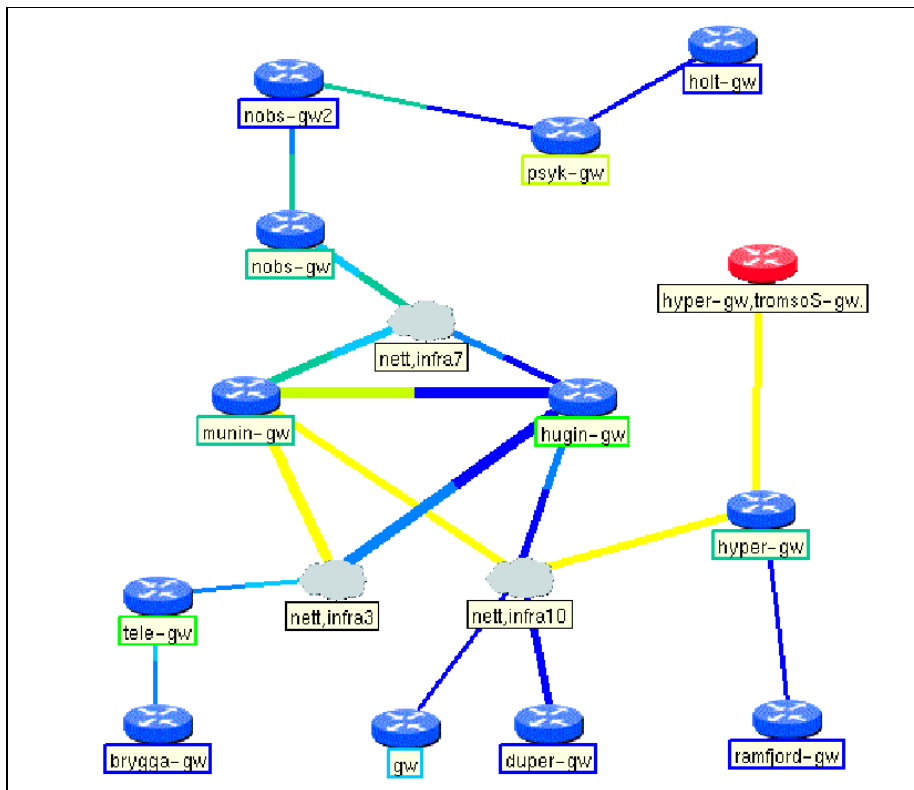


Sluttrapport

NAVMe



Pilotinstallasjon av NAV ved UiTø

UNINETT Prosjekt 353101.13

Prosjektleder

Vidar Faltinsen

Prosjektgruppe

John Magne Bredal – Kristian Eide – Sigurd Gartmann

Gro-Anita Vindheim – Knut-Helge Vindheim

ITEA, NTNU

30. november 2001

Sammendrag

UNINETT prosjekt 353101.13, NAVMe, ble forslått av UNINETT etter en demonstrasjon av NTNU's egenutviklede nettadministrasjonssystem, NAV¹, i januar 2001. Prosjektet har pågått fra 11. juni til 30. november 2001 og er utført av 3 ansatte og 3 studenter ved NTNUs IT-seksjon (ITEA). Totalt timeforbruk har vært 2055 timer.

NAVMe har med utgangspunkt i NTNU NAV laget en mer generell løsning, NAV v2, som lar seg distribuere til andre universiteter og høyskoler. En pilot er installert ved Universitetet i Tromsø (UiTø).

Den viktigste funksjonaliteten fra NTNU NAV er videreført:

- Kunnskapsdatabase
- Rapportgenerator
- Statistikkinnsamler
- Nettkart med visualisering av topologi og last
- Statusmonitor
- Terskelmonitor
- Hendelsessystem (herunder varsling på epost og sms)
- Maskinsporing (Mac- og IP-adresseoversikt)

NAVMe har ikke rokket ved NAV sine grunnprinsipper. Vi baserer oss fortsatt på manuelt vedlikeholdte kildefiler som inneholder et minimum av informasjon om utstyret som skal overvåkes. Natlige script henter ytterligere informasjon ved hjelp av SNMP og fyller NAV sin kunnskapsdatabase. Alle delsystemer i NAV jobber mot kunnskapsdatabasen.

Topologien i nettkartet avledes automatisk basert på ARP-data fra rutere, samt brotabeller og spanning tree tilstand fra svitsjene. I tillegg benyttes CDP-informasjon fra Cisco-utstyr.

NAVMe har styrket kunnskapsdatabasen radikalt. Mange nye tabeller og relasjoner er innført. Vi har også tatt steget fra MySQL til PostgreSQL. Kunnskapsdatabasen fremstår nå som en mer fullverdig modell av nettverket:

- Inventarinformasjon om alle svitsjer og rutere m.m.
- Oversikt over IP-adresserom med knytninger til organisatorisk tilhørighet og anvendelse.
- Informasjon om alle ruterporter. Forståelse av subinterfacer og HSRP.
- Informasjon om alle svitsjeporter med vlaninformasjon (herunder *alle aktive vlan* på trunker).
- Historiske data som viser oppetiden til alle IP- og macadresser på hele nettverket.

¹Network Administration Visualized.

Alle delsystemene er tilpasset den nye databasen. Vi har også styrket funksjonaliteten i mange tilfeller, særlig gjelder dette rapportgenerator og hendelsessystem.

NAVMe har innført CVS. Det gir bl.a. ryddigere versjonshåndtering og enklere distribusjon av systemet. Vi har jobbet med filstruktur og lagt til rette for lokale tillegg. En NAV-installasjon kan lett føye til lokal informasjon i web-grensesnittet. NAV kan således bli det naturlige samlingsstedet for en total datamessig driftstatus.

Nettverket ved UiTø har gitt NAVMe mange nye utfordringer. Vi har støtt på nye utstyrstyper, nye konfigurasjoner og nye designløsninger. De viktigste problemene er løst, resultatet kan sees på <http://po.uit.no> (tildels passordbeskyttet). Pr dato ligger 93 enheter inne i kunnskapsdatabasen, flere vil komme til. Vi har kartlagt 275 ruterporter og 3300 svitsjeporter. Topologiavlederen har funnet struktur og satt det hele sammen i kunnskapsdatabasen.

NAV v2 er ikke noe endelig produkt. Vi peker i rapporten på mange forslag til forbedringer. Det er viktig å gjøre systemet enda mer robust og brukervennlig. Ny funksjonalitet bør komme til. Spesielt nevnes maskinsporing, rundreise- og pakketapsmålinger, integrert tjenesteovervåkning, samt sysloganalysator. Vi foreslår et nytt UNINETT prosjekt som løper i hele 2002. Endelig målsetning for det nye prosjektet bør være å installere NAV på alle universiteter og høyskoler som ønsker det.

Innhold

1	Forord	1
2	Innledning	2
2.1	Hva initierte dette prosjektet?	2
2.2	Hva er prosjektets hovedoppgave?	2
2.3	Hvilke effekter vil prosjektet ha for UNINETT og UNINETTs brukerorganisasjoner?	2
2.4	Rapportering	3
2.5	Arbeidet med prosjektet	3
2.6	Dokumentasjon	3
3	NAV før NAVMe	4
3.1	Historikk	4
3.2	NAV grunnprinsipper	4
3.3	NAV forutsetninger	5
3.4	NAV v1 (NTNU NAV)	5
4	Arbeidet i NAVMe	7
4.1	Nettverksdesign og Konvensjoner	7
4.1.1	Design av lokalnett	7
4.1.2	Design på ruternivå	8
4.1.3	Navnsetting av utstyr	9
4.1.4	Valg av utstyr	10
4.1.5	Stedsinformasjon	10
4.2	Utvidelser i kunnskapsdatabasen	11
4.2.1	Generelle forbedringer	11
4.2.2	Inventar	12
4.2.3	IP adresserom og ruterporter	13
4.2.4	Vlan og svitsjeporter	14
4.3	Topologiavlederen	14
4.3.1	Fysisk sammenkobling	15
4.3.2	Vlan på trunker	15
4.3.3	Spanning tree tilstand	16
4.4	Konsistenssjekker / feilrapporter	16
4.5	Andre delsystemer	16
4.5.1	Rapportgeneratoren	17
4.5.2	Trafikkinnsamling (RRD og Cricket)	18
4.5.3	Nettkartet (vlanPlot)	18
4.5.4	Statusmonitor	19
4.5.5	Terskelmonitor	19
4.5.6	Hendelsessystem	19
4.5.7	Statussiden	20
4.5.8	Mac- og IP-adresse oversikt	21
4.6	NAV omgivelsene	21
4.6.1	Installasjon av hardware / software	21
4.6.2	Filstruktur og webstruktur	21
4.6.3	Dokumentasjon	22

5	UiTø piloten	23
5.1	Samarbeidet	23
5.2	Fremdriften	23
5.3	Spesielle problemstillinger	23
5.4	Resultatet	24
6	Oppsummering	25
7	Videre arbeid	26
7.1	Distribusjon	26
7.2	Kunnskapsdatabasen	26
7.3	Rapportgeneratoren	27
7.4	Trafikkinnsamling (RRD/Cricket)	27
7.5	Topologi og last-visualisering (vlanPlot)	27
7.6	Statusmonitoren (Live)	27
7.7	Terskelmonitoren	27
7.8	Hendelsessystemet og statussiden	28
7.9	Brukerprofil	28
7.10	Maskinsporing	28
7.11	Utvidelser, ny funksjonalitet	28
8	Konklusjon	29
A	Ordliste	30
A.1	NAV ord	30
A.2	Andre forkortelser	30
B	Spørsmål til UiTø	32
 Figurer		
1	Fra kildefil til database	5
2	Typisk nettstruktur ved NTNU	8
3	Kunnskapsdatabasen	12
4	NAV delsystemer	17

1 Forord

Prosjekt NAVMe er utført av en prosjektgruppe ved ITEA/NTNU på oppdrag fra UNINETT ved Olav Kvitem. Prosjektet har et timeforbruk på 2055 timer og er utført i perioden juni-november 2001. Prosjektdeltagere har vært:

- Vidar Faltinsen (prosjektleder, nettverksansvarlig NTNU)
- John Magne Bredal (student, 5.klasse telematikk/NTNU)
- Kristian Eide (student, 3.klasse dataing./HiST)
- Sigurd Gartmann (student, 3.klasse datateknikk/NTNU)
- Gro-Anita Vindheim (overingeniør, ITEA/NTNU)
- Knut-Helge Vindheim (overingeniør, ITEA/NTNU)

Prosjektet har gjort en pilotinstallasjon av NAV ved Universitetet i Tromsø. Primær kontaktperson/samarbeidspartner har vært:

- Børge Brunes (senioringeniør, IT-avdelingen/UiTø)

Prosjektleder benytter her anledningen til å takke hele prosjektgruppen for en kjempemessig innsats! Dere slutter aldri å imponere, jeg er såre fornøyd :) Vi får håpe oppdragsgiveren har (minst et snev av) samme oppfatning.

Jeg vil også takke Børge for et positivt samarbeid som helt sikkert vil fortsette utover dette prosjektet! Det har vært lærerikt på flere måter.

Vidar Faltinsen
Prosjektleder

2 Innledning

2.1 Hva initierte dette prosjektet?

ITEA har siden våren 1999 gradvis utviklet sitt eget nettadministrasjonssystem, NAV, Network Administration Visualized. Systemet slik det var ved prosjektstart (NAV v1) er nærmere beskrevet i kapittel 3.

UNINETT fattet interesse for systemet etter en gjennomgang januar 2001. De lurte på hvorvidt systemet var NTNU-spesifikt, eller om det mer generelt lot seg installere andre steder. UNINETT hadde spesielt andre høyskoler og universiteter i tankene. Prosjektet NAVMe² fikk da sitt mandat:

2.2 Hva er prosjektets hovedoppgave?

Med utgangspunkt i NTNUs egenutviklede nettadministrasjonssystem, NAV v1, skal prosjektet lage en generell løsning, NAV v2, som lar seg distribuere til andre universiteter og høyskoler.

Prosjektet vil initielt kartlegge hvordan et representativt utvalg av høyskoler/ universiteter har realisert sine lokalnett. Herunder hva slags utstyr som brukes, hvordan nettdesign og fysisk infrastruktur er realisert. Således kan man ved generalisering og kodetilpasning ta høyde for et bredt spekter av særegenheter/problemstillinger/ utfordringer.

Prosjektet skal videre gjøre en installasjon ved en utvalgt pilotinstitusjon, nemlig Universitetet i Tromsø. I prosessen vil vi revurdere NTNU sine NAV-konvensjoner. Vi vil også åpne for en mer generell og omfattende datastruktur og implementere et nytt NAV v2 rundt dette, primært med følgende basiskomponenter:

- Kunnskapsdatabase
- Topologiavleder/konsistenssjekker
- Rapportgenerator
- Statistikkinnsamler
- Grafisk visualisering av topologi og last (nettkart)
- Statusmonitor
- Terskelmonitor
- Hendelsessystem (herunder varsling på epost og sms)
- Maskinsporing (Mac- og IP-adresseoversikt)

2.3 Hvilke effekter vil prosjektet ha for UNINETT og UNINETTs brukerorganisasjoner?

NAVMe generaliserer NAV slik at det lar seg installere ved interesserte høyskoler/ universiteter. Systemet kan erstatte/supplere eventuelt eksisterende nettadministrasjonsverktøy og tilby viktige basistjenester for nettverksansvarlig. Systemet er utvidbart og vil ved videreutvikling kunne dekke ytterligere nettadministrasjonsoppgaver. Det vil også kunne integrere seg mot systemovervåkningsverktøy.

²NAVMe uttales på engelsk 'NAV me'. Løselig oversatt blir dette: 'installer NAV hos meg!'

2.4 Rapportering

NAVMe har rapportert tre ganger gjennom statusrapport. Prosjektet har ført timer og dette er dokumentert som vedlegg til statusrapportene. Statusrapportene er tilgjengelig under prosjektsiden <http://metanav.ntnu.no/NAVMe>. Totalt timeforbruk i prosjektet har vært 2055 timer.

2.5 Arbeidet med prosjektet

Prosjektgruppen har hatt jevnlige møter der vi har diskutert designvalg og tatt opp status. Det har vært fortløpende diskusjoner/oppfølging på mailingliste og individuelle eposter og innbyrdes møter. Det har vært møter med UiTø, primært Børge Brunes, både i Tromsø og i Trondheim, samt over telefon og på epost. Vi har også hatt flere oppfølgingsmøter med UNINETT, primært Olav Kvitem. Vi har demonstrert resultatet på UNINETT sitt høyskoleseminar, 31. oktober. Vi merket her oppløftende stor interesse for systemet.

2.6 Dokumentasjon

En deloppgave i prosjektet har vært å styrke dokumentasjonen (som var tildels fraværende i NTNU NAV). Vi har laget både brukerdokumentasjon, dokumentasjon for NAV administrator og mer detaljert systemdokumentasjon. NAV v2 tilbyr også hjelp-sider direkte i brukergrensesnittet, slik at brukeren kan få hjelp der og da. I tillegg er koden kommentert.

All dokumentasjon er tilgjengelig på web. Her er en oversikt over sentrale lenker:

NAV lenker	http://metanav.ntnu.no/NAVlinks
NAVMe prosjektside	http://metanav.ntnu.no/NAVMe
NAV v2 brukerdokumentasjon	http://metanav.ntnu.no/v2/brukerdok
NAV v2 admindokumentasjon	http://metanav.ntnu.no/v2/admindok
NAV v2 systemdokumentasjon	http://metanav.ntnu.no/v2/systemdok

3 NAV før NAVMe

Vi beskriver her hvordan NAV oppsto og hvordan det gradvis har utviklet seg frem mot dette prosjektet.

3.1 Historikk

ITEA begynte å jobbe med nettadministrasjon i 1997 da vi gradvis overtok driftsansvaret for NTNUs nett fra RUNIT. Vi startet med en installasjon av HP Openview, noe som har vist seg å være en svært lærerik erfaring. Vel så lærerikt var et påfølgende oppdrag fra UNINETT, der Vidar Faltinsen våren 1998 i et forprosjekt evaluerte ulike kommersielle og ikke-kommersielle NMS systemer. Konklusjonen i rapporten var todelte: HP Openview ble anbefalt som beste NMS, men det ble oppfordret til en parallel aktivitet rundt egenutvikling av verktøy. Det ble presisert at nettadministrasjon er et svært omfattende felt, der et produkt vanskelig kan favne alle aspekter.³

Forprosjektet utløste på mange måter ITEA sin egenutvikling. Vi startet med en automatisert MRTG-installasjon for alle NTNUs ruterporter våren 1999 (Vidar Faltinsen). Påfølgende sommer ble denne erstattet av en mer omfattende Cricket-løsning (Trygve Lunheim). Vi startet da også de første byggeklossene med kildefiler og rapporter (Tor Arne Kvaløy). Vi laget videre en ARP-innsamler (Stig Venås) og vår første grafiske lastvisualisator (Tor Arne Kvaløy).

Parallelt laget Eric Sandnes en NT-basert statusmonitor, Falkeblikk. Denne ble våren 2001 erstattet av Live (Gro-Anita Vindheim). Svært mange ideer og erfaringer ble her videreført. Vi laget da også vår integrerte SMS-løsning (Knut-Helge Vindheim).

Arbeidet med maskinsporing tok til høsten 1999 (Gro-Anita Vindheim). Våren 2000 etablerte vi de ulike sikkerhetsnivåene (Knut-Helge Vindheim) og integrerte oss etterhvert mot NTNUs brukerdatabase. Vi startet også så smått med en overgang fra flate filer til MySQL database (Gro-Anita Vindheim).

Sommeren 2000 oppsto trapdetect, vårt hendelses- og varslingssystem (John Magne Bredal). Vi laget også vlanplot, første utgave av dagens last-/ og topologi frontend (Kristian Eide), herunder med en generell RRD lasttuthenter (John Magne Bredal).

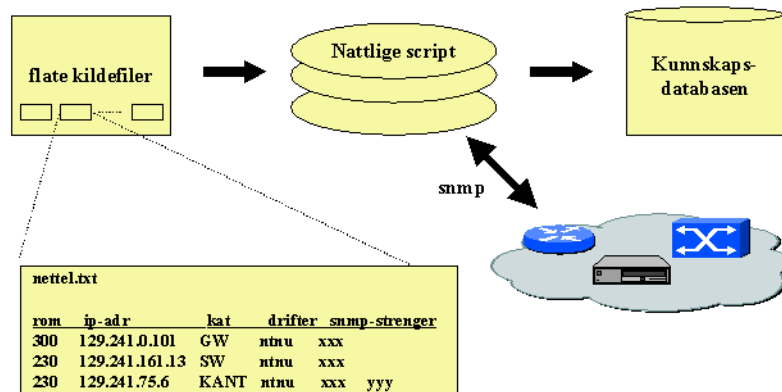
NAV ble demonstrert for mange interesserte januar 2001 og dette trigget UNINETT sin interesse. NAVMe prosjektet startet juni 2001 og slutter med denne rapporten.

3.2 NAV grunnprinsipper

NAV har noen urokkelige grunnprinsipper:

- Alt utstyr som skal overvåkes registreres manuelt i kildefil. Et absolutt *minimum* av informasjon registreres pr enhet.
- Natlige script avleder ytterligere informasjon ved hjelp av SNMP og fyller/oppdaterer kunnskapsdatabasen (se figur 1).
- Kunnskapsdatabasen danner en modell av det virkelige nettverket med informasjon om utstyr, sammenkobling, topologi, samt informasjon om organisasjon og anvendelse.
- Alle øvrige delsystemer i NAV bygger på informasjon fra kunnskapsdatabasen.

³Hele rapporten kan sees på <http://metanav.ntnu.no/prosj98/nettadm.pdf>



Figur 1: Fra kildefil til database

3.3 NAV forutsetninger

NAV setter noen krav til en gitt installasjon. Disse må være oppfylt for at systemet skal virke fullt ut:

1. Unike navn på enhetene som skal overvåkes.
2. Unik bruk av vlannummer, også på tvers av VTP-domener.
3. Ved konfigurasjon av nettelektronikk må NAV-konvensjonene følges.
4. NAV kildefiler må fylles og kontinuerlig vedlikeholdes.

NAV v2 gir langt flere frihetsgrader enn NAV v1 hva angår konvensjoner.

3.4 NAV v1 (NTNU NAV)

Før NAVMe startet besto NAV av følgende delsystemer:

1. Kunnskapsdatabase: MySQL-tabeller som inneholder oppdatert informasjon om utstyr, lokalt nett, topologi m.m. Data samles inn nattlig v.hj.av snmp. Kildefiler (nettel.txt og server.txt) angir utstyret som skal overvåkes. 4 sentrale MySQL tabeller: nettel, subnet, swport og trunk. Topologi forstås ut i fra navnsetting av svitsjeporter.
2. Rapporter: Ulike webrapporter som gir informasjon fra kunnskapsdatabasen. Herunder ruterporter, svitsjeporter, spredning av vlan, informasjon om rutere, svitsjer og kantutstyr. Script er skreddersydd for hver rapport.
3. Statistikkinnsamler (basert på Cricket/RRDTool): Gir statistikkdata med god oppløselighet for siste døgn/uke. Historiske data trykkes sammen for å gi gode trendindikasjoner. Krever et absolutt lagringsbehov. Statistikk for bl.a. last, feil, dropp, cpulast m.m. Cricket konfigurasjonstre bygges hver natt basert på kunnskapsdatabasen.

4. Topologi frontend (vlanPlot): Javabasert løsning som visualiserer topologi og last. Gir mulighet for å vandre i lokalnettstrukturene og forfølge laststrømmer i et vilkårlig tidsvindu. Henter topologi fra kunnskapsbasen og lastdata fra RRD-Tool.
5. Terskelmonitor: Trigger på høye last/cpu-verdier m.m. ved å “lukte på” RRD dataene og sende varsel til hendelsessystemet.
6. Statusmonitor (Live): Sjekker om alt utstyr er operativt, rapporterer nedetilstand til hendelsessystemet. Bruker kunnskapsdatabasen for å forstå topologi og avdekke “skyggeforhold”.
7. Hendelsessystem (TrapDetect): Tar i mot meldinger i form av SNMP traps fra andre delsystem, samt fra eksterne system/elektronikk. Ved hjelp av et konfigurerbart varslingsystem får brukere, om ønskelig, beskjed enten på epost eller sms. Hendelsessystemet forstår tilstand (der tilstand forekommer) og varsler både med “sykmelding” og “friskmelding”.
8. Statussiden: Viser oppetilstand (og skyggetilstand) akkurat nå for alle enheter som statusmonitoreres. Viser også historiske data 30 dager tilbake i tid. Kan summere total nedetid for en enhet.
9. Maskinsporing (arp og cam): Henter arp-data fra rutere og brotabelldata fra svitsjer. Lagrer data i MySQL tabeller som i essens viser levetiden til ulike ip/mac i siste 30 dagers periode (eller mer). Muliggjør sporing av ip- og macadresser helt ut til siste svitsjereport.

En forutsetning for at NAV v1 skal virke fullt ut er at et sett med konvensjoner følges. Konvensjonene omfatter description-felt i rutere, portnavnsetting på svitsjer, samt navnsetting av nettelektronikk (rutere, svitsjer, kantenheter). Fullstendig dokumentasjon er på <http://metanav.ntnu.no/v1/konv>.

4 Arbeidet i NAVMe

NAVMe prosjektplan og statusrapporter kan sees på prosjektsiden, <http://metnav.ntnu.no/NAVMe/>. Som det fremgår har vi arbeidet med følgende felt (antall timer pr deloppgave vist i parentes):

1. Kartlegging av ulike høyskolenett/universitetsnett (27 timer).
2. Dialog/opplæring/oppfølging kunden (pilotinstitusjonen) (29 timer).
3. Tilpasning: Innhenting av data (642 timer).
4. Tilpasning: Rapportgenerator og varlse front end (121 timer).
5. Videreutvikling: Hendelsessystemet (394 timer).
6. Justering: Cricket/RRDTool/terskelmonitor (63 timer).
7. Tilpasning: vlanPlot (80 timer).
8. Videreutvikling: navAdmin (239 timer).
9. CVS (47 timer).
10. Installasjon (34 timer).
11. Bruker- og systemdokumentasjon (97 timer).
12. NAVMe Sluttrapport (79 timer).

Det er svært ulik vektning på disse punktene. Den desidert tyngste deloppgaven har vært innhenting av data (deloppgave 3), dernest hendelsessystemet (deloppgave 5).

I dette kapitlet beskriver vi hva som er gjort, samt hvilke avveininger og designvalg som er tatt. Vi følger ikke deloppgavene i prosjektplanen slavisk.

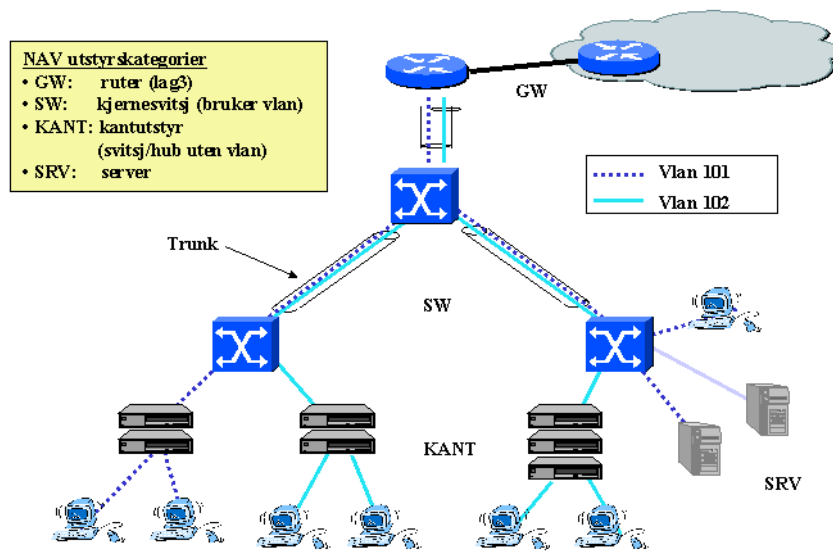
4.1 Nettverksdesign og Konvensjoner

Det har vært nyttig å diskutere konvensjoner og ulike designscenarier med UNINETT og UiTø. Detaljert har vi sett hvordan ulike høyskoleinstallasjoner er implementert, hvilke utstyrstyper som inngår, hvordan design og navnsetting er gjort. Vi har videre studert UNINETT sine ruterport konvensjoner, samt strukturen i UNINETT sitt medlemsregister (DRS).

Vi innser at NAV v1 har en alt for tett knytning mellom managementsystem, utstyrvalg, design og konvensjoner. Det er viktig at en gitt installasjon har flest mulig frihetsgrader. Managementsystemet må ikke gi for mange føringer. Denne erkjennelsen er helt sentral, den har vært en ledetråd for vårt arbeid.

4.1.1 Design av lokalnett

NTNU har en gjennomført design av sine lokalnett med en lik struktur (nesten) overalt. Det er med få unntak en trestruktur med trunker mellom kjernesvitsjene, og kantsvitsjer/huber ut mot brukerne, se figur 2. På kantsvitsjene benyttes ikke vlan. I NAV deler vi disse to lokalnettkomponentene i to kategorier: SW (kjernesvitsj) og KANT (kantsvitsj/hub).



Figur 2: Typisk nettstruktur ved NTNU

UiTø har i svært liten grad KANT-utstyr. De har valgt å benytte vlan helt ut til kanten for optimal ressursutnyttelse/fleksibilitet. Dette vil også være tilfelle andre steder. For NAV er ikke det noe problem, det betyr simpelthen at man har mer SW-utstyr, mindre KANT. (På sikt vil vi kanskje se at kategorien KANT blir helt overflødig, den har en historisk opprinnelse med huber i nettverket).

Topologien/strukturen kan også variere fra sted til sted. Noen vil velge mer redundante modeller med ringstrukturer i den fysiske topologien og basere seg på STP-basert redundans. Det kan være flere parallelle koblinger mellom to svitsjer, eller en svitsj og en ruter, kanskje på ulike vlan, men også på samme vlan (fast ether channel). NAV v2 er nå langt mer generell enn sin forgjenger og vil håndtere de fleste scenarier. I NAV v1 var navnsetting av svitsjeporter helt avgjørende. NAV v2 har løst seg helt fra dette og avleder i stedet topologien selv. Les mer om topologiavlederen i 4.3.

4.1.2 Design på ruternivå

NAV v1 baserer seg på tolkning av description-felt satt på ruterporter for å kunne kategorisere nett i ulike nettyper. Dette er videreført i NAV v2. Viktigste nettyper er:

- lan: Et lokalt nett som går ut mot sluttbrukere. Kan også være et dedikert servernett. Kan ha flere rutere tilkoblet, f.eks. for å støtte HSRP, men primært er ikke et 'lan' brukt for transittrafikk mellom rutere.
- stam: Et stamnett for transittrafikk der fler enn 2 rutere er tilkoblet.
- link: Et punkt-til-punkt nett mellom 2 rutere.
- elink: Som link, men her er motenden en ekstern ruter/brannmur. Dette er altså grensen mot et eksternt nett som er utenfor NAV sitt overvåkningsområde.

NAV v1 bruker videre description-feltet til å si noe om hvem som benytter et gitt lokalnett. Dette gjøres ved å angi organisatorisk tilhørighet (fakultet, institutt, enhet, etc) og anvendelse (student, ansatt, etc). NAV v2 viderfører også dette.

NAVMe har imidlertid gjort noen justeringer. Vi har ryddet bort noen redundante og tildels ulogiske ting. Følgende er styrket:

- Informasjon om organisasjon og anvendelse er strukturert bedre. Vi har laget kildefiler som vedlikeholdes av NAV-administrator. Kildefilene blir til tabeller i kunnskapsdatabasen som igjen muliggjør ryddige relasjoner fra adresseromstabelen.
- NAV v2 kan håndtere en vilkårlig hierarkisk organisasjonsstruktur (f.eks. gruppe under institutt under fakultet).

Siden UiTø har egne rutere har piloten fulgt NAV-konvensjonene fullt ut. Det vil *ikke* være mulig ved en høyskole, som typisk har UNINETT-rutere. UNINETT har sine egne svært innarbeidede konvensjoner for description felt. NAVMe har studert disse. Vi har belyst en del svakheter og foreslått noen forbedringer innenfor rammen av dagens konvensjonsverk:

- Sambands-identifikasjonen har liten mening i mange tilfeller (eks: lokal-ether). Man kan her vurdere å angi nettype.
- Den hierarkiske strukturen for en organisasjon introduserer det litt forvirrende begrepet stedsid, men når dette forstås har man her en entydig modell. Denne kan NAV understøtte, men vi trenger tilleggsinformasjon fra DRS.
- Angivelse av anvendelse er lite gjennomført og bør struktureres bedre.

Med de foreslåtte forbedringene vil NAV kunne tolke UNINETT descriptionfelt. Selv uten en slik tilpasning vil vi kunne få til en del. Dette bør sees nærmere på ved første høyskoleinstallasjon.

NAVMe har også studert muligheten for automatisk avledning av nettype. Det vil gjøre oss helt uavhengig av description-setting for å kunne forstå lag3 topologien! Vi mener en slik avledning vil være mulig, men den involverer en del ny kompleksitet. Eksempelvis må vi ha en måte å skille et lokalnett som bruker HSRP fra et transittnett. Det er altså ikke nok å telle antall rutere som grenser til et gitt subnett, man må også forstå om transitttrafikk er ment å gå denne veien. Vi foreslår at dette arbeidet videreføres i et nytt prosjekt.

4.1.3 Navnsetting av utstyr

NTNU har en gjennomført konvensjon for navnsetting av nettelektronikken (altså ruterne, kjernesvitsjene og kantsvitsjene). Navnet skal i størst mulig grad reflektere funksjon, plassering og hvilke miljøer enheten betjener. Som et eksempel heter en kantsvitsj som betjener studenter på Bygg-fakultetet plassert i Materialteknisk verksted for byggstud-370-h. Vi har da i tillegg en romdatabse som sier at telematikkrom 370 er et rom i Materialteknisk verksted. Telematikkrombenevnelsen er for NTNU gjennomført helt ut med punktmerking og følger Norsk Standard.

I NAVMe har vi sett at UNINETT har en lik navnsettingsfilosofi i bunn, men de har munnet ut i litt avvikende konvensjoner. UiTø har hatt en mindre konsekvent navnsettingspraksis, noe som sikkert er tilfelle også andre steder.

Vi finner det mest fornuftig å løsrive navnsetting av utstyr *helt* fra NAV. Det gir en optimal fleksibilitet. NAV vil ikke lenger diktere denne typen detaljer (som det typisk hersker mange betemte meninger om). Altså, dør barnet hva du vil, NAV gjør ingen tolkninger av navnet.

Vi vil uansett kunne si mye om en gitt enhet. Dette p.g.a. plasseringsinformasjon, kategori og type, samt beskrivelse av subnett som utstyret befinner seg på.

Det vi sitter igjen med er et fornuftig krav om unike navn. Videre har vi jobbet med problemstillinger rundt plassering av utstyr i ulike dns-domener, samt avvik mellom sysname (navnet konfigurert på enheten) og dns-navnet. NAVMe foreslår at det endelige navnet velges ut i fra en sammenligning av sysname og dnsnavn, der vi rapporterer feil ved inkonsistens. Vi lar navnet være det fullt kvalifiserte dns-navn, der en konfigurert endelse utelates (for ikke å få for langt navn). Typisk vil endelsen ntnu.no utelates ved NTNU, uit.no ved UiTø. NAV v2 støtter en utelatelse av flere slike domene-endelser ved en installasjon.

4.1.4 Valg av utstyr

NTNU sitt nett er i all hovedsak realisert ved hjelp av Cisco rutere og Cisco kjernesvitsjer, samt 3Com kantsvitsjer. NTNU NAV har således støtte for en lang rekke Cisco utstyr, samt 3Com svitsjer/huber.

Det viser seg at Cisco er en svært dominerende gjenganger ved andre høyskoler/universitet. Ved UiTø er det omtrent bare Cisco. Vi støter imidlertid på noen nye varianter av Cisco-utstyr.

For å lettere supportere nye utstyrstyper har NAVMe gjort noen ombygninger:

- Vi angir ikke lenger utstyrstypen i kildefilen nettel.txt. Innlesingsscriptet finner nå typen selv. Dette gjøres ved å se på standard-MIBen system.sysObjectId. Denne gir en eksakt typebetegnelse.
- Vi har utvidet med en kildefil type.txt som fyller typetabellen. Her gis typebetegnelse og menneskelig forklaring av sysObjectId. Denne filen utvider vi manuelt etterhvert som vi støter på nytt utstyr.
- Et viktig felt i typetabellen er typegruppe. Alle utstyrstyper som vi kan behandle likt blir samlet i samme typegruppe. F.eks. favner typegruppen cat-sw Catalyst 2926, 4003, 4006, 5000, 5500, 5505, 5509, 6509 med evt flere. Alle disse typene er like sett fra et snmp-messig perspektiv.
- I alle script baserer vi oss på typegruppe, ikke på en enkeltstående type. Dette blir mer generelt og er mye mer robust mot feil.
- Videre benytter vi i størst mulig grad standard MIBer, ikke private MIBer, ved uthenting av data. Da er vi sikret en størst mulig lik behandling. Eksempler på bruk av standard MIBer er ved uthenting av arp-data, brotabell-data, tilkoblede nett, lastellere osv.

4.1.5 Stedsinformasjon

Stedsinformasjon har også medført en del hodebry. Som nevnt baserer NTNU seg på en gjennomført nummering av telematikkrom. Vi har videre en kildefil og tilhørende databasetabell som beskriver lokasjonen til et gitt romnummer. UiTø hadde ingen slik løsning på plass.

UNINETT har sentralt et mer overordnet begrep med stedsid'er. En stedsid beskriver unikt en lokasjon til et gitt UNINETT-medlem. Stedsid'en er videre angitt med UTM-koordinater for å kunne tegne geografisk riktige kart. UNINETT har imidlertid ikke sett på nummering av telematikkrom, mest fordi UNINETT typisk er i kun et rom ved en gitt stedsid.

NAVMe foreslår to kildefiler og to tilhørende tabeller for å angi telematikkrom og område. Område vil typisk være ekvivalent med UNINETT sin stedsid, men må ikke være det. En gitt NAV installasjon står fritt til å samle sine telematikkrom i en områdeinndeling de finner fornuftig. De kan også gruppere alt i samme område.

UiTø har vært opptatt av beskrivende romnummer (tre siffer beskriver dårlig). Vi har da tillatt 10 tegn i romnummeret. I NAV v2 benyttes dette som nøkkel i databasen. I NAV v3 vil vi foreslå en liten ombygning der vi her som ellers har skjulte nøkler.

4.2 Utvidelser i kunnskapsdatabasen

Deloppgaven som har tatt desidert mest tid i NAVMe er arbeidet med kunnskapsdatabasen. Denne har jo også en grunnleggende viktig rolle. Alt annet er bygd over den. I 4.1 har vi allerede vært inne på noen utvidelser i kunnskapsdatabasen, de gjentas ikke her.

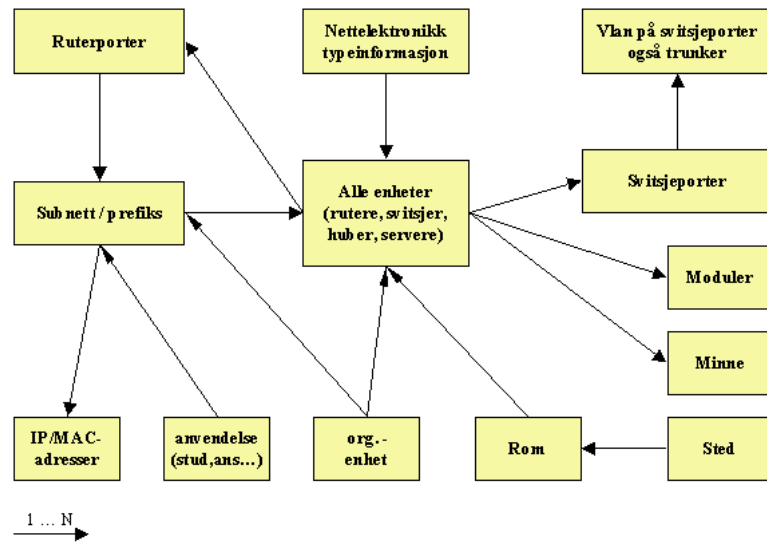
4.2.1 Generelle forbedringer

NAV v1 har en relasjonsdatabase på plass, implementert i MySQL. Den har vokst frem gradvis og bærer preg av det. I NAVMe har vi revurdert hele løsningen. Vi fant at vi ville beholde grunnprinsippene, men styrke løsningen i mange ledd. Dette har ført til flere kildefiler og mange flere tabeller, men også ryddigere relasjoner mellom tabellene. Vi fikk tidlig i prosessen et ER-diagram på plass (som dog har vært justert underveis). Hovedtrekkene er vist på figur 3.

En avveining har vært hvorvidt vi bør legge inn data ved hjelp av kildefiler, eller ved hjelp av web-skjemaer. Vi mener flate filer som inneholder alle inndata gir en bedre oversikt og gjør innlegging mer rasjonelt. Det gjør også at vi når som helst kan starte fra grunn av og bygge opp kunnskapsdatabasen på nytt (ved feil, datatap, andre forhold).

Et annet valg er omfanget av kildefilene. Dersom volumet blir for stort kan det virke avskrekkende og komplekst for NAV-administrator. Her er vi spente på tilbakemeldinger fra pilotbrukerne. Med NAVMe har vi vokst fra 3 kildefiler (nettel.txt, server.txt og rom.txt) til 8 kildefiler som må redigeres av NAV-administrator. De fordeler seg på fire kategorier:

1. Bokser (nettelektronikk/servere)
 - nettel.txt: Inneholder all nettelektronikk som overvåkes.
 - server.txt: Alle servere som statusmonitores/topologiavledes.
2. Geografi
 - rom.txt: Definerer telematikkrom/skap hvor utstyr er plassert.
 - sted.txt: Definerer områder. Et rom tilhører et område.
3. Kategorisering av nett
 - org.txt: Organisatoriske enheter. Tillater en hierarkisk struktur.



Figur 3: Kunnskapsdatabasen

- anv.txt: Anvendelsestypene for nett (ansatte, studenter etc).

4. Tilleggsinformasjon om nett

- vlan.txt: Definerer vlnnr som er i bruk og angir hvilke subnett de tilhører.
- prefiks.txt: Lister spesielle prefiks som ikke kan hentes fra ruterne.

Kildefilene og kunnskapsdatabasen er dokumentert (og diskutert) i mer detalj i system- og admindokumentasjonen, se lenker under <http://metanav.ntnu.no/v2>.

Et svært drastisk valg var overgangen fra MySQL til PostgreSQL. PostgreSQL har vesentlig mer avansert databasefunksjonalitet innebygd, samtidig som den fortsatt er open source. Helt avgjørende er støtten for fremmednøkler og dertil mulighet for "cascading delete". Når f.eks. en ruter fases ut, ønsker vi at alle ruterporters, all minneinformasjon og all modulinformasjon også blir slettet. Med PostgreSQL trenger vi ikke skrive særskilt kode for slike forhold. Databasen tar seg av en slik konsistenssletting selv.

4.2.2 Inventar

NAVMe har styrket inventarfunksjonaliteten vesentlig. Vi har nå en egen tabell for minne, som ivaretar informasjon om ulike minnetyper, som flash og hovedminne. Vi har også en tabell for moduler, som bryter ned en modulbasert svitsj eller en modulær ruter i moduler/port adaptere. Minneenheter og moduler relaterer seg til en gitt enhet (ruter/svitsj).

Med denne løsningen kan rapportgeneratoren nå gi oversikt over en gitt catalyst-modul, hvor mange man har av den, hvor de er etc. Eller en total minneoversikt, eller flashoversikt, osv.

4.2.3 IP adresserom og ruterporter

NTNU NAV opererte med en tabell for alle ruterporter. Den ga for såvidt et komplett bilde over alle subnett i bruk, men modellen hadde en del svakheter. Det var blant annet redundant informasjon rundt et gitt subnett som flere rutere henger mot, og hva verter er, det var komplekst å se strukturen.

NAVMe fant det fornuftig å lage en egen prefikstabell i tillegg til ruterporttabellen. Prefikstabellen inneholder alle subnett i bruk (med nettadresse og maske). Ruterportene har en relasjon til en gitt prefiksrad. Vi har da en ryddig måte å se hvor mange rutere som er tilkoblet et subnett. Vi kan videre gi alle enheter vi overvåker en prefiksreferanse. Vi kan sågar gi alle arp-innslag en slik referanse. Sistnevnte gjør blant annet telling av antall maskiner på et gitt subnett lettere.

NTNU NAV hadde kun oversikt over de subnettene som er direkte tilkoblet en NAV-overvåket ruter. NAV v2 støtter også subnett som er statisk rutet ut bak en ruter/brannmur som ikke overvåkes. Vi kunne løst dette ved å finne de statiske rutene ved hjelp av SNMP, men siden volumet er lite, fant vi det mer rasjonelt å benytte en kildefil for dette (prefiks.txt). Samme fil benyttes også for å angi tildelte adresseblokker, samt totalt adresserom. Vi får således en del manuelt innsatte nettyper (statisk, tildelt, adresserom) som sammen med de SNMP-uthentede danner et komplett bilde av forbruket av adresseblokker. I NAV v3 kan vi lett lage en grafisk visualisering av dette forbruket, NTNU NAV gjør dette, men da på en mindre generell måte.

NAV v2 er også utvidet til å støtte HSRP. Vi legger inn en ekstra ruterportrad for HSRP IP-adressen og knytter denne til ruterens som er aktiv (ved polling). Vi har videre innført et begrep, rootgwid, for alle subnett. rootgwid er den ruterens som NAV har valgt som rot for subnettet. Rota er startstedet for topologivandringen og således bestemmende for retningen. Vlanet sees fra denne ruterens. I tilfellet HSRP velges HSRP IP-adressen som rot, ellers velges laveste IP-adresse. Merk at valget av rot ikke må samstemme med faktisk rot i spanning-tree, det har bare betydning for hvor vi skal starte å utforske topologien.

En annen viktig utvidelse i NAV v2 er støtten for subinterfacer. Denne konfigurasjonen finnes ikke på NTNU, mens den er helt sentral ved UiTø og ved UNINETT forøvrig. Vi har studert MIB-støtten rundt dette, avhengig av hardwareplattform og softwareversjon har Cisco varierende støtte for subinterfacer. Det optimale er naturligvis at hvert subinterface har både interfacellere og ifAlias (description). Dette har lenge vært et problem, men ser ut til å være løst for de viktigste plattformene nå.

Dersom interfacellere ikke kan hentes velger NAV v2 å vise lasten fra masterinterfacet. Dette er naturligvis ikke helt riktig, men det er det beste vi kan gjøre. For Cisco 8540-plattformen er dette eneste mulighet og den er pr dato meget sentral ved UiTø (men vil bli byttet ut i januar 2002). Uansett, for subinterfacer som ikke selv har lasttelling har ruterportraden en peker til masterinterfacet. Videre har vi utvidet til å samle inn ruterporter som ikke har IP adresserom (som jo er tilfellet for slike masterinterface). Her var det en del andre fallgruver, som f.eks. individuelle isdn B-kanaler. Vi har nå logikk for å unnta en del slike uinteressante interface.

8540 gir heller ikke ut description for subinterface ved SNMP. Da har vi heller ingen måte å få satt organisasjon og anvendelse på vanlig måte. Det er svært uheldig, vi har løst det ved å statisk dublere denne informasjonen i kildefilen vlan.txt. Vi er ikke tilfreds med denne løsningen, men velger å skylde på en svært spesiell hardwareplattform som heldigvis ikke har noen utbredelse i UNINETT.

4.2.4 Vlan og svitsjeporter

Også i den svitsjede delen av nettet har NAVMe gjort fundamentale forbedringer. NTNU NAV hadde to tabeller for denne informasjonen, swport og trunk. Vi hadde da forsåvidt all informasjon om alle svitsjeporter og hvilke vlan som gikk på hvilke trunker. Men sistnevnte var implementert via en del “hack” som virket på NTNU. Det var altså ikke helt generelt løst. Vi kommer inn på NAV v2 sin vlanavledning i 4.3.

Topologiforståelsen i NTNU NAV sto og falt på riktig portnavnsetting. Svitsjeportene måtte navnes ut i fra hvilken enhet som var koblet til porten. Dette er jo en fornuftig navning som vi foreslår fortsetter, men da primært for å gi administrator en god oversikt når han er inne på svitsjen og gjør arbeid. Portnavnsetting er altså ikke lenger et NAV-krav.

NAV v1 konvensjonen spesifiserte også en måte å gi retningsangivelse ved portnavnsetting. Nettkartet brukte dette for å kunne tegne opp topologien. I tillegg til opp og ned hadde vi retningen horisontal. Horisontal var typisk trunker i kjernestrukturen der man hadde vlan som hadde sitt ruteropphav fra forskjellig hold. NAVMe kommer dette til livs ved å støtte en retning pr vlan, ikke pr trunk.

Et annet aspekt vi har endret litt er koblingen mellom subnett og vlan. I NTNU NAV satte vi vlan-navn tilsvarende description-felt og dernest hentet vi vlan informasjon fra alle VTP-serverne (vi anga VTP-serverne i kildefilen). Dette fungerte, men løsningen var ikke helt intuitiv fra et driftsperspektiv. Uttak av et nytt vlan var litt komplisert da man var avhengig av et komplett bilde på tvers av VTP-domenene. En implisitt betingelse er unikhed av vlan for hele NAV-installasjonen. Dette kravet fortsetter, vi mener det er udiskutabelt fornuftig. Fra et operativt synspunkt er gjenbruk mulig på tvers av VTP-domener, men det er uryddig og forvirrende.

NAVMe er her inspirert av UiTø og foreslår at vlan kildeinformasjon samles i en kildefil, vlan.txt. Uttak av et nytt vlan skal da skje her. Det vil enkelt hindre uheldig dublering. Navningen av vlan på VTP-server er da ikke viktig, sett fra et NAV ståsted.

Det som er viktig er at vlan.txt gir limet som binder vlannummer til subnett/prefiks. Dette gjøres ved å angi nettype og nettident som er (omtrent) ekvivalent med description-beskrivelse på ruterport. Alternativt kan IP-adresserom brukes for å gi dette limet, men husk at man kan ha flere parallelle adresserom for et gitt vlan/broadcast domene.

4.3 Topologiavlederen

Topologiavlederen er en av de største fremskrittene i NAVMe og utgjør muligens den mest komplekse delen av systemet. Algoritmen som brukes er generell og kan avlede strukturen til en vilkårlig sammensetning av svitsjer og rutere i et lokalnett.

Avlederen baserer seg på en oppdatert kunnskapsdatabase med informasjon om alle svitsjer og rutere, herunder svitsje- og ruterporter. Macadresseinformasjon for alle NAV-enheter må også være tilgjengelig (ARP-innsamleren har samlet inn dette).

Algoritmen kan oppsummeres slik:

1. Avled fysiske sammenkoblinger mellom enhetene på lokalnettnivå.
2. Avled hvilke vlan som *kan gå* på hvilke trunker. Finn således den totale utbredelsen til alle vlan.
3. Finn den faktiske tilstanden for hvert enkelt vlan. Sett retning ut i fra primær ruterport (rootgwid) og angi alle veier som spanning tree blokkerer.

4.3.1 Fysisk sammenkobling

Primært hjelpemiddel for å avlede fysisk sammenkobling er brotabellene til svitsjene. Ved å hente brotabelldata med SNMP og ta vare på de interessante macadressene, dvs macadressene til NAV-registrerte enheter, bygger vi en hjelpetabell. For hver svitsj lagrer hjelpetabellen hvilken svitsjeport en annen enhet befinner seg bak.

Hjelpetabellen sier ingenting om direkte naboskap. Men ved å se den totale informasjonen i sammenheng kan vi i prinsippet klare å avlede det direkte naboskap. I praksis er det imidlertid en del faktorer som kompliserer dette:

- Dersom en enhet ikke har blitt tilsnakket på lang tid kan denne være borte fra ditto brotabeller. Dette motvirkes i mange tilfeller av vår egen statusmonitor som jevnlig pinger alt utstyret.
- Svitsjeporten til en kantsvitsj A som henger bak en kjernesvitsj B vil normalt ikke ha B i sin brotabell, da det er usannsynlig at maskiner bak A snakker med B, altså managementinterfacet til B. De sender naturligvis pakker gjennom B, men det er noe annet.
- Det viser seg at macadressen til ruterer (som jo alle på et gitt subnett kommuniserer med) ikke legger seg på den interne svitsjeporten til en hybridsvitsj av typen Cat5k/RSM.
- Vi har påvist bugs i en rekke softwareversjoner på Catalyst 3524XL plattformen som gjør at *kun* macadresser som starter på 00 lar seg hente ved hjelp av SNMP. Vi har rapportert inn denne feilen til Cisco TAC og en bugfix foreligger av 30/10 2001.
- Vi har en del andre uforklarlige mangler i brotabellinformasjon på Catalyst 5000 plattformen som gjør at macadresser som burde vært der tidvis mangler. Vi har ikke funnet noe system på denne feilen.

Gitt disse problemene har vi valgt å supplere brotabellinformasjonen med CDP (Cisco Discovery Protocol). CDP har også sine begrensinger:

- CDP virker bare på Cisco utstyr.
- CDP forteller bare om direkte *Cisco naboskap*, utstyr fra en annen leverandør kan ligge i mellom uten at dette blir oppdaget.
- CDP annonsering kan også være avskrudd på Cisco utstyr. NAV anbefaler å ha det påskrudd.

Siden dominansen til Cisco er såvidt stor både på NTNU og UiTø har imidlertid CDP vært forløsende på våre brotabellproblemer.

4.3.2 Vlan på trunker

Nattlige script sørger for å oppdatere svitsjeporttabellen. I tillegg settes vlanverdien på alle porter som ikke er trunk. Det som da gjenstår er å avlede hvilke vlan som går på hvilke trunker og her kommer topologiavlederen inn.

Først avgrenses det øvre settet av tillatte vlan på en gitt trunk Dernest traverserer vi rekursivt den fysiske topologien for et gitt vlan og ser på svitsjer der aktuelle vlan er lokalt definert. Vi sier at vlnet har lov til å gå på alle trunker fra primær ruterport

(rootwid) som gir en forbindelseslinje til svitsjer som har vlanet satt lokalt. Dette gir et korrekt bilde.

Merk at vi bevisst ikke setter vlanverdier basert på det som er administrativt konfigurert. Årsaken er at mange installasjoner bevisst velger av administrativ enkelhet å tillate alle vlan på alle trunker. Andre kan igjen basere seg på VTP pruning der svitsjene selv beskjerer vlanstrømmen dynamisk basert på spanning tree informasjon. Merk videre at en slik aktiv vlan tilstand på trunken ikke kan hentes via SNMP på de Cisco svitsjene vi har sett på.

4.3.3 Spanning tree tilstand

Siste oppgave for topologiavlederen er å pr vlan se på den faktiske tilstanden til spanning-treet. Informasjonen hentes ved hjelp av SNMP og porter som står blokkert får denne tilstanden satt i databasen.

Videre setter vi retning for svitsjeportene som danner vlanet ved å traversere trestrukturen som nå danner seg. Retningen settes ut i fra primær ruterport (rootwid). Denne retningen benyttes av nettkartet som visualiserer topologien.

4.4 Konsistenssjekker / feilrapporter

Konsistenssjekkeren er tett integrert med topologiavleder. Den gir en rekke feilrapporter:

- Portnavn som ikke er satt, eller er satt feil (ut i fra en topologisk tolkning).
- Misforhold mellom administrativt satte vlanbegrensninger på trunker og reelt behov (gitt at VTP pruning ikke benyttes).

Flere ting bør komme med NAV v3, f.eks:

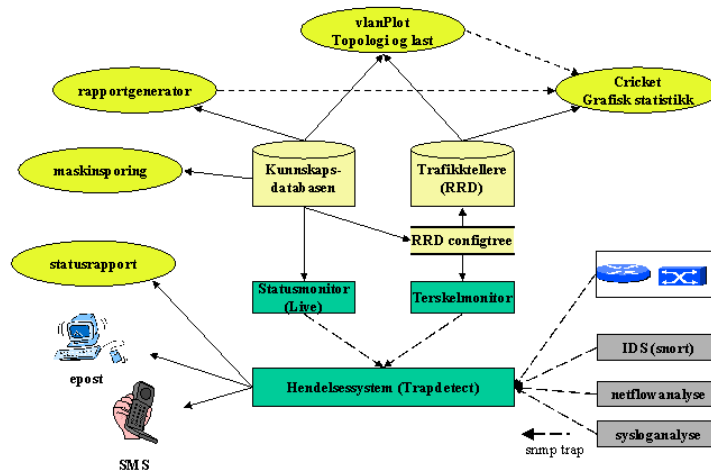
- Dupleks misforhold (half vs full) mellom sammenkoblede svitsjer.
- Oppdaget uregistrert enhet (typisk med CDP).

Vi sjekker også konsistens og finner feil i de natlige scriptene som fyller kunnskapsdatabasen. Vi har ikke strømlinjeformet feilrapporteringen her enda. Eksempler på feil kan være:

- Ukjent utstyrstype (ukjent sysObjectID).
- Misforhold mellom sysname (konfigurert på enheten) og dnsnavn.
- Feil descriptionfelt syntaks på ruterport.
- Manglende definisjon av rom, organisasjon eller anvendelse.
- og så videre, og så videre...

4.5 Andre delsystemer

Som vist på figur 4 er hele NAV bygd rundt kunnskapsdatabasen. Alle øvrige delsystem henter data her, direkte eller indirekte. De store forandringene i kunnskapsdatabasen medførte naturlig nok forandringer i alle grensesnittene. Utover det har vi også gjort store funksjonelle forbedringer. Det gjelder først og fremst rapportgeneratoren (som er helt ny) og hendelsessystemet, men også nettkartet.



Figur 4: NAV delsystemer

4.5.1 Rapportgeneratoren

NAV v1 har en rekke rapporter som viser oversikt over alle rutere, svitsjer, subnett og svitsjeporter. Rapportene er skreddersydd til hvert formål og ny kode må skrives for å lage en ny rapport.

I NAVMe ønsket vi en mer generell rapportgenerator der man i en konfigurasjonsfil definerer de rapportene man ønsker. Dette har vi realisert og resultatet er en stor forbedring.

Et eksempel viser det NAV v2 støtter:

```
gw {
  $sql="SELECT romid,sysname,typeid,orgid,boksid FROM boks WHERE kat='GW' ";
  $overskrift = "Rutere";
  $navn_sysname = "Navn"
  $order_by="romid";
  $ekstra="ruterporter,moduler,minne";
  $skjul = "boksid";
  $url_romid="?rapport=rom&romid=$romid";
  $url_typeid="?rapport=type&typeid=$typeid";
  $url_orgid="?rapport=org&orgid=$orgid";
  $url_ruterporter="?rapport=gwport&boksid=$boksid";
  $url_minne="?rapport=mem&sysname=$sysname";
  $url_moduler="?rapport=modules&sysname=$sysname";
}
```

Denne konfigurasjonen gir en rapport over rutere som viser plassering, type og driftsansvarlig enhet. Rapporten kan for UiTø sees på <https://po.uit.no/rapportgenerator/?rapport=gw>

Mulighetene i konfigurasjonsfilen er blant annet:

- Man kan skrive en generell select-spørring mot databasen. Det er noen begrensinger ved avanserte spørringer.
- Man kan definere rapportoverskrift og individuelle feltoverskrifter.
- Man kan lage URLer videre fra en rad, der vilkårlig feltverdi kan inngå som parameter.
- Man kan skjule rader i visninger (typisk nøkkelfelt som må med for å kunne gjøre relasjoner til andre tabeller ved URLer).
- Rapportene støtter sortering i alle kolonner, samt avanserte søkemuligheter.

Vi har forslag til utvidelser. Vi ønsker f.eks. en mulighet for å lage rapporter som en funksjon av sikkerhetsnivået til NAV-brukeren som er innlogget. Videre ønsker vi ytterligere fleksibilitet ved URLing bl.a. for å få til lenking over til Cricket-statistikk. Vi ser for oss at rapporter overtar som frontend mot Cricket og således erstatter/supplerer standardmenyene som følger med Cricket.

4.5.2 Trafikkinnsamling (RRD og Cricket)

Funksjonaliten i Cricket er stort sett den samme i NAV v2 som i v1. Det natlige scriptet som genererer Cricket sitt konfigurasjonstre er tilpasset den nye kunnskapsdatabasen. Vi har også endret kriteriene noe. Vi samler f.eks. ikke inn last for subinterfacer som ikke har lasttellere, men derimot masterinterface (som ikke har IP adresserom).

Vi har videre styrket visningen av grafer ved å gjøre det mulig for brukeren å angi skalaen interaktivt. Det har lenge vært en svakhet med Cricket at store maksverdier i dataene gir en uheldig skalering av grafen. Nå kan brukeren selv justere slike ting etter eget ønske.

Vi har også løst floken med 32bits lasttellere, 5 minuttts polleinterval og høy last (over 115 Mbps). Problemet er her at telleren kan rekke å ta en hel runde før ny poll og RRD vil bli lurt. Vi har løst dette ved å polle hvert minutt⁴ for gigabitinterfacer og dernest beregne et gjennomsnitt av 5 sammenhengende poll for å få 5 minuttts gjennomsnittet. Dette settes inn i RRD-basen. Det er ikke den mest elegante løsningen, men det var den eneste måten vi fikk RRD til å samarbeide på. En bieffekt er at vi får et litt rotete Cricket webmenysystem. Det vil vi igjen løse med en generell rapportgenerator overbygning i NAV v3.

Lastscriptet som nettkartet bruker er også omskrevet mot den nye databasen.

4.5.3 Nettkartet (vlanPlot)

Nettkartet (vlanPlot) er, av tekniske årsaker, realisert som en applet skrevet i Java v1.1. vlanPlot henter beskrivelse av nettverkstopologien fra en modul på serveren, også denne skrevet i Java, som bruker kunnskapsdatabasen som datagrunnlag. Last-data hentes fra et eget script; her er Cricket og RRD-filer grunnlaget. vlanPlot er fokusert på grafisk visualisering, og så mye som mulig av logikken befinner seg derfor på serveren; dette avlastet klienten og gir økt sikkerhet. Mer informasjon om vlanPlot finnes i systemdokumentasjonen, se <http://metanav.ntnu.no/v2/systemdok>.

Her beskrives kort de endringer som er gjort i NAVMe: vlanPlot v2 er en videreutvikling av den eldre versjonen som ble skrevet for NTNU NAV. Gjennom denne

⁴Vi takler da datarater opptil 572Mbps.

prosessen har kildekode for både server-modulen og vlanPlot blitt restrukturert for å øke hastighet, pålitelighet og vedlikeholdbarhet.

Store endringer har vært nødvendig for å takle overgangen til det generaliserte NAV v2, og mye av logikken har i denne prosessen blitt flyttet fra klienten til serveren. Server-modulen har blitt skrevet om fra bunnen, spesielt for å imøtekomme kravet om at all tekstlig informasjon som vises i vlanPlot skal konfigureres på serveren; dette er nødvendig for at hver installasjon av NAV skal kunne velge individuelt hvilken informasjon som skal vises på de ulike sikkerhetsnivåene.

I NAV v2 angis retningen på forbindelser per vlan (se 4.2.4), og det som i NTNU NAV ble kalt for "horisontale linker" er derfor blitt borte. I vlanPlot v2 fokuserer man på kun et vlan av gangen når man vlan-vandrer, og alle forbindelser går dermed enten opp eller ned. Det er imidlertid raskt og enkelt å bytte hvilket vlan man ser på, dette er vesentlig styrket i den nye versjonen.

Det har også dukket opp flere nye problemstillinger ved UiTø. Der ser vi eksempler på flere fysiske forbindelser mellom samme ruter og svitsj (men som bærer forskjellige vlan). Når man bytter vlan vil man også bytte fysisk port og dertil last; vlanPlot v2 tar høyde for dette.

4.5.4 Statusmonitor

Det har ikke blitt gjort noen store endringer i statusmonitoren (Live) annet enn å tilpasse den til den nye databasen. Herunder har vi bygd om mekanismene for hvordan skyggeforhold avledes, siden topologiavlederen nå gir denne informasjonen. I NAV v1 bruker Live en tolkning av portnavn og satt sine egne hjelpefelt i databasen.

Et problem med dagens Live er at den er sekvensiell og ved mye nedetid i nettverket vil den ikke rekke rundt til alle boksene i løpet av sin 4. minutters periode. Som en neste forbedring foreslår vi en ombygging av Live der den i stedet pinger i parallell.

4.5.5 Terskelmonitor

Terskelmonitoren er ikke endret nevneverdig. Den startes fortsatt av cron hvert femte minutt og leter etter verdier som i gjennomsnitt siste kvarter er over 80% av maksverdien. Slike tilstander legges i en flat tekstfil. Når tilstanden inntreffer første gang sendes en trap til hendelsessystemet. Når verdien igjen er under 70% oppheves tilstanden (i tekstfilen) og friskmelding sendes.

I NAV v3 foreslår vi å gjøre terskelmonitoren mer generell, slik at man kan justere hvilke verdier terskelen skal slå inn på, samt hvilke variable den skal "lukte på". Helst bør alt dette være konfigurerbart på enhetsnivå.

4.5.6 Hendelsessystem

Hendelsessystemet er styrket radikalt med NAVMe. Vi har laget et nytt databasedesign der vi har innført nye tabeller og relasjoner. Vi har sågar skilt ut trapdetect tabellene i en separat database. Vi fant dette mest ryddig, da de har lite med kunnskapsdatabasen å gjøre. Databasen er (i likhet med kunnskapsdatabasen) nå implementert i PostgreSQL.

Trapdetectbasen holder rede på alle definerte hendelser (dvs SNMP traps med tilhørende subOIder). Dette leses inn fra en veldefinert kildefil, trapdetect.conf. Kildefilen knytter også trappen (hendelsen) til utstyrskategorier (GW,SW etc) og organisatoriske enheter. Dette er igjen representert som tabeller i databasen. Således avgrensers vi ef-

fektivt hvilket utstyr (basert på kategori) som hendelsen har gyldighet for og hvilke brukere som er kvalifisert for varsling.

Det bringer oss over på brukersiden av databasen. Her er NAV-brukere registrert med sin varslingsprofil. Brukerne er av NAV-administrator tilknyttet en, evt flere, organisatoriske enheter. Innenfor denne ytre rammen kan brukeren selv sette opp sin varslingsprofil, representert med ytterligere tabeller i databasen. Brukeren kan velge hvilke hendelser han ønsker varsel på. Han kan også definere unntak fra en gitt utstyrskategori, eller evt definere enkeltstående tillegg. I essens kan brukeren justere sin varslingsprofil helt ned på enhetsnivå. Dette er nytt i NAV v2, og det har kommet etter ønske fra NAV-brukere ved NTNU.

Webgrensesnittet for bruker og NAV-administrator er styrket i henhold til den nye funksjonaliteten. Vi har lagt vekt på brukervennlighet med rullegardinskjemaer der brukeren kan klikke og innstille sin profil. Det ligger mye ny kode med logikk for å sette inn, slette og oppdatere informasjon i trapdetectdatabasen. Det har her vært svært nyttig med tilbakemelding fra pilotbrukere ved UiTø som har avdekt en del “barnesykdommer”.

Vi har også vektlagt online hjelpinformasjon. Vi har prøvd å skjule tekniske termer for brukeren (som 'traps' og 'snmp').

For hver enhetsgruppe/enkeltenhet kan brukeren velge varsling på epost, sms eller dagtid sms⁵. Funksjonaliteten rundt dagtid sms er styrket ved at brukeren mer fleksibelt kan justere sin definisjon av dag.

I NAV v1 hadde vi svært begrensede muligheter til å justere utseendet på sms-meldingene. Formatet må jo være kompakt med kun relevant informasjon. Syntaks vil igjen variere fra hendelse til hendelse. NAV v2 har løst dette ved å innføre en egen konfigurasjonsfil der NAV-administrator kan formatere smsmeldinger pr hendelse.

Trapdetectprogrammet er i NAV v2 ombygd til å støtte den nye databasen. Programmet jobber videre mot status-tabellen i kunnskapsdatabasen som holder rede på tilstand til hver enkelt hendelse (samt annen relevant informasjon). Når en ny gjenkjennbar SNMP trap mottas fra den underliggende trapdemonen vil trapdetect vurdere denne mot eksisterende tilstand i status-tabellen. Dersom vi har en endring blir tabellen oppdatert og brukere blir varslet i henhold til sin definerte profil.

4.5.7 Statussiden

Statussiden viser hvilke enheter som er nede (evt i skygge) akkurat nå. Man kan også få informasjon om hvilke brukergrupper som er berørt. Videre er det en historiefunksjon der man kan se hva som har foregått opptil 30 dager tilbake i tid. NAV v1 funksjonalitet er her overført til NAV v2 med de nødvendige omskrivninger. Vi har også gjort en del ombygning fra perl til php.

NAV-brukeren har mulighet til å sette enheter på 'service' slik at de unntas fra varsling (typisk ved planlagt nedetid). NAVMe har styrket brukergrensesnittet ved at brukeren kan velge enheter fra en liste (hentet fra boks-tabellen i kunnskapsdatabasen).

I NAV v3 ønsker vi å gjøre statussiden mer generell, slik at den kan gi en status for alle hendelser, gjerne med mulighet for å avgrense visningen basert basert på organisasjonstilknytninger og evt andre kriterier. Vi ønsker også å utvide med en generell meldingstjeneste slik NTNU NAV har.

⁵Dagtid sms vil si at brukeren kun får sms-melding på dagtid. Dersom en hendelse inntreffer utenom dag vil den ikke bli sendt før neste morgen. Dersom hendelsen også er friskmeldt før den tid, blir den aldri sendt.

4.5.8 Mac- og IP-adresse oversikt

Denne baserer seg på søk i ARP-tabellen. Under fasaden ligger naturligvis cronjobben som henter arp-data fra alle ruterne og oppdaterer informasjonen i tabellen. Vi har også en slettejobb som sletter gamle data (nå 30 dager).

Brukergrensesnittet gir mulighet for å se et intervall med ip-adresser eller evt en macadresse og når den/disse var på nett.

Her snakker vi all hovedsak om videreføring fra NAV v1. Som ellers har vi ombygd til PostgreSQL. I tillegg har vi styrket integrasjonen ved å legge inn en relasjon mot adresseromstabelen i databasen. Vi kan da enkelt si hvor mange maskiner som er på et gitt subnett (og hvilke) til en gitt tid.

NAVMe har også løst en feilkilde forbundet med proxy ARP. Proxy ARP gir forvirrende ARP-innslag som vi ikke ønsker å ta vare på.

NAVMe har *ikke* videreført innsamlingen av brotabelldata fra alle svitsjene, dette er nevnt som en naturlig utvidelse under kapittel 7.

4.6 NAV omgivelsene

NAVMe har også styrket omgivelsene rundt selve programmet. Vi tenker da på installasjonsprosess, versjonshåndtering, filsystem og dertil muligheter for lokale tillegg, samt dokumentasjon.

4.6.1 Installasjon av hardware / software

Vi har installert NAV v2 både ved UiTø og NTNU. Her har vi høstet en del erfaring rundt hardwarekrav, partisjonering av filsystem og installasjon av programpakker. Dette er dokumentert i admindokumentasjon, se metanav.ntnu.no/v2/admindok/.

Vi presiserer at vi bevisst ikke har brukt tid på andre operativsystem enn Mandrake linux. Vi tror systemet kjører fint under andre unix-varianter, men har ikke prioritert dette. Det vil vi heller ikke gjøre i fortsettelsen, i hvertfall ikke i første fase. Vi møtte en viss motstand her ved UiTø. De har en forkjærlighet for en annen unix-variant (FreeBSD). Vi er takknemlig for at de føyde seg, slik at vi kunne fokusere på viktigere ting.

I pilotprosjektet er det prosjektgruppen som har gjort det meste rundt oppsett av maskin, og alt rundt installasjon av nødvendige programpakker. Vi anbefaler at et slikt sentralisert oppsett fortsetter ved neste installasjon, men at vi gradvis jobber mot et selvinstallerende konsept av NAV omgivelsene.

4.6.2 Filstruktur og webstruktur

NAVMe har innført CVS og det er en vesentlig mer ryddig måte for en prosjektgruppe å parallelt utvikle kode. De enkelte utviklerne sjekker ut CVS-treet til sitt lokale utviklingssted og jobber der. Når den nye koden er stabil så sjekkes en oppdatering inn i CVS-treet. CVS håndterer parallell utvikling av samme kode, skjønt det er ikke tema i NAV, her har vi stort sett adskilte programmer som det jobbes på (men felles databaseoppsett, for eksempel).

CVS gjør i utgangspunktet en individuell versjonsinkrementering på filnivå. Vi kan imidlertid løfte hele treet til samme versjon ved milepæler. Ved termineringen av NAVMe prosjektet løftes hele treet fra 1.x versjoner til 2.0, således foreligger NAV v2. Senere vil vi synkronisere på høyere tall. Det er nå svært enkelt å laste ned/oppgradere til en gitt versjon av NAV.

NAVMe har også ryddet i filstrukturen og laget et klart skille mellom NAV-filer og lokale filer. I essens snakker vi om to filtrær som kan oppdateres uavhengig av hverandre. Det ville vært uakseptabelt om en ny NAV-versjon forkludret lokale tilpasninger.

En slik konsekvent inndeling i en NAV-del og en lokal del har gitt en del merarbeid. For web hovedsiden syr vi nå sammen ulike HTML-filer fra ulike grener av filsystemet for å vise hovedsiden. Dette er dokumentert i mer detalj i admindokumentasjonen på web.

Vi ser også at en del konfigurasjonsfiler bør ha en NAV-del og en lokal del. Dette har vi ikke implementert enda, men vi ønsker å utvide til det i NAV v3. Vi tenker da spesielt på hendelsessystemet og rapportgeneratoren.

Et annet gjennomført aspekt er inndelingen i sikkerhetsnivåer. NAVMe har videreført dette konseptet fra NAV v1. På web-sidene opererer vi med åpen del (http) og begrenset del (https,systemfolk) og interndel (https,nettgruppen). Denne strukturen er reflektert i filsystemet og tilhørende passordfiler (.htpasswd) er lagt på relevante steder i strukturen. NAVMe har utvidet dette konseptet til å gjelde på lokal gren også.

Mekanismene for å legge inn brukere, herunder med mulighet for både manuell innlegging eller automatisk innlegging fra en ekstern brukerbaser, er i hovedsak videreført fra NTNU NAV (men noe ombygd). Dette er omtalt i mer detalj i admindokumentasjonen på web.

4.6.3 Dokumentasjon

De to siste deloppgavene i NAVMe har vært å skrive dokumentasjon og sluttrapport. Dokumentasjon var tildels fraværende i NAV v1. Vi har nå en tredelt inndeling, med brukerdokumentasjon, admindokumentasjon og systemdokumentasjon.

Brukerdokumentasjonen er å regne som ferdig. Admindokumentasjon er også i all hovedsak ferdig, men vil nok bli styrket ved behov.

Systemdokumentasjon blir jo aldri ferdig, det blir i så tilfelle en definisjonssak av detaljeringsnivå. Vi har stort sett bra med kommentarer i kode, men vi ønsker også overordnet beskrivelse av database, filstruktur, individuelle script, deres algoritme og eksterne grensesnitt. Vi har gjort en del, men må jobbe mer med dette.

All dokumentasjon sees under metanav.ntnu.no/v2/

Sluttrapporten kommenteres ikke her i fare for en uendelig rekursiv løkke ;)

5 UiTø piloten

Som nevnt i 2.2 har primær målsetning i prosjektet vært å videreutvikle NAV til et generelt system som også fungerer utenfor NTNU. Piloten ved UiTø blir da selve beviset på om vi har nådd vårt mål.

Piloten er allerede omtalt i de foregående kapitlene. Her gjentar vi hovedtrekkene og oppsummerer.

5.1 Samarbeidet

Sett fra prosjektgruppen har samarbeidet med UiTø vært upåklagelig. Primær kontaktperson (og primus motor) ved UiTø har vært Børge Brunes. Børge har vært akkurat passe NAV-religiøs(!), og det har vært en klar fordel. En pilot må være innstilt på at alt ikke virker med en gang, at man må gå noen omveier, gjøre noen forandringer, tåle en del feilmeldinger og en del ventetid (på feilfix) etc.

Det har vært en åpenbar fordel å ha en såpass stor og selvgående pilot. UiTø har tilført oss nye tanker og ideer og de har rimelig umiddelbart forstått vår (forunderlige) NAVske verden. UiTø har kommet med mange konstruktive innspill til forbedringer av NAV v2. Noe er implementert allerede, noe står på listen over videre arbeid.

Vi har også inntrykk av UiTø er fornøyd. For å kunne begrunne dette mer formelt sendte vi et spørreskjema med noen nøytrale spørsmål. Svarene er vedlagt i vedlegg B. Døm selv.

5.2 Fremdriften

Vi hadde et oppstartsmøte i Tromsø 20. juni. Her presenterte prosjektgruppen NAV i detalj og UiTø ga en detaljert gjennomgang av sitt nettverk (både logisk og fysisk).

Neste oppgave i Tromsø ble å få NAV-maskinen (po.uit.no) på lufta. Det gikk smertefritt. Den var oppe og kjørte i slutten av juni med Mandrake Linux. Hardwaremessig er dette en intel 1 GHz Pentium 3 med 512MB minne og 2x18GB SCSI disk med RAID1-kontroller.

Pga ferieavvikling i Tromsø laget ikke UiTø lokale kildefiler før i august. Det var imidlertid ikke hemmende for fremdriften. Vi fant det naturlig å først teste NAV v2 med NTNU data. Så vi startet faktisk med å overvåke NTNU-nettverket fra Tromsø. I august/september falt alle UiTø kildefilene gradvis på plass, men vi utsatte stadig å gå over til UiTø data. Hovedårsaken var forsinkelser med den svært så omfattende ombyggingen av kunnskapsdatabasen.

I september fant vi det fornuftig å etablere en midlertidig NAV v2 maskin ved NTNU. Her installerte vi databasen og jobbet med fylling av denne. Vi utvidet etterhvert til RRD-data og nettkart. Vi kunne da parallelt kjøre begge steder. I midten av oktober var vi rimelig fullverdig oppe med UiTø data. Da begynte for alvor feilretting rundt databasefylling. En del misforståelser rundt konvensjoner og fylling av kildefiler måtte ryddes opp i. Vi måtte også gjøre tilpasninger pga nye problemstillinger (de fleste var dog avdekket på forhånd). Fullt operative ble vi ikke før i begynnelsen av november, så piloten er relativt fersk.

5.3 Spesielle problemstillinger

Av spesielle problemstillinger som dukket opp ved UiTø nevnes her:

- Subinterfacer, herunder problemer med manglende octettellere, problemer med å hente ifAlias (description) m.m: UiTø viste seg å gi oss et worst-case-scenario med sine to sentrale 8540-rutere (jfr 4.2.3). Det var ingen ulempe for NAVMe, tvert i mot. Når det er sagt, så er vi glade for at 8540 ikke har noen stor utbredelse i UNINETT.
- Nye utstyrstyper: Vi har støtt på en del nye Cisco-varianter, herunder 2600 serien rutere, 7500 serien ruter og Catalyst 2948G L3-ruter. Videre Catalyst 2916m-XL og Catalyst 2912XL svitsjer. Ombygningen med egen NAV-deteksjon av type har her vist seg vellykket. De nye typene har heller ikke gitt oss store problemer, men noen tilpasninger har vi måttet gjøre.
- Konvensjonstilpasning: Dette har nok vært den største jobben for UiTø, men det har også vært en prosess de har ønsket. De er ved et driftsmessig veiskille, der alt driftsansvar av nettelektronikk sentraliseres. NAV er da en gylden anledning å rydde i struktur og konvensjoner. UiTø har i NAVMe laget en egen telematikkromplan (rom.txt) og de har strukturert informasjon rundt organisasjon (org.txt) og anvendelse (anv.txt). Videre har de satt description-felt på alle ruterporter i henhold til NAV-konvensjonen. De har også i noen utstrekning fulgt NTNU konvensjonene med navning av svitsjeporter. Dette er jo ikke lenger et NAV-krav, men UiTø har likefullt funnet det fornuftig med en gjennomført beskrivelse her.

Av NAV v2 kodemessige problemer som vi først merket i UiTø nevnes:

- Problemer med PostgreSQL. Vi har hatt en del problemer med stabilitet. Databasen har hengt seg opp og vi har måttet droppet tabeller og bygd opp på nytt mer enn hva logisk var. Vi har hatt tidvis problemer med respons fra databasen. Hovedproblemene er ryddet av veien, men vi forventer å måtte ta flere ting etterhvert som piloten modnes ytterligere.
- Segmenteringsfeil for apacheserveren har ført til at denne har hengt seg opp av og til. Dette problemet er ikke helt løst enda.
- Database design feil. Vi har gjort oss noen erfaringer rundt indeksering. Bruker man database på feil måte kan søk ta veldig lang tid. Det har vært morsomt å se de kolossale forbedringene etterhver som vi implementerte ting mer korrekt.
- Barnesykdommer i varslingsystemet: Dette er jo grunnleggende ombygd og her er vi fortsatt i en tidlig pilotfase. NAV v1 sitt varslingsystem har gradvis “satt seg” over år. Vi forventer en langt raskere prosess nå, men en del må justeres, det er klart. Vi ønsker å ta i bruk NAV v2 ved NTNU også, og det vil skyte ytterlige fart i respons fra brukere og dertil feilfiksing.
- Problemer med SMS-demonen. Vi har støtt på noen problemer med at SMS-demonen hang seg opp. Dette er løst og fungerer bra nå.

5.4 Resultatet

Det endelige resultatet sees på <http://po.uit.no>. Vi overlater til leseren å dømme selv.

6 Oppsummering

Prosjekt NAVMe har satt et kritisk søkelys på NTNU NAV. Målsetningen har vært å generalisere, å løsrive NTNU-spesifikke ting og generelt lage et mer robust og brukervennlig system. På veien har vi i essens demontert hele NAV v1, ristet det hele lenge og vel, for dernest å sile ut de gode bitene. Vi har *ikke* rørt ved grunnprinsippene (se 3.2) eller de grunnleggende forutsetningene (se 3.3). Men kunnskapsdatabasen er radikalt forandret, og de fleste av NTNUs mange konvensjoner er borte, vi sitter igjen med krav til ruterport description-felt (men også dette foreslår vi modereres i NAV v3).

Oppsummert fremstår NAV v2 slik:

- Løsrevet fra NTNU-konvensjoner, herunder:
 - Uavhengig av portnavnsetting
 - Uavhengig av navnsetting av utstyr (nettelektronikk)
- Styrket databasestruktur. Kunnskapsdatabasen er nå en fullverdig modell av nettverket.
- Håndterer nye scenarioer:
 - Subinterfacar på rutere
 - HSRP
 - Nye utstyrstyper
 - Mer vilkårlig nettverksdesign, ikke avgrenset til typiske NTNU-konfigurasjoner.
- Ny topologiavleder finner:
 - Fysisk struktur på lokalnett
 - Vlan på trunker
 - Spanning tree tilstand pr vlan
- Hendelsessystemet er langt mer generelt. NAV-brukeren kan svært detaljert justere sin varslingsprofil.
- Filstrukturen er ryddigere. Det er lagt til rette for lokale tillegg på NAV sin web-side(r).

7 Videre arbeid

NAV begynner å få en anseelig størrelse og et slikt produkt vil alltid ha behov for kontinuerlig videreutvikling. Vi ser det som overordnet viktig å være nøktern i forhold til utvidet funksjonalitet. Vi anbefaler først å styrke den eksisterende funksjonaliteten ytterligere.

Vi har diskutert hvorvidt vi skal støtte et sentralisert UNINETT NAV som kan overvåke en rekke høyskoler. En slik modell har flere besnærende fordeler, men også mange ulemper. Det medfører blant annet en masse kodemessige utvidelser spesiallaget for et slikt ekstra overbygningsnivå. I den fasen vi nå er i blir dette feil bruk av ressurser. Det er andre ting vi bør fokusere på først.

Her følger prosjektgruppens liste over forslag til videre arbeid. Vi har strukturert den innenfor de ulike delsystemene, men den er ikke prioritert. Et nytt prosjekt bør ta tak i dette og forme en fornuftig fremdriftsplan:

7.1 Distribusjon

Det må jobbes mer med hvordan vi skal distribuere NAV, hvordan det bør driftes, hvordan oppgradering skal skje. Dersom løsningen skal kjøre under andre operativsystem enn Mandrake Linux ligger det arbeid også her. Men her ville vi anbefalt en konservativ modell.

7.2 Kunnskapsdatabasen

- Få bedre system på feilrapportering fra cronjobbene. Gode, *forklarende* rapporter.
- Styrke robusthet rundt endringshåndtering (gjelder også topologiavdekker).
- Generell logging av databasenendringer. Det er blant annet interessant med en generell historiefunksjon slik at man kan finne ut når en ruter ble faset ut, når en ny svitsjemodul ble operativ osv.
- Vurdere autotolkning av nettype slik at topologiavledning på lag3 kan automatiseres.
- Tolkning UNINETT description konvensjon, primært med hensyn på organisasjon og anvendelse (gitt autotolkning av nettype).
- Styrket elink håndtering, evt ved hjelp kildefil. Dette problemet vil melde seg ved første høyskole. Vi vil trenge en mekanisme som forteller oss hva som er grensenettene mot omverden. Description felt kan ikke gjøre dette i UNINETT-tilfellet, da disse ruterne med UNINETT sine øyne ikke er grenserutere.
- UTM koordinat utvidelse.
- Utvidelser rundt inventar, spesielt mer ruterdata, men også submoduler på catalyster.
- Databasetekniske endringer: endre nøkkel fra å være romnummer til å bli en teller, tilsvarende for org og anv. Støtte inet datatype for mer elegant sortering av IP-adresser.

- Utvide til svitsjeportinnsamling for alle KANT-porter (en radikal utvidelse).
- Parallellisere nattlige innhentingsscript.

7.3 Rapportgeneratoren

- Mulighet for å lage rapporter som en funksjon av sikkerhetsnivået til NAV-brukeren som er innlogget.
- Ytterligere fleksibilitet ved URLing. En anvendelse vi da vil legge inn er lenking over til Cricket-statistikk. Vi ser for oss at rapporter overtar som frontend mot Cricket og således erstatter/supplerer standardmenyene som følger med Cricket.
- Løse problemet med store rapporter (LIMIT-problemet). Flere tekniske detaljer, en del estetiske ting.

7.4 Trafikkinnsamling (RRD/Cricket)

- Parallellitet i innsamlingen, slik at den skalerer bedre.
- Innsamling av data også for KANT-enheter. Dette er en radikal utvidelse. På NTNU vil totalt antall porter gå fra 3000 porter i dag til 25000 porter. Ikke alle disse er oppe, men vi snakker om en mangedoblet vekst i innsamling.
- Legge bedre til rette for lokale utvidelser av Cricket-treet.
- Mulighet for å sortere RRD-dataene etter gitte kriterier (f.eks. maks last siste døgn) og vise en tabulær oversikt (finnes i NAV v1).

7.5 Topologi og last-visualisering (vlanPlot)

- Mulighet for inndeling i sikkerhetsnivåer, der intern bruker ser mer popup informasjon.
- Visualisering av last for en svitsj/kantenhet ved et grafisk vindu som viser last pr port.
- Mulighet for å se hele utbredelsen til et vlan i et vindu. Dette er realisert for NTNU NAV med en "windows explorer" lignende visning.
- Mulighet for visning av kun et område (basert på område-tabellen).

7.6 Statusmonitoren (Live)

- Innføre parallellpinging slik at tidsproblemet ved mye nedetid løses.
- Styrke rapportering rundt enheter som ikke er nede men som har variabel/lang responstid/pakketap.

7.7 Terskelmonitoren

- Mer fleksibelt system for å kunne justere grenseverdier som skal trigge varsling (i dag er dette flatt på 80%). Fleksibilitet både i forhold grenseverdier pr variabel, men også mulighet for å justere dette på enhetsnivå.

7.8 Hendelsessystemet og statussiden

- Styrket statussiden til å rapportere mer generell status og historikk for *alle* hendelser, ikke bare enheter som er nede/i skygge. Herunder mulighet for å se profil som en funksjon av organisatorisk enhet.
- Generell meldingstjeneste tilknyttet statussiden der NAV-brukere kan gi informasjon om planlagt/oppstår driftshendelse (finnes i NAV v1). Mulighet for å knytte meldingen til en enhet.

7.9 Brukerprofil

- Mulighet via web å få oversikt over SMS-meldinger som skal være sendt deg (finnes i NAV v1).
- Tilsvarende oversikt for eposter som er sendt brukeren.

7.10 Maskinsporing

- Generell brotabellinnsamling på alle svitsjeporter (SW og KANT) slik at vi vet oppetiden og plasseringen til en vilkårlig macadresse.

7.11 Utvidelser, ny funksjonalitet

Følgende utvidelser kan komme:

- Integrering mot UNINETT sin MPING-pakke slik at vi har gode målinger på rundreise og pakketap.
- Integrert tjenesteovervåkning (på vei for NTNU)
- Sysloganalysator (finnes i NAV v1)
- Grafisk visualisering av adresseromsforbruk (finnes i NAV v1)
- kontorsporing (i pilot på NTNU)
- RMON?

8 Konklusjon

Prosjektets hovedoppgave har vært:

“Med utgangspunkt i NTNUs egenutviklede nettadministrasjonssystem, NAV v1, skal NAVMe lage en generell løsning, NAV v2, som lar seg distribuere til andre universiteter og høyskoler.”

For å nå dette målet er mye tid og krefter brukt på å styrke kunnskapsdatabasen i bunn. NAV v2 kan nå modellere de fleste nettverksscenarioer og det er overkommelig å støtte nye utstyrstyper. Det rigide konvensjonenssett er kraftig redusert. Det stilles ikke lenger så mange krav til lokal navnsettingspraksis. Totaltsett er systemet langt mer feiltolerant og robust.

Funksjonelt lovet NAVMe å videreføre de fleste av NAV v1 sine delsystemer (men ikke alle). Det har vi også klart. Alt er oppe og kjører ved piloten på UiTø, men systemet er ferskt og bruken har enda ikke tatt helt av. Så feil vil nok fortsatt forekomme, de skal vi rette fortløpende.

“Barnesykdommer” til tross, vi ser konturene av et vesentlig bedre produkt. Ved UiTø kan vi vise til et NAV som står på egne bein, langt borte fra sitt barndoms nett⁶. Her har vi oppdaget 275 ruterporter⁷ og 3300 svitsjeporter. Uten portnavnsetting har vi funnet struktur og satt det hele sammen i kunnskapsdatabasen. Nettkartet illustrer resultatet, rapportgeneratoren likeså. Vi lager laststatistikk for alle aktive porter og vi statusmonitorer pr. dato 93 enheter. Flere enheter vil strømme på, NAV-brukere også. Alle kan få epost og SMS basert på sin personlige varslingsprofil.

Det er mange ting som fortsatt kan og bør styrkes. Det er også mange mulige utvidelser. Vi foreslår et nytt UNINETT prosjekt som løper i hele 2002. Endelig målsetning bør her være å få alle universiteter og høyskoler som ønsker NAV oppe og kjøre. For:

“The NAV must go on!”

⁶Trondhjem, Trondhjem, at æ reist i fra dæ...

⁷Inklusive subinterfacer.

A Ordliste

A.1 NAV ord

Vi omtaler i rapporten NAV i mange forskjellige sammenhenger. Her er en klargjøring av brukte NAV ord:

Cricket Open source løsning for grafisk visualisering av RRD-data.

NAV NAV står for Network Administration Visualized. NAV er et NMS.

NAV v1 Første versjon av NAV, også kalt NTNU NAV. Ble gradvis utviklet fra våren 1999 til våren 2001,

NTNU NAV = NAV v1

NAV v2 Resultatet av NAVMe prosjektet. Et mer generelt NAV som lar seg installere ved andre universiteter og høyskoler. NAV v2 er i pilot ved UiTø og NTNU pr november 2001.

NAV v3 Neste versjon av NAV. Slik vi ser det, resultatet av et nytt UNINETT prosjekt som løpet i hele 2002.

NAVMe NAVMe er kun et prosjektnavn, *ikke* et produkt. NAVMe leses på engelsk som 'NAV me'. Løselig oversatt blir dette: 'installer NAV hos meg!'

Live Delsystem i NAV, også kalt statusmonitor. Har til hensikt å sjekke om det er liv i alle overvåkede komponenter. Statussjekking gjøres ved å pinge.

RRD Round Robin Database. Open source løsning for innsamling og dertil lagring av statistiske data, som f.eks. lasttellere.

Statusmonitor se Live.

Terskelmonitor Sjekker terskelverdier og trigger når terskel overskrides. NAV v2 sin terskelmonitor trigger på verdier over 80% (gjennomsnitt siste 15 minutters periode).

Trapdetect Delsystem i NAV, også kalt hendelsessystemet. Tar i mot hendelser som SNMP trap og oppdaterer en statustabell. Varsler NAV-brukere på epost/sms.

vlanPlot Delsystem i NAV, også kalt nettkartet. Viser topologi og last grafisk ved hjelp av java.

A.2 Andre forkortelser

Andre forkortelser som brukes i rapporten er:

ARP Address Resolution Protocol. Grunnleggende protokoll i TCP/IP som resolver en ip-adresse til en macadresse. Alle maskiner cacher denne informasjonen for å slippe å spørre hver gang. Rute-ere har typisk en stor ARP-cache som gir IP/MAC informasjon for alle maskiner den snakker med.

DRS UNINETT Medlemsregister.

CDP Cisco Discovery Protocol. Cisco proprietær protokoll som brukes av Cisco rute-ere/svitsjer for å annonsere informasjon om seg selv til sine Cisco naboer.

- CVS** Concurrent Version System. Et system for revisjonskontroll av en trestruktur med filer (for oss NAV programkoden).
- HSRP** Hot Standby Routing Protocol. Cisco-proprietær løsning som muliggjør et redundant opplegg for default gateway. En virtuell IP- og macadresse innføres. Denne kan flyttes transparent mellom 2 (eller flere) rutere tilknyttet et lokalnett. Endemaskiner mister da ikke konektivitet selv om den primære ruterer går ned.
- MIB** Management Information Base. Et sett av variable som et NMS (f.eks. NAV) kan lese fra (evt skrive til) en SNMP agent (f.eks. en ruter eller en svitsj. Et eksempel på en MIB er den viktige standard MIBen MIB II som enhver SNMP agent skal støtte. Leverandører har typisk sine proprietære MIBer i tillegg.
- MPING** Mping står for Multi-ping og er et system som samler nettstatistikker for pakkeforsinkelser og pakketap. Utviklet hos Uninett.
- NMS** NMS står for Network Management System. Et NMS er et system som tar seg av oppgaver innen nettverks-feildetektering, datainnsamling og rapportering (svært grovt sagt).
- SNMP** Simple Network Management Protocol. En definert standard for uthenting av data fra utstyr i nettet. Finnes i v1, v2 og v3. NAV benytter v1 og benytter SNMP read, altså lesing av data. Vi benytter ikke SNMP write (konfigurering av utstyr). NAV sitt hendelsessystem baserer seg på SNMP trap, som er en "motsatt" mekanisme der utstyr eller andre delprogram sender varsel inn til systemet.
- STP** Spanning Tree Protocol. Brukes av svitsjer/broer og har til hensikt å bryte evt ringstrukturer i lokalnettene.
- UTM** Universal Transverse Mercator. Et globalt koordinatsystem som muliggjør en eksakt og utvetydig beskrivelse av en gitt posisjon på kloden.
- VTP** VLAN Trunk Protocol. Brukes for å spre vlaninformasjon fra VTP server til VTP klienter i et VTP domene. Et VTP domene er et administrativt avgrenset sett med svitsjer med trunker i mellom.

B Spørsmål til UiTø

Vi stilte 10 spørsmål til pilotkunden på slutten av prosjektperioden. Svarene er gjengitt i eksakt ordlyd, slik de ble formulert i epost fra Børge Brunnes mandag 26/11:

1) Hvorfor ønsket dere å delta i NAV-piloten?

Vi hadde lenge vært på jakt etter NMS som fungerte 100% i vårt nett. Intill nylig så hadde vi brukt HP NNM som ikke lenger hadde den funksjonaliteten vi trengte. Resultatet var at vi benyttet oss av hjemmesnekra systemer som fungerte rimelig greit isolert sett, men det var ingen sammenkobling mellom disse. Etter en prat med Vidar i vår hvor han fortalte om NAV fant jeg fort ut at dette var noe som kunne være nyttig for oss. Vi ønsket å komme oss i gang så fort som mulig med NAV for bla. å sikre oss at våre behov ble dekket. I tillegg så begynte nettet vårt å bli av en slik størrelse at vi måtte finne på noe lurt, kjapt. Forespørselen om være pilotinst. for NAV kom derfor meget beleilig.

2) Etterhvert som dere har blitt kjent med NAV, hva har overrasket mest i:

a) positiv retning

- Oppbyggingen av NAV som er modulbasert samt muligheten for å innkludere nye moduler og systemer.
- Samspillet mellom modulene
- Visualiseringen

b) negativ retning

- systemet er kanskje sentrert for mye rundt romnr.

3) Slik NAV v2 nå fremstår:

a) Hvilke delsystem vil ha størst verdi for UiTø?

Jeg tror ikke jeg vil fremheve et enkelt delsystem da styrken til NAV, slik jeg ser det, er helheten. Mange av delsystemene finnes i dag i form av andre programmer og systemer, men disse mangler helheten. Dersom jeg må velge så er status og varslingsdelen gode kandidater.

b) Hva står høyest på ønskelisten over utvidelser?

- sporbarhet -> Bak hvilken switchport befinner følgende macadr. seg.
- lag 2 traceroute.
- Bedre visualisering av lag 2 på nettkartet. Mulighet for å se alle bokser på et sted/bygg.
- Mulighet for å velge "script-friendly" rapporter (hvilke spørringer ligger bak rapportene i rapportgeneratoren?).
- Automatisk topologiavledning på lag 3
- Visualisering av spanning-tree
- Komme-i-gang oppskrift på integrering av av andre systemer som f.eks. snort og netflow
- Tjenesteovervåking

c) Har systemet totalt sett innfridd forventningene?

Ubetinget ja!

4) Hvordan har prosjektsamarbeidet vært?**a) positive aspekter**

Meget bra!

b) negative aspekter

- *Noe mangelfull info om endringer og veivalg som er gjort underveis*
- *For få sosiale treff :-)*

5) Vil dere anbefale NAV til andre universiteter/høyskoler?

Ja

6) Har dere andre kommentarer?

Vi håper selvsagt at vi får være med videre i utviklingen av NAV. Jeg vil tro at UiTø har ting å bidra med også i tiden som kommer. I den forbindelse så trenger vi en avklaring rundt drift av po ++

Ellers så er det kanskje en ide å opprette en epostliste for NAV-users når det blir flere som tar NAV i bruk?

Regner med at vi får avklart en del på møtet i januar.