Service Composition for Mobile Ad Hoc Networks using Distributed Matching

Unai Aguilera, Diego López-de-Ipiña

Deusto Institute of Technology - DeustoTech
University of Deusto
Avda. Universidades 24, 48007 Bilbao, Spain
{unai.aguilera,dipina}@deusto.es
http://www.morelab.deusto.es

Abstract. Ubiquitous computing envisions the existence of spaces which consist on objects with computational capabilities. These objects can be connected among each other forming wireless networks which could be very dynamic in their nature due to their mobility. Service composition aims to resolve user's needs by creating work-flows which use available functionalities, provided by devices, by assembling them into new services. This work presents an architecture for service composition applied to mobile networks which reacts to those changes which can arise in the network topology. All nodes taking part in the network contain the same architecture meaning that the solution is performed in a peer-to-peer manner distributing the processing load among different nodes of the network. Applications needing to compose services will be constructed on top of the proposed architecture. The architecture has been fully implemented and evaluated using a network simulator.

1 Introduction

In ubiquitous computing environments users should not only access services provided by devices in an single manner but also by means of complex services obtained by connecting other simpler ones. The environment, constituted by available devices, should be capable of helping user to automate composition.

In this work we propose an architecture for service composition which resolves the problem of composition in a mobile ad-hoc network. A mobile ad-hoc network is a type of infrastructure-less network where devices communicate with each other using wireless communication. Nodes in these networks communicate by discovering available routes and performing one-hop transmission of messages. Due to the mobility of the network, new routes can appear or disappear during the network life-time and have an impact over those protocols and applications constructed using them.

The proposed architecture solves the problems of service discovery and composition search while providing mechanisms for message forwarding and a set of functionalities which can be used by developers to construct their final applications. Due the nature of the mobile ad-hoc network, the solution is proposed in a way that all nodes take part in the different composition processes. This avoids the problem of broker nodes selection or the usage of central repositories which could produce a point of failure due to changes in the topology of the mobile ad-hoc network.

The rest of the paper is organized as follows. Section 1.1 introduces some questions about service composition in mobile ad-hoc networks. Section 2 presents the proposed architecture while Section 3 summarizes the evaluation performed to test the proposal. In Section 4 related work in the area is introduced and, finally, Section 5 concludes the paper and presents some future work.

1.1 Service Composition in Mobile Ad-hoc Networks

Mobile ad-hoc networks are useful in those situations where there is a need to rapidly deploy a network of devices and no previous communication infrastructure exists (e.g. emergency networks, vehicular networks or people agglomerations in airports, streets, etc.). Service composition provides a way to access functionality required by users' applications by assembling services which are directly available on the network. For example, let's suppose that an application executed by a user's devices needs to access some service which needs to obtain the current temperature in degree Celsius. In the current situation, the user's device does not have an integrated thermometer nor access to a any temperature service. However, there could be other devices in the network providing that information, for example, as a service offering the temperature in degrees Fahrenheit. These services cannot be directly used by the application, however, if other services which convert from temperature units exists, there could be possible to create a composed service. The proposed architecture will try to compose available services in order to create valid solutions for required services.

2 Distributed Service Composition Architecture

We propose an architecture to resolve service composition in mobile ad-hoc networks which is peer-to-peer in the sense that every node of the network contains the proposed architecture. Figure 1 depicts the different layers of the proposed architecture which are executed on each one of the nodes that participate in the mobile ad-hoc network. The architecture is built on top of the *IEEE 802.11* protocol. This protocol is the most common communication protocol used in wireless networks. Applications can be constructed on top of these layers using the functionalities they provide.

2.1 Dissemination

The proposed solution for composition does not rely in the existence of central repositories for service registration. However, information about services provided by each node must be propagated across the network in order to achieve the composition process. Service dissemination is performed by means of using

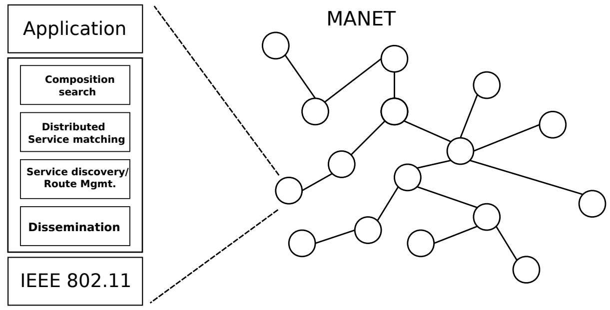

Fig. 1. Proposed layered architecture for service composition.

Table Update messages which are sent to neighbour nodes until the maximum propagation distance is reached. *Table update* messages start from the nodes which provide the services and are propagated through the network. Propagation of update messages is triggered when a previous message was received by a node and its local information table is updated, or when a new neighbour appears. The dissemination of service information has been previously studied by the authors of this paper in [1].

In order to reduce the number of messages sent, nodes group the information sent to neighbours thanks to the use of a shared taxonomy. This taxonomy could be specified using an ontology language such as OWL or RDF [11]. *Table update* messages do not contain the whole description of each service but only the type of their parameters (i.e. inputs and outputs) according to the defined taxonomy. Nodes receiving input or output information group them according to the shared taxonomy if parameters are related. Only the most generic type of related parameters will be propagated to neighbours. Parameters can be related, according to the shared taxonomy, in the following ways: *equality*, when two parameters have the same exact type, *subsumption*, when a parameter type is more generic than the other, or *no-relationship* among received parameters.

2.2 Service Discovery and Route Management

This layer uses the information which has been propagated by the *Dissemination* layer to search for available services. It offers functionality to search for services which have some specific input or output parameters. An application, or upper layer, which is trying to locate some service in the network must specify which are the types of the input or output parameters that the requested service provides. Once a search starts, a message containing the needed service description will be propagated from the searching node to its neighbours. Nodes receiving a search message will check their local services with the parameters contained in the search message. If a match is produced a message is sent to the searching node containing the network address of the node and the unique identifier of the located service.

Search messages are propagated through the network aided by the information distributed by the dissemination layer. Every node which has received parameter information from some service contains an entry which estimates the distance to the node which provides the service. This distance information, called *estimated distance*, is used by the search mechanism, in conjunction with a TTL counter, to determine if a search message could reach the needed service. If the TTL of a search message is lower than the distance to a compatible service, the search message will be dropped and not further propagation will occur. This optimization decreases the number of propagated messages which also reduces the congestion of the mobile ad-hoc network.

This layer also manages the creation and maintenance of communication routes among different nodes during the mobility of the ad-hoc network. Every time a search or a response message is received by a node, the layer creates or updates the node's routing tables accordingly. Routing tables are removed when neighbours disappear and a valid communication route is broken. The layer also provides functionality to sent *unicast* or *multicast* messages to those destinations which have been discovered thanks to previous message propagation.

2.3 Distributed Service Matching

Service composition needs to determine relationships among available services in order to discover what connections can be established. In the case of centralized solutions this can be done by matching all registered services in order to discover input and output compatibility among them. Due to the fact that services are distributed among different nodes of the mobile ad-hoc network there is no easy way to perform service matching in these networks. In the proposed solution, service matching is performed by intermediate nodes, i.e. nodes which are located in paths which connect compatible services. Nodes propagate, through the network, information about the parameters of the services they provide. Every time a node receives information about input and output parameters, which comes from other node, the information will be checked against other previously received. If a match among input and output parameters is discovered, a collision occurs.

The detection of a collision triggers a search process in order to exactly locate which services were matched and where they are located. The node which detects the collision sends a search message containing the colliding parameters using the underlying layer. The search message is propagated to the provider nodes which, upon reception, will return a response containing information about the required services. Response messages are re-propagated by collision node in order to create full communication paths among the discovered nodes. This process enables discovered nodes to communicate with each other if necessary. A graph connecting services distributed, which expresses the existence of communication routes among those services, is created in the network. The connections are maintained during network life-time and updated according to changes in the network topology.

2.4 Composition Search

Every time a node wants to search for a composed service it must specify the requirements of the desired service. This specification is performed by describing the inputs and outputs of the required service. The description will be used by the composition search layer to search for a solution, which using the available services in the network, provides a work-flow fulfilling the required service. The search for composed services consists on a search on the graph constructed by the underlying layers. The node searching for a service must create, and register, two special services which represent the *START* and *END* of the composed service. The *START* service is created as a service which has the inputs of the required service as its outputs and has no inputs. On the other hand, the *END* service has the outputs of the desired service as inputs and has no outputs.

The composition process is a search on the graph which starts from the *START* service and propagates through the service graph following the connections established among compatible services. A message is sent through the network using the discovered paths and every time a compatible service is reached it is added to the message. A message is only further propagated by a connected service if the receiving service is fully covered. A service is fully covered when the service have received a composition message from all its ancestors. One service is ancestor of other service if one or more of its output parameters are connected with one or more of the parameters of the other service. A composed service is found when message returns to the node which started the composition through the corresponding *END* service. This is only possible if a path, with all intermediate services covered, which connects the *START* and *END* services exist.

Figure 2 shows the different layers working to create the distributed graph and obtain a composition.

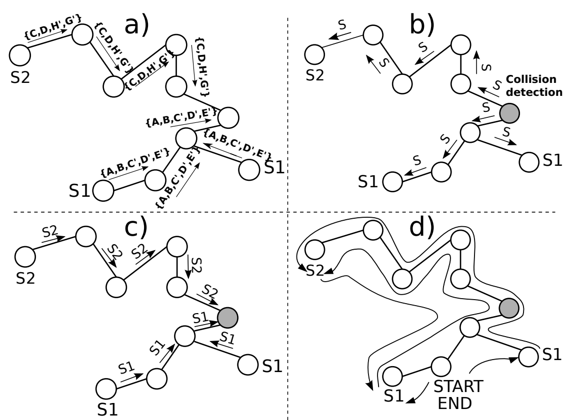

Fig. 2. Different phases of service composition: a) dissemination of services' parameters. b) collision detection by intermediate node and search of services. c) responses from matching services to search. d) connection of *START* and *END* services.

3 Evaluation

The proposed protocol has been fully implemented and evaluated using the NS-2 network simulator[1] extended with AgentJ[2]. An evaluation of the proposed solution (DSM) was performed with the following simulation parameters: 30 nodes, 300 x 300 area, 100 meters transmission range, *Random Waypoint* model with uniform speed distribution of 0-5 m/s, *IEEE802.11* as the MAC protocol with a transmission data rate of 54 Mb/s and a maximum packet size of 1500 bytes. The relation between the transmission range and the area tries to simulate a small network, conference or meeting room, where devices are carried by people which move at walking speeds (equivalent to 10 m transmission range on a 30 m x 30 m area). A 30 % of the nodes provide services and randomly selected nodes start composition searches with a frequency of 1 search per second.

Obtained results have been compared with a flood based composition. In the flood-based search (FBS), services which need to compose a service start the process by sending a message to all nodes in the network. Each time the composition message is received by a network node, the current service composition is checked against all the services provided by the node and added if matching exists. Message propagation finishes when the *END* services receive the composition search message. Figure 3 shows the results of the experimentation while finding compositions of increasing length (3, 5, 7 services). As can be seen, the proposed solution obtains compositions faster than the flood based one because the usage of the service graph avoids the sending of search messages through unconnected routes. However, it also reduces the ratio of found compositions because graph maintenance has a cost during network mobility. It could be possible for the search process to explore routes which are not longer valid, which produces that some composition search messages are lost.

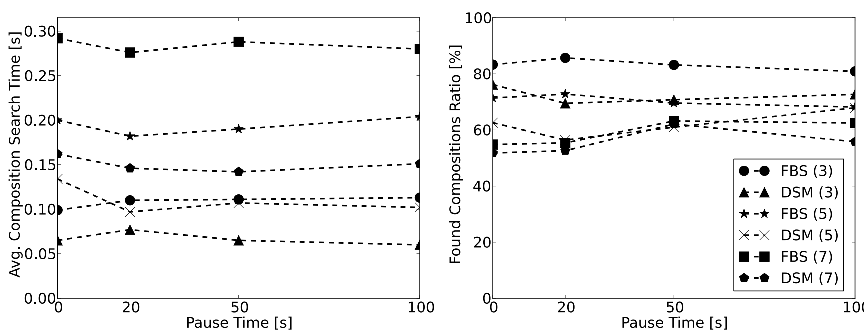

Fig. 3. Comparison of proposed architecture for service composition with a flood based solution for compositions of 3, 5 and 7 services.

[1] The Network Simulator - ns-2 - http://www.isi.edu/nsnam/ns/

[2] AgentJ Java Network Simulations in NS-2 - http://cs.itd.nrl.navy.mil/work/agentj/

4 Related Work

The application of dynamic service composition to mobile networks has been studied in [2] where the authors propose the usage of a Hierarchical Task Network to decompose the searched composed service into simpler parts, and in [3] which proposes the integration of a group-based service discovery protocol and a selective forwarding of services based on discovered paths. The usage of an overlay network for service composition was firstly proposed in [8]. Another solution is proposed in [7] where an specification of the required service as an abstract graph is instantiated using the services of the network. A more complete solution which covers the service discovery, composition and substitution of broken compositions is presented in [14]. Other solutions for service composition in mobile ad-hoc networks have been proposed. For example, in [13] fuzzy logic is applied to select among available services. Another solution is proposed in [10] where the usage of a Distributed Constraint Satisfaction Problem is applied to find connections among services. In [15] the authors study the composition in opportunistic networks by propagating composition services when connections among neighbour nodes exist. In those works the process is performed by means of instantiating a pre-constructed work-flow graph while our proposal creates the composition work-flow from scratch starting with the provided description.

The idea of performing dynamic service composition with the usage of a precomputed service graph has been firstly proposed in [4] or in [6]. However, the solution is only applied to static networks while our work is directed to mobile ad-hoc networks. On the other hand, the usage of taxonomies and semantic technologies has been previously explored in [3] and [5]. However, these two approaches treat the services as a whole and do not enable to search for services using the parameters they provide. In [9] the authors also propose a service discovery mechanism which uses ontology information during service search. The dissemination and search layers have some resemblances to a pub/sub system. Application of pub/sub systems to mobile ad hoc networks has been studied in [16] and in [12]. Our proposal includes the usage of a taxonomy of concepts while establishes and maintains communication routes across the network among the different nodes which provide services.

5 Conclusions and Future Work

This paper has presented an architecture for service composition which can be applied to mobile ad-hoc networks. It is designed as a layered solution which provides many features to the application layer. Our proposal takes into account the dynamic nature of mobile ad-hoc networks by reacting to the topology changes which could arise during its life-time. In addition, all nodes in the network share the same architecture design meaning that there are no differences among participant nodes. This way, the possibility of failure due to network changes is reduced because it avoids the existence of any centralization which could be a point of failure. For the future, we plan to test the proposed solution with larger networks and use real devices to compare results with the simulation.

References

1. Aguilera, U., López-de Ipiña, D.: A Parameter-Based service discovery protocol for mobile Ad-Hoc networks. In: ADHOC-NOW 2012, LNCS 7363. pp. 274–287. Springer-Verlag (2012)
2. Basu, P., Ke, W., Little, T.D.C.: Scalable service composition in mobile ad hoc networks using hierarchical task graphs. In: Proc. 1st Annual Mediterranean Ad Hoc Networking Workshop (2002)
3. Chakraborty, D., Yesha, Y., Joshi, A.: A distributed service composition protocol for pervasive environments. In: Wireless Communications and Networking Conf., 2004. WCNC. 2004 IEEE. vol. 4, pp. 2575–2580 Vol. 4. IEEE (2004)
4. Gu, Z., Li, J., Xu, B.: Automatic service composition based on enhanced service dependency graph. In: Proc. of the 2008 IEEE Intl. Conf. on Web Services. pp. 246–253. IEEE Computer Society (2008)
5. Helal, S., Desai, N., Verma, V., Lee, C.: Konark-a service discovery and delivery protocol for ad-hoc networks. In: Wireless Communications and Networking. vol. 3, pp. 2107–2113. IEEE (2003)
6. Hu, S., Muthusamy, V., Li, G., Jacobsen, H.: Distributed automatic service composition in large-scale systems. In: Proc. of the second international conference on Distributed event-based systems. pp. 233–244. ACM, Rome, Italy (2008)
7. Hu, Z., Tang, X., Wang, X., Ji, Y.: A distributed algorithm for DAG-Form service composition over MANET. In: Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. Intl. Conf. on. pp. 1664–1667 (2007)
8. Huang, J., Bai, Y., Zhang, Z., Kong, J., Qian, D.: Service forest: Enabling dynamic service composition in mobile ad hoc networks. In: Proc. of the The 2007 Intl. Conf. on Intelligent Pervasive Computing. pp. 174–177. IEEE Computer Society (2007)
9. Islam, N., Shaikh, Z.A.: Towards a robust and scalable semantic service discovery scheme for mobile ad hoc network. Pak. J. Engg. & Appl. Sci. Vol 10, 68–88 (2012)
10. Karmouch, E., Nayak, A.: Capability reconciliation for virtual device composition in mobile ad hoc networks. In: Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th Intl. Conf. on. pp. 27 –34 (2010)
11. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C Recommendation 10, 2004–03 (2004)
12. Paridel, K., Vanrompay, Y., Berbers, Y.: Fadip: Lightweight publish/Subscribe for mobile ad hoc networks. In: On the Move to Meaningful Internet Systems, OTM 2010. pp. 798–810. Springer-Verlag (2010)
13. Prochart, G., Weiss, R., Schmid, R., Kaefer, G.: Fuzzy-based support for service composition in mobile ad hoc networks. In: Pervasive Services, IEEE Intl. Conf. on. pp. 379–384 (2007)
14. Ruta, M., Zacheo, G., Grieco, L.A., Di Noia, T., Boggia, G., Tinelli, E., Camarda, P., Di Sciascio, E.: Semantic-based resource discovery, composition and substitution in IEEE 802.11 mobile ad hoc networks. Wireless Networks 16(5), 1223–1251 (2010)
15. Sadiq, U., Kumar, M., Passarella, A., Conti, M.: Modeling and simulation of service composition in opportunistic networks. In: Proc. of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems. p. 159–168. MSWiM '11, ACM, New York, NY, USA (2011)
16. Yoo, S., Son, J.H., Kim, M.H.: A scalable publish/subscribe system for large mobile ad hoc networks. Journal of Systems and Software 82(7), 1152–1162 (2009)