip unnumbered Loopback0
ip pim sparse-mode
clock rate 2016000
clock rate 2016000
dce-terminal-timing-enable
service-policy output QUEUE1
!
interface Serial0/0/2
description Transmission (max 8 Mbit/s)
ip unnumbered Loopback0
ip pim sparse-mode
clock rate 2016000
clock rate 2016000
dce-terminal-timing-enable
service-policy output QUEUE1
!
interface Serial0/0/3
description Transmission (max 8 Mbit/s)
ip unnumbered Loopback0
ip pim sparse-mode
clock rate 2016000
clock rate 2016000
dce-terminal-timing-enable
service-policy output QUEUE1
```

### 6.3 Konfiguration insamling mätdata

Detta skall ses som ett generellt exempel och inte som den faktiska konfigurationen på alla routrar under insamling.

På alla interface som skall kontrolleras anges:  
ip flow ingress

Ovanstående innebär att flödesdata samlas in för all inkommande trafik till interfacet.

Konfigurationsinställningar för inhämtningen:

```
ip flow-cache timeout active 2
ip flow-export source Loopback0
ip flow-export version 9
ip flow-export interface-names
ip flow-export destination [IP-Adress till logserver] 9999
```

## 7 Python script

Sparad data från nätverket måste göras om från det format som programvaran WhatsAppGold använder till ett format som JCSS kan läsa in. Detta görs med hjälp av en applikation skrivet i programmeringsspråket python.



### 7.1 Anvisning

DESCRIPTION:

pytr2 is a simple Python application for importing a WUG14 NetFlow Database into

MSSQL and exporting it into the JCSS compatible format tr2.

INSTALLATION/USAGE:

1. Install python-2.7.1.msi
  2. Install pyodbc-2.1.8.win32-py2.7.exe
  3. Extract pytr2-x.x.zip to C:\  
Will create a folder named pytr2.
  4. Copy NetFlow Backups to C:\pytr2\dbs  
No Folders, pytr2 will look for files ending with .dat.
  5. Create a new MSSQL user, default is uid: pytr2 with pw: passwd  
Default user and pw can be changed within the pytr2.cfg  
Make sure the new user has atleast permissions to erase the
- NetFlow db.
6. Check that C:\pytr2\tr2 is empty.
  7. Check that C:\pytr2\pytr2.log doesn't exist. Or your logs will be merged.
  8. Doubleclick on pytr2.py in C:\pytr2
  9. Check logs for errors.  
Make sure that Exit status for MSSQL always is 0.

12 DB backups with a total size of 1.5 GB takes about 30 minutes to complete.

AUTHOR:

Johan Aldor (C2Solutions AB) for FMV.  
johan.aldor@c2solutions.se  
+46706468370

## 7.2 Script

```
#!/usr/bin/env python
```

```
# -*- coding: utf-8 -*-
```

```
"""
```

Description:

pytr2 is a simple Python application for importing a WUG14 NetFlow Database into MSSQL and exporting it into the JCSS compatible format tr2.

vim:

```
tabstop=4 expandtab shiftwidth=4 softtabstop=4
```

Author:

Name: Johan Aldor for FMV  
Email: johan.aldor@c2solutions.se  
Phone: +46706468370  
Date: 24-05-2011

```
"""
```

```
import sys, os, os.path, re, pyodbc, time, ConfigParser
```

```
ImportCommand = """sqlcmd -S "localhost\whatsup" -U %s -P %s -Q " USE  
Master RESTORE DATABASE NetFlow FROM DISK='%s\\%s' WITH  
REPLACE, MOVE 'NetFlow' TO 'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Data\NetFlow.mdf', MOVE 'NetFlow_log' TO  
'C:\Program Files\Microsoft SQL  
Server\MSSQL.1\MSSQL\Data\NetFlow_log.ldf' """
```

```
FetchQuery = """SELECT dbo.ipIntToString(nSrcIPAddress) AS source,  
dbo.ipIntToString(nDstIPAddress) AS destination,  
dbo.ReverseProtocolNameServer(nProtocolId) AS protocol,  
nPortID AS source_port, nPortID AS destination_port,  
nTypeOfService AS tos,  
convert(varchar,datetime,101) + ' ' + convert(varchar,datetime,108) AS start,  
convert(varchar,DATEADD(mi,nTimeDelta,dTime),101)  
+ ' ' + convert(varchar,DATEADD(mi,nTimeDelta,dTime),108) AS 'end',  
replace(nPackets/(nTimeDelta*60),',','.') AS packets_sec,
```

```
replace(nBytes*8/(nTimeDelta*60),',,') AS bits_sec FROM KeyData''''
```

```
CreateFunctionRPNS = ''''create FUNCTION ReversePortNameServer
(
  -- Add the parameters for the function here
  @port int
)
RETURNS varchar(15)
AS
BEGIN

  Declare @ProtocolName char(15)

  -- Add the T-SQL statements to compute the return value here
  select @ProtocolName=sApplicationName from Port where
nPortID=@port

  return @ProtocolName
END
''''
```

```
CreateFunctionRProtoNS = ''''create FUNCTION
ReverseProtocolNameServer
(
  -- Add the parameters for the function here
  @port int
)
RETURNS varchar(15)
AS
BEGIN

  Declare @ProtocolName char(15)

  -- Add the T-SQL statements to compute the return value here
  select @ProtocolName=sProtocolName from Protocol where
nProtocolId=@port

  return @ProtocolName
END
''''
```

```
conffile = 'pytr2.cfg'
```

```
def GiveMeSomeDBs(logfile, dbdir):
  logstr = 'Imported the following DBs from %s' % (dbdir)
  files = os.listdir(dbdir)
  dats = []
  for file in files:
    if re.search('.dat', file.lower()):
      dats.append(file)
```

```
        logstr += ', %s' % (file)
    LogMe(logfile, logstr)
    return dats

def LogMe(logfile, string):
    logger = open(logfile, 'a')
    logger.write("%s - %s\n" % (str(time.strftime("%Y-%m-%d %H:%M:%S")),
str(string)))
    logger.close()

def ImportDBsToMSSQL(logfile, uid, pwd, dbdir, TheDataBase):
    LogMe(logfile, 'MSSQL NetFlow import Exit status: %s (0 = good,
otherwise check console for output.)' % (str(os.system(ImportCommand %
(uid, pwd, dbdir, TheDataBase))))))
    cnxn = pyodbc.connect("""DRIVER={SQL Native Client};
SERVER=localhost\\whatsup;
DATABASE=NetFlow;
UID=%s;
PWD=%s;"" % (uid, pwd))
    cursor = cnxn.cursor()

    cursor.execute(CreateFunctionRPNS)
    cnxn.commit()

    cursor.execute(CreateFunctionRProtoNS)
    cnxn.commit()

    del cursor
    cnxn.close()
    del cnxn

def main():
    config = ConfigParser.ConfigParser()
    config.read(conffile)
    LogMe(str(config.get('path', 'log')), 'Start of pytr2.')
    dbs = GiveMeSomeDBs(str(config.get('path', 'log')), str(config.get('path',
'dbdir')))
    for db in dbs:
        LogMe(str(config.get('path', 'log')), 'Begin work with: %s' % (db))
        ImportDBsToMSSQL(str(config.get('path', 'log')), str(config.get('mssql',
'uid')), str(config.get('mssql', 'pwd')), str(config.get('path', 'dbdir')), db)
        cnxn = pyodbc.connect("""DRIVER={SQL Native Client};
SERVER=localhost\\whatsup;
DATABASE=NetFlow;
UID=%s;
PWD=%s;"" % (str(config.get('mssql', 'uid')), str(config.get('mssql',
'pwd'))))
        cursor = cnxn.cursor()
        cursor.execute(FetchQuery)
```

```
i = 0
j = 0
for row in cursor:
    if i == 0 or i == int(config.get('misc', 'linesperfile')):
        if os.path.exists('%s\\%s.%s.tr2' % (str(config.get('path', 'tr2dir')), db,
str(j))):
            f.close()
            LogMe(str(config.get('path', 'log')), 'Writing to %s\\%s.%s.tr2' %
(str(config.get('path', 'tr2dir')), db, str(j)))
            f = open('%s\\%s.%s.tr2' % (str(config.get('path', 'tr2dir')), db, str(j)),
'w')
            f.write('traffic:\n')

f.write('source,destination,protocol,source_port,destination_port,tos,start,end,p
ackets_sec,bits_sec\n')
    i = 1
    j += 1
    f.write('%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\n' % (row.source,
row.destination, row.protocol, row.source_port, row.destination_port, row.tos,
row.start, row.end, row.packets_sec, row.bits_sec))
    i += 1

del cursor
cnxn.close()
del cnxn
LogMe(str(config.get('path', 'log')), 'Ended work with: %s, I wrote %s
files and now I'm sleeping for 60 seconds.' % (db, str(j)))
time.sleep(60)
LogMe(str(config.get('path', 'log')), 'And we are done..')

if __name__ == "__main__":
    main()
```

## 8 Källförteckning

För källförteckning se huvuddokumentet.