The Design and Simulation of Service Recovery Strategy Based on Recovery Node in Clustering Network

Hecun Yuan
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
yuanhecun@bupt.edu.cn

Biyao Li
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
libiyao@bupt.edu.cn

Huilin Liu
Industry Corporation Limited Network Information Department
China Aerospace Science
Beijing, China
qcgl_168@163.com

Yi Han
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
hanyi@bupt.edu.cn

Lanlan Rui
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
llrui@bupt.edu.cn

Ying Wang
Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
wangy@bupt.edu.cn

***Abstract*—In order to ensure users enjoying the services continuously and steadily, we need an efficient service recovery strategy to quickly recover the failed links and reconstruct the device set. In this paper, we introduce a service recovery strategy based on recovery node which can save and maintain service data flexibly. First, we give the definition of recovery node and the selection mechanism for it. Then we describe our recovery strategy in detail. At last, we make a simulation by NS-3. The effectiveness of the proposed methods is demonstrated by simulation results.**

Keywords—ubiquitous stub environment, service recovery service-oriented, NS-3 simulation

I. INTRODUCTION

In the ubiquitous environment [1], a single smart device has poor processing power with memory constraint and is not enough to support some services. In order to meet user's need for abundant services, multi-device cooperation is introduced by researchers to provide users with composite services anytime and anywhere. However, in this environment, network topology is dynamic, bandwidth and energy are limited, and thus those composite services are prone to failures. Therefore, it is of great importance to develop recovery strategies for composite services.

The existing research on recovery strategies for composite services mainly includes two aspects: backup device replacement strategy and retransmission strategy.

The backup strategy [2][3][4] sets up a backup method for the unstable or failed service and when the service interruption happens, a backup device can replace the device should have executed the service. Chen [5] proposed a dynamic monitor based service recovery strategy (DMBSRS) to reduce response time of services.

The retransmission strategy [6] lets the service provider re-execute the service or retransmit the result which has already saved if there is failure during the service transmission. Chen [7] introduces some typical global recovery strategies: atomic service timer with resending strategy (ASD) and atomic service timer with resending and cache strategy (ASDC).

The most researches give every device its due and leave out of consideration about difference among smart devices. Actually, devices in the ubiquitous environment are heterogeneous and there exists different computing power, storage capacity and so on. Therefore we try to assign different roles for different devices. Our contributions are mainly of two folds. First, we make full use of features in clustering network and set up a service recovery model. Moreover, we design a service recovery strategy based on recovery nodes to reduce the average service response time.

The outline for the rest of paper is as follows. In section II, we introduce the specific application scenario and abstract a service model. Next, we propose a service recovery strategy based on recovery nodes in section III. The simulation is given in section IV and the result shows that our strategy significantly improve the performance of service recovery. Finally, we conclude and summarize the future research in section V.

II. SERVICE MODEL

First, we give a specific application scenario to understand the clustering network architecture and how we apply the clustering network to the recovery of composite services.

In the disaster succor [8], the whole rescue task is made up of many sub-tasks and it needs to complete over multiple division of labor. This organization is similar to the clustering network architecture. These sub-tasks is similar to the atomic services in a composite service. The group member like the node in cluster provides information and executes task, and the group manager like the cluster head masters every member's location and how the tasks perform. When completing a composite task, with the mobility of members, some information is lost and it leads to the failure of executing task. Therefore, how to restore missed information timely in the process of executing task is a problem to be solved.

We abstract a model with service layer and network layer [9][10] as is shown in Fig.1.The service layer shows all the services that smart devices in the network can provide. The network layer shows the clustering network that smart devices make up and there are 5 clusters in the network. The correspondence between device and service is not one-to-one.

When requesting for a composite service, there is a process of service discovery and selection as follows. For a cluster, the cluster head has all the service information provided by every node in this cluster. When a node requests a composite service (e.g. The node in cluster 1 requests atomic service S1, S2, S3.), it sends the request to the cluster head of cluster in which it sits. After receiving the request, the cluster head will check if there is any node can provide these atomic services.

If so, the cluster head forwards request to the provider (e.g. the green node in cluster 1 provides S2.). But if not, the cluster head forwards this request to another cluster head of its adjacent cluster (e.g. the cluster head in cluster 2.). After such a process, we can know the green node in cluster 3 provides S1, the green node in cluster 1 provides S2 and the green node in cluster 4 provides S3.We assume that atomic services in the composite service don not need orderly executive process. And the cluster head of cluster 1 can return the execution results of all atomic services back to the requester by multiplexing.

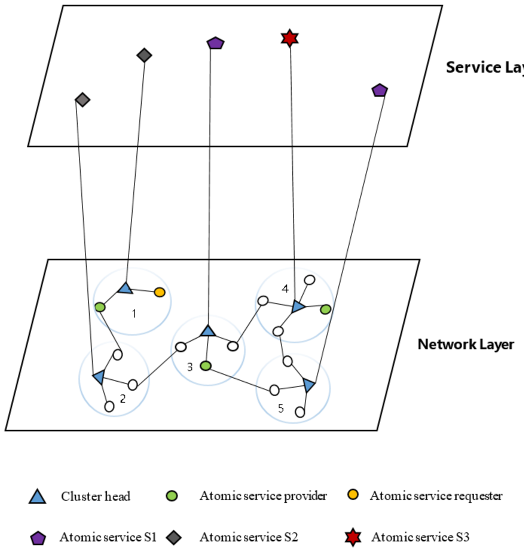

Fig.1. Service recovery model

III. SERVICE RECOVERY STRATEGY

A. Definition and setup of recovery nodes

According to the model introduced in Section II, we can see that the transmission path for atomic service S3 is relatively long, and if the path is broken, there may be a longer retransmission delay which effects the quality of service recovery. Consequently, we propose that setting up a series of relay nodes [11] to save data dynamically for retransmission. We define these nodes as recovery nodes, they are deployed in the transmission path and used for saving and forwarding the service data. Considering that the recovery node needs to save and forward data, cluster head is a proper choice for recovery node. Besides, the number of recovery nodes should be within limits. In this paper, we try to give an evaluation mechanism to select some of cluster heads in the transmission path. Only the evaluation reaches a certain value, the cluster head could be the recovery node. The evaluation formula for cluster heads is shown in (1)

$$F_i = (w_1 S_i + w_2 E_i + w_3 C_i + w_4 M_i) \quad (1)$$

$$\sum_{i=1}^{4} w_i = 1 \quad (2)$$

There are 4 evaluation indexes: S_i is denoted as the ability of the cluster head I to send data, the higher value of S_i means the higher reliability of cluster head i. It can be measured by the number of hops between the head and its adjacent nodes in cluster. M_i is denoted as the mobility of the cluster head i, and the lower speed of cluster head relative to nodes in cluster means the cluster is more stable. E_i is denoted as the energy of the cluster head i. C_i is denoted as processing capacity of cluster head. We choose entropy method to update the weight coefficient. We assume that weight coefficient should be updated after m times' interruption. According to the history data of evaluation index in m times, we calculate the new weight coefficient.

Moreover, we try to set up a threshold value V and only when the $F_i > V$, the cluster head could be the recovery node. Different value of V can set up different number of recovery node, therefore the V should satisfy the recovery strategy with better performance, and we would describe how to get a proper V in the latter Section IV with more detail.

B. Service recovery strategy

In the above parts, we define the meaning of recovery node and how to set up it. In this part, we explain our recovery strategy in detail.

We will introduce the process of requesting service and service transmission first. The requester requests the node which is the cluster head in its own cluster for one or more services simultaneously. Next, the cluster head will check if there are any smart devices which can provide these services. If exists, the cluster head requests the devices that provides the requested services and theses devices send the service data to requester. If not, the cluster head forwards the request to the other adjacent cluster heads.

According to the above description, we can set up several recovery nodes along the paths from the requester to the provider. We divided the failure that happens in the process of service data transmission into two types. They are the failure between recovery nodes and the failure of provider or recovery nodes.

With regard to the failure between recovery nodes, our processing flow is shown in Fig.2. If the nodes does not receive the ACK message until the timer T_1 expires, and then we can think the link is likely to fail and the nodes can retransmit the data has already saved to the downstream.

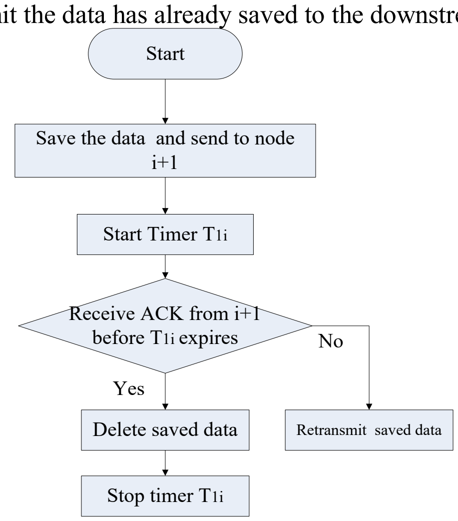

Fig.2. processing flow for failure between recovery nodes

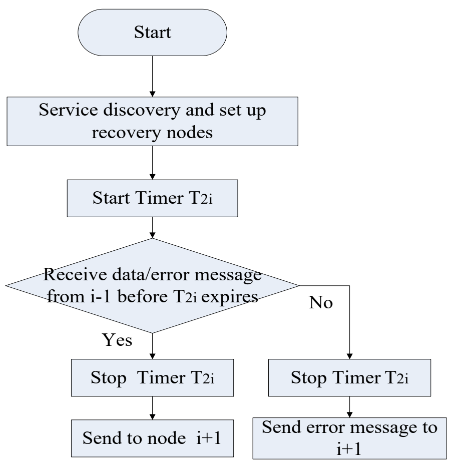

Fig.3. processing flow for failure of recovery nodes/provider

However when the provider or recovery nodes themselves fail, the above process is useless. If we just use timer deployed to the requester to discover the failure, it is too late. Therefore, we use recovery nodes to send failure messages. The processing flow is shown in Fig. 3.

Besides, as for requester, the process is shown in Fig.4.It is similar to the Fig. 3.The difference is that the requester needs to judge the type of received result. And if requester receives error message, it will stop timer and restart service discovery and composition.

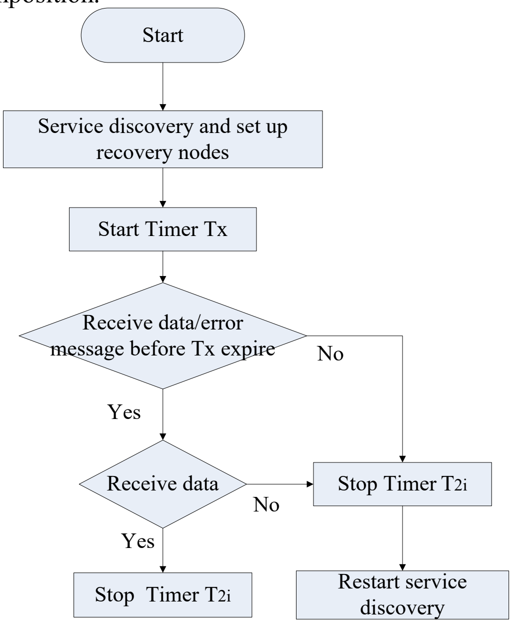

Fig.4. processing flow for requester

C. Overhead analysis of recovery nodes

Recovery nodes need spend time to handle received data, and setting up recovery node leads to the increase of service transmission delay. There should be a compromise between transmission delay and retransmission delay. In other words, the number of recovery nodes should have an optimal value in a transmission path. Here we give a particular situation to analyze.

Assuming that there are m recovery nodes in the transmission path of an atomic service. The probability of failure between every two adjacent recovery nodes is the same. The notations used to describe service transmission are listed by TABLE I.

TABLE I. Notations used to describe service transmission

Symbol	Meaning
m	The number of recovery nodes.
d	The transmission delay between provider and requester without any failure.
t	Time to handle the received data for every recovery node.
T1	Timer of provider.
T2	Timer of recovery nodes.
p_1	The probability of failure between two adjacent nodes.
P	The probability of failure in the whole path.

The relation between P and p_1 is

$$(1-p_1)^{m+1} = 1-P \qquad (4)$$

Definition: T_{sim} is the average transmission delay with simple retransmission way. T_{rep} is the average transmission delay with recovery node.

$$T_{sim} = \mathrm{PT1} + \mathrm{d} \qquad (5)$$

$$T_{rep} = (\mathrm{m}+1) * [1-(1-P)^{\frac{1}{m+1}}] * \mathrm{T2} + \mathrm{d} + \mathrm{m} * \mathrm{t} \qquad (6)$$

Let $T_{sim} = T_{rep}$, then,

$$k_1 * \left[1-(1-P)^{\frac{1}{m+1}}\right] - \mathrm{P} + k_2 \mathrm{m} = 0 \qquad (7)$$

$$k_1 = \frac{(m+1)T2}{T1} \geq 1 \quad 0 < k_2 = \frac{t}{T1} < 1$$

Corollary: If m = 1, $k_2 \leq \frac{P-1+(1-P)^{\frac{1}{2}}}{3-2*(1-P)^{\frac{1}{2}}} = c$, then always has $T_{sim} > T_{rep}$.

We can know when the percentage of the time to save data among timer is lower than c, recovery node strategy is necessary.

IV. Simulation Analysis

In this section, we evaluate the performance of our approach. In part A, we compare it with the simple retransmission recovery strategy in the same network. Moreover, we draw the graph of two functions in Part B with different parameters to have a better understanding of relation between recovery node and service execution time. We can get the proper number of recovery node in the specific scenario and this number value can be the reference value of V in Section III part A.

A. Simulation of recovery strategy

During the process of data transmission, the less packet loss means the higher reliability of transmission. And the less average service response time accounts for less effect caused by service recovery. In this paper, we use packet loss (PL) and average service execution time (ASET) to measure the performance of the different recovery strategies. ASET contains the normal execution time and the recovery time of service. And it is effected by retransmission timer.

All the experiments in this paper run on 25 mobile routing nodes in the scope of 400m*400m with NS-3.The wireless transmission radius is 100m.The nodes adopt RandomWalk2dMobility model and the speed of node goes from 2m/s to 20m/s.

Fig.5 shows the PL of different service recovery strategies with different speeds. As we can see, when speed of node goes from 2m/s~4m/s, there is no packet loss .Along with the increase of the speed of node, the PL of two strategies also increase. The decrease of PL when speed goes from 8m/s to 10m/s and 12m/s to14m/s is supposed to be reasonable. Because the nodes may move closer to each other. Besides, the PL of all speeds are overall lower than the simple strategy.

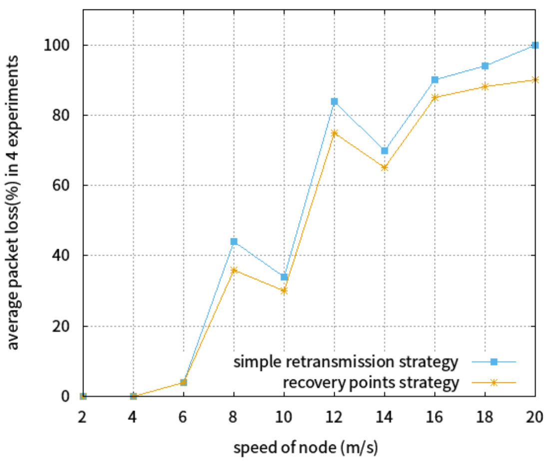

Fig.5. The packet loss of different speeds of node

Fig.6 shows comparison results of ASET of the two strategies. They are nearly equal when speed varies from 2m/s-6m/s and meet the change of PL. With the increase of speed, ASET of our strategy is slowing the growth and the gap between two strategies has becomes bigger. We consider the overall performance of our strategy is better than simple retransmission strategy.

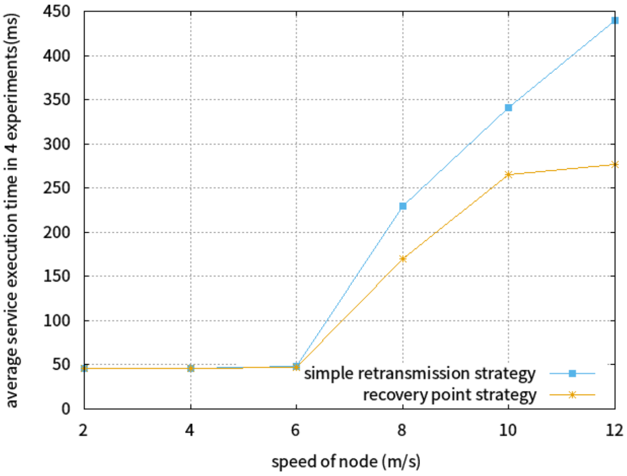

Fig.6. The ASET of different speeds of node

In conclusion, our strategy has a better performance than simple strategy in two indexes and gain more obvious advantages when speed of node is higher.

B. Overhead analysis of recovery node

In this part, we use MATLAB to draw graphs of two functions in section III, part C. We set various parameters to compare two strategy, Simple Retransmission Strategy (SRS) and Recovery Node Strategy (RNS). We set d (transmission delay) as 200ms, P (the probability of failure in the whole path as 0.5 and 0.9), and t (Time to handle the received data for every recovery node) as 2ms.

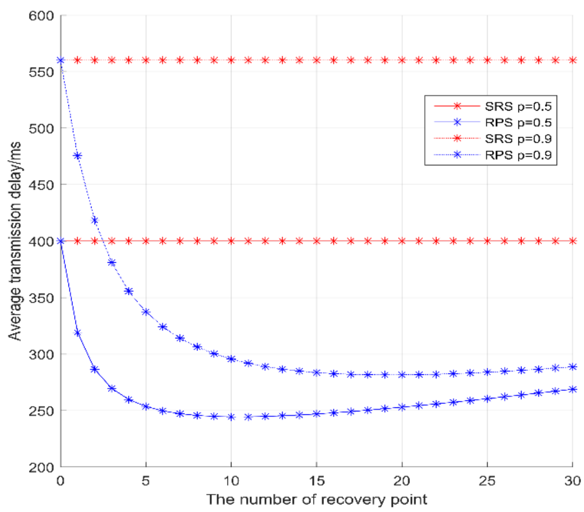

Fig.7. The number of recovery node and average transmission delay

The Fig.7 shows that with the increase of number of recovery node, the average transmission delay trended down at first, and then increased. The down trend is more apparent for higher probability of failure. Take the P=0.5, t =2ms for example, m=10 is the best number of recovery node.

When we set d = 200s, P=0.5, m=4 and 10, from Fig.8, we can see the influence of t to average transmission delay is linear. With m =10, t>= 15ms, recovery node strategy decreases its position. Therefore, ideally, the smaller value of t is better. Actually, when t decrease to some value, we can get an optimal number if recovery nodes. Besides, in above situation, the tolerable t is 80ms.

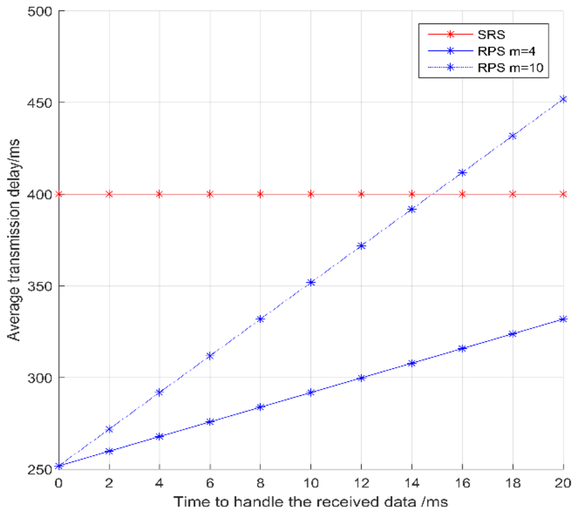

Fig.8. The time to handle data and average transmission delay

V. Conclusion and Expectation

In this paper, we studied the problem of service recovery. We propose a service recovery strategy and gain an advantage over simple retransmission method on overall performance. However, there are some deficiencies in our simulation. For example, we did not combine the recovery strategy with the selection mechanism which happens in the process of service discovery and composition, so it is still non-validated that whether our selection is reasonable. In our future work, we will try to improve it.

References

[1] Ping Z, Yang J, Feng Z. Mobile Ubiquitous Service Environment (4)[J]. ZTE Communications, 2007(3):55-59.

[2] Zhou X , Shi J , Xu Y , et al. A Backup Restoration Algorithm of Service Composition in MANETs[C]// IEEE International Conference on Communication Technology. 0.

[3] Jeon J , Lee K , Kim C . Fast route recovery scheme for Mobile Ad Hoc Networks[C]// Information Networking (ICOIN), 2011 International Conference on. IEEE, 2011.

[4] Su W T , Kuo Y H , Huang P C . A QoS-driven approach for service-oriented device anycasting in ubiquitous environments[J]. Computer Networks, 2008, 52(18):3342-3357.

[5] Chen W , Xu Y , Peng B , et al. Dynamic Monitor Based Service Recovery for Composite Service in MANETs[C]// IEEE International Conference on Communication Technology. 0.

[6] Ouyang B , Hong X , Yi Y . A comparison of reliable multicast protocols for mobile ad hoc networks[C]// Southeastcon, IEEE. IEEE, 2005.

[7] Chen W , He Z , Ren G , et al. Service Recovery for Composite Service in MANETs[C]// International Conference on Wireless Communications. IEEE, 2008.

[8] Bessis N , Asimakopoulou E , Conrad M . A mathematical analysis of a data-grid push service for disaster management response scenarios[M]// A Mathematical Analysis of a Data-Grid Push Service for Disaster Management Response Scenarios. 2009.

[9] Jiang S , Xue Y , Schmidt D . Disruption-aware service composition and recovery in dynamic networking environments[C]// Workshop on Automating Service Quality: Held at the International Conference on Automated Software Engineering. ACM, 2007.

[10] Jiang S , Xue Y , Schmidt D C . Minimum disruption service composition and recovery in mobile ad hoc networks[J]. Computer Networks, 2009, 53(10):1649-1665.

[11] Li J S , Liu I H , Kao C K , et al. Intelligent Adjustment Forwarding: A compromise between end-to-end and hop-by-hop transmissions in VANET environments[J]. Journal of Systems Architecture, 2013, 59(10):1319-1333.