Framework conditions and requirements for network monitoring in campus networks

Best Practice Document

Produced by UNINETT led working group
on network monitoring
(UFS128)

Authors: Vidar Faltinsen, Gro-Anita Vindheim
October 2011

Document No: GN3-NA3-T4-UFS128
Version / date: October 2011
Original language: Norwegian
Original title: “Rammebetingelser og krav til nettverksovervåking i campusnett”
Original version / date: 2011-10-03
Contact: campus@uninett.no

UNINETT bears responsibility for the content of this document. The work has been carried out by an UNINETT led working group on network monitoring as part of a joint-venture project within the HE sector in Norway.

The research leading to the results of this report has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 23 8875, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3)'.

Table of Contents

Executive Summary

This recommendation defines the requirements and framework conditions for network monitoring in campus networks. The recommendation has been written on the basis of many years of collective experience in the HE sector with the operation and monitoring of campus networks.

The document applies to the fields of fault management, accounting management and performance management. The functions which must or should be covered by the network monitoring system are specified for each field.

As is evident, a large number of tasks must be handled. When choosing between a single, complete system which encompasses all requirements and a set of smaller tools which are combined to produce a total solution, the latter is recommended. It is important to place emphasis on good integration. One should prioritise a central, flexible alarm system and a general, overall portal providing joint handling of authentication and authorisation.

The recommendation emphasises the need for robustness in the monitoring system. The location of the monitoring system must be carefully considered, the system itself should be monitored, its level of redundancy should be evaluated and effective routines should be in place for recovery in the event of failure of the system. It is recommended that the monitoring system is not virtualised for reasons of robustness.

Security must be a high priority. In this context, the use of SNMPv3 is recommended. If this is not supported by the selected tools, it is recommended that measures be taken to ensure satisfactory security.

1 Introduction

This recommendation defines the requirements and framework conditions for network monitoring in campus networks. The recommendation has been written on the basis of many years of collective experience in the HE sector with the operation and monitoring of campus networks. By way of the GigaCampus programme (2006-2009) [1], UNINETT has co-ordinated the technical field using a specialised monitoring work group. UNINETT has also deployed monitoring servers, so-called toolboxes [2] and beacons [3], in the HE institutes' campus networks. The toolboxes and beacons contain a collection of open-source applications developed by UNINETT and the HE sector [4,5,6] or other parties.

1.1 The need for network monitoring

An ICT department's principal job is to support the objectives of the insitutions's activities. In the case of a university or college, this means supporting the defined objectives for teaching, research and information dissemination. The ICT department provides a set of services (and an underlying infrastructure) to employees and students. The requirements placed on these services will vary, but a general trend is that there is increasing reliance on the underlying network functioning day and night, all year-round. Not only must the network function, but it must also have adequate capacity and acceptable, reliable response time. It must be capable of handling a range of different services with different characteristics, from real-time applications such as IP telephony and video conferencing to extremely capacity-demanding data transmission as used in high-performance computing and other research functions.

Such a network in itself is complex, consisting of a large amount of equipment and cabling assembled into in a system. Taking one of the major campus networks in the sector as an example, the network at NTNU consists of about 25 routers, 1100 switches and 1600 access points (with 20 controllers). Gigabyte traffic rates are in use round the clock, seven days a week, all year-round.

It is important to build fault tolerance into the network, at least in the most important parts of it. Ideally one should avoid the possibility of failure of a single component paralysing the whole network or parts of it. The degree of redundancy in the network will be subject to cost-benefit analysis. We refer here to UFS 114: Fault-tolerant Campus Networks [7]. However, irrespective of how much fault tolerance is built in, problems *will* arise. Components will fail and will need to be replaced. This calls for efficient monitoring which provides precise alarms in fault situations. Monitoring tools also play an important proactive role by providing operational personnel with indications of problems at an early stage and enabling them to solve them before they become critical.

1.2 Limitations

Network monitoring is closely related to and clearly overlaps with system and service monitoring. System and service monitoring play a corresponding role, but are directed towards servers and services. A third area is

client monitoring, the focus of which is the operation and maintenance of client servers in a network. In this document we will not consider system, service or client monitoring in detail.

Neither does the document deal with the underlying operational processes, i.e. the activities, methods and procedures necessary for efficient, proactive ICT operations. This is clearly described in the ITIL Best Practice framework [8]. It should be remembered that network monitoring has no intrinsic value. It is exclusively a tool to assist ICT operational personnel. The purpose of such tools is to support the ICT department's activities, methods and procedures to ensure efficient operation, maintenance and development of the ICT infrastructure. Although the tools *are* important, remember that organisation, personnel resources, work methods and routines are substantially more important.

Let us explain precisely what network monitoring actually is. Network monitoring represents a central element of Network Management, which is defined as activities, methods, procedures and tools which support the operation and maintenance of a network. A common way of characterising network management is FCAPS[1] - Fault, Configuration, Accounting, Performance and Security [9].

Network admin fields	Explanation
Fault management	Monitors the network and underlying components. Detects faults and transmits alarms.
Configuration management	Maintains an overview of all components in the network. Supports the configuration of network equipment. Maintains an archive of change logs.
Accounting management	Maintains an overview of traffic load and traffic types. Also evaluates the state of health of network equipment (CPU load, memory use, environmental data, etc.).
Performance management	Evaluates the performance of the network, including delay, packet loss, throughput and jitter (variation in delay).
Security management	Monitors access to the network and its components.

The Configuration Management and Security Management fields are considered to be outside the scope of network monitoring and will not be considered in this document. The main focus of network monitoring is "fault management", in other words the monitoring of the network and generation of alarm signals in the event of failure. According to our definition, accounting management and performance management are also included in the term "network monitoring". Chapter 2 provides a detailed description of these three functional areas.

1.3 Underlying technologies

The most commonly used method for obtaining data from network equipment is based on the IETF standard, SNMP (Simple Network Management Protocol), described in detail in Attachment A.

In 2006, IETF produced a new standard, NETCONF, which is intended to be the successor to SNMP, although it is not yet clear if this will happen in practice. NETCONF is described in Attachment B.

[1] FCAPS is a network management framework defined by ITU-T which was first introduced through ISO 10040 early in the 1980s.

2 Functional areas and requirements

We will here provide a detailed description of functional areas and requirements for network monitoring, using the classification given in Section 1.2, and dealing with the fields of fault, accounting and performance management.

2.1 Fault management

Fault management consists of monitoring the components in the network, detecting faults and sending alarms. These are the most fundamental and important network monitoring tasks. If a component fails, you want an alarm, and you want it immediately.

2.1.1 Monitoring

A network monitoring system will consist of a range of underlying monitors. The purpose of a monitor is to check regularly that everything is in order and, if not, transmit an alarm. When the fault has been rectified, the monitor will transmit a clean “bill of health”. A monitor is often dedicated to a particular function:

- A *ping monitor* checks that all equipment (routers, switches, wireless devices, servers etc.) is functioning.
- An *interface monitor* checks that interfaces and communications are operating.
- A *module monitor* checks that all modules in a modular switch or router are functioning. This includes monitoring of power supplies and fan modules.
- A *threshold module* transmits an alarm if traffic load, CPU load, etc. exceeds a pre-defined limit.

A ping monitor uses an ICMP echo (ping), while interface, module and threshold monitors should be based on SNMP. All these monitors send alarm signals to the alarm system.

Service monitoring also includes the *service monitor* which checks that all services are functioning. A good service monitor simulates the communication protocol of the service and verifies that it receives a reasonable response (checking that the TCP/UDP port is open is not enough).

The following coarse filtering should be carried out on the monitors:

- Build robustness into the monitoring so that insignificant disturbances are not reported as repeated down and up alarms which create unnecessary noise. A better approach is to report such short-term disturbances as separate “flap alarms”.

- Keep an eye on the network's condition and do not report a given down condition to the alarm system more than *once*. Ensure also that the associated up message can easily be related to the down message, so that the alarm system can confirm rectification of the event.
- In connection with threshold alarms, use hysteresis with two defined thresholds, one for bad system health and one for good system health[2]. An example is that an alarm is wanted when CPU use reaches 90 % and this alarm is not cancelled until CPU use drops below 80 %. If the alarm were also cancelled at 90 %, this could potentially result in a series of alarms if the load oscillates around 90 %.

2.1.2 Alarm system

Most monitoring systems provide simple mechanisms for transmitting an alarm signal when an incident occurs, for example by sending an e-mail to pre-configured e-mail addresses. A satisfactory alarm system must provide significantly more functionality than this. In Chapter 5 we argue that you will need *several different* tools in your monitoring portfolio. However, we recommend *combining alarm and notification functions into a single tool*, in which the various tools send their alarm signals to this one system. Figure 1 shows the recommended architecture.

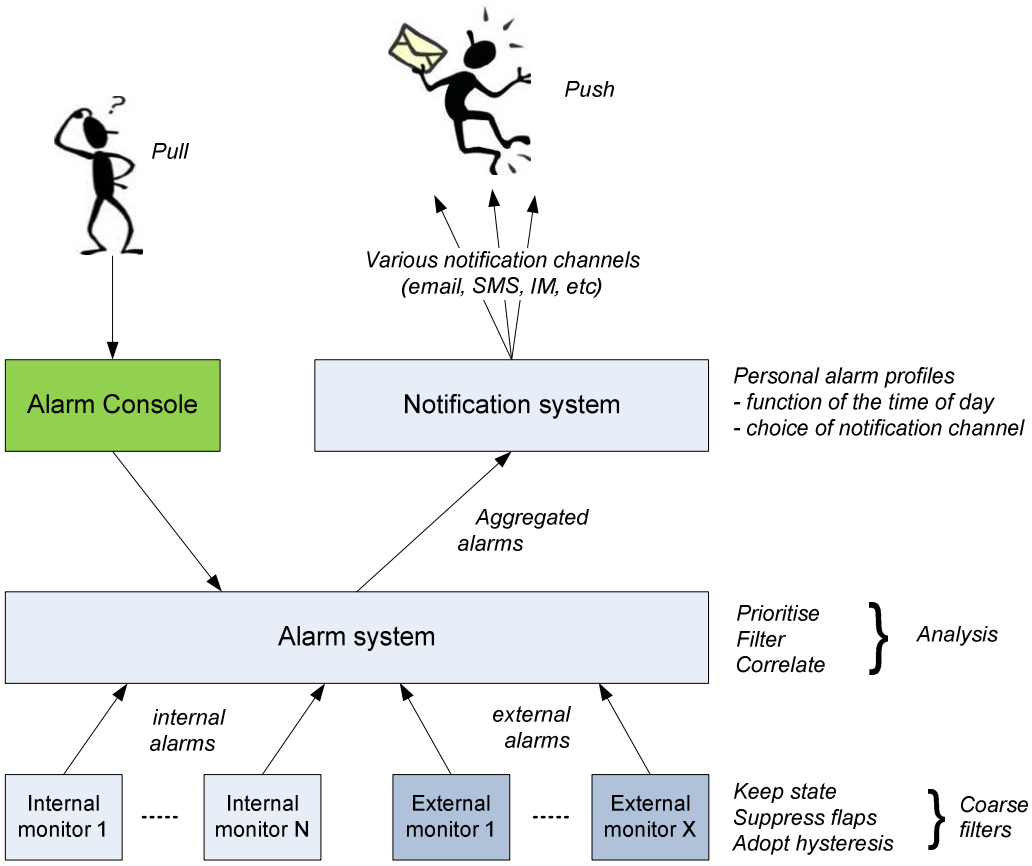


Figure 1: Monitors, alarm system and notification system

The tool which contains the alarm system usually also has its own monitors. *Internal* alarm reports can be achieved by different proprietary methods, via a database or similar arrangement, but in the case of *external* monitors a standard means of communication must be used. For reasons of flexibility it is recommended that the alarm receiver supports both SNMP traps and e-mail. The alarm system must moreover have flexible mechanisms for interpreting external alarms in order to be able to classify them. In many cases it is unrealistic to adapt the alarm format of the external system, and hence it is an advantage to be able to make system-specific interpretations in the central system.

[2] Applies to alarms for thresholds measured on the ascending flank.

The purpose of the alarm system is to handle incoming alarm signals and put the various alarms into context. This analysis phase is important to provide the best possible picture of the alarm situation to the network operator (by way of the notification system). The alarm analysis consists of the following tasks:

- Rank alarms according to degree of seriousness. In the event of a major failure a very large number of alarms may be received simultaneously and it will be important to be able to distinguish important alarms from less important ones.
- Correlate alarms transmitted by different monitors. If the alarms are related with the same incident, filter out superfluous alarms or combine them into one common alarm. This may be difficult to achieve in practice. The alarm system must *not* have a detrimental effect on system robustness. One alarm too many is preferable to one too few.
- Also correlate alarms in relation to network topology. This requires that the alarm system has knowledge of the topology of the network. In the event of a major failure, the alarm system can thus reveal the root cause. For example when an important router is down, the monitor will not be able to know whether underlying components are in fact down (unless there is a redundant path). A network operator does not want hundreds of down messages in this type of situation, but one clear message indicating which central component is down. A mechanism for distinguishing between a down message and an “unknown status” message is recommended (which assigns lower priority to unknown status).

2.1.3 Alarm console

Handled or aggregated alarms are forwarded to the notification system which in turn forwards them to the relevant system operators (see below). The operators are also provided with an alarm console with which to check status. An intuitive interface will indicate conditions using red, yellow and green lamps. A red lamp provides a clear message that something is wrong. If a very large number of components or incidents are being monitored, the alarm console must support the *grouping* and *hierarchical display* of alarms, using groups and subgroups. In this way the number of lamps at the highest level becomes manageable and a red or yellow alarm lamp here can be pursued to the level below, or to the level below that, in order to find the source.

2.1.4 Notification system

The purpose of the notification system is to transmit alarms to the operators. Typically, different operators will have different roles and/or areas of responsibility, so possibilities for configuration of individual alarm profiles are a reasonable requirement. An individual operator should be able to adjust his or her profile based on alarm priority, equipment type or category, or the area in which the alarm originates. One should also be able to flexibly select the notification channel for different categories of alarm. E-mail and SMS, at least, should be supported, and it would also be an advantage if instant messaging (IM), including, for example XMPP or IRC, was supported.

A network operator should also be able to set up different alarm profiles for different times of day or night. For example, it may be preferable not to receive alarms during the evening hours unless one is on standby duty (this applies particularly to SMS notification). Alternatively one might want to receive high priority alarms also in the evening and at weekends.

The notification system must be capable of queuing messages based on priority. This is essential to ensure that the most important alarms are received and are not lost in a queue of less important alarms.

2.2 Accounting management

Accounting management is understood to mean mechanisms for maintaining an overview of traffic load and traffic types, as well as the state of health of network components.

2.2.1 Collecting traffic counters

In order to keep track of the traffic volume in a network, one should regularly collect traffic counters from all router and switch ports (using SNMP). As a minimum requirement, the following data should be collected:

- Traffic in/out in terms of bytes (octets) and number of packets
- Error counts (faulty packets and dropped packets)
- Multicast/broadcast traffic volume

We recommend collecting data from *all* ports in the network. The vast majority will never be looked at, but when an incident occurs it is an advantage to be able to study various tables and graphs from different statistical data sets in order to establish the probable cause of the problem. Without such data, investigation will be difficult and the cause may never be found.

The data must be stored in an underlying database. Adequate data collection frequency is important to enable the identification of short periods of abnormal traffic patterns. The minimum recommended frequency is five minutes[3].

Data should be stored for several years to facilitate trend studies. It will be most practical to aggregate older data to ensure that the total storage requirement is manageable. When aggregating data, the maximum and minimum values must be kept. The system must be flexible enough to make it possible to define the rules (time intervals) for aggregation. As a minimum requirement, it must be easy to produce daily, weekly and monthly statistics.

Flexible mechanisms for searching the data and then presenting them in table or graphic form are essential. Good response time is also a basic requirement. The following components should be included:

- Topological network maps providing an overview of traffic flow in the network at a given time, with the present as the default time. The network map should be hierarchical, enabling one to concentrate on a particular part of the network.
- A flexible reporting system enabling the production of compiled reports and summaries. It must be possible to sort the reports by any column.
- Good graphical visualisation using various types of graph. It must be easy to plot different data sets in the same graph or in graphs presented one under another, with the same time axis. This will facilitate the examination of a given event by the operator.

Trend analysis (Capacity Management in ITIL), i.e. analysis of traffic growth with time, enabling prediction of future growth, may also be included in the network monitoring system.

2.2.2 Network equipment health checks

In addition to traffic counters, information should be collected from routers, switches and other network devices to provide a picture of their “state of health”. This includes:

- CPU load
- Memory use
- Disk use (NVRAM, flash disk)
- Power consumption, also at the PoE interface level
- Temperature sensors located in the equipment (preferably both at intakes and outlets)

These data are also important for determining whether the network is functioning optimally. The requirements for storage, aggregation and presentation of such data are the same as for traffic counters data.

[3] When using SNMP for data collection, a 5 minute interval will require 64-bit counters for traffic volume at gigabit rates.

2.2.3 Traffic types

These are data which provide an overview of what types of traffic exist in the network, including an overview of which IP addresses communicate with which (both IPv4 and IPv6), with what volume (number of packets and number of bytes), and in what time periods. It must be possible to study the data in detail down to TCP/UDP level. Because keeping data relating to all such “transactions” in the network is naturally resource-demanding, one will also here have to have routines in place for the aggregation and deletion of data. The storage time and access to such data must be carefully considered and adjusted according to current personal data protection legislation.

It must also be possible to combine the data to provide trend overviews with regard to:

- Top talkers (incoming and outgoing)
- Most used send and receive ports (TCP and UDP)
- Relationships between TCP, UDP, ICMP and other traffic
- Multicast traffic volume
- Traffic at autonomous system (AS) level.

A system for collecting such data may be passive measuring equipment which sees all traffic passing central points in the network. Alternatively, routers may export such data to the network monitoring system. The IETF standard IPFIX (RFC 3917) should then be used. Alternatively, Cisco’s proprietary NetFlow format may be used.

2.2.4 Machine tracking

Machine tracking is a special field included in accounting management. It entails collecting IP-MAC bindings from routers (ARP cache), both for IPv4 and for IPv6, and bridge table data from switches, in order to obtain an overview of when and where (at which switch port) a given machine is/was connected to the network.

When compiled, these data provide trends with regard to how many machines are in use in the different subnets, as well as the proportion of machines using IPv6.

These data are also very useful in the event of security incidents in which a complaint is made against a given IP address at a given time.

2.3 Performance management

The last area, performance management, evaluates the performance of the network, including delay, packet loss, output and jitter (variation in delay). It can be measured using probes located in the network which send test data to echo points in the network. Ping (ICMP ECHO) is often used for this purpose, but it is also possible to use UDP ECHO or other protocols. The following data must be saved:

- Round trip travel time
- Packet loss
- Jitter (variation in delay).

A disadvantage of measurements based on round trip travel time is that it is not possible to know if any delay occurred on the outward or return journey through the network. An improvement may be to use one-way measurements, but this entails accurate synchronisation of the sender and receiver (preferably using a GPS antenna). Another possible weakness is that one cannot know at which hop in the network the problem occurred. If measurements are made at every router in the network, it will be easier to determine where the delay occurred. A potential source of error is that some routers will give very low priority to such requests and hence give an unnaturally poor response time.

A system in which end-users can measure their performance relative to a measuring point in the network is also a useful tool. A so-called Internet speedometer performs such a function. It is recommended to use a system which is capable of providing details about network performance, including packet loss, maximum packet size, and allocated buffer space in end systems, etc.

Another important field is the measurement of quality. By studying the gap between packets and correlating it with time stamps in RTP headers, it is possible to assess the quality of the network with regard to the provision of audio and video streaming.

3 Robust monitoring

To make the monitoring as robust as possible, one must consider a number of important framework conditions, including the strategic location of the monitor, monitoring of the monitor, SMS configuration and redundancy in the monitoring itself.

3.1 Location of the monitor

The location of the monitoring service in a network must be carefully planned. A decentralised monitor has a higher probability of becoming isolated from the rest of the network and therefore unable to fulfil its function. The monitor should be located centrally in the network, close to the central servers. The monitor will then in most situations be capable of determining which services are operating and which users have lost communication with the network.

To improve reliability further, monitors should be located in a subnet with a redundant outgoing route (via a protocol such as VRRP). Moreover, the server should have redundant power supply, the different supplies receiving their current from separate sources, ideally separate UPS sources.

3.2 Monitoring the monitor

In spite of everything the monitor itself may die, or processes within the monitor may cease to function. Hence it is important to monitor the monitor. This is often overlooked, but should be given high priority. A monitor which ceases to function, unnoticed, may have unfortunate consequences if some undesirable incident should occur in the network. A simple way to monitor a monitor is to check that it responds to ping. A more advanced method is to check that all the necessary processes are operating, that the file system is not full, etc.

The monitor of the monitor should be located at a physically separate location and should be capable of sending SMS messages directly to operational personnel.

3.3 Use of SMS as a notification channel

As mentioned in Section 2.1, an alarm system should be capable of transmitting alarms using SMS as well as e-mail. SMS transmission can be achieved in a number of ways, for example via an SMS gateway on the Internet. However, this method is *not* recommended. The point of monitoring is that it shall be capable of providing a warning when cut off from the outside world. Hence, the most robust method is to have a mobile phone or similar GSM unit directly connected to a serial or USB port on the monitoring server. Be aware that the degree of coverage of the GSM network may be poor in a machine room which is located in basement premises or in the centre of a building mass. An additional external antenna may solve this problem.

3.4 Redundant monitoring

Monitoring servers, like all other servers, must occasionally be shut down for maintenance, and this must be taken into account. Is it acceptable to manage without monitoring during such a planned maintenance window? If not, is it possible to use the monitor of the monitor to provide at least rudimentary monitoring of the condition of the network?

One must be prepared for the possibility of a monitor failing, necessitating setting up a new server. Three alternative solutions to this problem are suggested:

1. *Cold standby:* Re-establish the monitor as quickly as possible.
   In this scenario one should monitor the monitor, have a standby server in storage and maintain a backup of the monitoring database and other data collected by the monitor, such as traffic statistics, server tracking data, logs, IPFIX and network flow data, etc. One should in addition have personnel on continuous standby and clear instructions for how such personnel shall re-establish the monitor if it should fail.

2. *Hot standby:* Replicate the monitor in a monitor which is kept on standby.
   This calls for continuous replication of monitoring data and monitoring of the primary monitor so that in the event of failure an alarm is sent to network operations who can then by means of simple manual operations put the secondary machine into operation. Such a manual operation would entail the secondary server taking over the IP address of the primary server and the monitoring process being manually started up on the secondary server. It may be possible to avoid changing IP address, but this will result in many potential problems, since a great deal is associated with the IP address or DNS name, including SNMP traps from network equipment, syslogs, e-mail alarms from external systems, SNMP filters in monitored equipment and the actual web interface of the monitor.

3. Use two active monitors which are mutually synchronised.
   This is a more complex configuration in which two servers monitor each other and are synchronised, and can seamlessly take over from each other. They should be located in two different places and should also monitor each other. Anycast[4] may potentially be used to avoid confusion with regard to IP addresses in external systems, but many factors must be taken into consideration if such mechanisms are to function. We will not go into this in detail here.

Our recommendation is that this should be done relatively simply. After all, it is more important to build in a good degree of redundancy in the central services than in the monitoring itself (monitoring has no intrinsic value). *Alternative 1 will be an adequate solution in most situations.*

3.5 Virtual monitoring server

If you choose to virtualise the monitoring servers, several new possibilities become available. One advantage is that it is easy to re-establish the monitoring service in the event of failure, although this does depend on having several blade systems, which themselves are redundant and have redundant, virtual architecture (based on VMWare, KVM, etc.). It also requires that one can continue using the same IP address for the replaced, virtual machine, cf. Section 3.4. By constructing a server environment which takes all this into account, one approaches a very good solution for service provision in general and network monitoring in particular.

However, it is important to evaluate the robustness of such a configuration. A free-standing server which does not rely on anything but itself is potentially a more robust monitor than one which is incorporated in a blade system and virtual systems. Remember that it is also important to maintain a reliable monitor if the most serious

[4] Anycast is a mechanism in which the same IP address occurs in two or more machines in different parts of a network. Routing announcements are given from all instances of the IP address. Other machines in the network will communicate with the machine that is closest to them. Anycast results in a natural load balancing and gives implicit redundancy. Stateless services such as DNS resolvers are very suitable for use as anycast services.

accident happens. Quickly identifying the fault source in an ongoing incident can significantly reduce the overall down-time for users.

4 Secure monitoring

The following is a description of SNMP and NETCONF. Further details can be found in Attachments A and B.

It is very important to maintain security with regard to your routers, switches and other network equipment. SNMPv3 is therefore recommended for network monitoring, as it ensures secure, encrypted communication between the monitor and SNMP agent. In future, NETCONF may be a good option, but at present there are too few implementations.

However, in reality, very many network monitoring systems on the market are based on SNMPv2c. This does not provide adequate security. Traffic is transmitted in plain text, including the SNMP password (community) in use. This may be sniffed by a “man-in-the-middle” who can potentially obtain access to the network equipment by way of SNMP. This has unfortunate effects on SNMP read access rights and catastrophic effects on SNMP write access rights. With the latter, one has the ability to restart a router or switch and to modify or delete an entire configuration.

In spite of these weaknesses we consider it advisable to use SNMPv2c for network monitoring. However, we do recommend implementing the following measures:

1. SNMP access to the network equipment should be limited as strictly as possible. No machines apart from the monitoring servers shall need to communicate with the network equipment by way of SNMP.
2. Management IP addresses of switches should be in a dedicated subnet with strictly controlled access. Wireless access points located in public areas should be located in yet another dedicated subnet (not mixed with switches).
3. Access to the monitoring system itself should be restricted to authorised personnel. This applies to network access via both SSH and HTTPS, and to physical access to the machine room. In the case of both SSH and HTTPS, user log-in does not provide adequate security. In addition the subnets which are permitted access to the server must be limited strictly. See also UFS122 [10] regarding recommended ICT architecture.
4. Be careful with autodiscovery of new hardware. This will typically imply probing all machines it finds in the network using the SNMP password (community). This makes it *very* easy for a third party to sniff this password. Autodetection in a restricted subnet without end-user access, such as a dedicated subnet for network equipment, is acceptable.

Irrespective of which SNMP version is used, the following security perimeter is also recommended:

5. The web interface of the monitoring system should have a built-in authorisation system which makes it possible to restrict access at group level to a selection of the underlying web tools.

5 Acquiring a network monitoring system

The review in Chapter 2 indicates that a network monitoring system must provide a very wide range of functions. Ideally, one should acquire a complete network monitoring system which supports all these functions. The alternative is a set of systems which in combination provide the required functions. We have a choice between two alternative system models as shown in Figure 2.

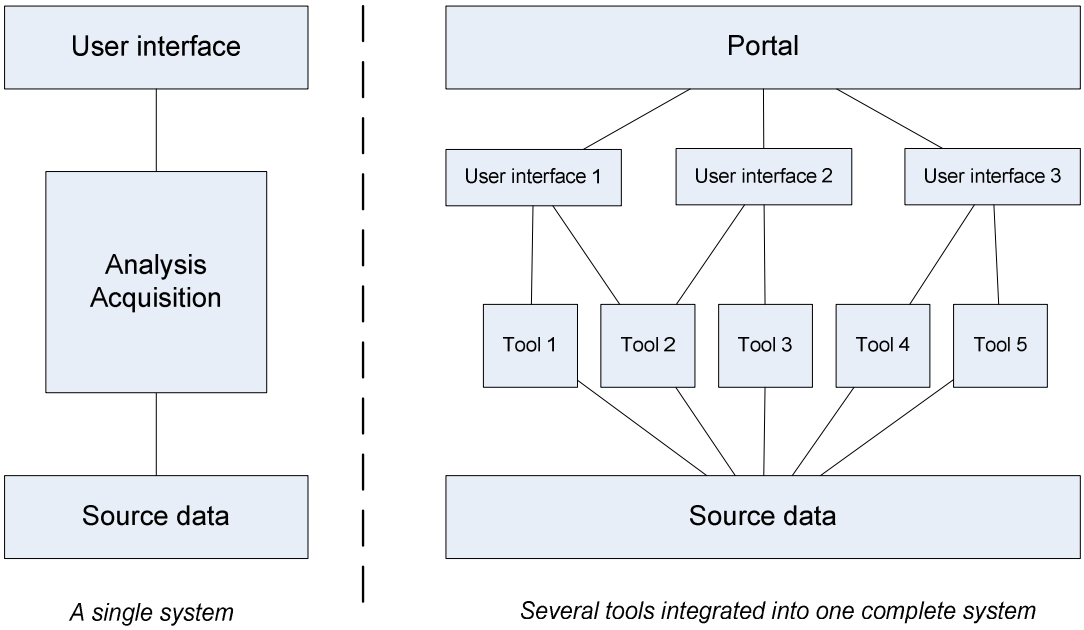


Figure 2: Two alternative system models for network monitoring applications

Let us look more closely at the advantages and disadvantages of the two alternatives before presenting our final summary and conclusions.

5.1 One complete system

An obvious advantage of acquiring a complete system is that instead of having to worry about the integration between applications, one can deal with a standard user interface. If the system is commercially produced, good support facilities will presumably also be provided, with reliable guarantees for future system development, regular software updates, and so on. The installation and updating of the system can also be very simple and user-friendly.

Before acquiring such a large system, one should consider such things as:

- Purchase price compared with the acquisition of a set of smaller systems (in which some or all may have open-source code and be free). Some large systems can be rather expensive.
- Implementation of a large system demands resources, including hiring external consultants, personnel training, and so on. The start-up costs will often be higher than the purchase price.
- A support agreement, as opposed to managing alone. These running costs can also be significant.
- Possibilities for carrying out local adaptations or expansion. This is an important aspect which should not be underestimated. Almost without exception it is necessary to adapt or adjust a system to local conditions. For many systems, such adaptation will cost more than it is worth. There may be several reasons for this, such as a lack of configurability of the software, difficulty in accessing the code, high complexity of the system itself, and so on. If modifications are made to the program code, this may in the long run lead to problems because one must maintain the programs manually and repeat one's modifications in connection with future system upgrades. Remember that working time involved in adaptation is part of the overall calculation of cost. In many cases, such local adaptations will also necessitate hiring consultants.

5.2 A set of smaller systems

If one decides to use a set of smaller applications, the initial investment and annual maintenance may be low or potentially free (open-source code). Moreover, provided they are modular and well documented, the result is a set of tools which can be adapted with a reasonable amount of effort. However, this is not free either, since one must allocate personnel to local adaptation under any circumstances. Nevertheless, replacing a partial system or a single application will be simpler.

A disadvantage of small applications may be that it is not possible to sign up for a support agreement and/or that support and software upgrades are unreliable. A small, open-source application also entails greater risk of failing than a tool from a major supplier. This risk must be considered seriously.

On the other hand, an active, open-source project with a large user community can function very well. The system will be well tested and code contributions can come from many sources. However, it is important that the project is backed by a reliable development team which can control and co-ordinate the development. However, if development should break down, one will, thanks to the open source, potentially be able to maintain and develop the system using one's own personnel.

If one decides to invest in a number of individual applications, one should always be cautious. It can be a disadvantage to have *too many* systems in operation. Systems put into operation by individuals, and not used by the entire operational staff should definitely be *avoided*. Neither should tools in the portfolio overlap, at least as regards the functions of the individual application which one chooses to use. For example, several systems may each have their own service monitor or alarm systems, but one should choose a single system for these functions.

One should endeavour to reduce complexity as far as possible. If the choice is between a distributed or a hierarchical set of monitors, each covering part of the network, as opposed to centralising this in a single machine, the latter is preferable. With the server power available today, this should not be a problem. More powerful CPUs, more memory and better disk I/O are parameters which can be adjusted relatively cheaply[5].

A clear disadvantage of having many systems in operation is that operational personnel have to relate to many different interfaces. If each system has its own system for authentication and authorisation, this will also increase requirements with regard to duplication of configurations for user names and passwords, and so on. One should prioritise efficient integration between the systems. A superstructure using *a common portal*, as shown in Figure 2, with common authentication and authorisation, is a good model. One of the applications selected should be adapted to this function.

[5] Note that for very large campus networks, it can prove to be difficult to achieve traffic collection of all switch ports from a single server. Hence, if a given task is to be distributed, a simple and tidy architecture should be chosen.

Another potential trap is that one duplicates or multiplies the effort required to maintain source data. Examples of such source data are information about the equipment to be monitored (with IP address, SNMP community, location, etc.), geographical information (name, location of machine or communications room, etc.) and organisational information (who is responsible for equipment). One should make an effort to achieve *a common authoritative source* (cf. Figure 2) for such data, so as to simplify maintenance and reduce the probability of inconsistency[6].

5.3 Summary and conclusion

Let us compare the arguments put forward in the sections above:

	One commercial NMS	Several open-source applications
Purchase price		Lowest
Support agreement		Lowest
Replacing system elements		Simplest in practice
Integration between applications	No problem	
Reliable management	Can have advantages	
Frequent updating		Can have advantages[7]

In practice, however, the choice is more complex. For example, the small tools do not need to be open-source applications and one may naturally choose a larger commercial system and attempt to integrate small applications with this. This is also a question of economics and budgetary limitations, as well as whether one is used to, and wishes to carry on, one's own development.

The latter is normally an option only for the larger communities, and this is also the way it has been in Norway. UNINETT and the major universities have benefited from communities where creative enthusiasts have had a free rein and have in time developed systems which have gradually improved day-to-day operations. For a smaller enterprise this is not a realistic option, and buying a complete system will therefore be an attractive option.

However, it is important here to learn from our own experience, meaning the collective history of the HE sector. There are examples of institutes which have tried out large, commercial network monitoring systems and been disappointed. Based on the product specifications the systems have seemed highly promising, but in an actual installation deficiencies and weaknesses have been apparent. Rectifying these, either with the help of the supplier or by one's own adaptation, has in several cases proved to be both time consuming and costly.

Of course, this is not necessarily the case. Products are being developed constantly and new ones are being added, but in general one should be sceptical of large systems which make ambitious promises and offer widespread functionality. Settling for several smaller applications with each individual one having a more specialised function, has been shown to be a more viable strategy[8].

[6] As an alternative or supplement to manually feeding the system with source data, new components in the network may be detected automatically. This may be an advantage, as it is a better way of ensuring that a router or switch is not overlooked by the monitoring system. On the other hand, it may happen that components which one does not wish to monitor are included automatically. A combination of automatic and manual approval is the best solution. However, be aware that autodetection based on SNMPv2c can be a security risk, cf. Chapter 4.

[7] Large NMSs can be cumbersome and may not be upgraded as frequently as active open-source applications.

[8] Precisely this was an important motivation behind UNINETT's toolbox product [2] which was introduced during the GigaCampus programme [1] in 2006. The fact that UNINETT, on behalf of the sector, assumed responsibility for adaptation and development, has made it possible also for smaller colleges with low risk to select this type of platform. Such a model provides clear economies of scale savings. This has also been established by an independent consultant's report [11].

A. SNMP

SNMP stands for Simple Network Management Protocol and was originally defined as SNMPv1 in the following RFCs:

- RFC 1155, May 1990: Structure and Identification of Management Information for TCP/IP-Based Internets.
- RFC 1157, May 1990: A Simple Network Management Protocol (SNMP).
- RFC 1213, March 1991: (Version 2 of) Management Information Base (MIB) for Network Management of TCP/IP-based Internets.

A long series of RFCs have subsequently been added, such as OSPF v2 MIB (August 91), BGP v3 MIB (October 91) and RMON MIB (December 91).

SNMP consists of three fundamental forms of communication (see Figure 3):

- `snmpget`: The monitoring system polls the agents and collects the requested information (required MIB variable(s)).
- `snmpset`: The monitoring system changes the MIB variable in the agent in order to change the agent's configuration.
- `snmptrap`: The agent itself sends a message to the monitoring system when a given event occurs.

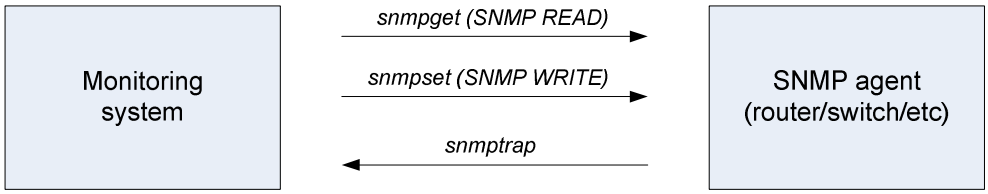


Figure 3: Means of SNMP communication

UDP is used as the protocol for SNMP, with the disadvantages this entails (UDP is connectionless and there is no guarantee that a packet will be delivered). It is also possible to implement SNMP using TCP, but this is in practice not very common.

The information which the agent (the network equipment) makes accessible via SNMP is organised in a tree structure described as a Management Information Base (MIB). The tree structure and the syntactical structure itself is a standard. Important branches of the tree structure, including MIB-II, are also standardised. The MIB structure also permits private, proprietary sub-trees, enabling the various suppliers to offer system-specific information. Parts of the tree structure are illustrated in Figure 4.

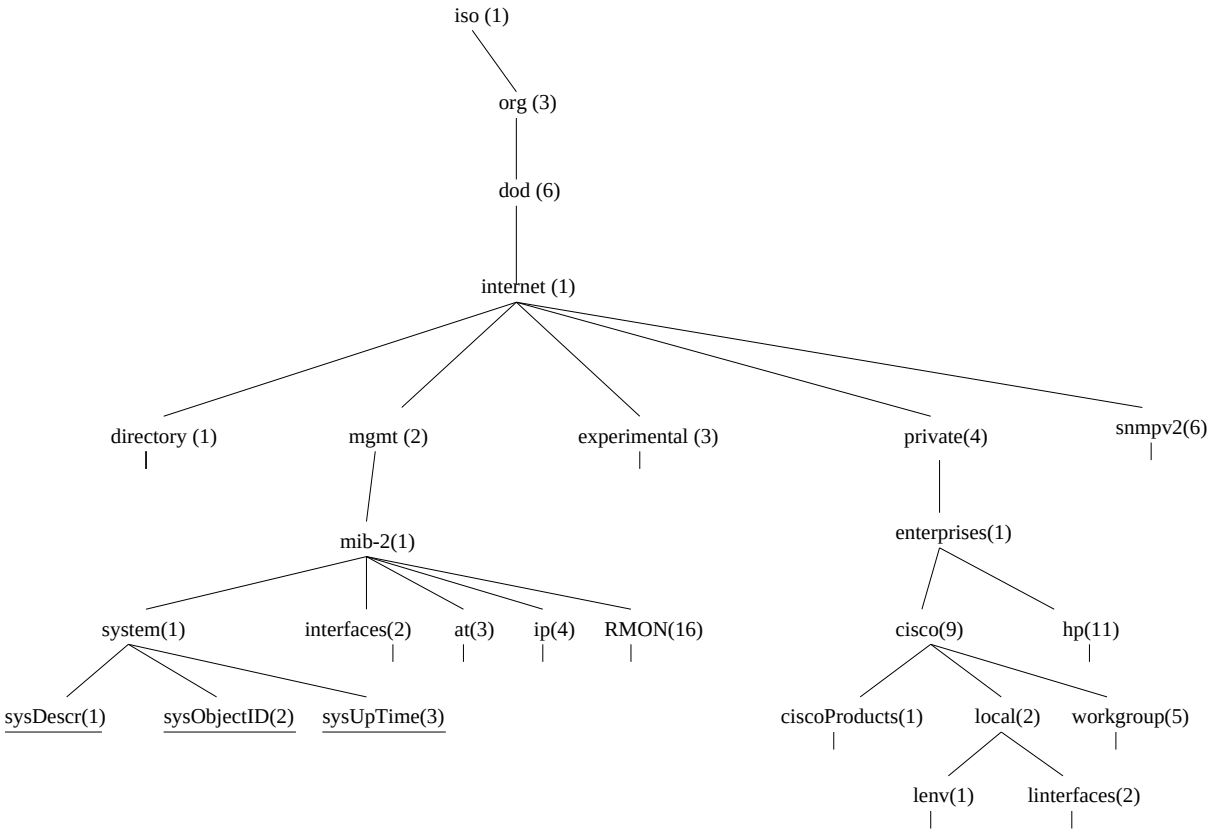


Figure 4: Excerpt from the Management Information Base (MIB) structure

The leaf nodes in the tree structure contain information which can either be read (`snmpget`) or written (`snmpset`). Figure 4 shows three leaf nodes, (`sysDescr`, `sysObjectID` and `sysUpTime`). As an example, the following query will return the uptime for trd-gw1.uninett.no:

```
snmpget trd-gw1.uninett.no .1.3.6.1.2.1.1.3.0
```

SNMP protocol has been improved several times. The security of SNMP is debatable and the first attempt at improvement came with Version 2, but the IETF work group could not agree and in January 1996, Version 2c (RFC1901) was adopted. In this, the SNMP password (community) is still transmitted unencrypted over the network and can potentially be sniffed (unless precautions are taken).

SNMPv2c made other changes to the protocol, among other things introducing `getbulk`, which enables the transmission of large amounts of data in a single SNMP query. SNMPv2c also supports 64-bit counters (as opposed to the original 32-bit counters), which are necessary for monitoring traffic with gigabit rates.

The most recent version of SNMP is Version 3 (RFC 3411-3418), which was finally approved in 2004 and supersedes versions 1 and 2c. In SNMPv3, security has finally been improved. Authentication is now encrypted and one can also choose to encrypt the data transport itself.

Although SNMPv3 has been an approved standard for many years, SNMPv2c continues to dominate in actual implementations.

SNMP requirements for network equipment

When purchasing routers, switches and other network electronics, it is important to consider the equipment's ability to support different MIBs. This will vary from one manufacturer to another. Most often manufacturers have their own proprietary MIBs in which data is made accessible. This is valuable as a supplement, but it is very important that the IETF standard MIBs are fully supported. The monitoring system should for its part as far as possible base itself on data from the standard MIBs. In this way the monitor can function as a generic SNMP collector which can monitor equipment from a number of manufacturers.

The following is a list of minimum requirements which network equipment should support:

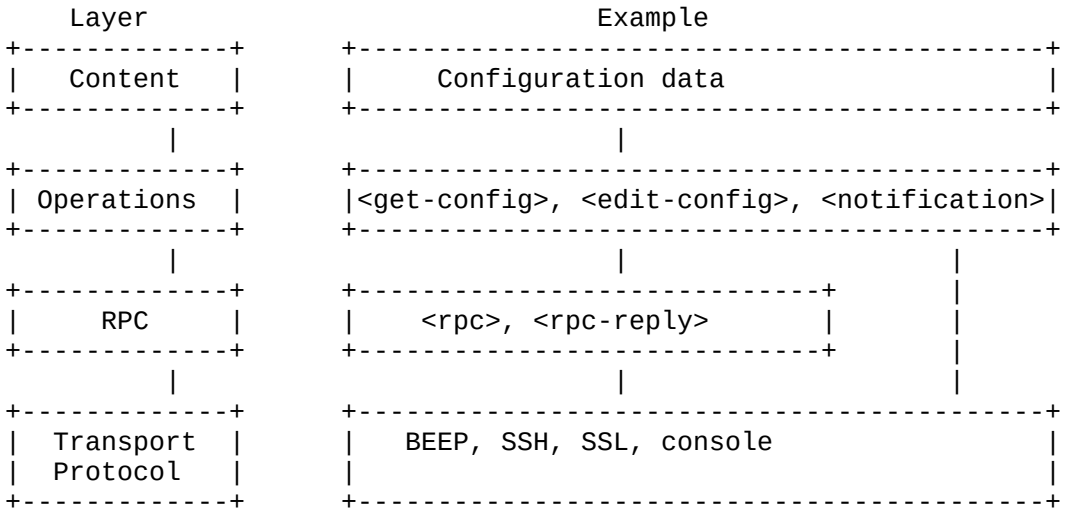
RFC 3418: MIB II	(system data)
RFC 2863: IF-MIB	(interface, including 64-bit traffic counters)
RFC 4293: IP-MIB	(IP interface and ARP; IPv4 and IPv6)
RFC 4133: ENTITY MIB	(modules, optics, software version, serial number)
RFC 4188: BRIDGE-MIB	(bridge table for switches)
RFC 4363: Q-BRIDGE MIB	(bridge table for VLAN, VLAN configuration)
RFC 3635: EtherLike-MIB	(duplex data for switch ports)
RFC 2368: MAU MIB	(physical medium for ports, for example twisted-pair cable, fibre optic, etc.)

B. NETCONF

One reason why the poor security of SNMP has been tolerated for so long is that the protocol in practice is mostly used for monitoring. For configuration of equipment, CLI is the most widespread method. There are several reasons for this, one being that CLI is text-based and easy to deal with. Besides, the majority of equipment suppliers do not implement full functionality through SNMP.

An IETF work group was set up to consider an alternative protocol for configuring network equipment in a secure, scaleable and flexible manner. In December 2006 the work resulted in the Network Configuration Protocol, RFC 4741 (NETCONF).

NETCONF provides mechanisms for installing, modifying and deleting the configuration of network equipment. The operations are implemented through an RPC (Remote Procedure Call) layer. NETCONF uses XML (Extensible Markup Language) based data encoding. NETCONF can be run with the help of several alternative transport protocols. Conceptually, NETCONF is divided into four layers:



XML can be difficult to work with and a separate IETF work group, NETMOD, defined a more user-friendly modelling language, YANG, in RFC 6020-6021 (October 2010). Work is in progress in the NETMOD work group to consider ways of improving NETCONF's compatibility with SNMP.

So far, there are few implementations of NETCONF and YANG.

References

[1] The GigaCampus programme (2006-2009): http://www.gigacampus.no

[2] UNINETT's toolbox products: https://ow.feide.no/gigacampus:verktoykasse

[3] UNINETT's beacon platform: http://forskningsnett.uninett.no/produkt/maalepale.html

[4] NAV (Network Administration Visualized): http://metanav.uninett.no/

[5] Stager: http://software.uninett.no/stager

[6] Software developed by UNINETT: http://software.uninett.no

[7] UFS 114 Fault-tolerant Campus Networks, https://ow.feide.no/gigacampus:ufs#nett

[8] ITIL (IT Infrastructure Library), http://www.itil-officialsite.com/

[9] FCAPS, http://en.wikipedia.org/wiki/FCAPS

[10] UFS122: Recommended ICT Security Architecture in the Higher Education Sector
https://ow.feide.no/gigacampus:ufs#sikkerhet

[11] GigaCampus profitability assessment, 4 July 2008
https://ow.feide.no/_media/gigacampus:gigacampus-lonnsomhet.pdf

Definitions

CLI	Command Line Interface
GBIC	**G**iga**B**it **I**nterface **C**onverter (fibre optics for GigaBit Ethernet)
HSRP	Hot Standby Routing Protocol (proprietary Cisco protocol)
HTTPS	HTTPS is a secure version of HTTP, which is the communication protocol of the World Wide Web
IM	Instant Messaging
IRC	Internet Relay Chat
ITIL	Information Technology Infrastructure Library
MIB	Management Information Base
NMS	Network Management System
RCS	Revision Control System
RRD	Round Robin Database
RSS	Really Simple Syndication
SFP	Small Form-factor Pluggable (same function as GBIC, but smaller form-factor)
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TFTP	Trivial File Transfer Protocol
UPS	Uninterruptible Power Supply
VRRP	Virtual Router Redundancy Protocol