International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 4, August 2019, pp. 2523~2533
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i4.pp2523-2533

Reliable fault tolerance system for service composition in mobile Ad Hoc network

P. Veeresh[1], R. Praveen Sam[2], C. Shoba Bindu[3]
[1,3]Department of CSE, Jawaharlal Nehru Technological University Anantapuramu, India
[2]Department of CSE, G. Pulla Reddy Engineering College, India

Article Info

Article history:

Received Nov 16, 2017
Revised Feb 18, 2019
Accepted Mar 4, 2019

Keywords:

Service composition
Fault tolerance
Mobile Ad Hoc network
Multi-service tree

ABSTRACT

A Due to the rapid development of smart processing mobile devices, Mobile applications are exploring the use of web services in MANETs to satisfy the user needs. Complex user needs are satisfied by the service composition where a complex service is created by combining one or more atomic services. Service composition has a significant challenge in MANETs due to its limited bandwidth, constrained energy sources, dynamic node movement and often suffers from node failures. These constraints increase the failure rate of service composition. To overcome these, we propose Reliable Fault Tolerant System for Service Composition in MANETs (RFTSC) which makes use of the checkpointing technique for service composition in MANETs. We propose fault policies for each fault in service composition when the faults occur. Failure of services in the service composition process is recovered locally by making use of Checkpointing system and by using discovered services which satisfies the QoS constraints. A Multi-Service Tree (MST) is proposed to recover failed services with O(1) time complexity. Simulation result shows that the proposed approach is efficient when compared to existing approaches.

Corresponding Author:

P. Veeresh
Jawaharlal Nehru Technological University Anantapuramu,
Anantapuramu, Andhra Pradesh, India.
+91-9440772419
Email: veeresh_kaly@yahoo.co.in

1. INTRODUCTION

Service composition is an emerging trend to create new services to solve complex needs of the users by aggregating several existing atomic services. Service composition creates efficient and cost-effective services [1]. Service composition approaches are categorized in two methods static and dynamic service composition. In static service composition, service discovery, interface matching, semantic and conditional matching is defined well before the composition. In this approach, more manual intervention is needed and it cannot be applied in MANETS due to its dynamic behavior. Dynamic service composition will do everything automatically at runtime without or little manual intervention [2] and can be easily adapted to MANETS.

Deploying service composition in an unreliable network like MANETs is a challenging task [3]. Fault tolerance is essential for service composition in MANETs for successful completion of the composition process. Due to dynamic nature of MANETs service composition may fail for the following reasons

1. Node mobility
2. Drain of energy source
3. Link Failure
4. Dynamic topology
5. Low communication bandwidth

It is essential to maintain fault tolerant system in service composition due to error-prone nature of MANETs to reduce the failure rate of the composition process. But wireless mesh networks are strong enough to withstand for faults and fault tolerance [4]. Checkpointing is a novel technique for fault tolerance in service composition in MANETs.

The fault is a defect or abnormal condition which may cause failure in service composition. Existing fault recovery mechanisms of services are categorized in two ways. They are forward recovery [5] (Centralized approach) and backward recovery approach [6]. In Forward recovery approach, a centralized coordinator monitors the service composition process. Centralized coordinator invokes the consistent services for the process of composition. In case of service failure, composition process continues with the equivalent service. If the coordinator itself fails, composition process is re-executed from the beginning. The centralized coordinator is not possible for MANETs due to dynamic nature. Hence forward recovery approach will not be suitable for MANETs. In backward recovery approach, service composition is restored with the previously saved checkpoint when a failure occurs at runtime. Composition execution process is continued from the saved checkpoint with the equivalent service.

The checkpoint is a saved program status and can be used when the regular process is interrupted. Checkpointing is efficient in case of transient faults which may happen frequently in MANETs due to link failures. By the checkpoints, service composition execution process will resume from the last consistent saved state thus reducing the cost of re-execution of all services from the beginning [7].

Service composition is a process of combining atomic services to create a solution for the complex task. In [8] a distributed approach is proposed to find services composition path by using path filtering and path combination. These two methods control forwarding messages and reduce the searching efficiency. In [9] distributed architecture is proposed for peer-to-peer MANETs for service composition. Table update messages are used for service discovery. The composition process load is distributed among the network for balancing. The problem with this approach is, a new architecture is implemented at every node which leads to great changes in MANET.

In [10] Distributed broker mechanism protocol and distributed service discovery is proposed for dynamic service composition. A distributed composition manager (CM) is elected based on device-specific potential values and QoS Parameters. Each composition request is independently assigned to CM. It works on the principle of single composition manager for single composition request. In [11] hierarchical graph-based service composition architecture is proposed, where a node represents logical service and edge represent data flow between corresponding nodes. In paper [12] based on service description, dynamic service composition is proposed. The probability of successful combination can be improved by considering the semantic relationship between services.

Checkpointing techniques are categorized three types: coordinated, uncoordinated, and communicationinduced checkpointing techniques. Coordinated checkpointing will maintain a consistent global state. Each node is maintained only one consistent checkpoint. The coordinator will issue control message to nodes to take checkpoint [13]. In [14] a non-intrusive coordinated Checkpointing algorithm is proposed for distributed computing system where the number of messages for piggybacking is minimized. Only one bit overhead is employed for odd and even checkpoint intervals. Uncoordinated checkpointing [15-16] allows each node to maintain multiple checkpoints and creates local states independently. No additional control messages are generated for maintaining of checkpoints. The objective of this approach is to minimize the number of messages for checkpointing and maximize the performance of the normal operation. A rollback dependency graph is used to restore from failure. This approach suffers from domino effect [17]. Communication-induced checkpointing [18] are designed to overcome the limitation of domino-effect. Each message is transmitted with a sequence number and uses piggybacked with timestamp [19] for unique identification of messages. A checkpoint is created only after the reception of control message. Each checkpoint is identified uniquely by the sequence number for avoiding a domino effect. Roll back process starts from the recent checkpoint to restore the application to the consistent state. Communication induced checkpointing are classified into two categories: a model based and induced based checkpointing. Major challenges of Checkpointing and fault recovery approaches are described in the paper [20].

In [21] a cluster based distributed QoS routing algorithm proposed for MANETs with aim of achieving fault tolerance is described. This mechanism reduces disruption time under network failures and avoids rerouting of QoS traffic. Various routing protocols performance comparison is discussed in [22]. In [23] Fault Tolerance QoS Routing Protocol (FTQRP) is proposed for fault tolerance by maintaining backup routes using multipath routing algorithm. Each backup route is maintained up to date. When a failure occurs, the data is to be forwarded by using backup routes and uses H-ARQ (Hierarchical ARQ scheme") for reducing the feedback implosion. The limitation with this approach is if any route is failed a new route is discovered by discovering and selecting of remaining subsequent nodes.Energy-Aware Smart Protocol for

Routing (EASR) is presented and uses the geographical location of nodes and graph theory to route packets in network. Fixed anchor nodes are used to route packets in congested and heavy traffic. Path Redundancy Metric is used for multihop routing for large traffic scenario for fault tolerance [24].

In [25] graph based fault tolerance system is proposed by adding the optimal relay nodes. For optimization of relay nodes, residual energy and density of the network is considered. The nodes in routing path are selected based on residual energy along the path of the intermediate nodes. The number of relay nodes is calculated based on distance and residual energy required between a pair of nodes. In [26] proposed a transport layer solution for fault tolerance routing in MANET, hybrid nodes are introduced in the network which uses traffic splitting mechanism for sending and receiving of data.

In [27] fault-tolerant routing algorithm is proposed based on ant colony and further optimizes with the lookahead property. The forward agents FANT and backward agents BANTS are used to calculate various QoS parameters. Path preference probability is calculated by using these parameters. From all available paths, higher preference probability path is selected between source and destination for transmitting of data.

In [10] "Distributed Fault-Tolerant Service Composition Protocol" (FTSCP) for MANETs is discussed. FTSCP is a distributed framework, it has two main features. First, all atomic services are discovered in one service discovery session and stored in runtime database (DB). Second, a centralized Execution Coordinator (EC) monitors entire service composition process and faulty services are rediscovered transparently. FTSCP doesnt differentiate the failure of nodes and services separately. QOS constraints are not considered at both service/node levels. A mechanism to store discovered services in Runtime DB is not discussed in detail.

A significant amount of existing research has done in the area of fault-tolerant mechanisms for various routing protocols in MANETs. Few researchers have explored applying fault tolerant mechanisms in service composition in MANETs. The problem with the existed approaches is if any one of the services is failed in service composition plan the service composition plan is reconstructed from the failed service instead of finding an alternate single service. The existed approaches consider only node failures but not consider service failure. The fault recovery mechanism is limited to node level QoS constraints only which is a bottleneck. These limitations motivate our work to enhance and combine fault tolerance mechanism at both service and node level QoS constraints in service composition of MANETs.

To counter the problems of existing approaches, in this paper we propose the following contributions

1. Fault policies are defined at both service and node layers by considering various faults.
2. A low overhead Checkpointing technique is proposed to resume the service composition process from thefailed service.
3. An MST is proposed to recover the failed services with O(1) time complexity.

The rest of the paper is organized as follows. In Section 2 describes the proposed approach Fault ToleranceSystem for Service Composition in Mobile Ad Hoc Network. The section 3 presents simulation results and section 4 describes conclusion and future work.

2. FAULT TOLERANCE SYSTEM FOR SERVICE COMPOSITION IN MOBILE AD HOC NETWORK

Our proposed approach will improve the performance of FTSCP and FTQRP by considering QoS parameters of nodes and services separately, identified various faults and associated fault policies in case of failure at node/ service layers and a novel recovery mechanism to select alternative services using MST.

Our proposed Reliable fault tolerant system for service composition in MANETs (RFTSC) edifice consists of 6 components as shown in Figure 1. Each component is described as follows.

a. Service Discovery - It discovers the related services required for service composition.
b. QoS Context - QoS context takes discovered services as input and pre-processes the services by applying Fuzzy Rating System as described in [28] and generates a set of qualified services as output.
c. Services Composer - It creates composition plan and builds Multi Service Tree (MST) for fault tolerance. Services Composer also updates MST to the Recovery Manager.
d. Checkpointing - Checkpointing is a reliable database which stores the updated status of services execution as per composition plan.
e. Execution Monitor - Execution monitor issues service execution request to the service providers. It monitors the execution of services. On successful execution of service, it updates checkpoint, in case failure event is occurred, it communicates with the Recovery Manager.

f. Recovery Manager - Recovery Manager identifies the alternate services by querying the MST and communicates back to the execution monitor with the alternate service as per composition plan. The main responsibility of the recovery manager is to identify alternate service if there is any failure occurred in the composition process.

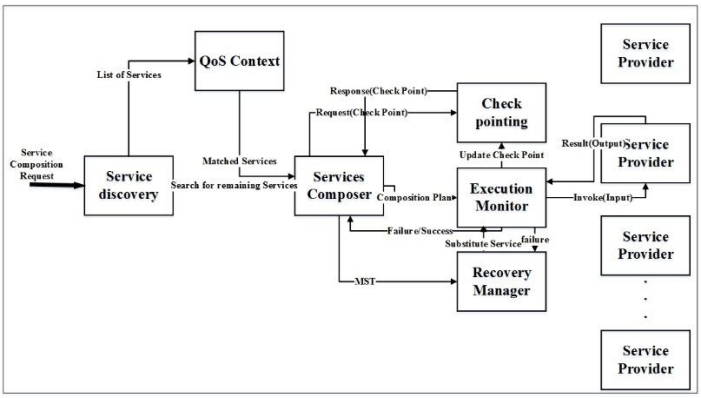

Figure 1. Fault tolerant service composition architecture

2.1. Service composition process

Service discovery manager discovers the list of related services as per the service composition request. The list of services is transferred to QoS context. QoS context process the services and generates a rating for each service using Fuzzy Logic System [28]. QoS context will update the optimal related services to the Services Composer. Service Composer creates a Multi-Service Tree (MST) structure (for more Details section 3.5) with the optimal services. Each tree contains a list of similar services and ordered in the max-heap property. In each service tree, root node contains a max rated service. With all the root nodes, the services composer generates composition plan and transferred to execution monitor. Services composer transfers MST to the recovery manager to recover services in case of failure occurred.

Execution monitor invokes each service with required inputs as per the composition plan. After completion of each service execution, execution monitor will collect the output results from the service provider and same updated to checkpointing database. Checkpointing is a consistent database which maintains updated the status of the composition process. If any failure occurs, the execution monitor issues failure request to the recovery manager. The recovery manager selects maximum rated service as an alternate in place of failed service with O(1) time complexity and transferred to execution monitor to invoke the new alternate service. Recovery Manager returns a NULL response to the Execution Monitor if the related service is not available in MST. In this case, Execution Monitor communicates with failure/ success status to the Services Composer. Service Composer will query the Checkpointing database for the latest checkpoint. The Checkpointing database responds with the latest checkpoint to the Services Composer. Service composer requests the Service Discovery to search for the remaining services in composition plan. All these steps are iterated until the service composition completes.

2.2. Node and a Service definition in MANET

In this section, we define a Node and Service in MANET.

2.2.1. Service in MANET

Service can be defined as

$$S_j = \left\{ S_{id}, S_{ip}, S_{op}, E_{Ser}, R_{Ser}, F, C \right\} \quad (1)$$

where,

- S_{id} - Unique service identifier
- Sip - Service Input
- Sop - Service Output
- ESer - Energy Index Value of a Service, amount of energy needed to execute a service.
- RSer - Service Response time
- F- Fault response. Response type as described in Table 1.
- C- Other Constraints

A service can be uniquely identified by $N_i.S_j$, N_i represents i^{th} node and S_j represents J^{th} service.

Table 1. Fault response codes

S.No	Response	Meaning
1	1	0 Retry
2	0	Dont Retry
3	>1	>1 Retry with n attempts

2.2.2. Node in MANET

A node in MANET can be defined as a triplet:

$$N = \left(N_{id}, \{S_1, S_2, \ldots, S_n\}, E_{Node}\right) \quad (2)$$

where,

- Nid- Unique identifier of a Node.
- S1, S2……., Sn- Services present in that node
- ENode - The amount of energy needed to maintain its properties like bandwidth, mobility, and reliability.

2.3. Fault policies for service composition.

A *fault* is defined as an incorrect process, step, or data definition in a computer system which influences the system to perform in an unexpected behavior. An interruption in normal composition process is termed as composition failure. Service composition process can be failed in MANETs due to either node failure or service failure. A *fault* policy is defined as the action to be taken when a fault occurs. In this section, we proposed various faults and associated fault policies in case of failure at node/service layers.

2.3.1. Fault policies for node failure

In MANETs, nodes can join and leave the network on the fly due to its mobility. Node faults will occur frequently in MANETs due to its dynamic movement of nodes. Node fault refers to the situation where a node is unavailable due to movement from its vicinity or link Failure or drain of battery. When a node fault occurs, the service present in that node is also not available. The recovery action is to substitute with another node to continue composition process.

2.3.2. Fault policies for Service failure

Service fault means unavailability of service. In this section, we discussed causes for service faults and fault policies to recover from failure.

a) Service unavailable - Service is unavailable due to following reasons

Service overloaded when the number of requests increases for the same service, the service gets overloaded. The recovery action is retrying the same service with n attempts. If the same service is available with n attempts then the service composition process continues with the same service else the service is substituted with another alternate optimal service to continue execution process.

Rejection of execution Service rejects execution request. The recovery action is substituted with another service.

Service crashed Service is crashed when an internal problem occurs in the service. In this case, the node is available and service is unavailable. The recovery policy doesn't retry and try for alternate service.

b) Service QoS faults - Service QoS faults will occur when the service violates predefined QoS requirements. In this case, recovery action is retrying with n attempts to restore the predefined QoS requirements.

c) Functional and behavioral faults Functional and behavioral faults refer to the situation where a service delivers incorrect results due to computational errors or cannot complete task execution. Additionally, behavioral faults can occur due to the improper order of service invocation. The recovery action is not to retry and substitute it with the alternate service.

d) Service operational faults Service operational faults refer to the faults happening due to communication congestion and Security Breaches. It is a transient failure and the recovery action is Retry with n attempts.

Table 2 summarizes various faults and fault handling strategies specified by the service provider and recovery manager.

Table 2. Faults and fault policies

S.No	Fault	Cause	Action Specified by Service Provider	Action Initiated by Recovery manager
1	Node unavailable	Node moved from vicinity	Don't Retry	Substitute
		Link Failure	Don't Retry	Substitute
		Drain off Battery	Don't Retry	Substitute
		Overloaded	Retry(n)	Retry(n)
2	Service unavailable	Rejection of execution	Don't Retry	Substitute
		Service Unavailable, Service crashed	Don't Retry	Substitute
3	Service QoS faults	service violates the predefined QoS	Retry(n)	Retry(n)
4	Functional and behavioral faults	Output not correct, Improper invocation order of service operations	Don't Retry	Substitute
5	Service Operational faults	Network congestion, Security breaches	Retry(n)	Retry(n)

2.4. Service response packet

After receiving a request from the service composition initiator, if the node contains related services, it initiates a si ngle response packet for all services. By reducing the number of packets required for service discovery will limits packet overhead and network congestion. Figure 2 shows response packet format.

Response Packet format is described as follows:

1. Service Composition Initiator Address represents Source node IP address.
2. Service Provider Address represent target node IP address.
3. Energy Index Value for a Node represents the amount of energy needed for mobility constraints like bandwidth, node mobility, and service discovery mechanisms.
4. Hop Count represents a number of hops between sources to a destination.
5. Timeout represents the packet alive time.
6. Sequence Number counters stale use of old packets.
7. Service ID Represents Unique identification of service
8. Energy Index Value for Service represents amount of energy needed to process the Service
9. Service response time is the time difference between initiation of a request to first response from the service.
10. Service throughput is the number of requests processed per unit of time by the service provider.
11. Fault action is considered while failure occurred. Fault actions are represented in Table 1.

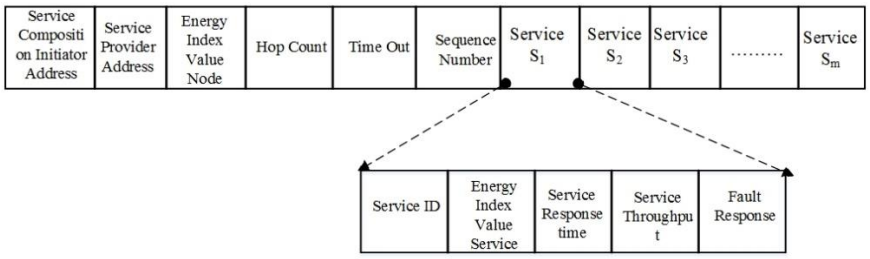

Figure 2. Service response packet

2.5. Multi-service trees (MST)

Here we propose a Multi-Service Tree (MST) to store similar services information into a single tree. Service composer builds MST with max heap property. For example, all S_1 services are grouped and constructed a S_1 service. Each service is rated with Fuzzy Logic System. An MST is constructed based on the rating of a service. MST is a complete binary tree in which root node value is greater than both left and right child nodes as shown in Figure 3. Max rated service is available at the root node of the MST. The service composition plan is created with all the optimal services available at root nodes of all the service trees. After creation of composition plan, all the root nodes are deleted and the max heap root node values are adjusted with the next max rated optimal service which is available at a left child or right child of the root node. If any failure occurred in service composition, alternate service is selected from the root of the service tree which is optimal rated among the available services. Failure service is recovered with O(1) time complexity.

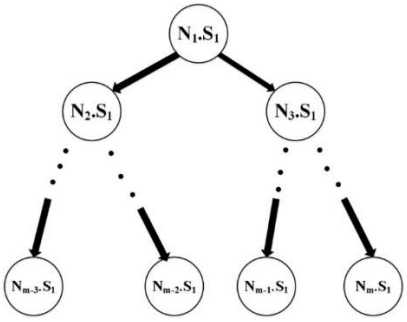

Figure 3. MultiService tree for service S_1 tree

Figure 4 shows example multiple services in a node and each service is rated with Fuzzy logic. The composition path is $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4$ and the order of optimal nodes selected for composition is $N_6 \rightarrow N_5 \rightarrow N_1 \rightarrow N_2$. Figure 5(a) shows an example service tree for service S_1. Max rated service (0.96) is available at the root node N_6. After the failure of service S_1 in Node N_6, alternate optimal max rated Service S_1 in node N_3 is selected for continuing service composition as shown in Figure 5(b).

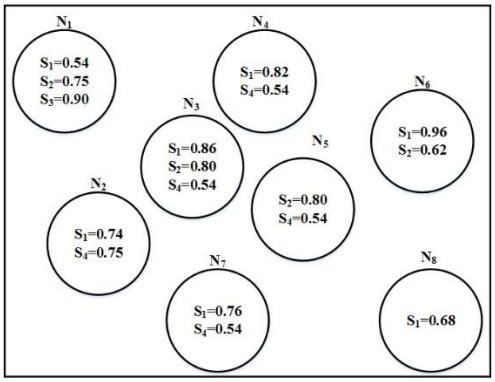

Figure 4. Nodes with multiple services with fuzzy rating

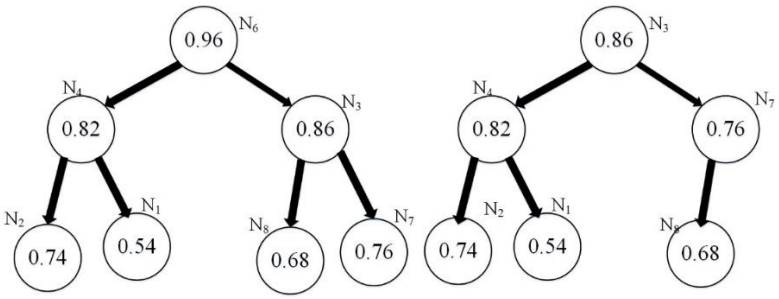

Figure 5. Example service tree (a) Before service failure S_1 (b) After service S_1 recovery

3. RESULTS AND DISCUSSION

In this section, we present results and discussion of our proposed methodology Reliable Fault Tolerance System for Service Composition in MANETs (RFTSC). Simulation of services is carried out using the tool [29], which is the extension framework of Network Simulator NS-3. Firstly, we organized an infrastructure less MANET with 80 mobile devices with wireless capabilities and each device can communicate within their proximity with other devices. Even if there is no direct link between different devices, the underlying routing protocol provides an indirect link enables to communicate. Unpredictable

node mobility causes path breaks occurs as nodes can move out of range at any time. We integrated NS-3 Mobility model to capture the effects of mobility realistically. Simulation is conducted for 160 seconds. For each service composition, we considered 5 abstract services i.e., the minimum 5 services are involved in service composition.

- Network *Model* In given area, an ad hoc network is generated. It consists of 80 nodes and that are placed in a random manner in a given area. Each node shared its limited bandwidth with their neighbors.
- *Channel Model* It expects the error-free channel and free space propagation model. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is used to access the channel to reduce subsequent packet drops and collisions.
- *Mobility Model* - Random way-point (RWP) mobility model adopted in the simulation.
- *Traffic Model* - We employed Constant Bit Rate (CBR) model that transmits a certain number of fixed size packets in a flow.

The major research outcome of our proposed method is node failure and service failure which can be recovered within O(1) time complexity. If any service failed in service composition plan, only alternate failed service/node is recovered and resumes the service composition process as per the existing composition plan. Also, the performance of the RFTSC is compared by using fault-tolerant routing protocol in MANET's FTQRP and FTSCP. To achieve this we have injected faults explicitly both at the network level and service level and simulation is carried out. Simulation results are compared with each other and our proposed methodology performs better than the FTQRP and FTSCP. Here we are comparing Number of services participated in composition(Abstract Services) Verses Service Composition Efficiency, Service Composition failure rate and Service Failure recovery Time. Table 3 overviews the simulation setup.

Table 3. Simulation setup

Parameter	Value
Number of nodes	80
Simulation Time	160 Seconds
Wifi standard	802.11b
Wifi rate	DsssRate1Mbps
Transmission range (R)	45m
Routing protocol	AODV
Number of concrete services	120
Size of composition plan	5 (Abstract Services)
Number of node faults injected	3
Number of service faults injected	3

3.1. Service composition efficiency

Service Composition Efficiency (n) is measured as the ratio of the number of successful compositions c to the number of composition requests r,

$$\eta = \frac{c}{r}$$

As shown in Figure 6, Abstract services in X-axis and service composition efficiency in Y axis is presented in the graph. The RFTSC performs better than the FTQRP and FTSCP approaches in terms of service composition efficiency. In the proposed method, least energy consumed services are selected which improves liveliness of the network and least hop count services are selected which reduces the packet transmitting time. These optimal services in composition increase the number of successful compositions. Hence the Service Composition Efficiency is better than the FTQRP and FTSCP.

3.2. Service composition failure rate

Service composition failure rate is defined as the number of service compositions failed with respect to the number of services involved in composition. Service composition failure rate is increased if a number of nodes/services is involved in composition path. The proposed methodology considers a node can provide more than one service which reduces the number of nodes involved in the composition process. In addition to this, more stable nodes (least energy and hop count) and quick response services (less response time and more throughput) are selected. As shown in Figure 7, the proposed methodology is more stable in failure rates of service composition compared with the FTQRP and FTSCP.

3.3. Service failure recovery time

Service failure recovery time is defined as the amount of time needed to recover failed service. Here we assumed that a service can be recovered in 350 Milli seconds. The proposed methodology recovers only the failed service whereas the FTQRP approach finds a backup path to transmit data and FTSCP approach finds all the remaining subset services from the failed service. Hence the proposed methodology recovers the failed service with constant time O(1) and performs better than the FTQRP and FTSCP as shown in Figure 8.

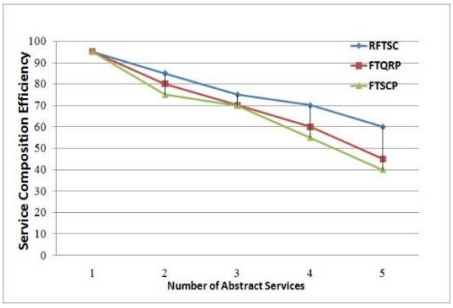

Figure 6. Service composition efficiency

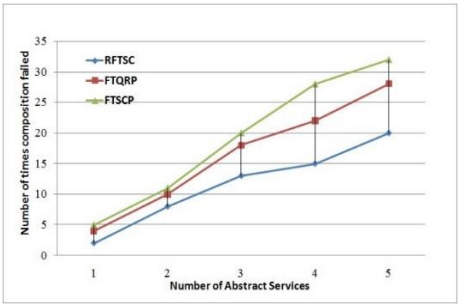

Figure 7. Service composition failure rate

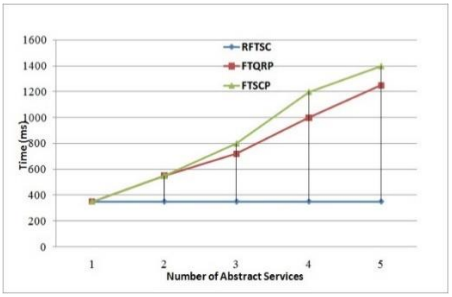

Figure 8. Service failure recovery time

4. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a Reliable fault tolerance system for service composition (RFTSC) to recover failed services in the service composition process for MANETS. We have considered the failures at both node and service layers and identified various faults and associated fault policies to address the failures both at node and service level. The discovered services are selected as per the QoS constraints by applying fuzzy rated system and then the filtered services are stored in MST as per the service composition plan. In case of service failure, we try to recover from the MST for an alternative service. If in case it is not present, then checkpointing mechanism is considered to know the last service that is executed successfully and the left-over services in the composition plan are rediscovered. Simulation results show that proposed methodology is efficient when compared to existing approach and recovers the failed services efficiently in MANETS. As a part of our future work we would like to consider more complex constraints, optimize the composition plan and overcome the single point of failure, and apply to real-world problems.

REFERENCES

[1] M. P. Papazoglou and W. J. Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol/issue: 16(3), pp. 389-415, 2007.
[2] I. Guidara, *et al.*, "Dynamic selection for service composition based on temporal and qos constraints," *Services Computing (SCC), 2016 IEEE International Conference* on, pp. 267-274, 2016.
[3] J. Bronsted, *et al.*, "Service composition issues in pervasive computing," *IEEE Pervasive Computing*, vol/issue: 9(1), 2010.
[4] L. Hui, "A novel qos routing algorithm in wireless mesh networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol/issue: 11(3), pp. 1652-1664, 2013.
[5] Y. Cardinale and M. Rukoz, "Fault tolerant execution of transactional composite web services: An approach," *Proc. Fifth Int, Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 158-164, 2011.

[6] A. Liu, *et al.*, "Fault-tolerant orchestration of transactional web services," *International Conference on Web Information Systems Engineering*, pp. 90-101, 2006.
[7] K. B. Ferreira, *et al.*, "Understanding the effects of communication and coordination on checkpointing at scale," *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pp. 883-894, 2014.
[8] B. Randell, "System structure for software fault tolerance," *IEEE Transactions on Software Engineering*, vol. 2, pp. 220-232, 1975.
[9] B. Bhargava and S. R. Lian, "Independent checkpointing and concurrent rollback for recovery in distributed systems-an optimistic approach," *Reliable Distributed Systems, 1988. Proceedings, Seventh Symposium on*, pp. 3-12, 1988.
[10] G. Cao and M. Singhal, "Mutable checkpoints: a new checkpointing approach for mobile computing systems," *IEEE transactions on parallel and distributed systems*, vol/issue: 12(2), pp. 157-172, 2001.
[11] R. H. Netzer and J. Xu, "Necessary and sufficient conditions for consistent global snapshots," *IEEE Transactions on Parallel and distributed Systems*, vol/issue: 6(2), pp. 165-169, 1995.
[12] W. Yuemin, "Web service composition algorithm based on description logic," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol/issue: 12(1), pp. 852-858, 2014.
[13] G. Janakiraman and Y. Tamir, "Coordinated checkpointing-rollback error recovery for distributed shared memory multicomputers," *Reliable Distributed Systems, Proceedings, 13th Symposium on*, pp. 42-51, 1994.
[14] P. Kumar, *et al.*, "A non-intrusive minimum process synchronous checkpointing protocol for mobile distributed systems," *Personal Wireless Communications, ICPWC 2005. IEEE International Conference on*, pp. 491-495, 2005.
[15] C. Liu, *et al.*, "A low-latency service composition approach in mobile ad hoc networks," *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 509-511, 2014.
[16] U. Aguilera and D. L´opez-de Ipina, "Service composition for mobile ad hoc networks using distributed matching," *Ubiquitous Computing and Ambient Intelligence*, pp. 290-297, 2012.
[17] D. Chakraborty, *et al.*, "Service composition for mobile environments," *Mobile Networks and Applications*, vol/issue: 10(4), pp. 435-451, 2005.
[18] P. Basu, *et al.*, "Scalable service composition in mobile ad hoc networks using hierarchical task graphs," *Proc. 1st Annual Mediterranean Ad Hoc Networking Workshop*, 2002.
[19] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol/issue: 21(7), pp. 558-565, 1978.
[20] L. K. Awasthi, *et al.*, "Minimum mutable checkpoint-based coordinated checkpointing protocol for mobile distributed systems," *International Journal of Communication Networks and Distributed Systems*, vol/issue: 12(4), pp. 356-380, 2014.
[21] L. C. Llewellyn, *et al.*, "Distributed fault-tolerant quality of wireless networks," *IEEE Transactions on Mobile Computing*, vol/issue: 10(2), pp. 175-190, 2011.
[22] R. M. Desai, *et al.*, "Routing protocols for mobile ad hoc network-a survey and analysis," *Indonesian Journal of Electrical Engineering and Computer Science*, vol/issue: 7(3), pp. 795-801, 2017.
[23] M. R. Chandra and P. C. S. Reddy, "Fault tolerance qos routing protocol for manets," *Advanced Computing (IACC), IEEE 6th International Conference on*, pp. 623-628, 2016.
[24] R. Havinal, *et al.*, "Easr: Graph-based framework for energy efficient smart routing in manet using availability zones," *International Journal of Electrical and Computer Engineering*, vol/issue: 5(6), pp. 1381-1395, 2015.
[25] S. Joshi and A. Srinivas, "Power aware fault tolerance in wireless networks with heterogeneous nodes," *Electronic System Design (ISED), Fifth International Symposium on*, pp. 176-181, 2014.
[26] K. Kiran, *et al.*, "Fault tolerant beehive routing in mobile ad-hoc multi-radio network," *Region 10 Symposium, 2014 IEEE*, pp. 116-120, 2014.
[27] S. Surendran and S. Prakash, "An aco look-ahead approach to qos enabled fault-tolerant routing in manets," *China Communications*, vol/issue: 12(8), pp. 93-110, 2015.
[28] V. Poola, *et al.*, "Fuzzy based optimal qos constraint services composition in mobile ad hoc networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol/issue: 9(3), 2017.
[29] P. Novotny and A. L. Wolf, "Simulating services-based systems hosted in networks with dynamic topology," *Tech. rep., Technical Report DTR-2016-2, Department of Computing, Imperial College London*, 2016.

BIOGRAPHIES OF AUTHORS

Veeresh Poola received his Ph.D degree in Department of CSE from Jawaharlal Nehru Technological University Anantapur with Master of Technology from Visvesvaraya Technological University, Belguam, India (2008). He obtained Bachelor Degree in Computer science and Information Technology from Jawaharlal Nehru Technological University, Hyderabad (India) in 2005. His research interests in fields of Mobile computing and web services. He has around 12 years of experience in Teaching and he is life Member in CSI.

R. Praveen Sam received his Ph.D degree in Computer Science and Engineering from JNT University, Anantapur. He is currently working as Professor in Computer Science & Engineering, G. Pulla Reddy Engineering College (Autonomous): Kurnool. His research interests are Mobile and Ad Hoc Networks, Network Security, Computer Organization, Design and Analysis of Algorithms, Big Data and Cloud Computing. He has around 15 years of experience in Teaching and Research. His Research projects are sanctioned by UGC. He has presented papers at National and International Conferences and published articles in National & International Journals. He is life Member in CSI, ISTE, IAENG, and IE.

C. Shoba Bindu, received her Ph.D degree in Computer Science and Engineering from JNT University, Anantapur. She is currently working as Professor in the Department of Computer Science & Engineering, JNTUA College of Engineering, Anantapuramu. She has good contributions in reputed journals. Her research interests are Mobile and adhoc networks, network security, data mining and cloud computing. She has around 17 years of experience in teaching and Research.