Paper

Probes for fault localization in computer networks

Wiesław Traczyk

Abstract—Fault localization is a process of isolating faults responsible for the observable malfunctioning of the managed system. This paper reviews some existing approaches of this process and improves one of described techniques—the probing. Probes are test transactions that can be actively selected and sent through the network. Suggested innovations include: mixed (passive and active) probing, partitioning used for probe selection, logical detection of probing results, and adaptive, sequential probing.

Keywords—fault localization, probes, computer networks, partitions, logic design.

1. Introduction

As computer networks increase in size, heterogeneity and complexity, effective management of such networks becomes more important and more difficult. Network management is essential to ensure the good functioning of these networks.

The International Standards Organization has divided network management tasks into six categories, as part of their Open System Interconnection Model. One of these categories—the fault management—can be characterized as detecting when network behavior deviates from normal and formulating a corrective course of action. Fault management deals with [1]:

– *fault detection*, to know whether there is a failure or not in the network;

– *fault localization*, to know which is(are) the component(s) that has/have failed and caused the received alarms;

– *fault isolation* so that the network can continue to operate, which is the fast and automated way to restore interrupted connections;

– *network (re-)configuration* that minimizes the impact of a fault by restoring the interrupted connections using spare equipment;

– *replacement* of the failing component(s).

Fault localization is the core of fault diagnosis and means a process of analyzing external symptoms of network disorder to isolate possibly unobservable faults responsible for the symptoms' occurrences. Traditionally, fault localization has been performed manually by experts but, as systems grew larger and more complex, automated fault localization techniques became critical.

The terms used so far (and in the future) require more precise definitions [1, 2].

Object is a part of the network that has separate and distinct existence. An object can be a node, a layer in a protocol stack, a software process, a virtual link, a hardware component, etc. Objects in a communication system consist of other objects, down to the level of smallest objects which are considered indivisible and called *elements*.

Fault (also referred to as *root problem*) can be defined as an unpermitted deviation of at least one characteristic parameter or variable of a network object from acceptable or usual or standard values. Faults may be classified according to their duration time as *permanent, intermittent* and *transient*.

Error, a consequence of fault, is defined as a discrepancy between observed and correct value. Fault may cause one or more errors.

Failure is an error that is visible to the outside world. Errors may propagate within the network causing failures of faultless hardware or software.

Symptoms are external manifestation of failures. They are observed (and send to the network manager) as *alarms*.

Communication networks are built on several layers, performing each fault management functions independently. When a failure occurs, several symptoms are issued to the network manager from the different management layers, and fault management functions start in parallel. Research is carried out to allow interoperability between different layers, to avoid task duplication and increase efficiency.

This paper discusses some approaches to automated fault localization and presents the new method based on probes.

2. Fault localization techniques

All techniques performing fault diagnosis rely on analysis of symptoms and events (such as warnings and parameters of the network elements) that are generated or detected during the occurrence of the fault. One can divide them in two main categories. The first ones are *passive approaches*, which compute fault location hypotheses on the basis of signals, generated by network elements by oneself and sent to management centers. The second ones are *active approaches*, which periodically check the state of the network elements, whether they are correct or not.

Among **passive fault localization techniques** two families of methods are of special interest: artificial intelligent (AI) methods and fault propagation methods.

Artificial intelligent techniques for fault localization. This the most widely used family contains a lot of methods and appropriate systems [1–4].

Model-based systems construct an abstract model of the network. The model represents the network topology and is able to generate predictions of the normal behavior of the system. This predictions are compared with network observations and used for obtaining fault hypotheses. Depending of the kind of model, different approaches can be used: deterministic, probabilistic, temporal, finite state machines, etc. The advantages of these systems are that they are able to cope with incomplete information and with unforeseen failures. The drawback is the difficulty of developing good model for large networks and computation complexity.

Rule-based systems describe human expert knowledge in the form of decision rules, linking logical description of the network state (rules conditions) with partial or final localization hypotheses (rules conclusions). These systems do not require profound understanding of the architectural and operational principles of the network, and can effectively take human expertise into account. The disadvantages of rule-based systems are: the translation of human expertise into the set of rules, which cover all cases in an exhaustive manner is hard, and the need to search for all possible fault hypotheses slows down the global functioning of the system.

Case-based systems make their decisions based on experience and past situations. They try to acquire relevant knowledge of past cases and previously used solutions to propose solutions for new problems. If these solutions can not be taken directly from the case-base and need special reasoning on the base of closely matched situations, case-based systems are computationally complex. Their advantages are efficiency and speed when the submitted problem was previously solved, and on-line learning that allows storing newly solved cases.

All three described systems are the special cases of *expert systems* or *knowledge-based systems* but since the most popular expert systems are rule-based, sometimes only these systems are known as expert systems.

Some other AI techniques (neural networks, decision trees, etc.) are rarely used in these applications.

Fault propagation methods [2, 5, 6]. This family of techniques require a priori specification of how a failure condition in one object is relevant to failure condition in other object. It is important because some errors can propagate failures through the network, generating many different alarms.

Code-based techniques use causality graph model to describe the cause-and-effect relationships between network events. For each problem and each symptom a unique binary code is assigned, and fault propagation patterns are represented by a codebook. Fault localization is performed by finding a fault whose code is the closest match to the code of symptoms. For small systems this technique is very effective.

Bayesian networks take into account uncertainty about dependencies within the managed network and about the set of observed symptoms. Uncertainty is represented by probabilities in a believe (Bayesian) network. The best symptom explanation is a result of Bayesian inference. The method is computationally complex and needs many values of events probability.

Dependency graph is a directed graph whose nodes correspond to objects and whose edges denote the fact that a fault in starting object may cause a fault in ending object. Probabilities may be assigned to nodes and edges, describing uncertain relationships and events. Comparing a state of the graph with known state of the network one can find the source of fault symptoms.

Active fault localization techniques construct managing tools which, instead of waiting for symptoms from the network, ask objects about their state and parameters. These techniques are not so popular as passive approaches but in some cases they may be very useful and therefore deserve attention.

Intelligent agents [8] are simply software processes that live on every managed node, collecting, forwarding and setting management information, either at predefined intervals or when requested to by management station.

Monitoring technique [9] locate in some network nodes the computers *(monitors)* which are guaranteed by self-testing. Each monitor tests the adjacent nodes and links, and sends results of testing to the management station. Proper number of monitors can cover all nodes and links in the network. More advanced technique starts from only one monitor. Its adjacent nodes that pass the tests can became new monitors, then test their non tested adjacent nodes and connected links, and so on.

Probing technique [10, 11] use an active measurement approach, called *probing*. A probe is a program that executes on a particular machine (called a *probe station*) by sending a command or transaction to a server or network element, and measuring the response. The objects represented by nodes may be physical entities such as routers, servers and links, or logical entities such as software components, database tables, etc. It is assumed that each node of the tested network can be either "up", functioning correctly, or "down", not functioning correctly. A probe either succeeds or fails: if it succeeds, then every object it tests is up; it fails if any of the objects it tests are down.

Fault localization attempts to determine the state of the system from the probe results, so effectiveness of localization depends on the number of probes and their paths. For practical networks the problem of achieving the minimal set of effective probes is solved only approximately. In the next section known probing technique will be modified, with the goal to simplify its practical applications.

Each of presented above (and many others) approaches has some advantages and drawbacks thus further research, improving existing methods, is still needed.

3. Probes for locating failures

Probes technique is already used by IBM's EPP technology and seems to be promising. The main problem with it is the need for effective algorithm of probes generation. Minimizing the number of probes is important because probing increases network overhead, probe results must be stored and analyzed, and modifications enforced by changes in network configuration are simpler for smaller set of probes. Active probe selection [12] gives some positive results but is based on the prior probability distribution over system states, difficult to achieve.

Approach proposed here tries to improve probing by:

1) application of mixed (passive and active) technique;

2) partitioning used for probe selection;

3) logical detection of probing results;

4) adaptive probing;

it is assumed that symptoms received by managing center refer to high level of the network topology, signaling defects of the whole path, with many nodes (objects).

The set of A tested nodes will be denoted by $\mathbf{N} = \{N_1, N_2, \ldots, N_A\}$ with node name N_i from the set of natural numbers (for simplicity). Binary state $n_i \in \{0,1\}$ of each node N_i equals 1 if node N_i is correct and equals 0—if it is not. The state of the whole network is described by the binary vector $\mathbf{n} = \langle n_1, n_2, \ldots, n_A \rangle$.

A probe S_j is represented by the set of tested nodes $S_j = \{N_{j_a}, N_{j_b}, \ldots, N_{j_z}\}$, and the set $\mathbf{S} = \{S_1, S_2, \ldots, S_B\}$ designates all probes used for tests.

Mixed technique. In the conventional method the set $\mathbf{N}$ consists of all nodes in the network, what makes construction of the probes set $\mathbf{S}$ very difficult and prolongs the time of testing. Instead, we can wait for symptoms generated by network equipment (passive part) and on this base to fix the set of nodes, suspected for malfunctioning. Usually it is not difficult, especially if symptoms concern communication paths or channels. Suspected set of nodes is much smaller then the whole set of nodes, even if it is defined approximately and in excess. Only this smaller set is traversed by probes (active part of the technique).

Partitions for probe selection. To describe probes needed for a fault localization one can use calculus of partitions.

Partition $\pi(X)$ of a set X is a family of subsets X_k (*blocs*), such that $\pi(X) = \{X_1, X_2, \ldots X_K\}$ and for each i, j there is $X_i \cap X_j = \emptyset$ and $X_1 \cup X_2 \cup \ldots \cup X_K = X$. A block of partition π_l of the set X is denoted as X_{π_l}. *Product* of two partitions π_1 and π_2 of the same set X is a partition $\pi(X) = \pi_1(X).\pi_2(X)$ such that for all blocks X_{π_1}, X_{π_2} there exists a block X_π such that $X_\pi = X_{\pi_1} \cap X_{\pi_2}$. Partition with K blocks of the set with K elements is marked as π_0.

Each probe may be considered as partition of the set of nodes from $\mathbf{N}$, with two blocks: the first block consists of all nodes tested by this probe and the second block contains all remaining nodes from $\mathbf{N}$. If $\pi_j(\mathbf{N})$ refers to a probe S_j then $\pi_j(\mathbf{N}) = \{N_{j_a}, N_{j_b}, \ldots, N_{j_z};\ \mathbf{M}_j\}$, where $\mathbf{M}_j$ is a set of remaining nodes.

The results of B probes have to separate one faulty node from A nodes. It can be done if the product of partitions related to probes gives the partition π_0, i.e., $\pi_1.\pi_2 \ldots \pi_B = \pi_0$. It is important advice for probe selection.

Usually managers are able to define the set of probes easily generated by probe stations. From this set a special algorytm should select these probes which contain nodes from $\mathbf{N}$. Desirable probes contain the number of nodes nearing the value A/2, because in this case their informative power is the highest. Since B probes can distinguish 2^B nodes, in the optimal case $A \approx 2^B$. When the product of partitions describing the best probes is not equal π_0, elements of blocks with more then one node should be separated by additional probes with appropriate partitions.

For example if $\mathbf{N} = \{1,2,3,4,5\}$ and the two primarily selected probes have partitions $\pi_1 = \{1,2,3;\ \mathbf{M}_1\}$ and $\pi_2 = \{3,4,5;\ \mathbf{M}_2\}$, then

$$\pi = \pi_1.\pi_2 = \{1,2;\ 3;\ 4,5\} \neq \pi_0.$$

Two probes can separate node 1 from 2 and node 4 from 5. Choosing $S_3 = \{1,4\}$, i.e., $\pi_3 = \{1,4; \mathbf{M}_3\}$ we have

$$\pi_1.\pi_2.\pi_3 = \{1;\ 2;\ 3;\ 4;\ 5\} = \pi_0.$$

It means that probes S_1, S_2, S_3 create the minimal set localizing faults in 5-object network.

Detection of probing results. Each probe S_j from the set $\mathbf{S}$ may give positive or negative result of testing. Having all these results we should compute the number(s) of node(s) with a fault. Logical functions will help in this task.

A probe $S_j = \{N_{j_a}, N_{j_b}, \ldots, N_{j_z}\}$ will be described by conjunction of logical variables v^α:

$$\eta(S_j) = V_j = v_{j_a}^{\alpha_a}.v_{j_b}^{\alpha_b}.\ldots.v_{j_z}^{\alpha_z}.$$

Here $\alpha \in \{0,1\}$, $v^0 = \overline{v}$, $v^1 = v$, and v_x^1 means that a node N_x is correct and v_x^0 means that it is not. Similarly V_j is used if the result of probe S_j is positive and $\overline{V_j}$—if it is negative.

Total result of testing with the set of probes $\mathbf{S} = \{S_1, S_2, \ldots, S_B\}$ will be described by conjunction $\mathbf{V} = \eta(\mathbf{S})$, with differentiated formulas $\mathbf{V}$, depending on the result of probing:

$\mathbf{V}_0 = V_1.V_2.\ldots.V_B$—if all probes gave positive result,

$\mathbf{V}_1 = \overline{V_1}.V_2.\ldots.V_B$—if only first probe gave negative result,

$\mathbf{V}_{1,2} = \overline{V_1}.\overline{V_2}.V_3.\ldots.V_B$—if two first probes gave negative result,

and so on.

Decimal pointers can be obtained by the set $\Gamma(\alpha) = \{i|\alpha_i = 0\}$, taking values of α from notations V^{α}:

$$\mathbf{V}_{\Gamma(\alpha)} = V_1^{\alpha_1}.V_2^{\alpha_2}.\ldots.V_B^{\alpha_B}.$$

When all symbols V^{α} are substituted by the appropriate conjunctions and reformulated, final formula shows the nodes with a fault.
Continuing the example—if probes

$$S_1 = \{1,2,3\} \qquad S_2 = \{3,4,5\} \qquad S_3 = \{1,4\}$$

are used for testing, results can be computed from the following equations:

$$\begin{aligned}
\mathbf{V}_0 &= V_1.V_2.V_3 = v_1.v_2.v_3.\,v_3.v_4.v_5.\,v_1.v_4 = \\
&= v_1.v_2.v_3.v_4.v_5,\\
\mathbf{V}_1 &= \overline{V}_1.V_2.V_3 = (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3}).\,v_3.v_4.v_5.\,v_1.v_4 = \\
&= \overline{v_2}.v_1.v_3.v_4.v_5,\\
\mathbf{V}_2 &= V_1.\overline{V_2}.V_3 = v_1.v_2.v_3.\,(\overline{v_3} \vee \overline{v_4} \vee \overline{v_5}).\,v_1.v_4 = \\
&= \overline{v_5}.v_1.v_2.v_3.v_4,\\
\mathbf{V}_3 &= V_1.V_2.\overline{V_3} = v_1.v_2.v_3.\,v_3.v_4.v_5.\,(\overline{v_1} \vee \overline{v_4}) = 0,\\
\mathbf{V}_{1,2} &= \overline{V_1}.\overline{V_2}.V_3 = (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3}).(\overline{v_3} \vee \overline{v_4} \vee \overline{v_5}).\,v_1.v_4 = \\
&= \overline{v_3}.v_1.v_4,\\
\mathbf{V}_{1,3} &= \overline{V_1}.V_2.\overline{V_3} = (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3}).\,v_3.v_4.v_5.\,(\overline{v_1} \vee \overline{v_4}) = \\
&= \overline{v_1}.v_3.v_4.v_5,\\
\mathbf{V}_{2,3} &= V_1.\overline{V_2}.\overline{V_3} = v_1.v_2.v_3.\,(\overline{v_3} \vee \overline{v_4} \vee \overline{v_5}).\,(\overline{v_1} \vee \overline{v_4}) = \\
&= \overline{v_4}.v_1.v_2.v_3,\\
\mathbf{V}_{1,2,3} &= \overline{V_1}.\overline{V_2}.\overline{V_3} = \\
&= (\overline{v_1} \vee \overline{v_2} \vee \overline{v_3}).(\overline{v_3} \vee \overline{v_4} \vee \overline{v_5}).(\overline{v_1} \vee \overline{v_4}) = \\
&= \overline{v_1}.\overline{v_3} \vee \overline{v_1}.\overline{v_4} \vee \overline{v_1}.\overline{v_5} \vee \overline{v_2}.\overline{v_4} \vee \overline{v_3}.\overline{v_4}.
\end{aligned}$$

From these formulas we can conclude: negative result from first probe means that node 2 is not correct, negative result from second probe means that node 5 is not correct, etc. Negative result from probe 3 is impossible, because probes 1 and 2 gave positive result.
Table 1 summarizes the results.

Table 1
Tests results

Tests	Probes				
Negative probes	1	2	1, 2	1, 3	2, 3
Incorrect nodes	2	5	3	1	4

Adaptive probing. In all approaches described above the set of probes **S** was defined on the basis of the whole set of nodes **N**. But the probes can also be defined sequentially:

- Step 1—probe S_1 is fixed for the set $\mathbf{N_1} = \mathbf{N}$.
- Step 2—probe S_2 is fixed for the set $\mathbf{N_2} = \mathbf{M_1}$ if the result of S_1 is positive and for the set $\mathbf{N_2} = \mathbf{N_1} \setminus \mathbf{M_1}$, if it is negative.
- Step 3, 4, ... as above, until all nodes are tested.

Such adaptive probing can be useful if the algorithm defining probes is fast enough.

4. Conclusion

Four suggestions presented in this paper can improve a procedure of probing, but some further research is needed. For the total diagnose of the large set of nodes it will be required to have:

– automatic generation of a set of probes from the network topology,

– additional probe selection for the case with more than one fault,

– additional probe selection for the case with dynamic routing.

References

[1] C. Mas and P. Thiran, "A review on fault location methods and their application to optical networks", *Opt. Netw. Mag.*, vol. 2, no. 4, 2001.

[2] M. Steinder and A. S. Sethi, "The present and future of event correlation: a need for end-to-end service fault localization", in *World Multi-Conf. Syst., Cyber. Inform.*, N. Callaos *et al.*, Ed., Orlando, USA, 2001.

[3] K. Hashimoto *et al.*, "A new diagnostic method using probabilistic temporal fault models", *IEICE Trans. Inform. Syst.*, no. 3, 2002.

[4] S. Bibas *et al.*, "Alarm driven supervision for telecommunication network: i-off-line scenerios generation", *Ann. Telcommun.*, vol. 51, no. 9–10, pp. 493–500, 1996.

[5] I. Katzela and M. Schwartz, "Schemes for fault identification in communication networks", *IEEE Trans. Netw.*, no. 3(6), 1995.

[6] M. Steinder and S. Sethi, "End-to-end failure diagnosis using belief networks", in *Proc. Netw. Oper. Manag. Symp.*, Florence, Italy, 2002.

[7] Chi-Chun Lo *et al.*, "Coding-based schemes for fault identification in communication networks", *J. Netw. Manag.*, no. 10, 2000.

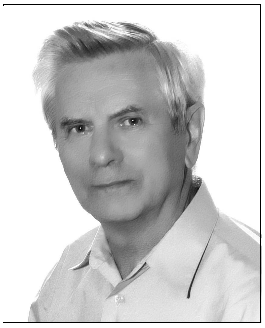
[8] E. U. Ekaette and B. H. Far, "A framework for distributed fault management using intelligent software agents", in *CCEC*, Monteral, Canada, 2003.

[9] H. Masuyama *et al.*, "A diagnosis method of computer networks", http://mylab.tottori-u.ac.jp

[10] M. Brodie *et al.*, "Optimizing probe selection for fault localization", in *Distr. Syst. Oper. Manag.*, Nancy, France, 2001.

[11] M. Brodie *et al.*, "Intelligent probing: a cost-effective approach to fault diagnosis in computer networks", *IBM Syst. J.*, vol. 41, no. 3, 2002.

[12] M. Brodi *et al.*, "Active probing strategies for problem diagnosis in distributed systems", http://research.ibm.com/piple/r/rish



Wiesław Traczyk is a Professor of the National Institute of Telecommunications and also of the Warsaw University of Technology, Institute of Control and Computation Engineering. His research interests include expert systems, approximate reasoning, failures in computer networks and data mining.

e-mail: W.Traczyk@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland
e-mail: traczyk@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland