2nd IEEE International Conference on Cloud Computing Technology and Science

# A Novel Approach for Cooperative Overlay-Maintenance in Multi-Overlay Environments

Chin-Jung Hsu[†] Wu-Chun Chung[†] Kuan-Chou Lai[¥] Kuan-Ching Li[§] Yeh-Ching Chung[*]

[*†]Dept. of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan
[¥]Dept. of Computer and Information Science, National Taichung University, Taichung 403, Taiwan
[§]Dept. of Computer Science and Information Engineering, Providence University, Taichung 433, Taiwan
Email: {[†]oxhead, [†]wcchung}@sslab.cs.nthu.edu.tw, [¥]kclai@mail.ntcu.edu.tw, [§]kuancli@pu.edu.tw, [*]ychung@cs.nthu.edu.tw

*Abstract*—**Overlay networks are widely adopted in many distributed systems for efficient resource sharing. Recently, issues in overlay network have also been introduced into cloud systems, in order to organize thousands of virtualized resources. In parallel, the explosion of P2P applications introduces the multi-overlay environment in which a number of nodes simultaneously participate in multiple overlays. When multiple applications running over a large set of nodes, some of nodes may take repeated efforts to preserve multi-overlay networks. Therefore, maintaining these co-existing overlays brings the redundant maintenance overhead. This paper presents a cooperative strategy to analyze the overlay maintenance of multi-overlay environments and to elaborate multiple overlays for simplifying the overlay maintenance. The proposed strategy exploits the synergy of co-existing overlays to handle their common overlay-maintenance, so that the redundant maintenance overhead could be eliminated while keeping performance. To evaluate the system performance, this paper not only analyzes several overlays but also considers realistic multi-overlay environments by varying the intersection ratio of diverse overlays and the combination of multiple overlays. Experimental results show that the proposed cooperative strategy significantly decreases the redundant overlay-maintenance overhead, where the reduction ratio of maintaining multiple overlays is higher than 60 percent in some of cases.**

***Keywords. peer-to-peer; overlay network; multiple overlays; cooperative overlay maintenance; multi-overlay environment***

## I. Introduction

The overlay network is a logical computer network built over the underlying Internet. On the benefits of scalability and reliability, overlay networks are widely adopted in applications of distributed systems, such as P2P systems [1], [14], [16] and resource discovery systems [3], [4]. With the emergence of Cloud computing [2], [7], co-existing multiple overlays are introduced with the explosion of application services. In such a multi-overlay environment, nodes participate in multiple overlays simultaneously. For instance, when one node executes a P2P file sharing application and a P2P streaming application simultaneously, such an environment is created.

In computing systems as Cloud, multi-overlay environments incur large cost to maintain multiple overlays of applications. However, some of these multiple overlay-maintenance costs are redundant and are possible to be eliminated. For instance, the failure detection operation is necessary for every overlay in order to ensure the overlay resilience [20] and the network-proximity estimation operation is also important one to enable the locality-aware overlay [5], [14], [15], [18]. Since some overlay-maintenance operations are identical, they imply large redundant overhead for maintaining similar operations.

Some recent studies tried to reduce the redundant overlay-maintenance costs. Maniymaran et al. [13] consider the co-existence of the Pastry and the Interest-based Gossip Protocol, while Lin et al. [12] exploit the synergies between multiple co-existing overlays. However, their approaches are only applied to specific overlays, mainly focused on Gossip-based protocols. Moreover, both previous research works did not consider the realistic multi-overlay environment that affects the practical effectiveness, e.g., different intersection ratio of overlays.

This is the approach we adopt in our proposal, focusing at the elimination of redundant overlay-maintenance operations while preserving the system performance in multi-overlay environments. The master-slave model is proposed so that one master overlay handles the common overlay-maintenance operations for other slave overlays, so that the overall maintenance costs could be reduced. Additionally, the inter-overlay and intra-overlay protocols are proposed to coordinate the co-existing overlays; more specifically, the former provides a communication channel for the master and the slave overlays while the latter defines how to handle the common overlay-maintenance.

Based on the above-mentioned cooperative strategy, cooperative failure detection (CFD) and cooperative network-proximity estimation (CNPE) are proposed to eliminate the redundant failure detection and the redundant network-proximity estimation in a multi-overlay environment. Also in this paper, analysis of the mechanism and the maintenance cost model are presented by comparing the maintenance cost of the original multi-overlay environment with that of the multi-overlay environment applying the proposed cooperative strategy. Multi-overlay environments are examined by varying overlay combinations of unstructured, ring and tree overlays. The realistic multi-overlay environments are also considered, as overlays have different sets of nodes which is not considered in previous researches [12], [13].

The rest of this paper is organized as follows. Section II introduces the related works of multiple overlays, while in Section III the proposed cooperative strategy and the master-slave model are presented; detailed CFD and CNPE

978-0-7695-4302-4/10 $26.00 © 2010 IEEE
DOI 10.1109/CloudCom.2010.12

mechanisms are described, as also analysis of cost model are reported. Section IV shows the experimental results to illustrate the system performance, and finally, conclusion remarks and future work are given in Section V.

## II. Related Work

The rapid development and widespread applicability of overlay-based distributed systems lead to the arising of multiple overlays co-existing over the Internet. In the literature, researches in multi-overlay have focused on the multi-overlay framework [8], [19], the race condition problem [6], [10], [11], [17] and the maintenance cost reduction [12], [13].

The problem of the multi-overlay framework emphasizes the framework supporting the co-work of multiple overlays. MOSAIC [19] presents an extensible infrastructure with the ability of automatic selection and composition of multiple overlays. GRIDKIT [8] proposes the framework supporting the interaction among multiple overlays for Grid computing. Another research that tries to solve the performance degradation suffered with the presence of resource competition in a multi-overlay environment, and a number of approaches that discusses such a problem can found in [6], [10], [11], [17].

There are research studies that targets at ways to reduce the maintenance cost of co-existing overlays; that is, as these overlays work separately, large amount of redundant maintenance cost is introduced. Maniymaran et al. [13] proposed a novel approach to leverage the coexistence of the Interest-based Gossip protocol and Pastry by constructing the joint overlay. They have shown that the routing table in the Pastry could be replaced by the cluster view routing table in the Interest-based Gossip protocol, and the RPS view routing table in the Interest-base Gossip protocol also can be replaced by the leaf set table in the Pastry. However, their approach has only focused on these two specific overlays. In [12], it exploits the synergies among multiple co-existing overlays focused on the Gossip protocol, where the synergy of overlays benefits the system performance instead of causing negative impact introduced by overlay competition, and different types of synergies are also analyzed, in order to compare the potential benefit.

From previous investigations, few studies have reported a general approach to leverage co-existing overlays. Besides, most of previous researches do not consider the practical network environment with the different intersection ratio of overlays. In a multi-overlay environment, the nodes in each overlay may not be the same; hence, the intersection ratio is a key factor affecting the overall system performance. Accordingly, our paper not only proposes a general strategy for leveraging multiple co-existing overlays to simplify the redundant maintenance, but also takes realistic multi-overlay environments into consideration.

## III. Cooperative Strategy

In a multi-overlay environment (MOE), most of overlay-maintenance operations are redundant. Aiming at this problem, we propose a cooperative strategy to exploit the synergy among co-existing overlays. In such a MOE, one of these overlays is dedicated, namely as the master overlay, to taking charge of the common overlay-maintenance of other slave overlays. This section presents the proposed cooperative strategy and corresponding mechanisms.

### *A. Master-Slave Model*

In a given MOE, one of the co-existing overlays acts as the master overlay to handle the common overlay-maintenance for other slave overlays. For example, the overlay-maintenance cost of a three-overlay environment, e.g., in a cloud environment with multiple overlays, is given as $C = C_{O1} + C_{O2} + C_{O3}$. After adopting the master-slave model, the expected cost would be $C' (= C'_{O1} + C'_{O2} + C'_{O3}) < C$, where $C_{Oi}$ represents the cost for overlay $i$.

To support the cooperation among the master and slave overlays, the inter-overlay and intra-overlay protocols are introduced. The former defines the interaction mechanism while the latter defines the mechanism in which the master overlay deals with the request from the other slave overlays. In this paper, two types of interaction mechanisms are supported. The subscription/notification protocol enables the master overlay to handle the event-driven overlay-maintenance and the query/response protocol makes the master overlay able to answer the query from slave overlays. Regarding the intra-overlay protocol, it depends on the type of overlay-maintenance. In essence, there are various common overlay-maintenance operations. Two important sorts are emphasized: the failure detection and the network-proximity estimation. More details about cooperations will be given in the proposed cooperative mechanisms.

### *B. Cooperative Failure Detection (CFD)*

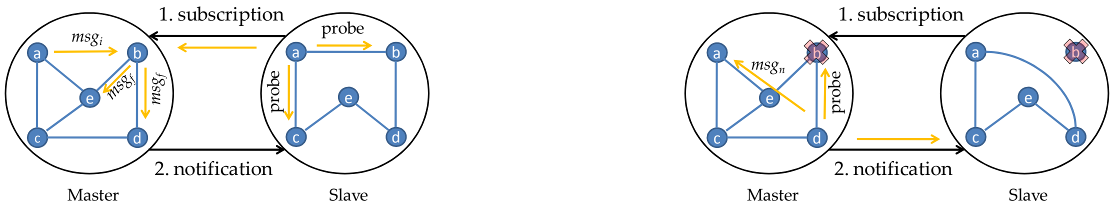
The failure detection is necessary for each overlay, in order to ensure the overlay resilience. In the following, we detail the CFD mechanism and demonstrate how to eliminate the redundant operations of failure detection in a MOE. After that, we present the analysis of maintenance cost model.

#### *1) Concept and Mechanism*

Within the master-slave model, one of the multiple overlays acts as the master overlay to handle the failure detection, so failure detection operations in the slave overlay are exempt from the maintenance process. Since failure detection is an event-driven overlay-maintenance operation, CFD adopts the subscription/notification inter-overlay protocol for the communication between the master overlay and the slave overlay. The node in the slave overlay subscribes to the same node in the master overlay for the required failure detection operation. After the node in the master overlay receives the subscription request, it deals with the request by adopting the intra-overlay protocol.

The following presents two main schemes, reducing redundant operations of failure detection.

*a) Elimination:* The concept of elimination approach is inspired by exploiting the duplicated links between two nodes in master and slave overlays. If such links exist, failure detection operations can be executed only once (instead of twice, one per each overlay), so that the redundant operations are free from handling.



(a) (b)

Fig. 1. Mechanism of CFD. (a) The subscription process. (b) The notification process.

*b) Cooperation:* Duplicated links do not always exist. Contrary to the elimination approach, this reduction approach exploits the duplicated probing from different nodes to the same node. Both master and slave overlays maintain the node status; hence, the master overlay can assist the slave overlay to handle the failure detection operations. However, it may incur extra overhead in the master overlay to handle such a case.

Fig. 1 gives the complete view on how the CFD mechanism works under a two-overlay environment. In this figure, the unstructured overlay constructs the topology as a random graph while the ring overlay connects nodes using the ring structure. The CFD mechanism includes four steps: request, inform, forward and notify. The following description takes Fig. 1 as an example and some of notations used next are listed in TABLE 1.

REQUEST: $a^s$ must handle failure detection operations monitoring the status of $b^s$ and $c^s$ originally; however, with the master-overlay model, $a^s$ requests $a^m$ for assistance instead. Hence, $a^s$ sends two requests to $a^m$. Each request includes the target node's id that $a^s$ needs to monitor. In this case, these two requests include the information of $n_t$ ($b^s$ and $c^s$) respectively.

INFORM: In general, once $n_h$ receives a request from $n_r$, $n_h$ starts dealing with the request. Next, $n_h$ compares $n_t$ information carried in the request with its own local routing table and decides the next action. If $n_t$ is in the local routing table, the subscription step is done; otherwise, $n_h$ informs $n_t$ about the new request. After that, $n^m$ adds a new record ($n_t$, $n_r$) to the subscription table. In the case of Fig. 1a, $a^m$ receives two subsequent requests from $a^s$. The first request received has the target node information ($b^s$). $a^m$ checks its own routing table, and finds no records about $b^s$. Next, $a^m$ informs $b^m$ about the request, and then $a^m$ adds ($b^s$, $a^s$) to its subscription table. The second request received has the target node information ($c^s$). $a^m$ checks the routing table again and finds a record in it. Then, node $a^m$ adds ($c^s$, $a^s$) to the subscription table and finishes the subscription step.

FORWARD: After $n_t$ receives the request from $n_h$, $n_t$ forwards this request to numerous $n_c$ chosen from the routing table of $n_t$. The remaining action is to record an entry ($n_h$,) in the forward table. In this example, $b^m$ receives the request informed by $a^m$. Then, $b^m$ forwards the request to $d^m$, $e^m$ and records the entry ($a^m$, { $d^m$, $e^m$}) in its forwarding table.

NOTIFY: While $n_c$ receives $msg_f$ from $n_t$, $n_c$ just adds a record to the notification table. Once $n_c$ detects that $n_t$ fails, $n_c$ notifies the mapping $n_h$ of the event. The final action to complete the subscription/notification protocol is to notify $n_r$. As shown in Fig. 1b, $d^m$ notifies $a^m$ that $b^m$ is failed and then, $a^m$ notifies $a^s$ of this failure.

The following analyzes related factors affecting the performance.

*Number of Cooperators* (*NC*): When $n_t$ sends $msg_c$ to numerous $n_c$, the amount of $n_c$ cannot be too small in order to avoid simultaneous failures of all $n_c$. Experimental results presented in section that follows next show that the minimum number of cooperators to ensure the CFD performance is two. Therefore, we use two cooperators in the CFD mechanism.

*Re-forward:* In a dynamic environment, nodes may fail or leave arbitrarily. Hence, as $n_t$ detects node $n_c$ is failed, the re-forward procedure needs to be executed to ensure CFD works properly. The re-forward procedure of $n_t$ involves detecting the failed $n_c$ and re-forwarding the request from $n_h$. The detection can be integrated with the original failure detection mechanism of $n_t$, and once $n_t$ detects the failed $n_c$, it re-forwards the request to new $n_c$.

*Link Intersection Ratio* (*LIR*): When $n_h$ receives a subscription request, it decides whether it can handle the request itself. If $n_h$ can handle such a request, it must contain the link ($L^m_{n_h n_t}$). In the case that the link intersection ratio is high, the number of $msg_f$ to $n_t$ is relatively small. For example, when *IR* is 100% in a two-overlay environment and the master overlay forms a fully connected topology, there is no additional overhead, and therefore, the cooperation approach is no more required.

TABLE 1. NOTATIONS

| Symbol | Explanation | Example |
|---|---|---|
| $n$ | the node $n$ | $a$, $b$ |
| $m$, $s$ | the superscript indicating the master and slave overlay | |
| $n^m$, $n^s$ | $n_m$ is the node $n$ in the master overlay and $n_s$ in the slave overlay | $a^m$, $b^s$ |
| $L^m_{\boldsymbol{n_1 n_2}}$, $L^s_{\boldsymbol{n_1 n_2}}$ | the link from $n_i$ to $n_j$ in the master overlay and the slave overlay | $L^m_{ab}$, $L^s_{cd}$ |
| $n_r$ | the requester node in the slave overlay | $a_r$ |
| $n_h$ | the handler node in the master overlay handling the request from $n_r$ | $a_h$ |
| $n_t$ | the target node in the master overlay that $n_r$ is interested in | $b_t$ |
| $n_c$ | the cooperator node in the master overlay that helps $n_h$ handle the request from $n_r$ | $d_c$ |
| $msg_i$ | the message produced by $n_h$ for informing $n_t$ about the request from $n_r$ | |
| $msg_f$ | the message produced by $n_t$, forwarding $msg_i$ to $n_c$ | |
| $msg_n$ | the message produced by $n_c$, notifying $n_h$ of failed $n_t$ | |
| $msg_e$ | the message produced by $n_h$ to explore more satisfied answer for the request from $n_r$ | |

*Overhead:* The overhead arises from the subscription message, the re-subscription message, the notification message and the maintenance message. The subscription message is introduced as one new node joins the slave overlay; the joining action triggers a subscription request to $n_h$. Regarding to the re-subscription message, it is triggered by the subscription request, while the notification message is the message notified from $n_c$ to $n_h$. The maintenance message is introduced by the maintenance procedure of the CFD mechanism, e.g., the re-forward procedure. In this way, the subscription message includes $msg_i$ and $msg_f$, the re-subscription message is comprised of the re-sent $msg_i$ and $msg_f$, the notification message is $msg_n$ and the maintenance message is the re-sent $msg_f$.

### *2) Maintenance Cost Model*

The total cost of a MOE is modeled by calculating the number of failure detection messages and the amount of extra overhead. We assume the network size $N$ and the duration of failure detection $D$. The total cost is calculated within the time period $T$. Besides, the overlay dynamics is controlled by $R$, representing $N\times R$ nodes join and leave every second. As matter of simplicity, the maintenance cost is modeled as discrete model, where nodes are identical in each overlay. Each node in the overlay runs the operation of failure detection every $D$ second. Hence, the node joining at time $t$ will execute the failure detection at time $t+D$, $t+2D$, $t+3D$, … Note that the node joining at time $t+D$ will also execute the failure detection at time $t+2D$, $t+3D$, $t+4D$, … Therefore, $[t/D]$ sets of nodes that joined at different time ($t-D$, $t-2D$, $t-3D$, ...) will execute the failure detection operation at time $t$; the number of nodes that execute the failure detection at time $t$ ($t \bmod D \neq 0$) is

$$G(s) = \sum_{i=1}^{s} N \times R \times (1-R)^{i\times D}, \qquad (1)$$

where $s$ is the number of node sets executing the failure detection. Since the initial network size is $N$, those initial nodes performs failure detection at $D$, $2\times D$, $3\times D$, and so on. The number of nodes executing the failure detection at time $t$ can be calculated as

$$\mathrm{S(t)}=\begin{cases}\mathrm{N}\times(1\text{-}\mathrm{R})^{\mathrm{t}}+\mathrm{G}([\mathrm{t/D}]\text{-}1), \text{ if t mod D} = 0\\ \mathrm{G}([\mathrm{t/D}]), \text{ otherwise}\end{cases}. \qquad (2)$$

To sum up, the number of times of failure detection at time $t$ is $F(t) = S(t) \times L(t)$, where $L(t)$ is the average number of links of all nodes in the overlay at time $t$. Therefore, the number of messages of failure detection within period $T$ is given as $F = \sum_{t=1}^{T} F(t)$.

As for a MOE without CFD, the total maintenance cost is

$$M = F_{MOE} = \sum F^{O}, O \in MOE, \qquad (3)$$

and the cost model of the MOE with CFD is

$$M_{CFD} = F_{MOE} + C_{CFD}, \qquad (4)$$

where $C_{CFD}$ is the overhead introduced by the CFD mechanism. In this model, we assume that the size of communication message of the overhead is equal to the message size consumed by failure detection.

Based on the preliminary analysis found at section III-B.1, the overhead comes from subscription, re-subscription, notification and maintenance messages, and these costs are analyzed next.

*a) Subscription message:* When a node joins the slave overlay, the node sends numerous subscription messages to the handler node ($n_h$). So, the number of subscription messages at time t can be calculated by

$$C_s(t) = J(t) \times W \times (1 - LIR(t)) \times (1 + NC), \qquad (5)$$

where $J(t)$ is the number of nodes joining the overlay, $W$ represents the number of links created per joining operation, and $LIR(t)$ means the link intersection ratio between $n_r$ and $n_h$. TABLE 2 lists different $W$ values corresponding to different overlays.

*b) Notification message:* When $n_c$ detects the failed $n_t$, it notifies $n_t$, and thus, the total number of notification messages introduced at time $t$ can be calculated as

$$C_n(t) = F(t) \times (D \times R) \times H(t), \qquad (6)$$

where $D\times R$ is the number of expected failed nodes of being detected and $H(t)$ is the average number of the sent notification messages when $n_c$ detects the failed $n_t$.

*c) Re-subscription message:* When $n_r$ receives a notification from $n_h$, $n_r$ executes the repair process based on different overlay mechanisms. The number of re-subscription message at time $t$ can be modeled as

$$C_r(t) = C_n(t) \times Y \times Z \times (1 + NC), \qquad (7)$$

where $Y$ is the probability that $n_r$ executes the repair process, $Z$ is the number of new links created by the repair process, and $NC$ is the number of cooperators as described in section III-B.1.

*d) Maintenance message:* The number of maintenance messages is the number of re-sent $msg_f$, also representing the number of failed nodes detected by $n_t$. Hence, the number of maintenance messages, at time $t$ , is

$$C_m(t) = F(t) \times (D \times R). \qquad (8)$$

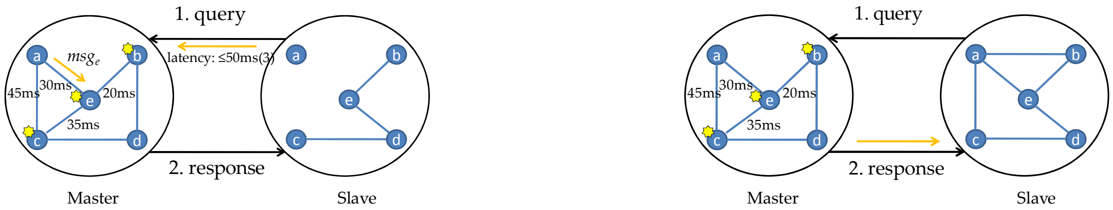
Accordingly, the total overhead can be calculated by $C_{CFD} = C_s + C_n + C_r + C_m$. The reduction ratio of maintenance cost of the CFD mechanism is

$$RR = \frac{M - M_{CFD}}{M} \times 100\% = \left(1 - \frac{M_{CFD}}{M}\right) \times 100\%. \qquad (9)$$

Clearly, the reduction ratio of CFD depends on the overhead introduced by CFD mechanism.

TABLE 2. PARAMETERS OF MAINTENANCE COST MODEL

| | Unstructured | Ring | Tree |
|---|---|---|---|
| $W$ | $2\times K$ | $4\times K$ | $2\times(K+H)$ |
| $L$ | $\leq 2\times K$ | $\cong 4\times K$ | $\leq (2+H)$ |
| $Y$ | $< 1$ | 1 | $< 1$ |
| $Z$ | $\leq 1$ | 1 | $\cong (1+H)/(1+K+H)$ |



(a) (b)

Fig. 2. Mechanism of CNPE. (a) The query process. (b) The response process.

### *C. Cooperative Network-Proximity Estimation (CNPE)*

Since the network-proximity estimation is important and costly, the proposed CNPE mechanism is helpful in reducing the redundant network measurement operation in a MOE.

#### *1) Concept and Mechanism*

The CNPE mechanism hands the common network-proximity estimation. Similar to CFD, CNPE is also based on the master-slave model and the query/response protocol is adopted. In a MOE, an overlay acts as the master overlay to deal with the network-proximity estimation for slave overlays, so that redundant operations of network-proximity can be replaced by information queries. While a node joins multiple overlays, it needs to estimate the network proximity separately, so redundant maintenance operations are introduced. With CNPE, the dedicated master overlay helps with elimination of those redundant operations. The node in slave overlays queries the needed information from the node in the master one.

CNPE has two critical schemes to reduce the redundant operations of network-proximity estimation. One is the elimination and the other is the exploration.

*a) Elimination:* This idea is similar to the elimination in CFD. Maintained by multiple overlays, the requisite network-proximity information can be shared. Having to estimate network-proximity (e.g., to obtain nodes geographically closed to it), the node asks the same node in the master overlay via the query/response protocol, so redundant estimation operations are reduced.

*b) Exploration:* Although the elimination approach is useful, the master overlay cannot handle the request from the slave overlays in some of cases. For example, the number of nodes requested by slave overlays exceeds that provisioned by master overlay, so it should adopt an enhanced intra-overlay protocol to satisfy the various needs of the slave overlay. Since the master overlay maintains the network information, the node in the master overlay can explore additional required information within the master overlay when this node cannot fit the query from the slave overlay. This node sends an exploration request to some of its neighbors for retrieving the network information. Since the message cost of sending a request is smaller than that of estimating the network, this exploration approach eliminates the operation of network-proximity estimation.

Detailed CNPE mechanism is presented next and related notations are listed in TABLE 1. Fig. 2 shows how CNPE works under the two-overlay environment composed of two unstructured overlays. The CNPE mechanism includes three steps: query, exploration and response.

QUERY: $a^s$ is ready to join the slave overlay, what means that $a^s$ must execute the network-proximity estimation. In this way, $a^s$ fetches the necessary information from $a^m$ via the query/response protocol. In this case, $a^s$ sends a query request to $a^m$ (e.g., *amount=3, latency≤50ms*).

EXPLORATION: After $n_h$ receives a query requested from $n_r$, $n_h$ finds the satisfied information in the local routing table. If the local information cannot fit the needs of $n_r$, $n_h$ sends the exploration request $msg_e$ to some of its neighbors $n_c$; otherwise, the exploration will not be executed. After $n_c$ receives the request from $n_h$, $n_c$ returns the information of the node latency stored in it to $n_h$, and then the exploration action ends. As shown in Fig. 2a, $a^m$ receives a request (e.g., *amount=3, latency≤50ms*) from $a^s$, and $a^m$ finds that it does not have enough satisfied information for $a^s$. After that, $a^m$ starts the exploration action and then it sends a request to $e^m$. In this case, $e^m$ returns the information (*{*$b^m$*, 20ms}*) to $a^m$.

RESPONSE: After $n_h$ gathers enough information, it returns the final result to $n_r$, so that the query/response process ends. As shown in Fig. 2, $a^m$ responds with the information *{{b, 50ms}, {c, 45ms}, {e, 30ms}}* to $a^s$. After $a^s$ obtains the required information, $a^s$ creates links to these nodes (*b, c, e*).

According to the proposed CNPE mechanism, analysis of relative factors affecting the performance is presented next.

*Network-proximity estimation:* the network latency/bandwidth/utilization can be used to estimate the network proximity. In this paper, our approach relies on network latency as the estimation metric. On the other hand, in the estimation process, each node measures $A \times K$ nodes and selects $K$ nodes from them, where $A$ is a factor used for estimating more candidate nodes. Note that we do not aim at proposing an efficient estimation approach in this paper, so the type of metrics used does not affect the proposed CNPE.

*Exploration:* The exploration approach in CNPE adopts the neighbor's neighbor method [15] to eliminate the costly network-proximity estimation operation. Although there are other methods that are available to be adopted in the exploration approach, the neighbor's neighbor method is adopted in order to simplify the proposed approach. To evaluate the exploration approach, our experiments compare the performance of CNPE with and without the exploration approach. Although the exploration approach introduces an additional overhead, this exploration communication cost is smaller than that of estimation.

*Overhead:* The proposed CNPE without the exploration approach has no extra overhead because the master overlay does not have routing messages; otherwise, the exploration approach brings extra overhead (i.e., $msg_e$) to explore additional network information for the slave overlay.

### 2) Maintenance Cost Model

The CFD mechanism is modeled by the times of failure detection and CNPE by the number of network-proximity estimation. An estimation operation is created as a node joins an overlay or repairs its routing tables (i.e., deletes the record of failed neighbor and adds the record of new neighbor). Therefore, the number of such operations introduced by an overlay at time $t$ is defined as

$$E(t) = A \times (J(t) \times W + F(t) \times (D \times R) \times Y \times Z), \quad (10)$$

where $J(t)$, $W$, $F(t)$, $Y$ and $Z$ are the same as presented in section III-B.2. Besides, $A$ is a weight factor used to decide the number of candidate nodes as estimated. For instance, when $A=2$, one node links to the size of $K$ new neighbor nodes selecting from $2\times K$ estimated nodes. The total number of network-proximity estimation in time $T$ is $E = \sum_{t=1}^{T} E(t)$.

The overhead of the CNPE mechanism comes from $msg_e$ issued from $n_h$ to $n_c$. When $n_h$ cannot satisfy the query request from $n_r$, $n_h$ starts the exploration procedure. Hence, the overhead at time $t$ can be modeled as

$$C(t) = E(t) \times \left(1 - P_{success}(t)\right) \times Q(t), \quad (11)$$

where $P_{success}(t)$ represents the success ratio that $n_h$ can handle the query request by itself, and $Q(t)$ is the average number of $msg_e$ sent for completing the request from $n_r$.

Similar to CFD, the maintenance cost of the MOE with CNPE can be defined by

$$M_{CNPE} = E_{MOE} + \frac{1}{B} \times C_{CNPE}. \quad (12)$$

Since the message size of network-proximity estimation is relatively larger than that of the exploration procedure, in this paper, we assume that the message size of estimation is $B$ times as large as that of exploration overhead. The reduction ratio on network-proximity estimation is

$$RR = \left(1 - \frac{M_{CNPE}}{M}\right) \times 100\%, \quad (13)$$

where $M = \sum E^{O}, O \in MOE$.

## IV. Experimental Results

The proposed CFD and CNPE are evaluated by diverse MOEs, and also evaluated by a realistic MOE with different sets of nodes in each overlay. Moreover, a hybrid strategy integrating CFD and CNPE is applied to a four-overlay environment of higher complexity.

### A. Experimental Environment

This paper adopts the PeerSim [9] simulator for performance evaluation. PeerSim is written by Java with cycle-based and event-based simulation engines, and the former simulation engine is applied in our set of experiments.

Three overlays are involved: an unstructured overlay ($O_{unstructured}$), a ring-based structured overlay ($O_{ring}$) and a tree-based structured overlay ($O_{tree}$). Each of them has a parameter $K$ specifying the number of neighbors. In our experiments, we set $K$=4 for $O_{unstructured}$, $K$=2 for $O_{ring}$ and $K$=3 for $O_{tree}$. Furthermore, $O_{tree}$ has an additional parameter $H$ specifying the number of links to other same-level nodes for avoiding the long repairing process. This paper evaluates the proposed cooperative strategy in various MOEs. These diverse environments are simulated through different combinations of three overlays, listed as $\{O_{unstructured}, O_{ring}\}$, $\{O_{unstructured}, O_{tree}\}$ and $\{O_{ring}, O_{tree}\}$. In each of combinations, there are two choices of master overlay, so six cases are all examined.

### B. Cooperative Failure Detection (CFD)

In this section, the performance evaluation of the CFD mechanism is presented and the reduction ratio of maintenance cost is used as the performance metric.

*Dynamics:* Each overlay must handle node joining and leaving; therefore, CFD is evaluated under various dynamic environments in terms of churn rates (e.g., $R$=0.002, 0.0008, 0.0004, 0.0002, 0.0001). The parameter $R$ is interchangeable with session time ($ST$) [15], so the above parameters are equal to $ST$=5.8, 14.4, 28.9, 57.8 and 115.5 minutes respectively. Fig. 3 shows that the reduction rate is positively correlated with the session time, and the CFD has performance degradation as the session time is small enough (e.g., < 10 minutes). To explain this, such a dynamic condition generates largely useless messages. This result echoes the maintenance model, as presented in section III-B.2. The smaller the session time is (i.e., the larger churn rate), the larger the overhead it produces. Nevertheless, the proposed CFD mechanism still benefits from decreasing the redundant overhead in various dynamic environments.

*Scalability on network size:* We also evaluate the CFD mechanism by varying the network size ($N$) ranging from 1000 to 10000. Fig. 4 reveals that the network size has no impact on the reduction rate of CFD. The reason is that the network size does not incur the degree of overhead, and thus, the reduction rate keeps unchanged in different network sizes.

*Probe frequency* (*PF*)*:* In general, as the probe frequency increases, the reduction ratio also increases because more redundant operations are saved. Fig. 5 shows the reduction ratio positively correlates to *PF*. This result shows that the proposed CFD is helpful to eliminate redundant failure detections, due to large amount of redundant overlay-maintenance operations.

*Number of cooperators* (*NC*)*:* This parameter plays an important role on the resilience of the CFD mechanism. This paper adopts the detection time of failed nodes and the survival rate of neighbors as metrics. The former represents the detection speed and the latter represents the detection accuracy. This evaluation takes a two-overlay environment as example in which $O_{tree}$ is the master overlay and $O_{unstructured}$ is the slave overlay. Fig. 6 demonstrates the evaluation of the detection speed, where the results show CFD ($NC$=2) outperforms the one without CFD. Regarding the detection accuracy, $NC$=2 also achieves the accuracy of detecting failures as well as ensures the resilience of CFD.

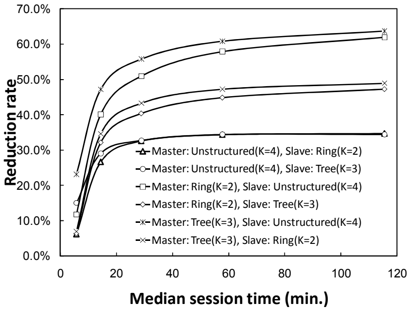


Fig. 3. Reduction rate when $N$=10000.

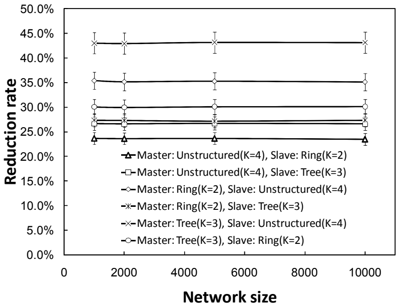


Fig. 4. Reduction rate when $ST$=11.6 minutes.

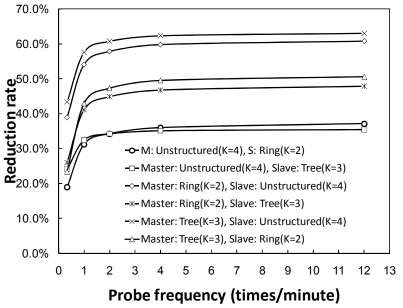


Fig. 5. Reduction rate when $ST$=57.8 minutes and $N$=10000.

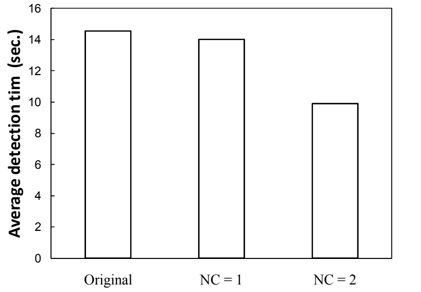


Fig. 6. Average detection time when $ST$=5.8 minutes, $N$=20000, and $PF$=2 times/minute.

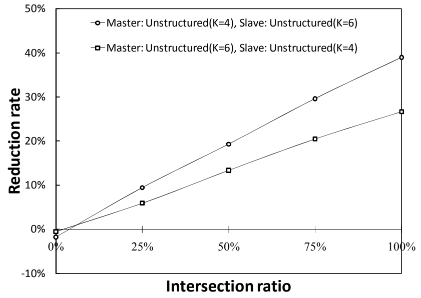


Fig. 7. Reduction rate when $ST$=11.6 minutes.

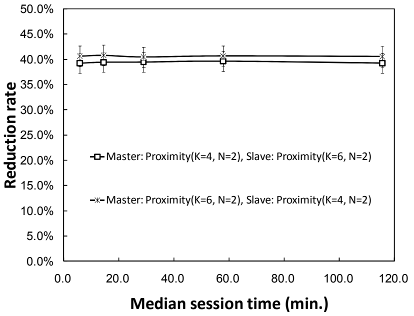


Fig. 8. Reduction rate when $N$=10000.

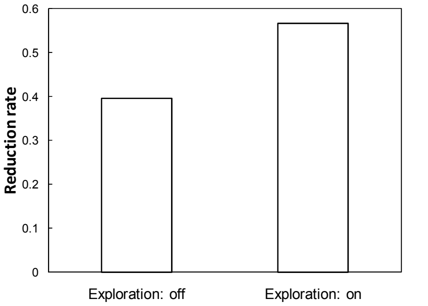


Fig. 9. Effect of the exploration method on the reduction rate.

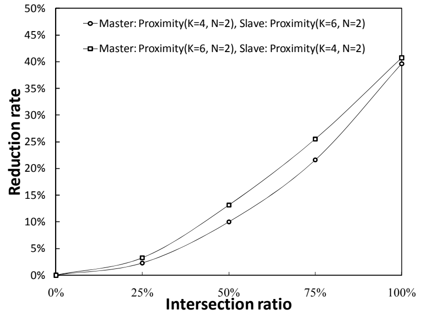


Fig. 10. Reduction rate when $ST$=11.6 minutes.

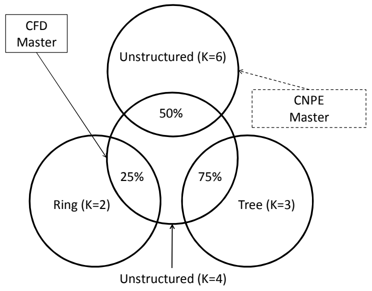


Fig. 11. Scenario of the hybrid cooperative maintenance.

*Intersection ratio:* Fig. 7 demonstrates the evaluation of intersection ratios (i.e., 0%, 25%, 50%, 75%, and 100%). The results present a positive correlation between the reduction rate and the intersection ratio. An interesting fact is that the intersection ratio leads a negative reduction rate. The reason is that most of $msg_i$ sent from $n_h$ to $n_t$ is useless when $n_t$ does not exist. However, this never happens because CFD would not be used if intersection ratio is 0%.

### C. *Cooperative Network-Proximity Estimation (CNPE)*

To evaluate the performance of CNPE, we have built two unstructured overlays with $K$=4 and $K$=6. This experiment does not take the structured overlay into account since it constructed the topology by hash key order instead of network proximity. Nevertheless, CNPE can be applied to structured overlays as well. The slave overlay may interact with the master via the query/response protocol. As a matter of simplicity, this paper only evaluates CNPE by two unstructured overlays. On the other hand, the estimation operation of the network-proximity executes when a new node joins or a participated node detects failed neighbors. The experimental results are presented as follows.

*Dynamics:* This experimental environment is very similar to that of the CFD experiment; however, the results are completely different. As shown in Fig. 8, the session time has no impact on the performance of CNPE. Contrary to CFD, the CNPE does not incur extra cost while the master overlay handles those requests.

*Scalability on network size:* The experimental results on various network sizes are similar to those of the CFD mechanism; hence, the experimental result is omitted.

*Exploration:* The exploration approach promotes the success rate of handling the request from a slave overlay. However, the exploration procedure produces extra overhead on sending $msg_e$. The message size of $msg_e$ is relatively smaller than that of network-proximity estimation. As in section III-C.2, the size ratio of $msg_e$ to the estimation message is $B$. In this experiment, we assume $B$ equals 10%. As shown in Fig. 9, the reduction rate is improved by nearly 20%.

*Intersection ratio:* The performance of CNPE under different intersection ratios is similar to that of CFD. However, the intersection ratio of 0% does not lead the negative reduction rate, as in Fig. 10. If the exploration approach is adopted, the reduction ratio can be negative due to extra $msg_e$, but yet this never happens since CNPE would not be used if intersection ratio is 0%.

### *D. Hybrid Cooperative Maintenance*

The above evaluations considered solely the two-overlay environment and one type of overlay-maintenance. The next experiment demonstrates the generality of master-slave model and evaluated under a realistic four-overlay environment. Additionally, this evaluation also shows CFD and CNPE can both co-work, as shown through four-overlay environment illustrated in Fig. 11 that includes an unstructured overlay ($K$=4), an unstructured overlay ($K$=6), a ring overlay ($K$=2) and a tree overlay ($K$=3). Note that each contains 1000 nodes, and the network size of the four-overlay environment is 2100 nodes, and the intersection ratios are varied. For the CFD mechanism, the unstructured overlay ($K$=4) is dedicated as the master overlay and the CNPE mechanism the unstructured overlay ($K$=6) acts as the master. Experiment results show that the reduction rate of CFD is 24.9% and that of CNPE is 14.5%, so the total reduction rate approximates 40%.

## V. Conclusions and Future Work

This paper addresses redundant operations and presents a cooperative strategy to handle the common overlay maintenance in diverse multi-overlay environments. Based on the master-slave model, CFD and CNPE enable co-existing overlays to eliminate redundant operations of failure detection and network-proximity estimation. Experimental results show our strategy significantly contributes with lowering overlay-maintenance costs as well as keeps performance, and the reduction rate is more than 60% in some cases. Moreover, a realistic four-overlay environment is considered in which the intersection ratio is not 100%, and the reduction rate approximates 40% in such an environment.

As future work, studies will focus on exploiting the synergy of other types of common overlay-maintenance in the multi-overlay environment. A comprehensive understanding on building an automatic system supporting the selection of the master overlay and applying to an implementation of the cloud computing are also considered.

## Acknowledgment

This paper is partially supported by National Science Council (NSC), Taiwan, under research grants NSC99-2218-E-007-013 and NSC99-2218-E-007-014.

## References

[1] "The Gnutella Protocol Specification," http://www.gnu.org/philosophy/gnutella.html.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report: UCB/EECS-2009-28, EECS, University of California, Berkeley, 2009.

[3] R. B. Ashwin, A. Mukesh, and S. Srinivasan, "Mercury: supporting scalable multi-attribute range queries," in *Proc*. SIGCOMM, August-September 2004, pp. 353-366.

[4] M. Cai, M. Frank, J. Chen, and P. Szekely, "MAAN: A Multi-Attribute Addressable Network for Grid Information Services," *Journal of Grid Computing,* vol. 2, no. 1, pp. 3-14, March 2004.

[5] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Exploiting network proximity in peer-to-peer overlay networks," Technical Report: MSR-TR-2002-82, Microsoft Research, 2002.

[6] B. F. Cooper, "Trading Off Resources Between Overlapping Overlays," in *Proc*. Middleware, November-December 2006, pp. 101-120.

[7] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research," *IEEE Internet Computing,* vol. 13, no. 5, pp. 10-13, September/October 2009.

[8] P. Grace, G. Coulson, G. Blair, L. Mathy, W. K. Yeung, W. Cai, D. Duce, and C. Cooper, "GRIDKIT: Pluggable Overlay Networks for Grid Computing," in *Proc*. DOA, October 2004, pp. 1463-1481.

[9] M. Jelasity, A. Montresor, Gian P. Jesi, and S. Voulgaris. "The {Peersim} Simulator," http://peersim.sf.net.

[10] W. Jiang, D. M. Chiu, and J. C. S. Lui, "On the Interaction of Multiple Overlay Routing," *Performance Evaluation,* vol. 62, no. 1-4, pp. 229-246, October 2005.

[11] M. Kwon, and S. Fahmy, "Synergy: An Overlay Internetworking Architecture," in *Proc*. ICCCN, October 2005, pp. 401-406.

[12] S. Lin, F. c. Ta¨ıani, and G. Blair, "Exploiting Synergies between Coexisting Overlays," in *Proc*. DAIS, June 2009, pp. 1-15.

[13] B. Maniymaran, M. Bertier, and A. M. Kermarrec, "Build One, Get One Free: Leveraging the Coexistence of Multiple P2P Overlay Networks," in *Proc*. ICDCS, June 2007, pp. 33.

[14] A. Rowstron, and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proc*. Middleware, November 2001, pp. 329-350.

[15] R. Sean, G. Dennis, R. Timothy, and K. John, "Handling Churn in A DHT," in *Proc*. USENIX, June-July 2004, pp. 127-140.

[16] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Transactions on Networking,* vol. 11, no. 1, pp. 17-32, February 2003.

[17] C. Wu, B. Li, and Z. Li, "Dynamic Bandwidth Auctions in Multioverlay P2P Streaming with Network Coding," *IEEE Transactions on Parallel and Distributed Systems,* vol. 19, no. 6, pp. 806-820, June 2008.

[18] Z. Xin Yan, Z. Qian, Z. Zhensheng, S. Gang, and Z. Wenwu, "A construction of locality-aware overlay network: mOverlay and its performance," *IEEE Journal on Selected Areas in Communications,* vol. 22, no. 1, pp. 18-28, January 2004.

[19] M. Yun, L. Boon Thau, I. Zachary, and M. S. Jonathan, "MOSAIC: Unified Declarative Platform for Dynamic Overlay Composition," in *Proc*. CoNEXT, December 2008, pp. 1-12.

[20] S. Q. Zhuang, D. Geels, I. Stoica, and R. H. Katz, "On Failure Detection Algorithms in Overlay Networks," in *Proc*. INFOCOM, March 2005, pp. 2112-2123.