Peer-to-Peer Netw. Appl. (2011) 4:439–452
DOI 10.1007/s12083-010-0100-4

Probabilistic and reactive fault diagnosis for dynamic overlay networks

Yongning Tang · Guang Cheng · Zhiwei Xu

Received: 8 August 2009 / Accepted: 27 December 2010 / Published online: 26 January 2011

Abstract Overlay networks have emerged as a powerful and flexible platform for developing new disruptive network applications. The attractive characteristics of overlay networks such as routing flexibility and overlay topology dynamics bring to overlay fault diagnosis new challenges, which include the dynamical overlay symptom-fault correlation, multi-layer (i.e., underlay vs. overlay) abstraction, and unregulated overlay symptoms. To address these challenges, we propose a novel user-level probabilistic and reactive fault diagnosis technique, called *ProFis* for overlay networks, which can seamlessly integrate passive and active fault reasoning to develop an optimal fault diagnosis framework. *ProFis* uses observable overlay symptoms as reported by overlay applications to dynamically correlate overlay symptoms and faults. *ProFis* diagnoses overlay faults passively and selects optimal actions (i.e., with the least cost) to enhance the passive diagnosis whenever necessary. Our evaluation study shows that *ProFis* can efficiently (i.e., low latency) and accurately localize the root causes of overlay faults, even when symptom loss rate is high.

Keywords Fault diagnosis · Probabilistic reasoning · Reactive probing · Overlay networks

Y. Tang (✉)
School of Information Technology, Illinois State University, Normal, IL 61790, USA
e-mail: ytang@ilstu.edu

G. Cheng
School of Computer Science and Engineering and the Ministry of Education Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing, People's Republic of China
e-mail: gcheng@njnet.edu.cn

Z. Xu
Computer and Information Science Department, University of Michigan-Dearborn, Dearborn, MI, USA
e-mail: zwxu@umich.edu

1 Introduction

Overlay networks [22, 23] have emerged as a powerful and flexible platform for developing new disruptive network applications [1]. With more and more overlay networks deployed, overlay fault management is playing a crucial role in successfully provisioning overlay services. Fault diagnosis is the most critical component in a fault management system because it identifies the root causes (i.e., faults) that can best explain observed network disorders (i.e., symptoms) [27, 28]. For instance, in an overlay multicast application [7] as shown in Fig. 1, a multicast forwarding tree is created in which n_0 is the source forwarding network traffic via n_1 to n_2, n_3 and n_4. Intuitively, if network performance degradation (e.g., high packet loss or latency) is observed from both n_2 and n_3 but not from n_4, then it is more likely that AS_b (i.e., Autonomous System) rather than n_1 is the root cause. The simple example clearly shows that the symptom-fault correlation based on passively observed symptoms may lead to an accurate root cause analysis without requiring active probing. However, if the observation is insufficient (e.g., the observation from n_3 is missing), it may significantly reduce the fault analysis accuracy and even mislead to a wrong conclusion.

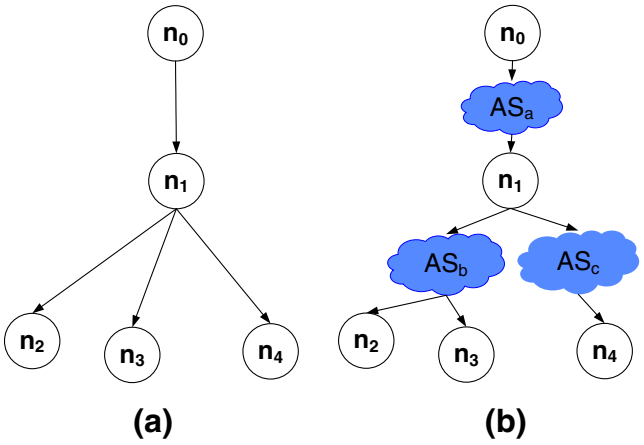

Fig. 1 Overlay multicast application: **a** multicast forwarding tree on the overlay network layer; **b** multicast forwarding tree on the underlying network layer

Passive diagnosis [4, 16, 27, 28] and active probing [2, 5, 11, 24] are two commonly used fault reasoning and localization approaches. In the passive approach, all symptoms are passively collected and processed to infer faults. In the active approach, faults are detected by conducting a set of probing actions. The passive approach is less intrusive but may incur long processing latency especially in a large-scale distributed network system (e.g., overlay applications), and less accuracy when observed symptoms are insufficient. On the other hand, the active probing approach is more efficient to pinpoint faults, but it may be highly intrusive to the monitored networks. Both the existing passive and active fault localization approaches are insufficient for overlay networks because of the following facts:

- System Scalability: An overlay network typically possesses a large number of nodes (e.g., end hosts). Even adopting certain optimization techniques (e.g., [35]), either massive network probing or a large number of network monitors are required.
- Fault unpredictability: A flexible and dynamic overlay network infrastructure (e.g., service node selection, topology changing) makes the interaction and correlation between overlay and underlying networks unpredictable.
- Symptoms irregularity: With unpredictable faults, it is impractical to define a static causal relationship between faults and symptoms. In overlay networks, symptoms cannot be designed for monitoring specific faults.

In this paper, we propose a probabilistic and reactive fault diagnosis technique called *ProFis* for overlay networks. *ProFis* can incrementally and interactively conduct fault diagnosis for both overlay and underlay network layers. Moreover, *ProFis* can seamlessly integrate passive and active monitoring techniques into one framework, where *ProFis* first analyzes the observed overlay symptoms on the overlay layer for a coarse-grained diagnosis to rapidly narrow down the fault search scope in a large-scale overlay network. Then it conducts a passive heuristic fault reasoning with finer granularity to pinpoint the root faults that highly likely occurred. Since the passive diagnosis may not be sufficient (e.g., due to missing symptoms), a set of optimally selected probing actions are conducted to collect the most relevant but unobserved symptoms. The feedback from active probing can be incrementally integrated with the previous reasoning results until the confidence (i.e., fidelity as explained in Section 4.3) on the diagnosis results is satisfactory. Our main contribution in this paper is to propose a probabilistic overlay fault diagnosis framework that can passively analyze observed overlay symptoms without using any monitoring infrastructure. When the passive diagnosis cannot be satisfactory, the system can reactively select a set of optimal actions to enhance the previous diagnosis. Unlike most of the existing fault diagnosis approaches, *ProFis* does not assume any static symptom-fault correlation graph. Instead, such a correlation graph can be dynamically created based on the current overlay application topology.

ProFis consists of four modules: Symptom Mining (SM), Fault Reasoning (FR), Fidelity Evaluation (FE), and Action Selection (AS). The symptom mining module discovers the inherent correlations among observed symptoms fed by various independent overlay applications to create a Dynamic Fault Reasoning Graph (DFRG). The fault reasoning module analyzes the observed symptoms based on DFRG to generate a fault hypothesis that may include multiple faults to explain the observed symptoms. The fault hypothesis is then sent to the fidelity evaluation module to check if the reasoning result is satisfactory (i.e., high fidelity). The action selection module finds a set of optimal actions (i.e., with least cost) to verify the existence of symptoms.

The rest of paper is organized as follows. In Section 2, the related work is discussed. In Section 3, we formalize the problems and give an overview on the *ProFis* system. In Section 4, we elaborate the four modules in the *ProFis* system. In Section 5, we present the system evaluation on the performance and accuracy of the *ProFis* system. Finally, we conclude our work in Section 6.

2 Related work

There is a significant amount of work in the literature in the area of fault localization and diagnosis. In our related work study, we focus on user-level network fault diagnosis tools/approaches, particularly those for overlay networks. We classify the user-level fault diagnosis related work into the following categories:

Passive approach Various passive monitoring and event correlation models have been proposed including rule-based analyzing systems [17], model-based systems [15], case-based diagnosing systems and model traversing techniques. In [4], a model-based event correlation engine is designed for analyzing the dependency across multiple protocol layers to locate faults. In [16], a coding approach is used to correlate a predefined symptom-fault causality to reduce the reasoning time and improve system resilience. An incremental event-driven fault reasoning technique is presented in [27] and [28] to improve the robustness of fault localization system by analyzing lost, positive and spurious symptoms. Unfortunately, all these techniques, in which a static symptom-fault correlation graph is assumed for fault reasoning, are inapplicable to overlay fault diagnoses.

Diagnosis tools Many end-to-end network probing tools have been designed for monitoring packet loss and other network path properties for network fault diagnosis [3, 12, 13, 19, 25]. These tools are effective in diagnosing a specific network property by a single end-user, but not adequate as a general fault diagnosis solution for large-scale distributed overlay networks. Some researchers also proposed to incorporate active probing into different fault diagnosis systems. In [5], an active probing fault localization system is presented, in which pre-planned probing actions are associated with the monitored system status via a dependency graph. An on-line action selection algorithm was studied in [24] to optimize action selection. In [11], a fault detection and resolution system was proposed for a large distributed transaction processing system. However, these active probing based techniques are not capable of utilizing passive fault analysis results, and thus may still incur unnecessary intrusiveness to the monitored systems even with optimal action selection. Moreover, periodic active probing may not discover intermittent problems.

Diagnosis framework Several network tomography-based approaches have been proposed to estimate network performance based on traffic measurement results from a minimum subset of the system nodes [6, 8, 35]. However, this type of fault diagnosis approach essentially belongs to an active approach and may still cause significant intrusiveness in a large overlay network. For example, the monitoring task for 51 overlay nodes as presented in [6] still requires approximately 76.5 K $\times$ 40 B/3 $\simeq$ 10 MB probing traffic sent three to four times every hour. Such monitoring intrusiveness may not be suitable for a large Overlay Service Provider such as Akamai [1], which has more than 25,000 deployed overlay nodes. An active and passive integrated system called PlanetSeer [34] has also been proposed, which can locate network faults by selectively and periodically invoking "traceroute" from multiple vantage points. However, this system has to be manually managed to monitor specific parameters (e.g., TTL change) and only can match the application domain direction of data flow.

To the best of our knowledge, *ProFis* is the first fault diagnosis framework that integrates active monitoring with passive fault reasoning based on a dynamically generated symptom-fault correlation graph.

3 Problem formalization and system overview

The current Internet consists of tens of thousands of autonomous systems. Each autonomous system is a collection of networks owned by a single authority. Among these autonomous systems, some of them are owned by different Internet Service Providers (i.e., ISP) to provide packets forwarding services to their customers. The others belong to different Internet users such as universities and companies. An overlay network is a logical network built on top of another network (e.g., the Internet), where overlay nodes (e.g., end-hosts, servers) are connected by logical links, each of which corresponds to a path, typically through many physical network links in the underlying network.

A generic overlay network infrastructure can be represented by a graph $G =< V, E >$, where V is the set of overlay (V^o) and underlay (V^u) nodes (i.e., $V = V^o \cup V^u$), and E is the set of overlay (E^o) and underlay (E^u) links (i.e., $E = E^o \cup E^u$). We assume every overlay node can be virtualized as multiple slices (e.g., the nodes in PlanetLab [23]) for hosting multiple overlay applications simultaneously. One slice of each overlay node is designated as an administrative slice that is controlled by an Overlay Network Operation Center (*OvNOC*). An overlay link e_i^o ($e_i^o \in E^o$) may span multiple underlay links $e_i^o = \{e_1^u, e_2^u, \cdots, e_N^u\}$ ($N \geq 1$). Multiple overlay applications can run independently and simultaneously on G.

In our framework, when an overlay application observes any problem (e.g., unreachability or performance degradation), it sends to the OvNOC a report that includes the symptom S_i (e.g., high packet loss or latency) and the unique path identifier (UPI) of the corresponding overlay path. An UPI consists of a sequence of unique component identifiers (UCI), which is a hash value (e.g., using MD5) representing a series of connected components (e.g., routers or end hosts) that share a common network prefix and autonomous system number (ASN) in underlay networks. Formally, we define an identifiable component's UCI as $C_i = H(p_i, a_i, o_i)$, where p_i, a_i and o_i denote the common network prefix, the autonomous system number and the overlay flag of each component i in an overlay path. We use the method as described in [18] to find the corresponding autonomous system number of the intermediate nodes (i.e., routers) in the overlay path. There are several other solutions that can provide highly accurate IP to ASN mapping results as well. For example, the *whois* web service offered by [32]. Mapping the components' IP addresses to their corresponding ASNs provides a logic clustering on the overlay and underlay components. However, it is not a prerequisite for our system to work.

Since these symptoms are observed by overlay applications, we call them overlay symptoms (denoted as S^o). However, an ISP might also participate in the *ProFis* diagnosis framework and forward to the OvNOC the symptoms (called underlay symptoms denoted as S^u) directly observed on the underlay components (e.g., routers). As shown in Fig. 2, the task of an OvNOC is to correlate the observed symptoms and identify their root causes. When the received symptoms

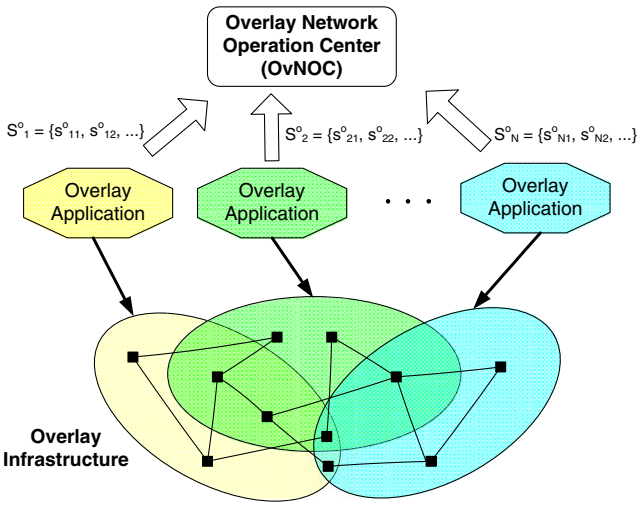

Fig. 2 System infrastructure

are insufficient to pinpoint faults, the OvNOC will selectively conduct a set of active probing to diagnose the problem. The probing actions (denoted as A), such as retrieving MIB values [20] or active probing [3, 12, 19, 25], can be conducted at any overlay nodes or between a pair of overlay nodes, respectively. We use s^o_{ij} to represent a symptom j from an overlay application i, which can be represented as the UCIs of a set of overlay C^o_j ($C^o_j \in V^o$) and underlay C^u_k ($C^u_k \in V^u$) components.

A symptom could be either positive or negative. A negative symptom s^o_{ij} indicates a network disorder that involves at least one of identifiable components (overlay or underlay). The negative symptoms can be generated as alarms from overlay applications or from probing action results. However, the positive symptoms $\overline{s^o_i}$ can only be resulted from probing actions to indicate the status of all identifiable components in a diagnosed path. There are two types of actions: overlay actions A^o, which are end-to-end network measurements covering a sequence of identifiable components along the same overlay path, and underlay actions A^u, which is application-specific for verifying the status of an individual underlay component [3, 12, 19, 25]. Each action A_i has an estimated cost w_i. There are different ways to choose the action cost. Here, we provide an intuitive idea on how to estimate the cost in an overlay environment. The action cost is a function of *benefit* minus the *overhead*. The *benefit* of an action is determined by the criticality and impact of an associated fault. The criticality reflects the severity of this fault on business value and SLA (i.e., service level agreement) that could be estimated from previous history or business policies. The fault impact is used to estimate the magnitude of the damage caused by this fault (e.g., like number of users/customers has been effected by this fault). The overhead of an action is estimated in term of bandwidth, delay (number of overlay component and underlay links), and labor cost. These factors can be converted to real cost (i.e., monetary values) based on business practices and policies, and then aggregated to calculate the total cost of an action. In our approach, the total cost is assumed as a weighted sum of number of normalized cost factors as explained.

An overlay fault diagnosis problem is defined as the following: given observed symptoms S_O, including overlay symptoms S^o generated from overlay applications and underlay symptoms S^u from ISPs if available (i.e., $S_O = S^o \cup S^u$), (1) to find a set of identifiable components that can best explain S^o and S^u with a satisfactory fidelity value, and (2) if necessary, to choose a set of optimal overlay (A^o) and/or underlay (A^u) probing actions to improve the passive root cause analysis. Our main objective in this work is to improve fault

reasoning by minimizing the detection time T and the false positive ratio β, and maximizing the detection ratio α.

The probabilistic overlay fault diagnosis framework (i.e., *ProFis*) as shown in Fig. 3 includes four functional modules: Symptom Mining (SM), Fault Reasoning (FR), Fidelity Evaluation (FE), and Action Selection (AS). The symptom mining module takes the observed overlay symptoms S^o to create a Dynamic Fault Reasoning Graph ($DFRG$). The fault reasoning module takes $DFRG$ as the input and returns a set of fault hypotheses Φ as the output. Φ might include multiple hypotheses $(h_1, h_2, \ldots, h_n)$, where each one contains a set of faulty components that can jointly explain all observed symptoms. Then, Φ is sent to the fidelity evaluation module to check if any hypothesis h_i ($h_i \in \Phi$) is satisfactory (i.e., with high fidelity). If the most correlated symptoms necessary to explain a hypothesis h_i are observed, then a satisfactory fidelity can be achieved and the fault reasoning process can terminate. Otherwise, a set of investigated components C_N that contribute to explain the hypothesis h_i is created and sent to the action selection module to verify. The conducted actions may return a result with a set of new observed overlay symptoms (i.e., $\overline{S^o_A}$ and/or S^o_A), respectively. The corresponding fidelity value might be

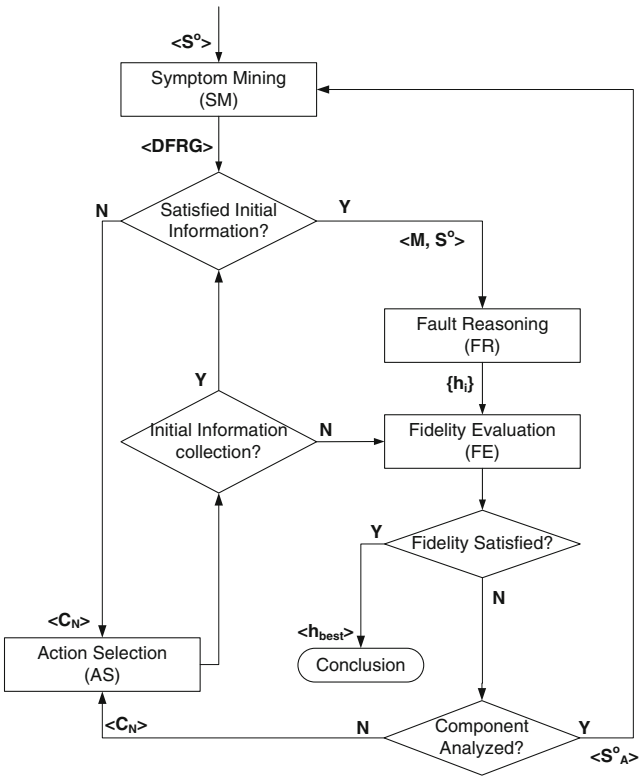

Fig. 3 Active overlay fault reasoning

increased or decreased based on the action results. If the newly calculated fidelity is satisfactory, then the reasoning process terminates. Otherwise, $\overline{S^o_A}$ and S^o_A are sent as the new input to the symptom mining module to update DFRG so as to create new hypotheses. This process is repeated until a hypothesis with a high fidelity is found. The fidelity calculation process will be explained in the next section. In the following, we first describe the four modules of *ProFis* in detail.

4 Probabilistic and reactive overlay fault diagnosis

4.1 Symptom mining

Overlay applications are logically constructed on the top of the Internet. Thus the correlation between overlay and underlay components depends on overlay applications and may be varying over time. Observed symptoms could be either completely irrelevant (i.e., related to different components) or closely related. Thus, the objective of the symptom mining module is to discover the logical (e.g., spatially, temporally) relationship among the observed overlay symptoms and create a symptom-fault correlation graph (i.e., DFRG).

As discussed in Section 3, an overlay symptom s^o_{ij} is represented by a sequence of components as $\{C_1, C_2, \cdots, C_n\}$ consisting of both overlay (e.g., end hosts, servers) and underlay components (e.g., routers). We define *Maximum Identifiable Component* (MIC) as the longest common UCI [29–31] shared by the maximum number of overlay symptoms. We calculate two types of MICs: overlay-MIC (denoted as o-MIC) and underlay MIC (denoted as u-MIC). For example, as shown in Fig. 1, we define symptoms with the following UPI as:

$$\begin{aligned} UPI(s_1) &= \{C^o_0, C^o_1, C^o_2\} \cup \{C^u_a, C^u_b\}, \\ UPI(s_2) &= \{C^o_0, C^o_1, C^o_3\} \cup \{C^u_b\}, \\ UPI(s_3) &= \{C^o_0, C^o_1, C^o_4\} \cup \{C^u_a, C^u_c\}, \\ MIC(s_1, s_2, s_3) &= \{(C^o_0, C^o_1); (C^u_a); (C^u_b); (C^o_2); (C^o_3); (C^u_c)\} \\ &= \{m^o_1, m^u_2, m^u_3, m^o_4, m^o_5, m^u_6\}. \end{aligned}$$

In the rest of the paper, we denote M_{s_i} as the set of MICs related to s_i. Thus, all symptoms can be represented using MIC as the following:

$$\begin{aligned} M_{s_1} &= \{m^o_1, m^u_1, m^u_2, m^o_3\}, \\ M_{s_2} &= \{m^o_1, m^u_1, m^u_2, m^o_4\}, \\ M_{s_3} &= \{m^o_1, m^u_1, m^u_5, m^o_6\}. \end{aligned}$$

In the following of the paper, we use s_i to represent an overlay symptom for simplicity. The probability $p(m_i|s_i)$ provides critical information on the likelihood measure that m_i is faulty when observing the symptom s_i. We assume the occurrence of any given symptom s_i is triggered by at least one faulty component m_k ($m_k \in M_{s_i}$). For a given m_i, $p(m_i|s_i)$ can be computed by Eq. 1. Item 1 as marked in Eq. 1 represents the probability of at least one component m_q ($m_q \in (M_{s_i} - m_i)$) is faulty. Item 2 is the probability of all components in addition to m_i is faulty. Item 3 shows the probability of all components except m_i are not faulty. By adding the three items together, it gives the probability of all possible scenarios of M_{s_i}. Thus, Eq. 1 provides the probability of all possible scenarios of all components in the s_i related component set M_{s_i} except m_i, while m_i occurs.

$$p(m_i|s_i) = p(m_i)\left[\sum_{h=1}^{|M_{s_i}|-1}\overbrace{\left\{\prod_{q=1,q\neq i}^{h} p(m_q)\prod_{k=h+1,k\neq i}^{|M_{s_i}|}(1-p(m_k))\right\}}^{\text{Item 1}} + \underbrace{\prod_{q=1,q\neq i}^{|M_{s_i}|} p(m_q)}_{\text{Item 2}} + \underbrace{\prod_{q=1,q\neq i}^{|M_{s_i}|}(1-p(m_q))}_{\text{Item 3}}\right] \quad (1)$$

The probability of m_i (i.e. $p(m_i)$) indicates the inherent likelihood that m_i becomes faulty. It is very challenging to obtain this parameter in a uncontrollable overlay environment. For an overlay component m_o, the practical approach we took in our experiments used the uptime of an overlay component collected by the CoMon system [9] to estimate the reliability of the overlay component. We empirically used 30 days uptime as the baseline for the overlay components in PlanetLab [23], and calculate the relative uptime of a component to estimate its reliability. For instance, if an overlay node m_o keeps on-line for more than one month, we set $p(m_o) = 0$. If its uptime is only 10 days, $p(m_o) = 1 - 10/30 = 66.7\%$. For any involved underlay component m_u, we initially assign a small prior fault probability (i.e., $p_0(m_u) = 0.1\%$). Then, a reward function $p(m_u) = p_0(m_u) + k\delta$ was used to adjust $p(m_u)$ based on previous related fault reasoning result if existed (Fig. 4). In our experiments, we set $\delta = 0.01\%$. $k = 1$ if a positive observation on m_u is obtained; $k = -1$ if a negative observation on m_u is obtained; otherwise, $k = 0$. We also set $p(m_u) \in [0, 15\%]$.

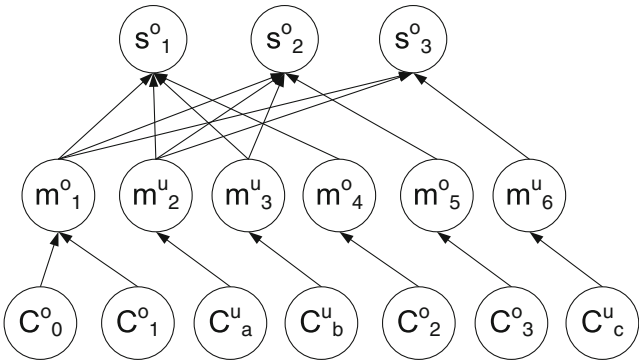

Fig. 4 Dynamic fault reasoning graph based on the example shown in Fig. 1

Reward function used about actually represents a learning process for *ProFis*. Supervised or unsupervised machine learning methods might be considered to improve the accuracy of the reward function, which we mark as the future work to extend *ProFis*.

Based on the relationship between m_i and s_i, Dynamic Fault Reasoning Graph (DFRG) can be constructed as the following:

- For every overlay or underlay maximum identifiable component (i.e., an MIC), associate a vertex m_i^o or m_i^u, respectively. For every overlay symptom, associate a vertex s_j^o. For every overlay or underlay identifiable component, associate a vertex C_i^o or C_i^u, respectively.
- For all m_i^o or m_k^u and their associated symptoms s_j^o, associate an edge e_{ij} or e_{kj} with a weight equals to $p(m_i|s_i)$ or $p(m_k|s_i)$, respectively.
- For all C_i^o (or C_k^u) and their associated MIC component m_j^o (or m_q^u), associate an edge e_{ij} (or e_{kq}).

4.2 Overlay fault reasoning

Given a DFRG, the next task is to find the most likely root causes that can best explain all observed symptoms. There is one fundamental difference between DFRG and a traditional symptom-fault casuality graph (SFCG). In SFCG based on both a deterministic and non-deterministic causality models, for a given monitored fault f_i, there is a set of observable symptoms associated (denoted as S_{f_i}). In other words, the strongest indication if f_i occurs is known. However in DFRG, it is infeasible or at least impractical to pre-define a set of correlated symptoms for each given network component due to the overlay flexibility and scalability. In order to tackle this challenge, we introduce a new parameter called *Symptom Contribution* (SC) that can

be used to evaluate the correlation likelihood between m_i and a set of observed symptoms S_{m_i}.

$$SC(m_i) = \frac{1 - \prod_{s_i \in S_{m_i}} (1 - p(m_i|s_i))}{|m_i|} \quad (2)$$

Symptom Contribution evaluates the possibility that we can diagnose faults with the given symptoms. The higher $SC(m_i)$ is, the more likely m_i is faulty if S_{m_i} observed. Since in *DFRG*, it is impossible to create a pre-determined set of symptoms S_{m_i} associated with a dynamically generated m_i. The value of *SC* can arbitrarily vary in $(\epsilon, 1)$ $(0 < \epsilon \ll 1)$. Achieving credible fault reasoning requires observing sufficient symptoms. Thus we introduce a threshold called Minimal Initial Information or MII. When $\max_{m_i \in M}[SC(m_i)] < MII$, we selectively conduct probing actions to increase $\max_{m_i \in M}[SC(m_i)]$ to meet the minimal reasoning requirement MII. In practice, we can set MII to a relatively low value to balance between analysis accuracy and network intrusiveness. In the following, we will discuss the corresponding algorithm for addressing the above two scenarios: (1) overlay fault reasoning when $\max_{m_i \in M}[SC(m_i)] \geq MII$; (2) active action selection when $\max_{m_i \in M}[SC(m_i)] < MII$.

A hypothesis may consist of o-MICs, u-MICs, or both of them. In general, the action cost to verify the status of an o-MIC is usually less than the cost needed for a u-MIC due to different information accessibility. Moreover, the prior fault probability of an overlay component is usually higher than an underlay component due to the different system manageability. In *ProFis*, we use two different credibility thresholds MII^o and MII^u ($MII^u = (1+\alpha)MII^o$ and $\alpha > 0$) such that an o-MIC has higher priority to be included into a hypothesis rather than a u-MIC. In other words, *ProFis* prefers to explain all symptoms by using o-MIC unless there is a strong indication for a u-MIC. Thus, the overlay fault reasoning essentially is a process of searching for MIC (m_i) with the maximum $SC(m_i)$. This process continues until all observed symptoms can be explained.

The overlay fault reasoning algorithm (i.e., Algorithm 1) initially finds a set of candidates of faulty components (i.e., $M^o_{\max}$ or $M^u_{\max}$) that may contain multiple overlay or underlay components with the same maximal *Symptom Contribution* (lines 1 and 2). The underlay components are only considered if their corresponding *Symptom Contribution* is $(1+\alpha)$ times over overlay components' (line 3). Based on the corresponding symptom contribution, the most likely component m^o_i or m^u_j is included into the hypothesis h_i and the corresponding set of explained symptoms S_{m_i} are removed from S_K. If multiple components show the same contribution, then multiple hypotheses are generated (lines 4–16). The searching process (OFR) will recursively run until all observed symptoms explained (lines 18–24). Notice that only the hypotheses using a minimum number of faults to explain all observed symptoms are included into Φ (lines 23 and 24).

Algorithm 1 Overlay Fault Reasoning Algorithm **OFR(**h, S_K, M^o, M^u**)**

Input: hypothesis h, observed but uncovered overlay symptom set S_K, overlay fault candidate set M_o, underlay fault candidate set M_u
Output: fault hypothesis set Φ

1: $M^o_{\max} \leftarrow \max_{m_i \in M^o}\{SC(m_i)\}$
2: $M^u_{\max} \leftarrow \max_{m_i \in M^u}\{SC(m_i)\}$
3: **if** $SC(m^u_i) > (1+\alpha)SC(m^o_j)$ & $m^u_i \in M^u_{\max}, m^o_j \in M^o_{\max}$ **then**
4: **for all** $m^u_i \in M^u_{\max}$ **do**
5: $h_i \leftarrow h \bigcup \{m^u_i\}$
6: $S_{K_i} \leftarrow S_K - S_{m^u_i}$
7: $M^u_i \leftarrow M^u - \{m^u_i\}$
8: $M^o_i \leftarrow M^o$
9: **end for**
10: **else**
11: **for all** $m^o_j \in M^o_{\max}$ **do**
12: $h_i \leftarrow h \bigcup \{m^o_j\}$
13: $S_{K_i} \leftarrow S_K - S_{m^o_j}$
14: $M^o_i \leftarrow M^o - \{m^o_j\}$
15: $M^u_i \leftarrow M^u$
16: **end for**
17: **end if**
18: **for all** S_{K_i} **do**
19: **if** $S_{K_i} = \emptyset$ **then**
20: $\Phi \leftarrow \Phi \cup \{h_i\}$
21: **end if**
22: **end for**
23: **if** $\Phi \neq \emptyset$ **then**
24: return $< \Phi >$
25: **else**
26: **for all** h_i **do**
27: $OFR(h_i, S_{K_i}, M^o_i, M^u_i)$
28: **end for**
29: **end if**

4.3 Fidelity evaluation

The fault hypotheses created using Algorithm 1 may not accurately determine the faults because of incomplete symptom observation. The task of the fidelity evaluation is to measure the fidelity of the hypothesis created in the passive reasoning phase. In order to

objectively evaluate the reasoning result, we design a fidelity function $FD(h)$ as Eq. 3 to measure the fidelity of hypothesis h given S^o, which essentially measures the likelihood of hypothesis h when S^o observed. We assume that the occurrence of each faulty component is independent.

$$FD(h) = \prod_{m_i \in h} \left(1 - \prod_{s_i \in S^o} (1 - p(m_i|s_i)) \right) \quad (3)$$

The maximal fidelity value can not be predicted as explained in Section 4.2. The more relevant symptoms observed, the higher fidelity value can be achieved. The initial observation is usually insufficient. Thus, probing actions are often required to be conducted to enhance or reduce the belief on the created hypothesis. Overlay network administrators can define a fidelity threshold $F_{\text{threshold}}$ based on their relatively long-term observation and previous experience. If the threshold is set too high, even correct hypothesis will be ignored; but if the threshold is too low, then less credible hypothesis might be selected.

The fidelity evaluation function is used to evaluate each hypothesis and decides if a hypothesis is satisfactory. If the best hypothesis (i.e., the highest fidelity) whose fidelity exceeds the fidelity threshold, the fault diagnosis process terminates. Otherwise, the best available hypothesis and a non-empty set of components (C_N) will be verified via the selected probing actions to find a satisfactory hypothesis in the next iteration.

4.4 Action selection heuristic algorithm

We take probing actions whenever the passively observed symptoms are insufficient to find a satisfactory hypothesis. The action results can directly collect more relevant symptoms (either positive or negative) that can be used (1) to gather more evidence to increase *Symptom Contribution* of the observed symptoms, or (2) to verify the correctness of a given hypothesis. Two kinds of actions are considered in *ProFis*: overlay actions (A^o) and underlay actions (A^u). An overlay action is an end-to-end network measurement involving two overlay nodes and a set of intermediate underlay components. If A^o returns a positive result, it indicates that all related overlay and underlay components are in positive (i.e., good) status. Otherwise, if A^o returns a negative result, it returns a negative symptom showing that at least one related component is in negative (i.e., bad) status. Every action A_i^o ($A_i^o \in A^o$) has its own coverage, which is a set of involved components. Every action also has its cost that can be measured differently as we discussed in Section 3. In *ProFis*, we simply use the number of hops that each A_i^o traverses as its cost denoted as T_i.

In the following, we use C to represent C_N for simplicity. Given a set of components $C = \{C_1, C_2 \cdots, C_N\}$, a set of actions $A = \{A_1, A_2, \cdots, A_M\}$, and the coverage of each action denoted as: $A_i = \{C_x, C_y, \cdots, C_z\}$ $(A_i \in A)$ $(C \subseteq \bigcup_{A_i \in A} A_i)$, the action selection module is to find a set of actions A' $(A' \subseteq A)$ such that: (1) all components in C are covered; (2) the total cost $T = \sum_{A_i \in A'} T(A_i)$ is minimized. Thus, the task of action selection is to find the least-cost actions to verify all components C that are included in the hypothesis with the highest fidelity; or improve the initial observation so as to provide a useful reasoning result. As the size of C could grow significantly, the process of selecting the least cost actions that can verify C becomes non-trivial. The Action-Component correlation graph can be represented as a 3-tuple (A, C, E) such that A and C are two independent vertex sets representing Actions and Components respectively, and every edge e in E connects a vertex $A_j \in A$ with a vertex $C_i \in C$ with a corresponding weight (w_{ij}) to denote that A_j can verify C_i with cost $w_{ij} = w(C_i, A_j) > 0$. If there is no association between C_i and A_j, then $w_{ij} = 0$. A consists of overlay action set A^o and underlay action set A^u. Here we use an underlay action A_i^u to associate a set of conjunctive actions to verify one underlay component. The cost of the corresponding underlay action is the total cost of those relevant conjunctive actions.

The Component-Action graph in Fig. 5 presents the verification relationship between symptoms $\{C_1, C_2, C_3\}$ and actions $\{A_1^u, A_2^o, A_3^o\}$. A symptom C_1^u can be verified by taking a joint action between a_1 and a_2, which causes an underlay action vertex A_1^u to be created with weight 6. After converting action combination to one underlay action, Component-Action correlation can be represented in a bipartite graph.

The goal of the action selection algorithm is to select the actions that cover all components C with minimal action cost. With the representation of Component-Action bipartite graph, we can model this problem as a weighted set-covering problem. Thus, the action selection algorithm searches for A_i such that A_i

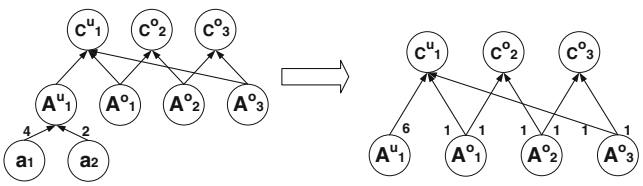

Fig. 5 Component-action bipartite graph

includes the set of actions that cover all the components in the Component-Action correlation graph with total minimum cost. We can formally define A' as the covering set that satisfies the following conditions: (1) $\forall C_i \in C, \exists A_j \in A'$ s.t. $w_{ij} > 0$, and (2) $\sum_{A_i \in A', C_j \in C} w_{ij}$ is the minimum.

The weighted set-covering problem is NP-complete [10]. Thus, we developed a heuristic greedy set-covering approximation algorithm to solve this problem. The main idea of the algorithm is simply first selecting the action (A_i) that has the maximum *relative covering ratio*, $R_i = \frac{|C_{A'}|}{\sum_{C_j \in C_{A'}} w_{ij}}$, where this action is added to the final set A_f and removed from the candidate set A_c that includes all actions. Here, $C_{A'}$ is the set of components that action A' can verify, $C_{A'} \subseteq C$. Then, we remove all components that are covered by this selected action from component set C. The searching process continues to find the next action A_i ($A_i \in A_c$), that has the maximum ratio R_i until all components are covered (i.e., C is empty). Clearly, this algorithm appreciates actions that have more component correlation or aggregation. If multiple actions have the same relative covering weight, the action with more covered components (i.e., larger $|C_{A_i}|$ size) will be selected. If multiple actions have the same ratio R_i and the same $|C_{A_i}|$, then each action is considered independently to compute the final selected sets for each action and the set that has the minimum cost is selected.

5 Evaluation

In this section, we describe our performance evaluation on the *ProFis* framework.

5.1 Evaluation metrics

The latency and accuracy are two critical factors to evaluate fault diagnosis techniques. Latency is measured by fault detection time T, which is the time between receiving the fault symptoms and identifying the root faults. The fault diagnostic accuracy depends on two factors: (1) the detection ratio (α), which is the ratio of the number of *true* detected root faults (F_d is the total detected fault set) to the number of *actual* monitored and occurred faults F_h. Formally, $\alpha = \frac{|F_d \cap F_h|}{|F_h|}$; and (2) the false positive ratio (β), which is the ratio of the number of *false* reported faults to the total number of detected faults. Formally, $\beta = \frac{|F_d - F_d \cap F_h|}{|F_d|}$ [28]. Therefore, the goal of any fault management system is to increase α and reduce β in order to achieve high accurate fault reasoning results. In overlay networks, fault diagnosis has to rely on end-user observations. Localization Granularity (LG) is also a very important factor in evaluating the accuracy of fault diagnosis, which is defined as the total number of detected faults by the total number of contained network components. Obviously, the best LG is 1 when each fault only contains one identifiable component. *ProFis* can be applied to solve various network faults, such as delay, loss, jitter etc. In our simulations, we assume the same type of overlay symptoms are collected and further correlated. However, correlating different types of symptoms into the same overlay fault reasoning framework can be interesting but out of the scope of this paper.

We consider the following dimensions and parameters for simulation.

- *Underlying Network Topology and Size:* We use a synthetic topology generator BRITE [21] with three types of topology model. In addition, we import to BRITE the real network autonomous system topology data from Skitter [26] with each set more than 20,000 autonomous systems for evaluation.
- *Overlay Network Topology:* For all given N overlay nodes, we create $\lceil 15\% N \rceil$ number of overlay applications. Each overlay application constructs a tree with 12 overlay pathes. The number of overlay links in each overlay path is uniformly distributed between [2, 6]. It is worth noting that ProFis doesn't have any restriction on the number of required overlay applications. The dynamic fault reasoning graph makes ProFis very flexible and scalable in terms of the number of overlay applications it can handle.
- *Overlay Symptom Coverage:* Each overlay application independently decides how many critical overlay paths should be monitored by configuring a parameter called Application Monitoring Ratio (AMR), which is the ratio of the number of monitored overlay paths over the total number of overlay paths.
- *Fault Ratio:* The prior fault probability of valid overlay nodes and valid underlay nodes are uniformly distributed in [0.01, 0.1] and [0.001, 0.01]. We assume underlay nodes (e.g., network routers) are commonly more reliable than overlay nodes (e.g., user workstations).
- *Passive and active symptom collecting rate (PSCR & ASCR)*: We define PSCR as one per 5 s; ASCR is one per 1 s. In our simulations, we use a discrete event simulation package called SimJava [14] to emulate symptom collections.

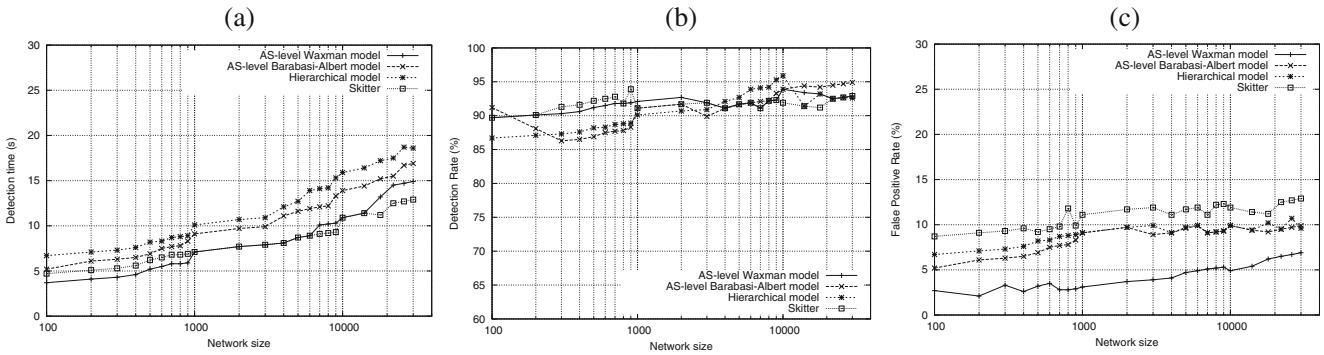

Fig. 6 The impact of network topology and size: **a** Detection time T; **b** Detection rate α; **c** False positive rate β

- *Symptom Loss Ratio (SLR):* Overlay symptoms can be lost during transmission. The higher the symptom loss ration, the less information available for a fault reasoning engine. Accordingly the longer time and more actions are required to locate the root faults.
- *Minimal Initial Information (MII) vs. Fidelity Threshold ($F_{\text{threshold}}$):* MII and $F_{\text{threshold}}$ are two configurable parameters that may significantly affect the latency and accuracy of *ProFis*.

The major contribution of this work is to present an efficient overlay fault reasoning technique that provides accurate results even when passive symptom collecting rate ($PSCR$) is low, and/or symptom loss ratio (SLR) is high. We show how these factors affect the latency (T) and accuracy (α and β) of our approach and passive fault reasoning approach. In our simulation study, we conduct experiments ten times for various setup parameters independently.

5.2 The impact of network topology and size

There are four type network topologies simulated using BRTIE [21]: (1) AS-level Waxman model, (2) AS-level Barabasi–Albert model, (3) hierarchical model, and (4) Skitter [26] based real network model. For the first three topology models, we create three different scenarios: (1) small-size networks (nodes between 100 and 1,000 placed in relatively small area of the plane in BRITE) to simulate regional collaborative ISPs and their overlay networks, (2) medium-size networks (nodes between 1,000 and 10,000 wide-distributively placed in the plane) to simulate national collaborative ISPs and their overlay networks, and (3) large-scale networks (nodes between 10,000 and 30,000) to simulate the Internet. For each generated topology, we select 10% nodes as overlay nodes with the fixed $MII = 40\%$ and $F_{\text{threshold}} = 75\%$. Then we run simulation ten times independently and use the average value as the final result.

The simulation results are shown in Fig. 6. The faults occur more frequently when the network size increases. With the properly integration of active actions and passive analysis, when the network size increased 1,000% from a small-sized network to medium-sized network, and further to a large-scale network, the corresponding detection time is only increased six times (approximately from 3 to 18 s) for all simulated network models. At the same time, both the detection rate and false

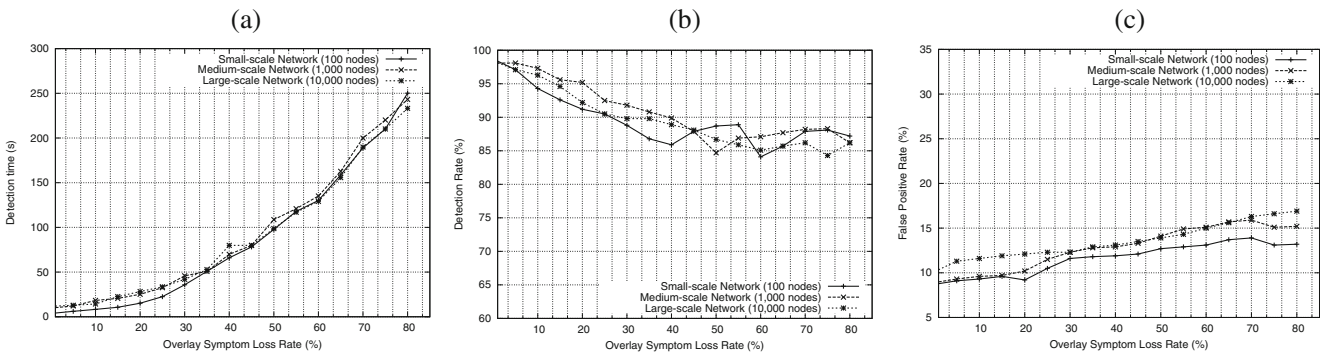

Fig. 7 The impact of overlay symptom loss: **a** Detection time T; **b** Detection rate α; **c** False positive rate β

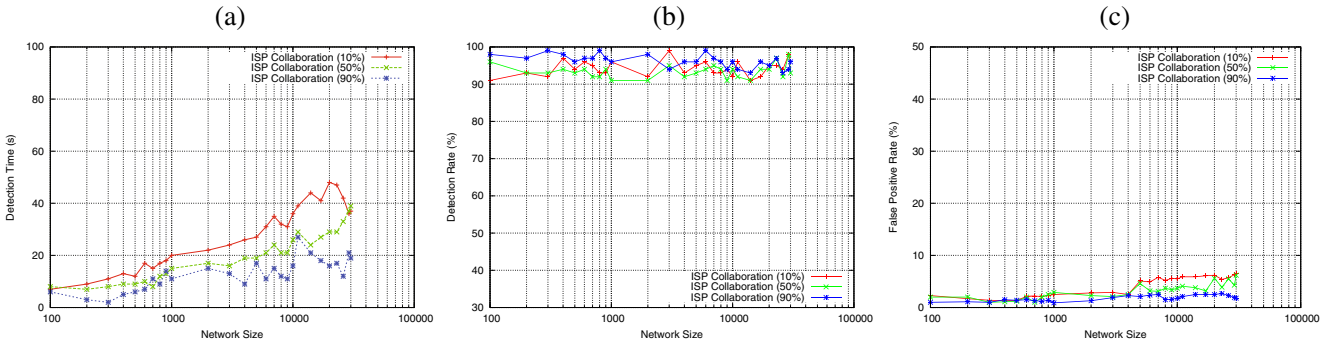

Fig. 8 The impact of ISP collaboration: **a** Detection time T; **b** Detection rate α; **c** False positive rate β

positive rate only vary within 10%. As shown in Fig. 6, there is no evident impact by various network topologies on the detection time, detection rate and false positive rate.

5.3 The impact of overlay symptom loss ratio

Since the identification and explanation of power laws has become an increasingly dominant theme in the recent body of network topology literature [21]. In the following simulation, we only use BRITE's Hierarchical Topology with ASBarabasiAlbert and RouterWaxman model, which can properly represent the power law node distribution. We use the fixed size of 10,000 nodes with $MII = 30\%$ and $F_{\text{fidelity}} = 80\%$. In our simulation study, we choose symptom loss ratio from 5 to 80% with an increment step of 5%. With the increase of symptom loss rate, the detection time could be significantly increased (e.g., increased by 50 times more) due to the insufficiently symptom observation. Accordingly, more actions might be required to collect relevant information. Moreover, incomplete observation may result to an incredible hypothesis that eventually causes more actions to enhance the reasoning result. However, the detection rate is relatively stable with a slightly decrease when the symptom loss rate increases. The false positive rate is also effectively controlled due to the actions reactively taken (Fig. 7).

5.4 The impact of ISP collaboration

The fundamental goal of overlay applications is to create disruptive services without relying on the support of network layer. In overlay applications, end-users can decide their data forwarding policies. However, ISPs may be willing to share some information for mutual benefit [33]. In this case, overlay applications may significantly benefit from the information provided by collaborative ISPs. In the following simulation study, we show that with certain level cooperation from ISPs, the latency and accuracy of overlay fault localization can be improved.

We still use three scenarios with different network sizes to represent possible overlay service models as follows. (1) Small size networks (100 nodes): Local ISPs (e.g., tier-3 ISPs) can collaborate with each other as well as their customers to achieve various service objectives such as redundancy or reliability. (2) Medium size networks (1,000 nodes): Reginal ISPs (e.g., tier-2 ISPs) can collaborate with each other to offer new services such as Web Caching. (3) Large scale networks (10,000 nodes): National ISPs (e.g., tier-1 ISPs) can collaborate with other ISPs or even among their own internal branches to enhance their overlay services. For each scenario, ISPs choose their collaboration rate (i.e., the ratio of shared symptoms over the total observed ones). As shown in Fig. 8, it is evident that such collaboration from ISPs by selectively providing their internal network information can greatly improve both the efficiency (i.e., detection time) and accuracy (i.e., detection rate and false positive rate) of *ProFis*.

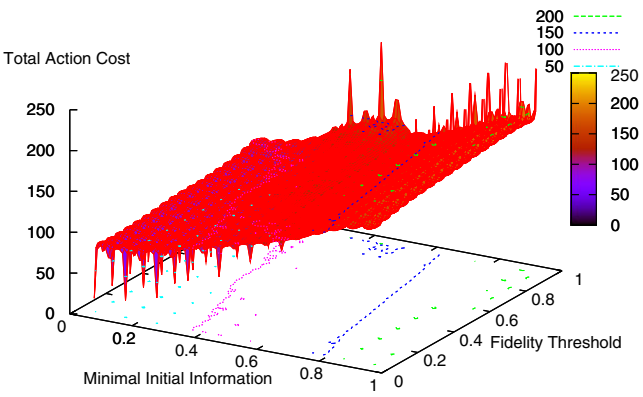

Fig. 9 ProFis intrusiveness evaluation

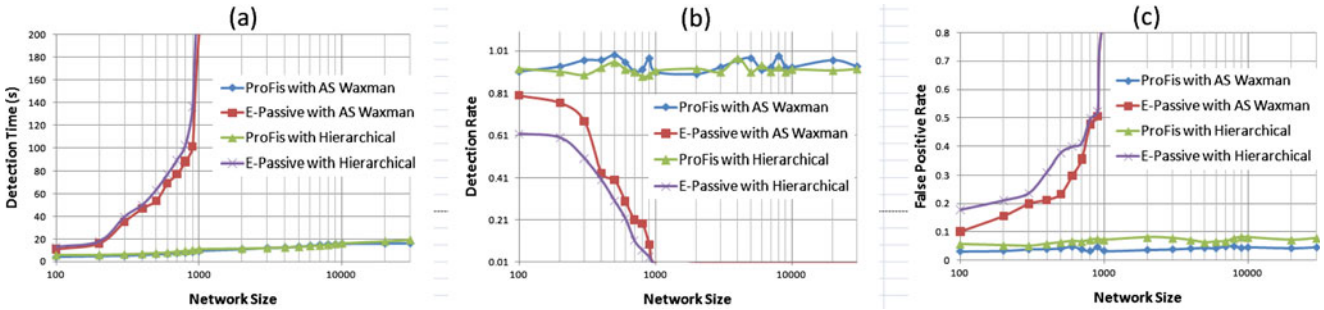

Fig. 10 The comparison to the enhanced passive approach: **a** Detection time T; **b** Detection rate α; **c** False positive rate β

5.5 The impact of MII and $F_{\text{threshold}}$ on ProFis intrusiveness

The intrusiveness of *ProFis* is measured by the total conducted action cost in various given network topologies with the corresponding given parameters such as MII and $F_{\text{threshold}}$. From the data trend shown in Fig. 9, it is clear that the increase of $F_{\text{threshold}}$ will incur the increasing cost of conducted probing actions. The choice of MII reflects a tradeoff between the action cost and detection time. In general, the higher MII is, the more actions are required. However, a higher MII may also reduce the cost for quickly satisfying $F_{\text{threshold}}$.

5.6 The comparison to the enhanced passive approach

ProFis is the first fault diagnosis framework that integrates active monitoring with passive fault reasoning to diagnose overlay faults based on a dynamically generated symptom-fault correlation graph. In the discussed existing work, there is no other solution that can be directly used to tackle all challenges existed in overlay fault diagnosis as *ProFis* can. The passive approach such as [28] is the closest one to *ProFis*. For more fair comparison purpose, we first extend [28] with the function of DFRG construction, referred to as Enhanced passive approach, or E-Passive in the following discussion.

We extensively compare *ProFis* to E-Passive with two network models (i.e., AS-level Waxman and hierarchical models) on all three evaluation aspects: detection time, detection rate and false positive rate. It is not surprise to see ProFis provides superior performance due to the seamless integration of active and passive diagnosis schemes. E-Passive even cannot conduct any level diagnosis when the network size is larger than 1, 000 nodes, in which the detection time of E-Passive is exponentially prolonged, the detection rate is significantly dropped, and the false positive rate is highly increased; all of these makes E-Passive unacceptable, as shown in Fig. 10.

5.7 The impact of symptom collecting rate

The PSCR and ASCR are two important system parameters, which directly affect the detection time of *ProFis*. In our experiment, we set PSCR and ASCR to low (1 symptom/5 s) and high (5 symptoms/s) collecting rates, respectively. Then, we have four scenarios shown as: (1) both PSCR and ASCR are low; (2) PSCR is low but ASCR is high; (3) PSCR is high but ASCR is low; and (4) both PSCR and ASCR are high. As the results shown in Fig. 11, in the first scenario, when the network size increasing from 100 to 30,000, the detection time is also evidently increased almost 600% since both PSCR and ASCR are low. In the second scenario, even the initial detection time is relatively high since PSCR is low, *ProFis* can still react quickly when faults occur as long as ASCR is high. In the third scenario, even the initial detection time is relatively low since PSCR is high, with the network size increasing, the overall detection time is significantly increasing similar to the first scenario since ASCR is high. The most ideal scenario is when both PSCR and ASCR are high, the overall fault detection time of *ProFis* can be greatly reduced.

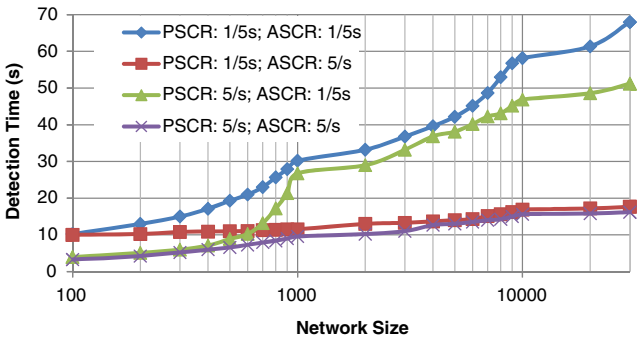

Fig. 11 ProFis symptom collection rate evaluation

6 Conclusion and future work

Fault diagnosis plays a critical role in managing and maintaining large-scale overlay networks. How to improve the efficiency and accuracy of overlay fault diagnosis is highly challenging due to the dynamics and scalability of overlay networks. In this paper, a novel overlay fault diagnosis framework called *ProFis* is proposed to dynamically create the correlation between overlay symptoms and faults, and seamlessly integrate the passive and active fault reasoning approaches in order to reduce fault detection time as well as improve the accuracy of fault diagnosis. *ProFis* is designed to minimize the intrusiveness of active probing via enhancing the passive fault reasoning and optimizing the probing action selection process. The evaluation via the simulation study and real-network experiments shows that *ProFis* is scalable even in a very large-scale overlay network and robust even when spurious symptom rate and symptom loss rate are high.

In our future work, we will study how to combine heterogeneous symptoms (e.g., packet loss and latency) into the same fault diagnosis framework. In addition, an information sharing protocol will be designed to facilitate the network symptom exchange among different overlay and Internet service providers.

Acknowledgements This research was supported in part by the National Grand Fundamental Research 973 program of China under Grant No. 2009CB320505, and the National Nature Science Foundation of China under Grant No. 60973123.

References

1. Akamai Global Content Delivery Technology Overview. http://www.akamai.com/html/technology/. Last access: October 20, 2010
2. Al-Shaer E, Tang Y (2002) QoS path monitoring for multicast networks. J Netw Syst Manag (JNSM) 10(3):357–381
3. Anagnostakis KG, Greenwald MB, Ryger RS (2003) Cing: measuring network-internal delays using only existing infrastructure. In: IEEE INFOCOM
4. Applety K et al (2002) Yemanja—a layered event correlation system for multi-domain computing utilities. J Netw Syst Manag 10
5. Brodie M, Rish I, Ma S (2001) Optimizing probe selection for fault localization. In: IEEE/IFIP (DSOM)
6. Chen Y, Bindel D, Song H, Katz RH (2004) An algebraic approach to practical and scalable overlay network monitoring. In: Proceeding of ACM SIGCOMM
7. Chu Y-h, Rao SG, Zhang H (2000) A case for end system multicast. In: Proceedings of ACM SIGMETRICS. Santa Clara, CA
8. Coates M, Hero A, Nowak R, Yu B (2002) Internet tomography. IEEE Signal Process Mag 19(3):47–65
9. CoMon—a monitoring infrastructure for PlanetLab. http://comon.cs.princeton.edu/. Last access: October 20, 2010
10. Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms, 2nd edn. MIT Press, Cambridge
11. Guo J, Kar G, Kermani P (2004) Approaches to building self healing system using dependency analysis. In: IEEE/IFIP (NOMS). Seoul, Korea
12. Hessler S (2006) BPB: a novel approach for obtaining network path characteristics in non-cooperative environments. In: Student's workshop at INFOCOM
13. Houck K, Calo S, Finkel A (1995) Towards a practical alarm correlation system. In: Integrated network management IV. Santa Barbara, CA
14. Howell F, McNab R (1998) Simjava: a discrete event simulation package for Java with applications in computer systems modelling. In: First international conference on Web-based modelling and simulation
15. Jakobson G, Weissman MD (1993) Alarm correlation. IEEE Netw 7:52–59
16. Kliger S, Yemini S, Yemini Y, Ohsie D, Stolfo S (1995) A coding approach to event correlation. In: Proceedings of the fourth international symposium on intelligent network management
17. Liu G, Mok AK, Yang EJ (1999) Composite events for network event correlation. In: Integrated network management VI, pp 247–260. Boston, MA
18. Mao ZM et al (2004) Scalable and accurate identification of as-level forwarding paths. In: IEEE Infocom
19. Mahajan R, Spring N, Wetherall D, Anderson T (2003) User-level Internet path diagnosis. In: Proc. ACM SOSP
20. McCloghrie K, Rose M (1991) Management information base for network management of TCP/IP-based Internets: MIB-II, RFC1213
21. Medina A, Matta I, Byers J (2000) On the origin of power laws in Internet topologies. ACM Comput Commun Rev 30:18–28
22. Peterson L, Anderson T, Culler D, Roscoe T (2002) A blueprint for introducing disruptive technology into the Internet. In: The proceedings of ACM HotNets-I workshop
23. PlanetLab. http://www.planet-lab.org. Last access: October 20, 2010
24. Rish I, Brodie M, Odintsova N, Ma S, Grabarnik G (2004) Real-time problem determination in distributed systems using active probing. In: IEEE/IFIP (NOMS). Seoul, Korea
25. Savage S (1999) Sting: a TCP-based network measurement tool. In: Proceedings of the 1999 USENIX symposium on Internet technologies and systems
26. Skitter. CAIDAs topology measurement tool. http://www.caida.org/tools/measurement/skitter/. Last access: October 20, 2010
27. Steinder M, Sethi AS (2002) Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms. In: Proc. of IEEE INFOCOM. New York, NY
28. Steinder M, Sethi AS (2004) Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. Comput Netw 45(4):537–562
29. Tang Y, Al-Shaer E (2008) Towards user-level collaborative overlay fault diagnosis. In: The 27th IEEE INFOCOM mini-conference. Phoenix, AZ
30. Tang Y, Al-Shaer E (2009) Sharing end-user negative symptoms for improving overlay network dependability. In: The 39th IEEE/IFIP international conference on dependable systems and networks (DSN). Lisbon, Portugal
31. Tang Y, Al-Shaer E, Boutaba R (2008) Efficient fault diagnosis using incremental alarm correlation and active investigation for internet and overlay networks. IEEE Transactions on Network and Service Management 5(1):36–49

32. Team Cymru IP to ASN Lookup. http://asn.cymru.com/cgi-bin/whois.cgi. Last access: October 20, 2010
33. Xie H, Yang YR, Krishnamurthy A, Liu Y, Silberschatz A (2008) P4P: portal for (P2P) applications. In: Proceedings of SIGCOMM
34. Zhang M, Zhang C, Pai V, Peterson L, Wang R (2004) PlanetSeer: Internet path failure monitoring and characterization in wide-area services. In: Proc. sixth symposium on operating systems design and implementation
35. Zhao Y, Chen Y, Bindel D (2006) Towards unbiased end-to-end network diagnosis. In: Proceeding of ACM SIGCOMM

Guang Cheng is a professor of Computer Science at Southeast University of Nanjing, P.R. China. He received his M.Sc. in Transportation Engineering from Southeast University, P.R. China in 1994, B.Sc. in Computer Science from Hefei University of Technology, P.R. China in 2000, and Ph.D. in Computer Science from Southeast University in 2003. From 2006 to 2007, he was a post-doctor at School of Electrical and Computer Engineering in Georgia Institute of Technology, USA. He has been working at Southeast University since 2003. His current research interests include active measurement and traffic sampling in network area.

Yongning Tang has been an assistant professor in the School of Information Technology at Illinois State University since 2007. He received his B.Sc. and M.Sc. in Electrical Engineering both from Hefei University of Technology, P.R. China in 1991 and 1994, respectively, and Ph.D. in Computer Science from DePaul University, Chicago, in 2008. His research interests are in fault diagnosis in large-scale distributed networking systems, overlay networks, network security, wireless networks, and network traffic classification, as well as various machine learning and artificial intelligence techniques.

Zhiwei Xu is an assistant professor at University of Michigan-Dearborn. He received his M.Sc. in Electrical and Computer Engineering from Guangxi University, P.R. China in 1997. He received his Ph.D. in Computer Engineering from Florida Atlantic University in 2001. His research interests are in software metrics, quality modeling, software V&V, data mining, pattern recognition, artificial intelligence, cost modeling, and optimization.