УДК 004.7

ACTIVE MONITORING OF COMPUTER NETWORKS WITH THE USE OF MULTI-AGENT APPROACH

ALEXEY M. IVANOV[1], BORIS A. ASSANOVICH[2]

[1]LLC "SDL-SOFT"
(Moscow, Russia)

[2]YK Grodno State University
(Grodno, Republic of Belarus)

E-mail: iam@sdl.ru, bas@grsu.by

Abstract. The greedy monitor allocation algorithm for analyzing computer networks based on iterative maximization of path coverage in a network graph was modified and the multi-agent cooperative network monitoring algorithm has been proposed.

Introduction

Communication networks such as wireless sensor networks (WSNs), Internet of Things (IoT) and different ad-hoc networks have been becoming more and more complex nowadays. The growth of the above networks size and the use of a combination of a wide data transmission channels (optical, microwave links, etc.) leads to increase of their analysis complexity and management [1]. When network failures occur, their recovery becomes much more difficult due to the complex nature of modern computer networks. Fault detection is the process of determining that one or more failures may have occurred and fault localization is the process of determining the location of exact failures in the network based on the observed and measurement of traffic parameters. The description of malfunctions, symptoms and methods for their detection has long been studied and is well presented in the survey [2].

Diagnostics of failures in network structures is often divided into two types: passive analysis and active probing. In the first case, passive methods are used to measure a number of traffic parameters in the network, and in the second case, monitoring of both nodes and links of the network structure is implemented. The classification of these fault localization techniques has been provided in [2] and includes several categories, including AI techniques and graph-theoretic techniques. In [3] authors presented a traffic engineering (TE)-based machine learning approach that adopts a passive mechanism to learn the network traffic behavior from its technical parameters (number of flows, propagation delay, etc.), on the basis of which the authors claim that they can localizes link failures with at least 90% accuracy using random forest algorithm. However, judging by the publications, the approach of active network monitoring, which is also called active probing [4], has received much greater use [5].

In this paper, we examined existing methods to the analysis of computer network problems using monitoring. Many publications on this topic prove the relevance of this topic. However, the results found in available works suggest the use of approaches containing improved diagnostics compared to exhaustive search of undirected graph paths based on the use of the greedy algorithm [6] and partitioning the graph into separate structures [7] to place monitors and analyze network topology. In this article, we are inclined to use a distributed agent approach for network diagnostics and propose an algorithm for multi-agent cooperative monitoring of network nodes and links. The remainder of the paper is organized as follows. In Section II we clarify the definitions used in this study and look into previous research within the area of computer network monitoring. Section III contains the description of multi-agent active monitoring approach. In Section IV the multi-agent cooperative network monitoring algorithm is presented. The conclusion of our study is shown in Section V.

Computer Network Structure

First of all, we shall introduce necessary definitions and notification. We model the network structure as an undirected graph *G*(*V*,*E*), where the graph nodes, *V*, denote the network nodes (routers) and the edges, *E*, represent the communication links connecting them. The number of nodes and edges is denoted by |*V*| and |*E*|, respectively. Further, we use *Ps,t* to denote the path traversed by the information packet from a source node *s* to a destination node *t*. We also assume that packets are transmitted using standard forwarding, that is, each node relies on the destination address in the packet to determine the next hop. We shall define by *x* a node that produces route *Px,t* and belongs to path *Ps,t*. Due to the fact that in real networks nodes (routers) are often numbered in some numerical way, for specific network structure we will assign fixed integer values $i \in N$ for *s* or *t*-node and denote them as No=i or Node_i.

We associate a positive cost *cs,t* with sending a message along the path *Ps,t* between any pair of nodes *s*, $t \in V$. For every intermediate node $x \in Ps,t$ both *cs,x* and *cx,t* are at most *cs,t* and *cs,x* +*cx,t* ≥ *cs,t*. Usually the message cost includes some timing (round trip time RTT) or statistical information for all hops defined by *hs,t* in path *Ps,t*.

Further, it is obvious that each path includes a set of links *fj*, between nodes *s* and *x*, which are also indexed by some integers $j \in N$, and for simplicity, we will mark them in the figure only through the *j*-index. In case a link *fj*, in the network fails, the routing protocol defines a new delivery tree, *Tj* , for every node $s \in V$. The tree *Tj* has the property that every path *Ps,t* in it, that does not contain link *fj* is also included in the tree *Tj* . The reason for this is that the failure of link *fj* only affects those routing paths in *Ts* that contain *fj*. Thus, we can infer the topology of a significant portion of *Tj* directly from *Ts* without any knowledge of the route computation algorithms.

Further, we shall analyze the placement of monitors to analyze network problems. At the same time, we will use the Controllable Arbitrary-path Probing (CAP) mechanism [7,8].

In complex computer networks we need a different approach for choosing the location of monitors and an appropriate network monitoring scenario, which will be discussed in the next section.

Multi-Agent Active Monitoring of Networks

The need for accurate and fast network monitoring methods has increased in recent years due to the complexity of networks and the need for quick troubleshooting. The term *network tomography* was introduced by Vardi [9] to encompass these class of approaches including the Path-oriented approach, which is interests us in this paper.

Obviously, we can relate the network paths *Ps,t* to the node states through the Boolean linear system [10].

The goal of Boolean network tomography is to invert this Boolean linear system to solve for *w* given *R* and *c*. Actually, node failures are identifiable if and only if (1) has a unique solution. Unfortunately, the number of accessible nodes is much smaller than number of links inside the network.

However, in real systems it is rather difficult to analyze a computer network and find the optimal solution. Therefore, in practice, often use suboptimal approaches with some restrictions and use so-called Greedy Partial Multi-Set Cover Algorithm [6] and Minimum Monitor Placement Algorithm [7]. However, in the considered works, the interaction of monitors with each other is not mentioned and the monitoring is performed by sending data to the network operations center (NOC). In this work, we will assume a multi-agent approach for monitoring a computer network, in which this task is solved in a distributed manner using agent-based approach.

Consider a more complex segment of an existing computer network, shown in Fig. 1.

It corresponds to an undirected graph with the following parameters |*V*| = 17, |*E*| = 30. To solve the problem of placing monitors, we will use the analog of the greedy algorithm [6] in terms of maximizing the *route coverage* (RC) (additive values of link metrics due to the integrated consideration of new nodes put into the topology) but in iterative manner that is discussed below.

To do this, first perform an analysis of the network topology and construct a Path Table of nodes *xi* ad links *fj*, in the paths *Ps,t*, which will contain information about the minimum required number of routes *Px,t* to bypass all network nodes (marked as Node_i in Fig.1). This can be done by virtual placing the monitor in each node $s \in V$ and finding a maximum number of hop-to-hops *hs,t* in the routes *Px,t* that relate to nodes, as well as the calculation of the RC for the selected monitor.

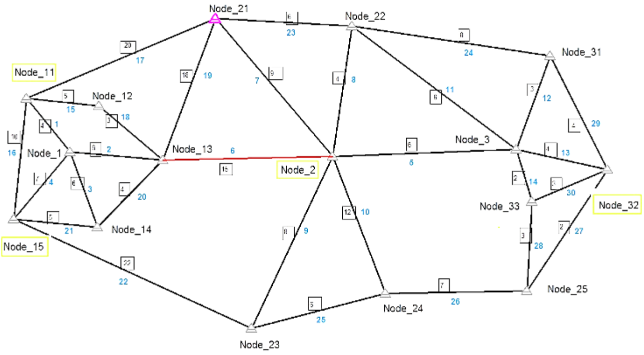

Fig.1. Topology of a computer network

Next, we find the node through which the minimum number of routs *Px,t* passes and select it as a monitor. Then we sort in descending order the candidates to become monitors by finding the RC of the remaining nodes by corresponding links for the selected monitor. We choose a node that is at the maximum distance from the first selected monitor and assign it to the next monitor. Observing the network topology we can visually verify the maximum spatial diversity of monitors in spatial structure of the network. We perform the next step in choosing monitors, taking into account the calculation of maximum coverage and maximum distance from previous monitors. Hence, we complete the algorithm when full network coverage and control of all links for all routes are completed.

Fig. 1 shows an example of a real network topology, where node No=11, through which a minimum number of routes passes, is selected as a monitor. The maximum number of remaining uncovered routes is RC=8. Then, after arranging the remaining candidates for coverage and distance from the previous monitor, No=32 is selected. In the next step, node No=2 is selected to place the monitor. Further, we see that two links ([4],[19]) remain uncovered. To increase RC, we can choose either placement of monitor in node No=1, or No=15. In reality, we can limit ourselves to the choice of three monitors considering the fact that other links in the routes during communications are not so critical to the network of failures.

Suppose we decide to completely control the network and select for the monitor node 15. Each selected node includes four links, which provides sufficient monitoring reliability in case of failure of one of the links. Thus, as a result of the algorithm, the following monitors (Mo): No=11 No=32, No=2, No=15 have been selected.

Agents Interaction Algorithm

In this section we shall present the interaction algorithm of agents-monitors (hereinafter referred to as *agents*) in detecting failures of nodes and links in a network. Agents work in standby and emergency mode.

After calculating the location of agents, we can draw up two general monitoring plans for all agents that they will use in their action: node monitoring plan (NMP) and link monitoring plan (LMP). NMP, LMP for a computer network from Fig.1 presented in Table 1 and Table 2 below. Actually these plans define the interaction algorithm to perform the network monitoring.

Each of the agents uses a column of the node table in standby mode. Choosing the next (highlighted by green in Table 1) node, it transmits a probe signal and, step by step, checks the route (path) obtained in control procedure with the one indicated in the plan. If the node does not respond at all, then we fix the probable failure of the node. If it was possible to reach the node step by step, but the route does not coincide, then we can define a hypothesis about the failure of one of the links in the route.

To evaluate the node functioning, we check in NMP the availability of routes to this node from the other monitors (we scan along the line of the plan corresponding to the failed node (see Table 1). If such routes exist, then we can a request for testing the node using the appropriate agent. To do this, we first send the probe signal to the Node where the agent is located in order to verify its operability. If such paths exist, then we can issue a request for testing the node using the appropriate agent. To perform it, we first send the probe to the point where the agent is located in order to verify its operability. And after a positive answer, we send him a task for test. If all monitors received a negative response when probing, we go to escalation of node failure.

Suppose in the next scan cycle node No = 2 using route 2 [7] 21, where 2 is the number of the outgoing node, 7 is the number of link, and 21 is the target node, a negative signal probe is received. In this case, agent No=2 sends test signals to agents capable of checking node No= 21. These are agents and corresponding messages:

- Agent_11 (2 [6] 13 [18] 12 [15] 11);
- Agent_32 (2 [5] 3 [13] 32);
- Agent_15 (2 [6] 13 [20] 14 [21] 15).

In response to agent availability, each of them is given a request to probe node No=21:

- Agent_11 (11 [17] 21);
- Agent_32 (32 [13] 3 [11] 22 [23] 21);
- Agent_15 (15 [21] 14 [20] 13 [19] 21).

If positive results are obtained from at least one agent, the hypothesis is considered incorrect and the agent proceeds to localize faulty links.

The above described algorithm can formally be written as a program script. We shall define it as Multi-agent cooperative monitoring (MACM).

The hypothesis test of a link failure is realized in a similar way. However, the LMP (see Table 2) is used as information to make a decision.

For example, we assume that in the process of scanning with node No=2 (2 [6] 13), link 6 became unavailable. To perform it, in the plan we find agents that can identify this failure and, as indicated above, after checking the availability of agents, we send request message to agents:

- Agent_11 (11 [15] 12 [18] 13 [6] 2);
- Agent_15 (15 [21] 14 [20] 13 [6] 2).

If a negative response came from two of these monitors when sensing, we take steps to escalation of link failure.

Conclusion

Thus, in this paper the known methods for finding network failures have been studied and a modification of the greedy monitor allocation algorithm for analyzing modern computer networks was iteratively modified, and the multi-agent cooperative network monitoring algorithm has been proposed. Further development of our multi-agent approach involves the correction of NMP and LNP plans in accordance with the current state of the network, which is performed by each monitor individually.

References

1. C. Savaglio, M. Ganzha, M. Paprzycki, C. Badica, M. Ivanovic, and G. Fortino,"Agent-based Internet of Things: State-of-the-art and research challenges", Future Gener. Comput. Syst., vol. 102, 2020, pp.1038-1053.
2. M. Steinder and A. S. Sethi, "A survey of fault localization techniques in computer networks," Sci. Comput. Program., vol. 53, no. 2, Nov. 2004. pp. 165-194.
3. M. Srinikethan, T. Truong-Huu, M. Gurusamy, "TE-Based Machine Learning Techniques for Link Fault Localization in Complex Networks", 6th Int. IEEE Conf. on Future IoT and CloudFiCloud 2018, Barcelona, Spain, August 6-8, 2018, pp.25–32.
4. M. Brodie, I. Rish, S. Ma, G. Grabarnik, and N. Odintsova, "Active probing". Technical report, IBM, 2002.
5. A. Dusia and A. S. Sethi, "Recent Advances in Fault Localization in Computer Networks," Com.. Surveys Tuts., vol. 18, no. 4, pp. 3030–3051, May 2016.
6. Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," IEEE INFOCOM, 2003.
7. L. Ma, T. He, K. K. Leung, A. Swami and D. Towsley, "Inferring Link Metrics From End-To-End Path Measurements: Identifiability and Monitor Placement," IEEE/ACM Transactions on Networking, vol. 22, no. 4, pp. 1351-1368, Aug. 2014.
8. L. Ma, T. He, A. Swami, D. Towsley and K. K. Leung, "Network Capability in Localizing Node Failures via End-to-End Path Measurements," *IEEE/ACM Trans. on Networking*, vol. 25, no. 1, pp. 434-450, Feb. 2017.
9. Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," J. of the American Statistical Association vol. 91, no. 433, 1996.
10. R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," Statistical Science, vol. 19, no.3, pp. 499–517, 2004.