Producing Quality Factors of LAN Interconnection Services

Harri Valimaa <harri.valimaa@tele.fi>

Tapani Honkanen <tapani.honkanen@tele.fi>

Abstract

LAN interconnections have considerably increased since the end of 80's. In most cases, the service is implemented by service providers, which has accelerated the growth of data networks.

As a result of the growth, network management programs have become essential to service providers. Quite often the monitoring capabilities of these programs have been found out to be insufficient. This fact has led to the need of development of more powerful tools.

The primary goal of this research has been to design a program that produces SNMP based quality factors which reflect the state of network devices.

1 Introduction

Most network management systems are based on Simple Network Management Protocol (SNMP) that makes them quite versatile and powerful. However, these programs cannot utilize the management data collected by SNMP as effectively as they should. Basically the network management programs can only display data or they add only minimal value to the raw SNMP data. The main problem is the capability of programs to further analyze collected SNMP data. For instance there are programs that cannot handle more than one MIB variable simultaneously.

Let us consider a simple example where we want to monitor the percentage of the discarded input packets that is given by a formula shown in Fig.1. The calculation may become a problem because the formula consists of five separate variables. Some programs can solve this, but they cannot join the results calculated from several functions to widen the view over a component or a net. The results are basically only a bunch of graphs or lists of numbers which are meaningless for persons who are not specialists in SNMP or network technologies. For them a single representative number reflecting the health of a network device would be more useful. Even though experts can easily follow calculated graphs, these formulas are too sensitive to network traffic profile changes. Thus oscillation or occasional sharp peaks in the output value of the formula may cause unnecessary alarms, if alarm thresholds have been set.

By using a carefully designed system, it is possible to take full advantage of SNMP and avoid problems described above.

1.1 Calculation of Quality Factors

Construction of such a system can be implemented in many ways. The easiest and also the most often used approach is based on the monitoring of only one function formed by several management variables [ref. Fig.1]. In the general case a form of the function could be given by

$$x = \frac{\Delta p_{err} / \Delta t}{\Delta p_{tot} / \Delta t}$$

where Δp_{err} is the number of error packets received or sent over the polling time interval Δt and Δp_{tot} is the number of total packets received or sent in the same direction over the polling time interval Δt. As stated in previous example this solution is not always the best one, because the oscillation of variables inflicts the function to oscillate, too.

To reduce the sensitivity of the system to variations in the values of management variables, the system has to be constructed to produce as an output a "quality factor" that is given by

$$h(t) = h(t-1) + e(t)$$

Now the system evaluates correction factor $e(t)$ from the function x and adds it to the previous quality factor $h(t\text{-}1)$. To get a usable quality factor, $h(t)$ must be limited to suitable range, like 0 to 10. However, this system does not fully eliminate the problem, because the output may still oscillate [Fig. 2]. The main advantage of this approach is that the range of the amplitude of oscillation can be defined beforehand and therefore the oscillation can be reduced.

With the inclusion of the rate of change of the function x in the calculations of correction factor $e(t)$ and with the provision that there is a nonlinear relationship between the inputs and the output, the oscillation problem can be solved. The rate of change $v[n]$ over a time interval from $t[n\text{-}1]$ to $t[n]$ is obtained as

$$\text{if\%InDiscards} = \frac{\text{ifInDiscards}}{\text{ifInDiscards} + \text{ifInErrors} + \text{ifInUnknownProtos} + \text{ifInUcastPkts} + \text{ifInNUcastPkts}} * 100\ \%$$

Figure 1 An example of calculation function.

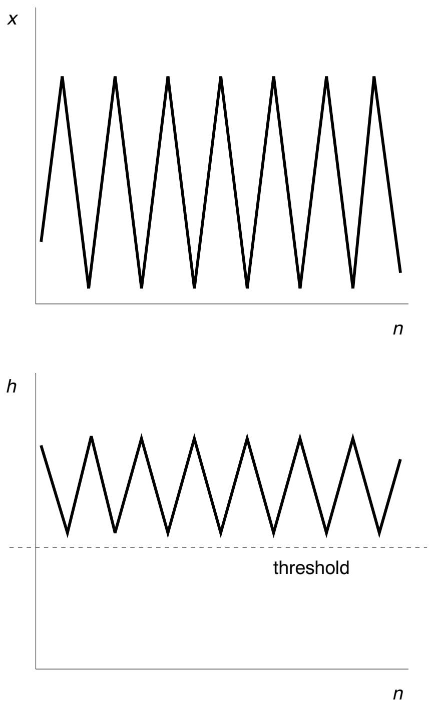


Figure 2 Oscillation of input may cause oscillation of output, too.

$$v[n] = \frac{x[n] - x[n-1]}{t[n] - t[n-1]}$$

where $x[n\text{-}1]$ and $x[n]$ are the values of the function x at sample times n-1 and n.

For example, let us suppose that there is an obvious non-linear relationship between inputs (x, v) and the output (e) as shown in Fig. 3. It can be expected that for the same values of function x as used in Fig. 2 the quality factor should behave as shown in Fig. 4.

The trend of the quality factor approaches the maximum or the minimum, depending on the exact values of x and the exact internal configuration of the system, over a finite period of time. There is still some oscillation, but in this case it is not so harmfull provided that the range of the amplitude is chosen to be small enough. One must notice that too narrow range slows down the system.

The implementation of the system described above is very difficult by using traditional mathematical methods or formal modelling technologies, because they require demanding calculations and therefore deep knowledge of these sciences. The problem becomes even worse, if several quality factors must be joined into a single factor.

For all these reasons a different approach was chosen. This research rests on fuzzy reasoning. A fuzzy reasoning system based on fuzzy reasoning method is discussed in the next chapter.

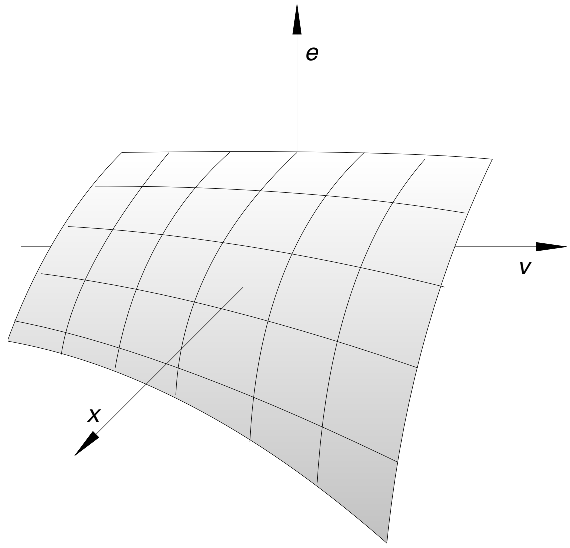


Figure 3 By using nonlinear relationship between the inputs (x, v) and the output (e), it is possible to avoid oscillation of the output.

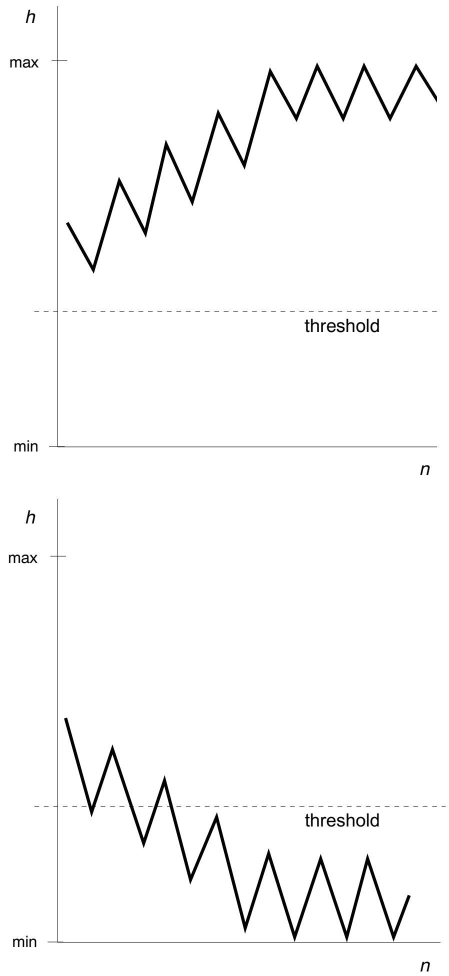


Figure 4 Influence of the consideration of the rate of change on the quality factor when using a same input (x) as shown in Fig. 2.

2 Fuzzy Reasoning System

Fuzzy reasoning is based on fuzzy logic which is quite a young area of technology. Fuzzy logic has nowadays become more and more popular and widely used in many branches. The first publications in the fuzzy set theory was introduced in 1965 by L. A. Zadeh. The fuzzy logic differs totally from conventional crisp dual on-off logic or formal modelling technologies, because it tries to behave like a human thinking process does [1].

In the classical dual logic the state of an element is true or false, which is often represented as degrees of truth 0 (false) and 1 (true). Fuzzy logic differs from this because the state could partially be true and false at the same time. Therefore, the degree of truth can take values between 0 and 1, as well. Several phenomena and matters of daily life, like "rich person" or "blue sky", are so vague that it is difficult to state them by using on-off logic or mathematical models. In the fuzzy set theory uncertainty that can be found in phenomena and matters is called fuzziness [4]. For instance, if we want to represent phenomenon "fast speed" by means of dual logic the result could be as shown in Fig. 5. There is a unnatural step between the

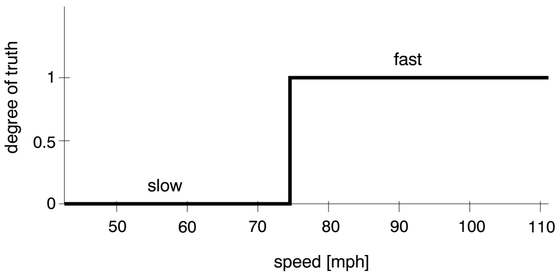


Figure 5 "Fast speed" represented by using dual logic.

slow and the fast speed. When representing the same phenomenon by means of fuzzy set theory the result could be as shown in Fig. 6. Many cases are much more complex. For instance in this research we want to produce quality factors. A quality factor can be defined by a number of descriptors like errors, loads and utilizations. On the other hand it can be found that these descriptors are defined by other descriptors; errors could be described by CRC and alignment errors. Selection of the right descriptors depends on a person. This is some kind of fuzziness, too [4].

Fuzzy reasoning system consists of four major parts which are fuzzification interface, knowledge base, decision-making logic and defuzzification interface (Fig. 7) [1]. There can be one or more inputs and outputs. However, in most cases a system with single output is used. The idea behind fuzzy logic is to represent different things by the help of linguistic variables which take linguistic values [1]. Here the word linguistic means a word or a sentence of natural language. For example, a rate of error packets could be represented as a linguistic variable "rate of errors"

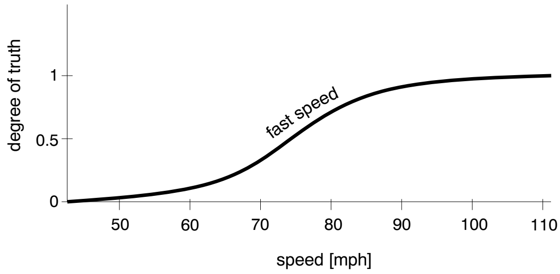


Figure 6 "Fast speed" represented by using fuzzy logic.

that might take values like "zero", "slow", "moderate", "fast", "very fast" etc. instead of "exact" numbers. Usually five to seven linguistic values are used to describe a variable.

The decision-making logic based on fuzzy reasoning makes decisions from linguistic input values of input variables by the help of linguistic decision-making rules. The output of decision-making logic is also linguistic. Conversion from numeric input values into linguistic values is done by the fuzzification interface. Reconversion of output back to the exact numeric value is handled by a defuzzification interface. The fuzzification interface and defuzzification interfaces use mapping information stored in a database called knowledge base when making conversions between linguistic and numeric values.

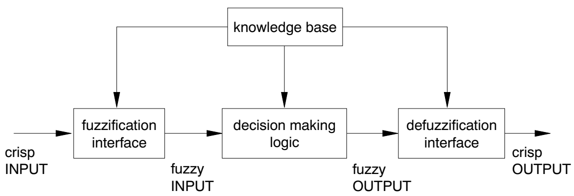


Figure 7 The basic model of fuzzy reasoning system.

2.1 Fuzzification

Mapping information used by fuzzification and defuzzification is defined by membership functions. For the fuzzification the linguistic values are characterized by fuzzy sets. Membership function μ_F denotes a grade of membership (=degree of truth) $\mu_F(u)$ that u belongs to a fuzzy set F [1]:

$$\mu_F:\ U \rightarrow [0, 1]$$

where U is a base set of objects $\{u\}$. Function $\mu_F(u)$ takes values in range from zero to one. Membership function can be continuous or discrete. Most common shapes of membership function are triangle, singleton, trapezoid and gaussian (Fig. 8) [3].

For instance the variable "rate of errors" could be represented by using membership functions shown in

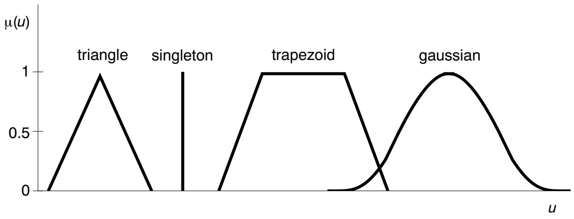


Figure 8 Different shapes of membership function.

Fig. 9, where it belongs to two fuzzy sets with different grades of membership. The rate of 10.2 error packets per second belongs to "moderate" with the grade of membership of 0.55 and at the same time to "fast" with the grade of membership of 0.15.

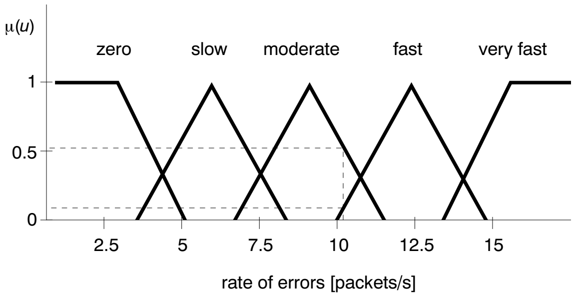


Figure 9 Linguistic variable "rate of errors".

2.2 Fuzzy Reasoning

As earlier discussed the reasoning is done from fuzzificated inputs. The decision-making rules are stored into the rule base which is part of the knowledge base. Generally these rules are in the IF-THEN form [1].

Fuzzy reasoning employs logical operators AND, OR and NOT and set operators AND, OR and NOT. Logical operators AND and OR between two fuzzy sets A and B are defined as [1]

$$\mu_{A \wedge B}(u,v) = \min\{\mu_A(u),\ \mu_B(v)\}$$

$$\mu_{A \vee B}(u,v) = \max\{\mu_A(u),\ \mu_B(v)\}$$

where $u \in U$ and $v \in V$. In the case of more fuzzy sets the formulas are equivalent. Operator NOT for set A is defined as [1]

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

Correspondingly set operators are defined as [1]

$$\mu_{A \cap B}(u) = \min\{\mu_A(u),\ \mu_B(u)\}$$

$$\mu_{A \cup B}(u) = \max\{\mu_A(u),\ \mu_B(u)\}$$

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

There is a difference between the use of the logical and the set operators. Set operators are only used for the fuzzy sets of same base set whereas logical operators are used for the fuzzy sets of different base sets.

Fuzzy reasoning process is clarified in Fig. 10, where $x \in X$ and $y \in Y$ are input variables and $z \in Z$ is an output variable; x_0 and y_0 are crisp input values; A_i, B_i and C_i are fuzzy sets of base sets X, Y and Z, respectively; $\mu_{A_i}(x)$, $\mu_{B_i}(y)$ and $\mu_{C_i}(z)$ are membership functions of these fuzzy sets. In this case it is assumed that we have only two rules (N=2):

1. IF x is A_1 AND y is B_1 THEN z is C_1
2. IF x is A_2 AND y is B_2 THEN z is C_2

To obtain an output there are several fuzzy inference methods that are possible to use. In this example, Mamdani's Minimum Operation rule is used [2]

$$\mu_{C_i'}(w) = \mu_{C_i}(w) \wedge \left(\mu_{A_i}(u) \wedge \mu_{B_i}(v)\right)$$

where the output variable is $w \in W$. By using operators described above and variables x, y and z the formula is given by [3]

$$\mu_{C_i'}(z) = \min\{\mu_{C_i}(z),\ \min\{\mu_{A_i}(x_0),\ \mu_{B_i}(y_0)\}\}$$

The total output is obtained as [3]

$$\mu_C(z) = \max\{\mu_{C_1'}(z),\ \mu_{C_2'}(z),\ \ldots,\ \mu_{C_N'}(z)\}$$

2.3 Defuzzification

To get a crisp numeric output value the fuzzy outputs have to be defuzzificated. As there were several fuzzy inference methods there are also several defuzzification methods [2]. The most commonly used method is the Center of Area method (COA). In COA the output z_0 can be obtained when using a discrete membership functions as [2]

$$z_0 = \frac{\sum_{j=1}^{n} \mu\left(w_j\right) \cdot w_j}{\sum_{j=1}^{n} \mu\left(w_j\right)}$$

where n is the numeric of quantization levels of the output; w_j is the middle of quantization level j and $\mu(w_j)$ corresponds to $\mu_C(z)$. COA method is clarified in Fig. 11.

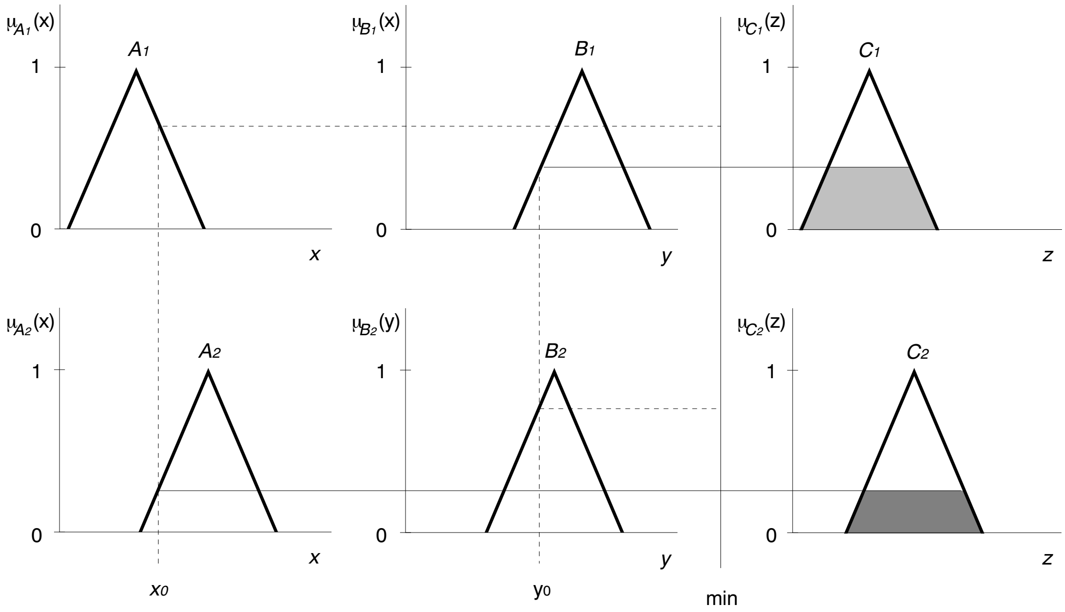


Figure 10 Fuzzy reasoning by using Mamdani's Minimum Operation rule.

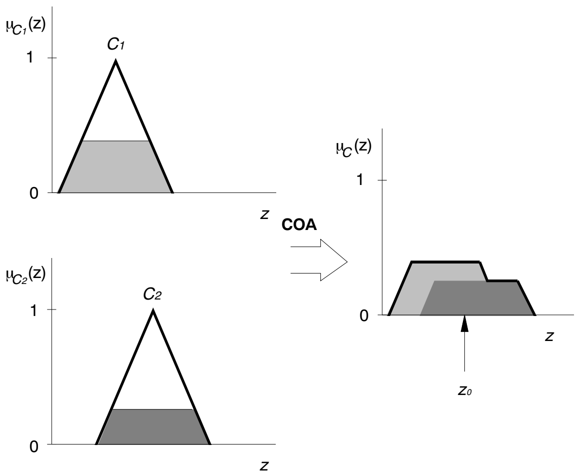


Figure 11 COA defuzzification method.

2.4 Fuzzy Reasoning System in the Calculations of the Quality Factors

Fuzzy reasoning system can be exploited in many ways in the calculations of the quality factors. The easiest way for the implementation is to use a two-input-single-output fuzzy reasoning system (FRS), where the inputs are the function *(x)* and its rate of change *(v)* (Fig. 12). The number of management objects that affect the quality factor depends on the function that is used.

To obtain a more powerful quality factor several functions must be used in parallel. For each function there has to be an independent fuzzy reasoning system. The results from these fuzzy systems are joined in a designed order by new fuzzy systems. This chain continues until there is only one output value left (Fig 13). The use of several fuzzy reasoning systems makes it also possible to get quality factors of different levels. For instance network operators could get quality factors of lower levels (1, 2, ...) whereas end users could get quality factors of more refined levels (..., *n*-1, *n*).

2.4.1 Allowed Management Objects

The calculation system of the quality factors is designed for use of SNMP management objects of different branches like SNMP MIB-II, RMON and vendor specific. However, there is not any restrictions for use of non-SNMP management data. It is even possible to join SNMP, RMON and vendor specific based functions when using the system shown in Fig 13.

The best results will be achieved when using SNMP COUNTER type management data.

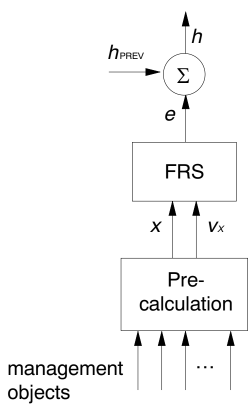


Figure 12 Two-input-single-output system.

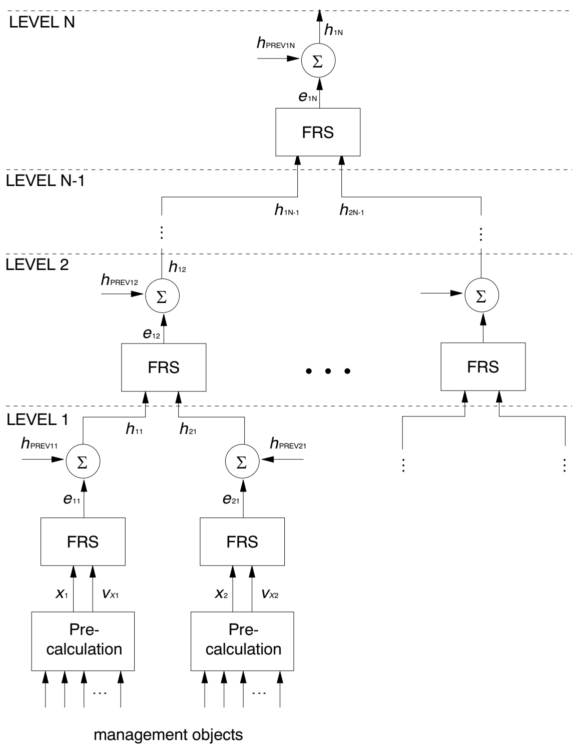


Figure 13 Multi-logic system.

3 Realization of the Calculation System

To become convinced of the plans a system was implemented as shown in Fig. 12. Hence, the basic building blocks of the system are pre-calculation, fuzzy reasoning system and configuration tool. The pre-calculation takes care of the calculation of the function (x) and the calculation of the final quality factor (h) from the correction value (e) returned by the fuzzy reasoning system. The configuration tool helps to set and change the function (x) and different parameters of the fuzzy reasoning system.

To simplify the program the number of membership functions and their shapes are fixed. To describe the function (x) six membership functions have been used. Both the rate of change (v) and the output (e) of the fuzzy system have been described by using seven membership functions. The shapes of the membership functions of the function and its rate of change have been chosen to be triangular. The number of quantization levels (n) has been set to 1000. To reduce the number of calculations needed in the defuzzification, the shapes of the membership functions of the output have been chosen to be singletons. Hence, when using COA method the crisp output can be obtained as [3]

$$z_0 = \frac{\sum_{j=1}^{n} \mu(x_j) \cdot x_j}{\sum_{j=1}^{n} \mu(x_j)}$$

where n is the number of singletons returned by the fuzzy reasoning process; x_j is the location of the membership function $\mu(x_j)$.

The precalculation program and the configuration tool are implemented as UNIX scripts. The fuzzy logic is implemented in C language. Programs have been tested both in practice and also by help of simulations.

3.1 Configuration Tool

The right configuration is very important for fuzzy reasoning system in order to return reliable quality factors. The blocks of fuzzy reasoning system needing configuration are the membership functions, the rule base and the calculation function. Hence, there is a large number of adjustable parameters. For these reasons the configuration tool had to be made as simple as possible.

The membership functions of the calculation function (x) are set by giving the vertexes of triangles and the corresponding linguistic values for each triangle (Fig. 14). The second input, the rate of change (v), is set in the same way. The output (e) is defined by giving the locations and the linguistic values of the membership functions (Fig. 15). The rules between the inputs and output are defined as shown in Fig 16. The calculation function (x) is defined by first giving the factors of function and corresponding management objects. Next the function is formed by using these factors (Fig. 17).

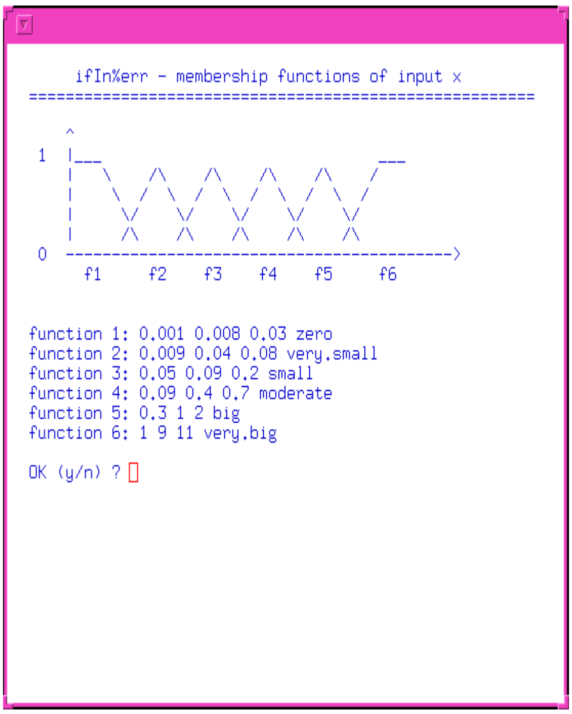
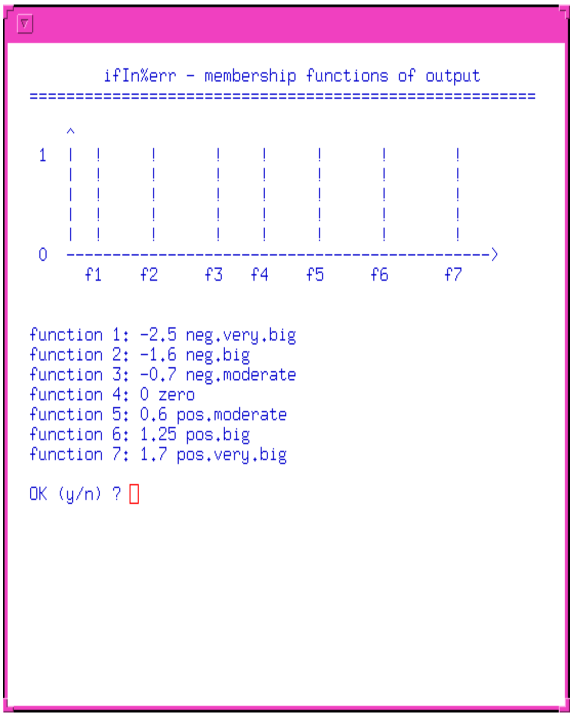
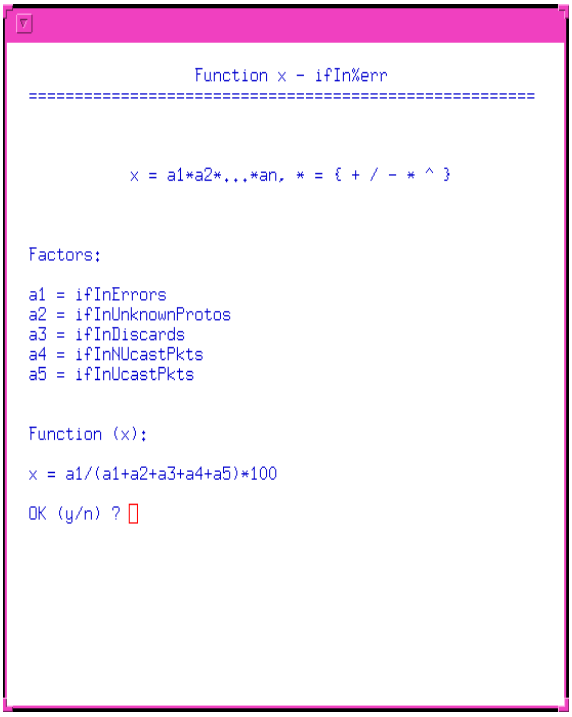


Figure 14 Configuration of the membership functions of x.



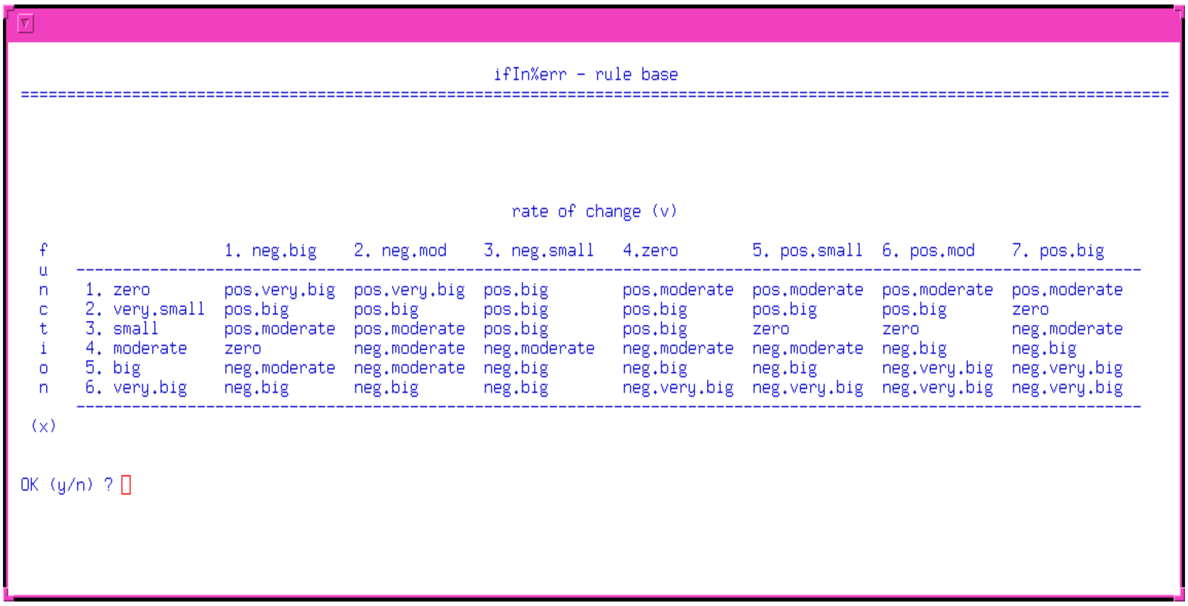
```
        ifIn%err - membership functions of output
=======================================================

    ^
 1  |  !     !      !    !     !      !       !
    |  !     !      !    !     !      !       !
    |  !     !      !    !     !      !       !
    |  !     !      !    !     !      !       !
    |  !     !      !    !     !      !       !
 0  ----------------------------------------------->
      f1    f2     f3   f4    f5     f6      f7

function 1: -2.5 neg.very.big
function 2: -1.6 neg.big
function 3: -0.7 neg.moderate
function 4: 0 zero
function 5: 0.6 pos.moderate
function 6: 1.25 pos.big
function 7: 1.7 pos.very.big

OK (y/n) ?
```

Figure 15 Configuration of the membership functions of the output.



```
                 Function x - ifIn%err
=======================================================


           x = a1*a2*...*an, * = { + / - * ^ }


Factors:

a1 = ifInErrors
a2 = ifInUnknownProtos
a3 = ifInDiscards
a4 = ifInNUcastPkts
a5 = ifInUcastPkts


Function (x):

x = a1/(a1+a2+a3+a4+a5)*100

OK (y/n) ?
```

Figure 17 Configuration of the calculation function.



ifIn%err - rule base

rate of change (v)

function (x)	1. neg.big	2. neg.mod	3. neg.small	4.zero	5. pos.small	6. pos.mod	7. pos.big
1. zero	pos.very.big	pos.very.big	pos.big	pos.moderate	pos.moderate	pos.moderate	pos.moderate
2. very.small	pos.big	pos.big	pos.big	pos.big	pos.big	pos.big	zero
3. small	pos.moderate	pos.moderate	pos.big	pos.big	zero	zero	neg.moderate
4. moderate	zero	neg.moderate	neg.moderate	neg.moderate	neg.moderate	neg.big	neg.big
5. big	neg.moderate	neg.moderate	neg.big	neg.big	neg.big	neg.very.big	neg.very.big
6. very.big	neg.big	neg.big	neg.big	neg.very.big	neg.very.big	neg.very.big	neg.very.big

OK (y/n) ?

Figure 16 Configuration of the rule base.

3.2 System Test Results

The tests were made by using Datanet wide area network of Telecom Finland. The network consist of over 1500 routers of several manufacturers like Cisco, Digital and IBM. The network is managed by the network operating center, which made it possible to collect easily data from real devices.

The tests were implemented by using the calculation function (x) shown in Fig. 17. The function implies the ratio between the error input packets and the total input packets. All management objects used are taken from the interfaces group of SNMP MIB-II. The settings of the fuzzy reasoning system were found by several trials and errors.

Input for the precalculation was simulated management data and data collected from few Cisco routers of a real network. The calculation interval of the quality factors was 15 minutes. Similarly, the polling interval was 15 minutes for the data collected from the routers.

The effect of an oscillating input on the system was tested with simulations. The input data used is shown in Fig. 18. The calculated quality factor is shown in the same figure. By changing the amplitude of the oscillation it was noticed that the angle of the trend of the quality factor was decreased or increased.

In the definitions it was a challange to find a way to avoid an uncontrollable oscillation of the quality factor. Anyhow, in the simulations the amplitude of oscillation of the input that inflicts the output to oscillate uncontrollably was found. This critical amplitude is shown in Fig. 19. With values of the amplitude of the oscillation of the input smaller than the critical amplitude, the trend of the output is rising. Correspondingly, with values larger than critical amplitude the trend of the output is decreasing. The remaining oscillation is very slight and is thus quite harmless in many cases. But if the oscillation crosses the threshold limit, it will cause great problems. To circumvent this situation the settings of the fuzzy reasoning system must be selected in such a way that this oscillation of the quality factor will take place inside acceptable limits.

Altogether, the simulations show that the settings of the fuzzy reasoning system have a very crucial effect on directions the trend of the quality factor will go and how fast the change will happen.

Figures 20a and 20b show realistic data collected from routers and the corresponding quality factors calculated from this data. In the first case the value of the function is so big that the trend of the quality factor decreases almost linearly. This clearly indicates errors and therefore the quality factor displays "bad". In Fig. 20b errors occur intermittently, but calculated quality factor keeps displaying "good". It can be easily seen that quality factor follows the behaviour of the function very well.

One should remember that collected data is not quite accurate, because the "samples" are 15 min counter values. During that time lots of things can happen and the error condition may be better or worse. Fortunately fuzzy reasoning digs the real trend out and displays it clearly.

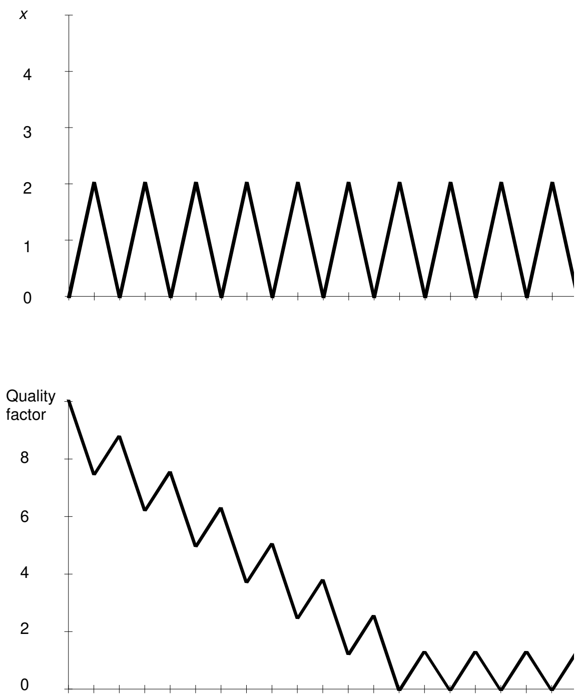


Figure 18 Behaviour of the quality factor when using a simulated input (x)

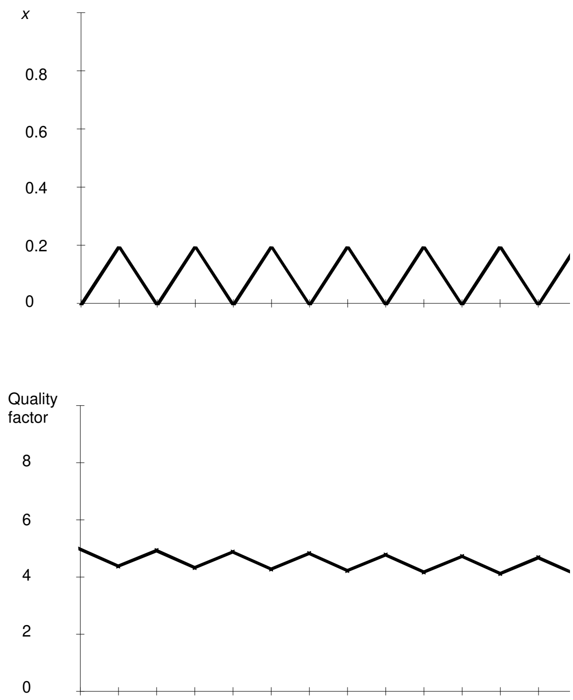


Figure 19 Behaviour of the quality factor when the oscillation of the input (x) occurs at the critical amplitude.

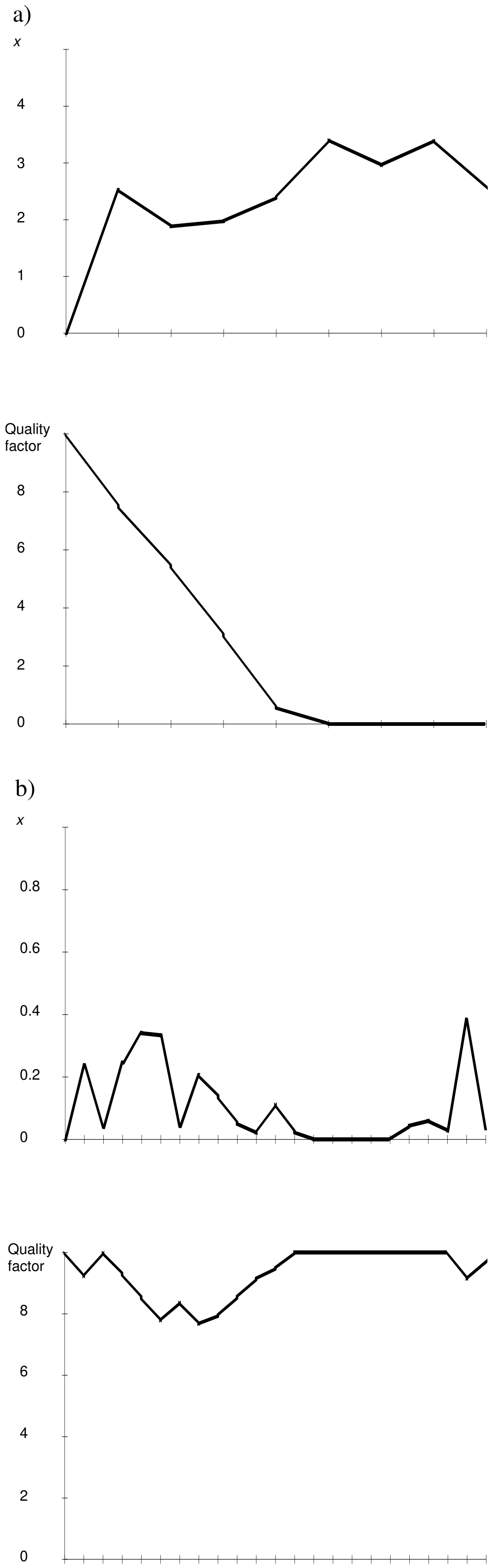


Figure 20 Quality factors calculated from functions (x) based on management data of real network.

4 Summary

Network management programs have become more and more important for operators of present networks. To take full advantage of the SNMP sophisticated programs must be used. These tools must have methods to analyze simultaneously values of functions formed from different management objects and they must produce common quality factors based on these analyses.

Fuzzy reasoning provides a good tool for this kind of analysis. It helps to avoid problems due to the dynamic changes in input functions. The main advantage is that it makes possible to produce quality factors based on SNMP management data of different branches like SNMP MIB-II and RMON. It is even possible to use non-SNMP data.

References

[1] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller - Part I," IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 2, pp. 404-418, March/April 1990.

[2] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller - Part II," IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 2, pp. 419-435, March/April 1990.

[3] P. V. Viljamaa, "Neuro ja fuzzy-laskennan perusteet" Tampere University of Technology, 1993.

[4] H. J. Zimmermann, "Fuzzy set theory and its applications," 2nd edition, Kluwer Academic Publishers, 1991.

Author information

Harri Valimaa received the M.Sc. degree in measurement technology from the Tampere University of Technology, Finland, in 1994. Since 1992 he has worked for Telecom Finland. His current assignments lie in the areas of network management systems of local and wide area networks.

Tapani Honkanen received his M.Sc degree in Electonics in 1981 and Tech.Lic. degree in Computer Science in 1988 from Tampere University of Technology, Finland. He is a long term influential in Finnish networking. Until 1994 he worked as an educator at Relevantum Inc. lecturing data communications, LANs, TCP/IP and network management. Now he is at Telecom Finland and works as a manager for LAN management services.