Advance Access publication on 3 June 2014
doi:10.1093/comjnl/bxu044

A Composition and Recovery Strategy for Mobile Social Network Service in Disaster

DANMEI NIU[1,2], LANLAN RUI[1], CHENG ZHONG[3] AND XUESONG QIU[1,*]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China*
[2]*Information Engineering College, Henan University of Science and Technology, Luoyang, China*
[3]*State Grid Hebei Electric Power Company of China, Shijiazhuang, China*
**Corresponding author: xsqiu@bupt.edu.cn*

Mobile social network service (MSNS) provides daily services for the user and can also be used in emergencies, such as natural disasters. How to conduct service composition and recovery among mobile devices quickly and efficiently is one of the important research areas of MSNS. This paper puts forward a comprehensive strategy applied to MSNS during natural disasters. When communication facilities are limited, several devices can work cooperatively to provide users with reliable composite service, also known as the service composition process. In addition, when some of the devices fail and the composite service interrupts, the presented recovery process reconstructs a service path quickly. Composition and recovery cost functions are used in the two processes separately. The goal is to find the service path or the recovery path with minimal cost function value in each process that satisfies the quality-of-service requirement. The simulation results show that the proposed strategy not only reduces the interrupt number and recovery time but also improves the success rate of the service request, making the performance of this strategy better than that of the other similar strategies.

Keywords: mobile social network service; composition; recovery; disaster

Received 14 September 2013; revised 14 April 2014
Handling editor: Zhiyong Zhang

1. INTRODUCTION

With the rapid development of mobile internet and smart phone, mobile social network service (MSNS) has become one of the hot development spots [1–4]. Various new mobile devices play important roles, which constitute MSNS by realizing local information sharing and producing abundant services [5–7]. In addition to providing daily services for the user, MSNS can also be used in some emergencies, such as natural disasters. It benefits the rapid spread of information and the formation of efficient decision-making.

Under the condition of the communication facilities damaged, rescue workers need to communicate through mobile devices. That is, multiple devices work cooperatively to provide users with reliable composite service. However, device failure or movement will cause composite service interruptions. The main problem, this paper discusses, therefore, is how to conduct service composition and recovery quickly and efficiently.

The existing typical solutions of this problem are resending service request strategy [8, 9] and backup service replacement strategy [10–13]. Based on the above two strategies, this paper puts forward a new kind of service composition and recovery strategy (SCRS). The simulation experiments show that this strategy can improve the success rate of the service request and reduce the recovery time efficiently.

2. BACKGROUND AND RELATED WORK

MSNS is the network in which participants with similar interests or goals interact with one another through their mobile devices [14–16]. MSNS is different from conditional social network services in which people interact via internet [17]. Each mobile device can act as host, router or gateway and connect with other devices in an *ad hoc* mode [18]. So MSNS can be applied to some special application scenarios, such as earthquakes, hurricanes, floods and some man-made disasters, such as fire and terrorist attack.

When a disaster occurs, the infrastructures and communication facilities are usually damaged. Rescue workers need to communicate and cooperate via mobile devices, such as smart phone, personal digital assistant and so on. To make statistics and support decision-making, a rescue manager needs accurate information provided by each rescue worker, making MSNS very attractive to such rescue personnel. Further, the research of service composition and recovery applied to dynamic network (such as MSNS) is very important.

The existing research is mainly for traditional static networks. A typical strategy for wireless or dynamic network is resending service request strategy [8]. A service is divided into several atomic services. The user selects some of the atomic services to constitute. When the composite service interrupts, the service requester will resend a request. In the process of re-executing service composition, caching technology is used to ensure the service quality.

Based on [8], Chen et al. [9] puts forward an improvement, which uses dynamic monitor. The service execution process is controlled by each monitor, rather than unified service requester. However, the dynamic network topology is unstable, some services provided by devices will never be used. If the service requester or service provider is always in a wait state, the network load and service execution time will greatly increase, and the availability and reliability of service composition cannot be guaranteed.

The other typical strategy is backup service replacement strategy. The principle of this strategy is when service interrupts, replacing the failing node with a backup node. Sun *et al.* [10] presents cold backup service replacement strategy (CBSRS) and hot backup service replacement strategy (HBSRS). In CBSRS, each node executes service in sequence according to the user's requirements. When a service node fails, the strategy sorts the available backup nodes with an algorithm. The node with the highest priority will be selected. In HBSRS, the backup service nodes are sorted continually whether or not the service interrupts. The backup node is ready to replace the failing node at any time. Compared with CBSRS, HBSRS reduces repair time, but adds cost. However, this reference does not specify how to calculate the priority of the backup service nodes, and the algorithm does not apply to the condition of most of the nodes failing.

Jiang *et al.* [11, 12] put forward a double framework of service composition and recovery, which consists of a service layer and network layer. Service composition and recovery need to coordinate between the two layers. The two references establish an evaluation mechanism about interrupt index. Through minimizing interrupt index and forecasting service link lifetime, the strategy makes the shortest recovery time and the longest lasting new path. This strategy, however, selects the service path through the substitutive node number between the old and new paths, which is unreliable.

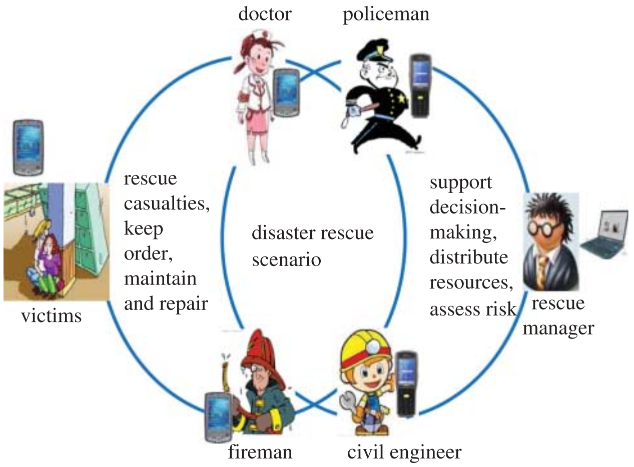

FIGURE 1. A disaster rescue scenario.

Lin *et al.* [13] proposes an iterative algorithm. It produces a reconfiguration area for each failing service process. The algorithm reconfigures service in the chosen area rather than in the whole service process. If the scale of the network is bigger, the reconfiguration area increases greatly, and the efficiency of the algorithm is reduced.

3. APPLICATION SCENARIO AND SYSTEM MODEL

3.1. Application scenario

Because the disaster situation changes very quickly and unpredictably, it is impossible to forecast every possible condition and related service requirements before or after the disaster. In a disaster condition, reliable internet access may be unavailable. However, there are many rescue tasks and cooperation applications needing to be executed dynamically. Users can communicate with adjacent people with mobile devices via MSNS. We use Fig. 1 to illustrate a disaster rescue scenario.

When a disaster happens, some victims will be concerned about their location, some will care about the nearest rescue center and some will have an urgent need to contact the nearest rescue workers. Rescue workers are not only emergency service people, such as doctors, firemen and policemen, but also civil engineers. These people rescue casualties, maintain social order, maintain and manage destroyed infrastructure. There are a large number of service requirements and cooperation between rescue workers and victims who need to communicate via mobile devices. To make statistics and analysis, support decision-making, allocate resource and assess risk, the rescue manager needs accurate information provided by the rescue workers, such as pictures, video and related data, which can be provided by MSNS.

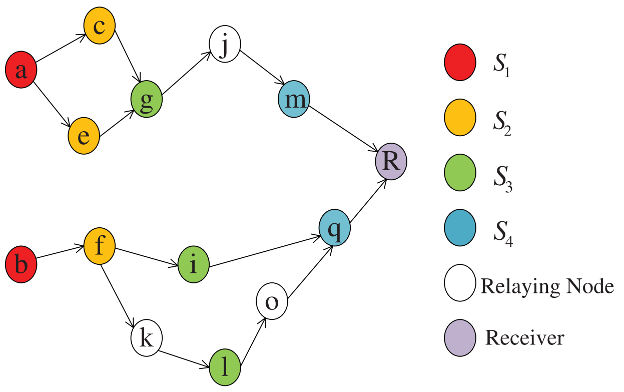

FIGURE 2. System model.

3.2. System model

Mobile *ad hoc* network (MANET) is a kind of typical network mode applied to MSNS. MANET is a peer-to-peer infrastructure-less communication network formed by short-range wireless enabled mobile devices. When a user sends out a composite service request, several services will be finished in order, and the result will be sent to the service receiver. Choosing the appropriate device for each service to form a best service path is the first problem we want to solve. In addition, in MANET, nodes have mobility, and network topology will change with the movement of the nodes. In the process of providing composite service for the user, service will probably be interrupted. How to reduce service interruptions, recover the service quickly and efficiently, provide high-quality service to user is the second problem we want to solve. Therefore, we set up a system model to solve these problems.

Figure 2 is the system model. Each node with one color represents that a device can provide a service. The node without color is a relaying device which does not provide service, only transmits data. Composite service refers to several nodes cooperate to provide service for user. When data flow through nodes, they will perform some operation, such as data processing or relaying. All the nodes connect to form a directed acyclic graph, known as the system model.

There are many kinds of executive ways of composite service. Figure 3a shows that service is executed by sequence, that is, $S_1 \rightarrow S_2 \rightarrow S_3$. Figure 3b shows that service is executed in parallel, that is, $S_1 \rightarrow S_3$ and $S_2 \rightarrow S_3$ being executed simultaneously. We mainly consider unicast transmission service in this paper, which is shown in Fig. 3a. The devices connect by sequence rather than in parallel or other ways.

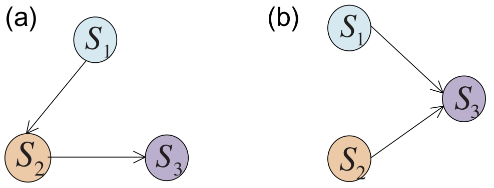

FIGURE 3. Executive ways of composite service.

Assume that the user knows how to decompose a composite service into a number of atomic services in advance. As described earlier in the scenario, if a user wants to print disaster scene images provided by someone else's mobile phone, the service request can be decomposed into three atomic services: download, unzip and print. Therefore, after sending the service request, first, the user needs to find several devices which can be used for each atomic service. The routing process depends on service discovery to find candidate devices. Owing to the service discovery not being the focus in this paper, we do not explain more here.

These atomic services will be executed in accordance with order, and the result will be sent to the service receiver. Such a composite service is called a service path. A service path can be expressed as $P = (S_1, S_2, \ldots S_i, \ldots S_K, R)$, S_i represents a service, R represents a receiver, K is the number of services in a path. Suppose that each node only support a service. Then, a service S_i can be represented by a set V_i which includes several nodes. For example, in Fig. 2, $V_1 = \{a, b\}$, $V_2 = \{c, e, f\}$, service S_i provided by node v_i can be expressed as $S_i[v_i]$, and the service path can be expressed as $P = (S_1[v_1], S_2[v_2], \ldots S_i[v_i], \ldots, S_K[v_K], R)$.

Take Fig. 2 as an example for the construction process of a service path. First, the user sends a service request to node a. When a receives the service request, it will finish service S_1 and send the service request to the next node c. Secondly, c does the same work as a, and nodes g and m do the same work continuously, j only transmits. Finally, node m will send the result to service receiver R, the service path can be expressed as $P_1 = (S_1[a], S_2[c], S_3[g], S_4[m], R)$. A service link (S_i, S_{i+1}) may consist of one or more network links, for example, service link $(S_3[g], S_4[m])$ consists of network link (g, j) and (j, m).

4. COST FUNCTION AND QUALITY-OF-SERVICE REQUIREMENT

4.1. Service process

According to time sequences, the service process can be divided into the following parts.

Service initialization process. A user sends a service request and sets the quality-of-service (QoS) requirement which mainly includes delay constraint of the service path. Then the service composition process is triggered.

Service composition process. This is a process of searching for a service path to satisfy the user's request and the specified QoS requirement. First of all, it should find several devices that can be used for each service, then select the appropriate devices to form a service path, which needs to use an appropriate algorithm.

Service interruption process. Service interruption is due to many reasons. For example, network topology changes with device movement, link fails in wireless communication path, and overloaded service and extended communication time may violate the QoS requirement. This paper mainly considers service interruption caused by device failure or movement.

Service recovery process. To provide the user with continuous service, if a service interrupts, the service path must be recovered immediately. The service recovery process is actually reconstructing a service path, in other words, finding new devices to replace failing devices and building a new path.

4.2. Cost function

Cost function consists of composition and recovery cost function. The composition cost function is used in service composition process. The recovery cost function is used after service interruption, which is more complicated. We discuss the recovery cost function first.

4.2.1. Recovery cost function

When a service interrupts, how to choose the best substitutive devices to constitute the best service path and provide the user with continuous service, we need to weigh the consumption and time problems involved in the recovery process. These problems should be described and calculated quantitatively, so we set up a comprehensive recovery cost function. There are many factors considered in the recovery cost function, such as the substitutive device number, the distance between the substitutive device and the failing device, the service path delay and the failure rate of device.

Suppose that there is a period of time T that is from a user sending a request to receive the result. First, recovery cost function can be achieved by the substitutive device number which is denoted by N. At time $t_i \in [0, T]$, N is the different devices between the new service path and the original service path. The substitutive devices include new devices providing services and transmitting data packets. Second, we must consider the distance h between the substitutive device and the failing device when calculating N. In this paper, the distance h will be calculated according to the hop. The calculating method of N is as follows: if h is 1, then N is 1. If h adds 1, then N adds μ (μ is usually set to 0.1). Every substitutive device should be calculated in this way, then all N add up. So we can get the substitutive device number $N(P(t_i))$ which is expressed in formula as follows:

$$N(P(t_i)) = \sum_{j=1}^{n} [1 + \mu(h_j - 1)]. \tag{1}$$

Here, $P(t_i)$ is the recovery service path at time t_i, n is the total substitutive device number, $h_j (h_j \geq 1)$ is the distance between the substitutive device and the failing device, namely hop. Calculating formula (1) will get substitutive device number of a candidate recovery service path at time t_i. If the number is smaller, the recovery cost is smaller.

Service path delay is another factor when calculating the recovery cost function. The calculation method is as follows: $d(v_i)$ is the calculation delay of device v_i, $d(v_i, v_j)$ is the link delay between device v_i and v_j. At time t_i, the delay calculation method of a candidate recovery service path is expressed as follows:

$$D(P(t_i)) = \sum_{j=1}^{K-1} [d(v_j) + d(v_j, v_{j+1})] + d(v_k) + d(v_k, R). \tag{2}$$

Device v_j measures the link delay to the next device v_{j+1}, K is the number of services in the new path. The calculation delay and the link delay will all be sent to the service receiver. On the premise of guaranteed service quality, if the delay value of a service path is smaller, the service provided to the user is faster.

In addition to the above two factors, service failure rate $R(t_i)$ should also be considered in choosing the best recovery service path. If a device failure means a device leaving or providing a service failed, we set a failure rate r for each device. At time t_i, the failure rate of a service path can be obtained through averaging all the device failure rates. If the failure rate is smaller, the service path is more reliable, and the interrupt possibility is smaller. The formula is expressed as follows:

$$R(P(t_i)) = \frac{1}{K} \sum_{j=1}^{K} r_j. \tag{3}$$

So we can get the recovery cost function of a candidate recovery service path through formula (1)–(3). When a service interrupts, the recovery cost function at time $t_i \in [0, T]$ is shown as follows:

$$\begin{gathered} F(P(t_i)) = \alpha N(P(t_i)) + \beta D(P(t_i)) + \gamma R(P(t_i)), \\ \alpha \geq 0, \beta \geq 0, \gamma \geq 0, \alpha + \beta + \gamma = 1. \end{gathered} \tag{4}$$

The goal of setting the recovery cost function is that when seeking a recovery service path, if the cost value is smallest, the interruption will have a minimal impact to the user. We should choose the service path with the smallest cost value as the best path.

4.2.2. Composition cost function

The service composition process is choosing a new service path, which does not involve service interruption and recovery. So we need to change the recovery cost function formula (4) when calculating the cost value of a service path. We only consider the delay and the failure rate of a service path and do not calculate the substitutive device number. We use formula (5) to calculate the cost value of a service composition process. Formula (5) is called the composition cost function.

$$F(P(t_i)) = \beta D(P(t_i)) + \gamma R(P(t_i)), \quad \beta \geq 0, \quad \gamma \geq 0, \quad \beta + \gamma = 1. \tag{5}$$

4.3. QoS requirement

The user who requests a service will require the service path satisfying certain QoS requirement. We consider the delay requirement D^* of a service path as the QoS metrics in this paper.

In each period of time $[t_i, t_{i+1})$, selecting or restructuring a service path must satisfy the QoS requirement, then the qualified path can serve as a candidate path. The cost function can be calculated and compared in the next step. In Fig. 2, suppose that the path delay requirement D^* is 7, all the link delay is 1 except for $d(e, g) = 4$, all the calculation delay of devices is zero. The delay of a service path $P_1 = (S_1[a], S_2[c], S_3[g], S_4[m], R)$ is 5, which satisfies the delay requirement D^*. If device c of path P_1 fails at this time, we cannot choose e as the substitutive device. Because the delay of service path $P_2 = (S_1[a], S_2[e], S_3[g], S_4[m], R)$ is 8, which is $> D^*$, the delay does not satisfy the requirement. The service path P_2 cannot act as a candidate path and compare with other service paths.

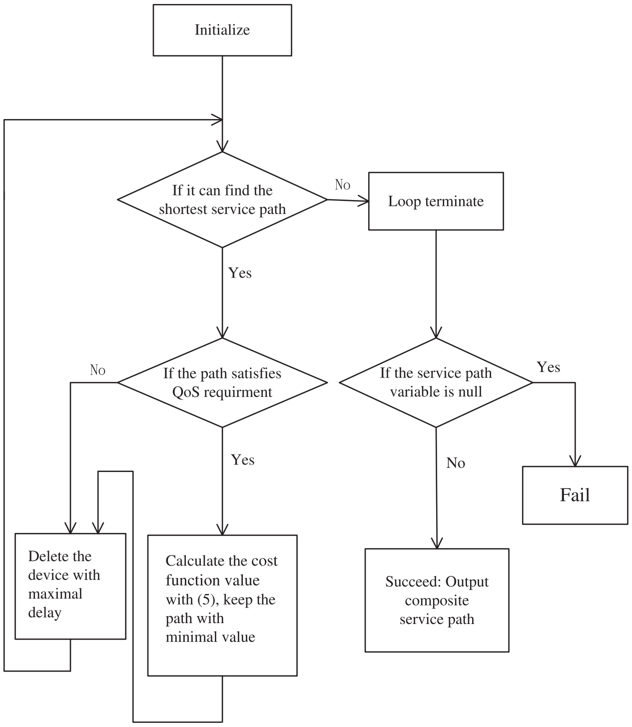

FIGURE 4. Service composition algorithm flow.

5. SERVICE COMPOSITION AND RECOVERY STRATEGY

5.1. Service composition strategy

When a user sends a service request, he triggers the service composition process. The user needs to initialize the system and set the QoS requirement and service path variable. The service composition process is divided into two sub-processes. Search for the service paths that satisfy the QoS requirement, then compare the composition cost function value of each path with formula (5). If we can find the path with minimal value, the process will succeed. The algorithm of this process can be expressed in Fig. 4.

5.2. Service recovery strategy

After the success of service composition, the receiver will receive a composite service provided by multiple devices. At time t_i, one or more devices maybe fail or move, which causes the composite service interruption. The service recovery process is triggered at this time. In the survival time $[t_i, t_{i+1})$, there are several feasible recovery paths. Our purpose is to find the best recovery path that satisfies the QoS requirement, which reduces the cost and the time of repairing the path as much as possible. In other words, we aim to find the service path with minimal recovery cost function value and let the user's interrupt perception reduce to a minimum.

In the service recovery process, we may find multiple devices or paths satisfying the substitutive requirement. The service composition and recovery processes are facing a common problem: how to choose the best devices to constitute a service path that provides the user with high-quality service. But there is something different between the service recovery and composition. We need to consider not only the quality of the new service path after repairing but also the failing service path used before. That is to say, we should reduce the consumption and recovery time of repairing the failing path as much as possible and prolong the life time of the new path. Therefore, when we are repairing a service path, we ought to reuse the possible working devices and change minimal devices of the failing path. In addition, we can use the devices nearest to the failing devices, which reduces the cost of recovery. This problem involves service local recovery and service global recovery.

When a composite service interruption is caused by the failure of devices or links, the service local recovery process is triggered. It will find substitutive devices nearby and the corresponding links to replace the failing devices and links, so service local recovery involves less substitutes and recovery time. The service global recovery process is to find a new service path for the user when the composite service interrupts. In contrast to the local recovery process, the global recovery process usually consumes more time in repairing path. In this paper, when a composite service interrupts, we should first consider the service local recovery. If it fails, the global recovery should be considered next.

The service recovery process is divided into several sub-processes. When a composite service interrupts, the service local recovery process is triggered. First, it sets the QoS requirement and service recovery path variable. Second, it replaces the failing devices locally to constitute a recovery path and see whether the path satisfies the QoS requirement. If satisfies, it will calculate the recovery cost function value of the new path using formula (4). There are several such paths in general. The process will find the path with minimal recovery cost function value. If such a path exists, the process succeeds. The algorithm flow of the service local recovery process is shown in Fig. 5. If such a path does not exist, the service global recovery process is triggered, which is similar with Section 5.1.

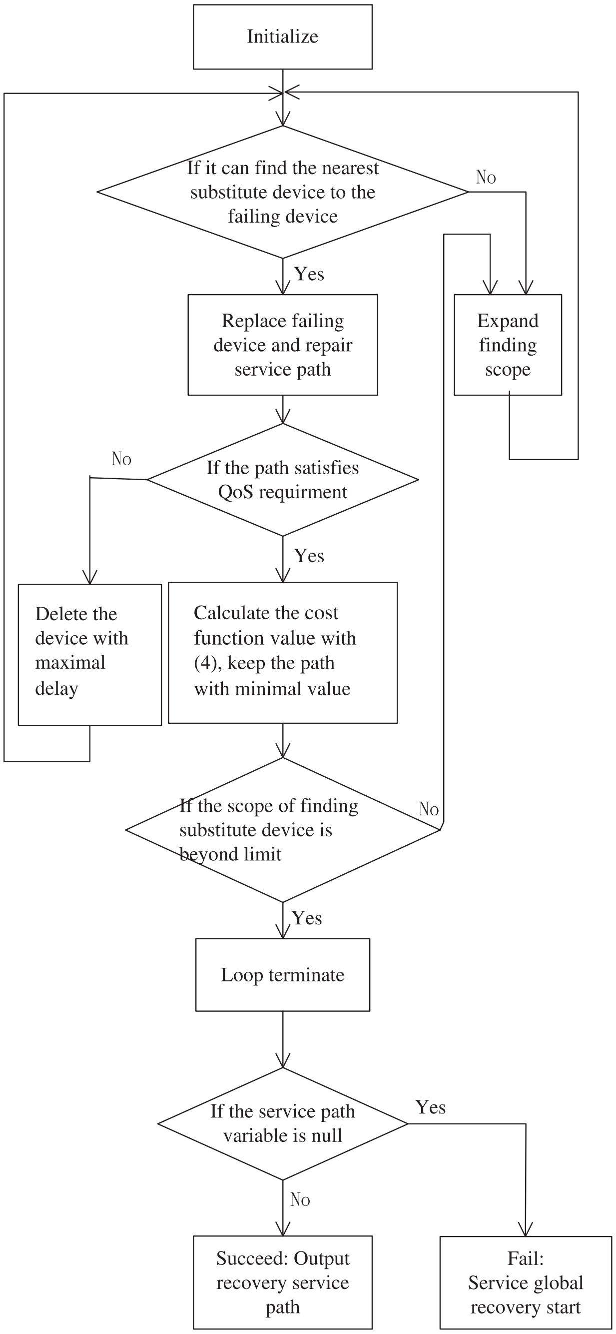

FIGURE 5. Service local recovery algorithm flow.

6. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

6.1. Simulation environment

In the simulation experiment, we use C++ language to program in VC++6.0 environment. We set up 60 nodes distributed in the range of $120 \times 120(\mathrm{m}^2)$ randomly, which follow the random way mobility model (RWP) [19]. The speed range of these nodes is from 1 to 20 m/s. After a node moves for 2 s, it will pause for 2 s. All the nodes have a transmission radius of 30 m. There are 10 kinds of atomic services distributed in the nodes, and the service number is from 0 to 9. For each type of atomic service, there are six providers. Suppose that the time of executing an atomic service is 2 s, the number of the concurrent requests to a node is 4. Ad hoc on-demand distance vector routing (AODV) protocol is used in the experiments, and the simulation time is set to 600 s. We select one node to send a service request in all 60 nodes constantly and randomly in each experiment. The process of selecting atomic services to constitute a service path is also random.

6.2. Simulation results analysis

In the simulation experiments, we compare our SCRS with resending request strategy (RRS) [8] and CBSRS [10]. We compare the strategies from several aspects, such as interrupt number, service recovery time and request success rate. Suppose that CSN means composite service number, we set up CSN × 2 s as the service response time threshold. The start time is a node sending a service request. If there is no response within the response time threshold, the service request will fail. We set up CSN × 7s as the service execution time threshold. If there is a response within the response time threshold, and some atomic services do not accomplish within the service execution time threshold, the service request will also fail.

1. Interrupt number

In this paper, interrupt number is the number of service interruptions for a service request. There are many service requests in a simulation experiment. We need to record the number of all the interruptions for every service request. Then, we should average the data. If the result is smaller, the service path is more stable, and the successful chance of executing a composite service is greater. Suppose that the composite service number is 4, Fig. 6 is the contrast figure of interrupt number changing with the increase of node max speed. Suppose that node max speed is 10 m/s, Fig. 7 is the contrast figure

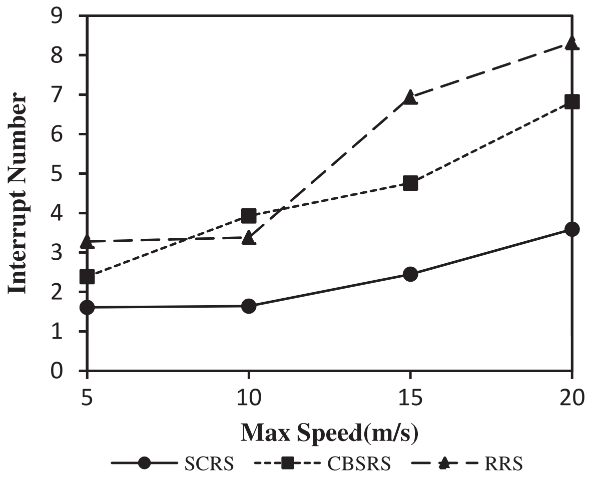

FIGURE 6. Impact of node max speed on interrupt number.

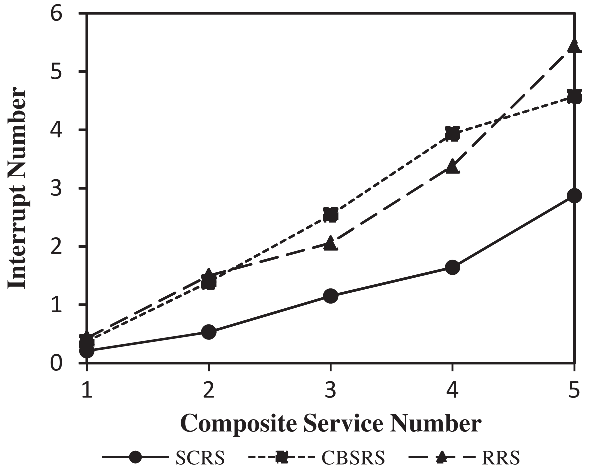

FIGURE 7. Impact of composite service number on interrupt number.

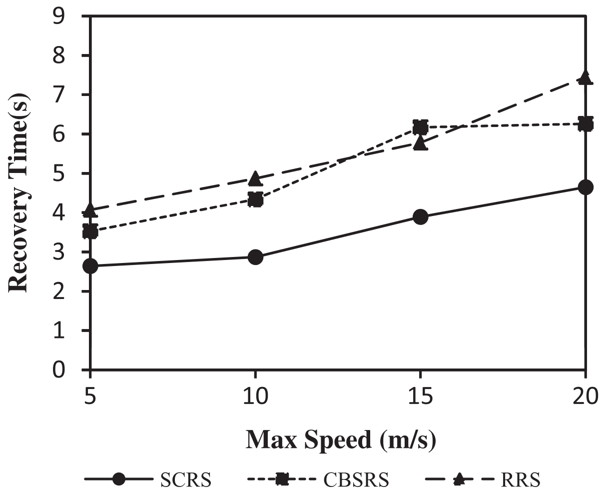

FIGURE 8. Impact of node max speed on recovery time.

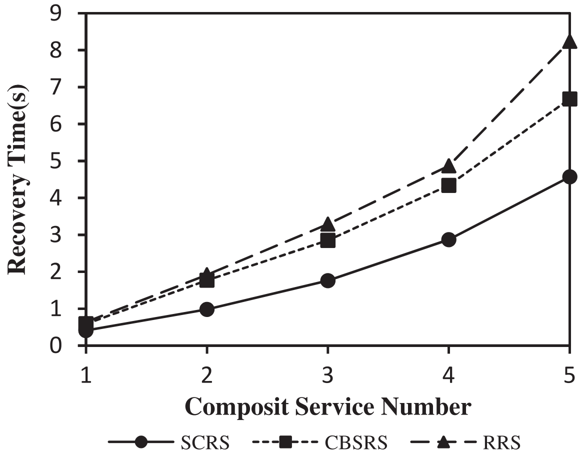

FIGURE 9. Impact of composite service number on recovery time.

of interrupt number changing with the increase of composite service number.

In general, the interrupt numbers of three strategies increase with the increase of node max speed and composite service number. If the node max speed is bigger, the service provided by the node is more unstable. If the composite service number is greater, more nodes will probably fail. In Fig. 6, the interrupt number of SCRS is less than CBSRS at ∼32–58% and lower than RRS at ∼50–65%. In Fig. 7, the interrupt number of SCRS is less than CBSRS at ∼43–62% and lower than RRS at ∼44–65%. So the interrupt number of SCRS is lowest.

2. Recovery time

Recovery time is the time needed to repair all the interruptions for a service request. Suppose that the composite service number is 4, Fig. 8 shows the contrast figure of recovery time changing with the increase of node max speed. Suppose that node max speed is 10 m/s, Fig. 9 shows the contrast figure of recovery time changing with the increase of composite service number. Here, the recovery time is an average value.

The recovery time of three strategies increase with the increase of node max speed and composite service number. In Fig. 8, the recovery time of SCRS is less than CBSRS at ∼25–37% and lower than RRS at ∼32–41%. In Fig. 9, the recovery time of SCRS is less than CBSRS at ∼30–45% and lower than RRS at ∼36–49%. So the recovery time of SCRS is lowest.

3. Success rate

Success rate is the ratio of successful requests number to all of the requests number in an experiment. If the success rate is higher, the performance of the strategy is better. Suppose that the composite service number is 4, Fig. 10 is the contrast figure of success rate changing with the increase of node max speed. Suppose that node max speed is 10 m/s, Fig. 11 is the contrast figure of success rate changing with the increase of composite service number.

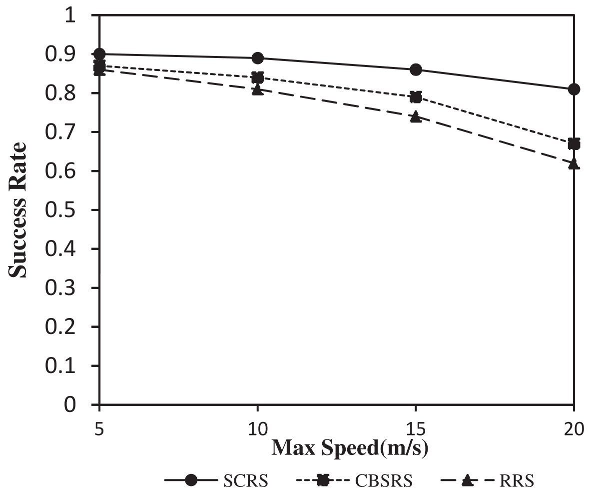

FIGURE 10. Impact of node max speed on success rate.

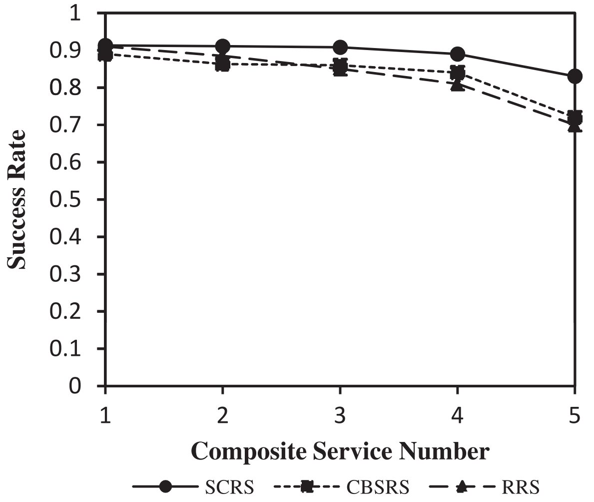

FIGURE 11. Impact of composite service number on success rate.

The success rate of three strategies decreases with the increase of node max speed and composite service number. In Fig. 10, the success rate of SCRS is higher than CBSRS at ∼3–20% and higher than RRS at ∼5–30%. In Fig. 11, the success rate of SCRS is higher than CBSRS at ∼3–15% and higher than RRS at ∼0–18%. So the success rate of SCRS is highest.

When looking for a recovery service path, SCRS considers various factors comprehensively, such as the substitutive device number, the service path delay and the device failure rate. The goal of SCRS is to choose the device nodes with higher reliability and stronger stability. But CBSRS does not consider factors that affect the whole recovery service path, and RRS just selects the device nods nearest to the request node after resending the request. Through simulation experiments, we can see that SCRS reduces interrupt number and recovery time and therefore enhances the success rate. SCRS attains outstanding performance and provides the user with better service experience.

7. CONCLUSIONS

This paper puts forward an improving strategy applied to MSNS in a disaster scenario. When communication facilities are limited, several devices can work cooperatively relying on MSNS. When some of the devices fail, this strategy conducts local recovery first. If the local recovery fails, the global recovery will be carried out. Compared with other strategies, this improving strategy is more efficient and has a stronger ability to adapt to various environments. The future research work will focus on how to calculate the failure rate of device scientifically and increase the efficiency of choosing the service path.

ACKNOWLEDGEMENTS

The work presented in this paper was partially supported by the Funds for National Natural Science Foundation of China (61302078, 61372108, 61370220), Creative Research Groups of China (61121061), Ph.D. Programs Foundation of Ministry of Education of China (20110005110011), Beijing Higher Education Young Elite Teacher Project (YETP0476), Plan for Scientific Innovation Talent of Henan Province Grant (134100510006).

REFERENCES

[1] Li, S., Wang, W.D., Cui, Y.D., Yu, K. and Wang, H. (2010) A Novel Prefetching Method for Scene-Based Mobile Social Network Service. *Proc. IC-BNMT2010*, Beijing, October 26–28, pp. 144–148. IEEE Computer Society, Piscataway, New Jersey.

[2] Bellavista, P., Montanari, R. and Das, S.K. (2013) Mobile social networking middleware: a survey. *Pervasive Mob. Comput.*, **9**, 437–453.

[3] Sanchez, F., Barrilero, M., Uribe, S., Alvarez, F., Tena, A. and Menendez, J.M. (2012) Social and content hybrid image recommender system for mobile social networks. *Mobile Netw. Appl.*, **17**, 782–795.

[4] Zhang, Z.Y. and Wang, K.L. (2013) A trust model for multimedia social networks. *Soc. Netw. Anal. Min.*, **3**, 969–979.

[5] Cui, Y.Q. and Honkala, M. (2013) A novel mobile device user interface with integrated social networking services. *Int. J. Hum.-Comput. Stud.*, **71**, 919–932.

[6] Yang, K., Cheng, X. and Pau, G. (2012) Special issue on: interdisciplinary and cross-layer design of mobile social networks and wireless networks. *Int. J. Commun. Syst.*, **25**, 1243–1244.

[7] Wan, H.Y., Lin, Y.F., Wu, Z.H. and Huang, H.K. (2012) Discovering typed communities in mobile social networks. *J. Comput. Sci. Technol.*, **27**, 480–491.

[8] Chen, W.Y., He, Z.Y., Ren, G. and Sun, W.W. (2008) Service Recovery for Composite Service in MANETs. *Proc. IEEE Int. Conf. on Wireless Communication, Networking and Mobile Computing*, Dalian, October 12–14, pp. 1–4. Inst. of Elec. and Elec. Eng. Computer Society, Piscataway, New Jersey.

[9] Chen, W.Y., Xu, Y.X., Peng, B. and Sun, W.W. (2008) Dynamic Monitor Based Service Recovery for Composite Service in MSNETs. *Proc. 11th IEEE Int. Conf. on Communication Technology, Hangzhou*, November 10–12, pp. 557–560. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey.

[10] Sun, L.L., An, J.W. and Yang, Y. (2011) Recovery Strategies for Service Composition in Dynamic Network. *2011 Int. Conf. on Cloud and Service Computing*, Hong Kong, December 12–14, pp. 60–64. IEEE Computer Society, Piscataway, New Jersey.

[11] Jiang, S.S., Xue, Y.and Schmidt, D.C. (2007) Disruption-Aware Service Composition and Recovery in Dynamic Networking Environments. *Proc. 2007 Int. Conf. on Automated Software Engineering*, Atlanta, November 5–9, pp. 28–33. Association for Computing Machinery, New York.

[12] Jiang, S.S., Xue, Y. and Schmidt, D.C. (2009) Minimum disruption service composition and recovery in mobile ad hoc networks. *Comput. Netw.*, **53**, 1649–1665.

[13] Lin, K.J., Zhang, J. and Zhai, Y.L. (2009) An Efficient Approach for Service Process Reconfiguration in SOA with End-to-End QoS Constraints. *IEEE Conf. Commerce and Enterprise Computing*, Vienna, July 20–23, pp. 146–153. IEEE Computer Society, Piscataway, New Jersey.

[14] Zhang, Z.Y. (2010) Security, Trust and Risk in Digital Rights Management Ecosystem. *2010 Int. Conf. on High Performance Computing and Simulation*, Caen, France, June 28–July 2, pp. 498–503. IEEE Computer Society, Piscataway, New Jersey.

[15] Zhang, Z.Y. (2012) Frontier and methodologies on digital rights management for multimedia social networks. *Int. J. Digit. Content Technol. Appl.*, **6**, 245–249.

[16] Boyd, D.M. and Ellison, N.B. (2007) Social network sites: definition, history, and scholarship. *J. Comput.-Mediat. Commun.*, **13**, 210–230.

[17] Zhang, Z.Y. (2012) *Security, Trust and Risk in Digital Rights Management Ecosystem*. Science Press, Beijing.

[18] Hu, X.P., Leung, V.C.M., Du, W.C., Seet, B.C. and Nasiopoulos, P. (2013) A Service-Oriented Mobile Social Networking Platform for Disaster Situations. *46th Hawaii Int. Conf. on System Sciences*, Wailea, Maui, HI, January 7–10, pp. 136–145. IEEE Computer Society, Washington D.C.

[19] Johnson, D.B. and Maltz, D.A. (1996) Dynamic source routing in ad hoc wireless networks. *Mob. Comput.*, **353**, 153–181.