Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 56 (2015) 133 – 138

Procedia Computer Science

The 12th International Conference on Mobile Systems and Pervasive Computing(MobiSPC 2015)

HARMS-based Service Discovery Protocol using Address-DNS

Kyuhwan Lee[a,c,*], Yunsuk Yeo[b], Tai-Myoung Chung[b]

[a] *Institute for Information & communications Technology Promotion, 1548 Yuseong-daero, Yuseong-gu, Daejeon, 305-348, Korea*
[b] *Sungkyungkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeong Gi-do, 440-746, Korea*
[c] *Purdue University, 401 North Grant Street, West Lafayette, IN, 47907, US*

Abstract

As much more robots are connected to IoT, finding the intended things becomes a more difficult and fundamental function. When people want to instruct a robot in doing its specialties, we can easily know a service name such as printing and moving but it's hard to get the information of intended service providers if either it's the first time to use or servide provider's information is not cached. To cope with this, in this paper, we suggested discovery protocols in two different circumstances. One is to use postal address and DNS system over Internet, and another is to adopt maximum hop counts based on AODV routing protocol, which showed both no effect on the network size and diminution of overhead with same success probability.

Keywords: HARMS; Service discovery; Routing protocol; Broadcasting; Domain Name Service;

1. Introduction

The growth of semiconductor and information technologies brings new services which can find and process the information anywhere or make intelligence by processing lots of data. Many small-sized devices can collect data surrounding us, transfer it to either each other or data center, and utilize that information in lots of ways. Many researchers continued to adopt these tries to things and named it as IoT(Internet of Thing). Technically, it is defined by the infra to provide service based on the communication process between physical things in the real world and virtual things in the cyber world. A similar concept historically emerged decades ago such as WSN(Wireless Sensor Networks), ubiquitous computing, pervasive computing and so on. Although many research has been done, it still has a great ripple effect throughout the economy and industry according to NIC(National Intelligence Council) in the US[1]. Most applications have been using the Internet, but because they can be sometimes in a situation without legacy wireless infrastructure like WiFi or LTE(Long-Term Evolution) networks, IoT is very closely relative to M2M(Machine-to-Machine) or D2D(Device-to-device) technology for communicating with each other.

Legacy robots have been improved and focused conventionally on hardware functions to do hazardous or strenuous tasks on behalf of humans. In the past, they had worked for simple and repeated tasks but robots recently replace

[*] Corresponding author. Tel.: +1-765-631-4152; fax: +1-765-496-1212.
E-mail address: lee1981@purdue.edu

doi:10.1016/j.procs.2015.07.183

human-specialized field little by little. Additionally, the more robots are connected to IoT, the earlier the all-things-automated world can be realized by the cooperation among them. It's necessary to use a common platform enable to connect, communicate and interact with each other even in different languages, communication technology and type of devices; hence, this research used layered architecture, HARMS(Humans, software Agents, Robots, Machines and Sensors)[2] model. Most platforms including HARMS model assume that the information which is necessary to make a connection is either known or specified by a website. However, when these services are generalized in everyday life, it is impossible to store all information on each connection on different devices. To solve this issue, we propose HARMS-based service discovery protocol using DNS(Domain Name System) and postal address, which are widely used and easy to remember by humans and same functionality to search something.

The remainder of this paper is organized with Section 2 describing the backgrounds of service discovery and routing protocols. Section 3 proposes our new scheme, HARMS-based Service Discovery Protocol using Address-DNS. Section 4 provides the performance evaluation with 3 cases and 7 difference manners. Finally, section 5 summarizes the conclusion and future works.

2. Backgrounds

2.1. Service discovery protocols

A number of researchers have found a way to optimize procedures and reduce overhead for service discovery. The key problem is who has service providers information and how to communicate with nodes, named server or directory, which have responsibility to advertise, reply and update the information. These procedures can be performed by consisting overlay network like virtual backbone or specific channels where only both service providers and nodes interested in those services can send or listen messages. Before proposing our scheme, we briefly introduce pioneering SDPs(Service Discovery Protocol) in terms of two points: who and how.

JINI is one of classical SDPs and designed for devices over JVM(Java Virtual Machine). It has lookup service, as a directory, both storing service descriptions advertised by providers and responding requests for providers or clients. Originally, to find lookup service, clients participate in the multi-cast channel and then send a message but if they cannot receive the address of a lookup service, it is necessary to use a broadcast manner generally restricted to the local LAN or business network. On the specification, there are no ways to make consistency among lookup services so even if the service exist, clients cannot find it[3]. The Universal Plug and Play(UPnP), developed by a industry initiative of vast leading companies, aimed to provide pervasive peer-to-peer network connectivity among connected devices and adopted SSDP(Simple Service Discovery Protocol) where services are announced by a service provider with multi-cast IP to join or withdraw services. Even if SSDP use multi-cast channels to discover services similar to JINI, it has no server such as lookup service. The other researches in wired networks such as SLP(Service Location Protocol) developed by IETF(Internet Engineering Task Force) and Apple-leading Bonjour use a multi-cast manner but the difference is whether servers exist or not.

However, in wireless networks, a multi-cast router or a server is very high cost due to the mobility of devices so lots of attempts adapted these technologies for use when the network topology can frequently change. In a survey[3], these were divided into two categories: directory-based and directoryless. The directory means a repository of available services in the network and can be either centralized or distributed. Most centralized approaches mainly used service discovery protocols in wired networks so it is very simple and provides well-optimized performance but known to be not suitable for IoT environment because of the assumption that a node has poor resources not enough to act as a server. Accordingly, some researches suggested the representative election where some nodes have better resource or location efficiently to respond to requests with lower overhead. These representatives can have either same information among them or a part of information. In the former case, a representative updates its change to all other representative so a client easily get all information form one nearest itself. The latter includes many researches using proximity or DHT(Distributed Hash Table) to disseminate service providers information[4].

In directoryless approaches, there are no nodes responsible for advertisements, requests and updates of services so a client have to gather information where a provider exist whenever finding out a service. This feature likes a routing protocol in ad-hoc networks so many attempts employ cross-layer technique to take advantage of information from another layer to alleviate redundant message transmissions and to manage it efficiently.

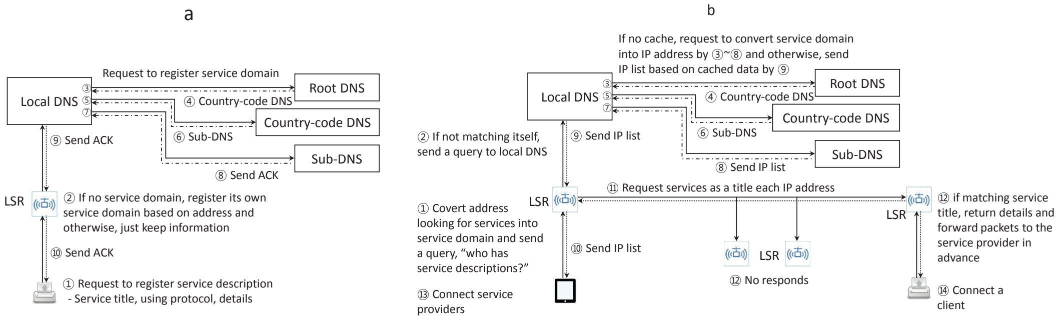
a
Local DNS
Request to register service domain
Root DNS
③
⑤
⑦
④ Country-code DNS
Country-code DNS
⑥ Sub-DNS
Sub-DNS
⑧ Send ACK
⑨ Send ACK
LSR
② If no service domain, register its own service domain based on address and otherwise, just keep information
⑩ Send ACK
① Request to register service description - Service title, using protocol, details
b
If no cache, request to convert service domain into IP address by ③~⑧ and otherwise, send IP list based on cached data by ⑨
Local DNS
Root DNS
④ Country-code DNS
Country-code DNS
⑥ Sub-DNS
Sub-DNS
⑧ Send IP list
② If not matching itself, send a query to local DNS
⑨ Send IP list
LSR
⑪ Request services as a title each IP address
① Covert address looking for services into service domain and send a query, "who has service descriptions?"
⑩ Send IP list
⑬ Connect service providers
LSR
⑫ No responds
LSR
⑫ if matching service title, return details and forward packets to the service provider in advance
⑭ Connect a client

Fig. 1. (a) Registration; (b) Service inquiry.

2.2. Difference between service discovery and routing protocols

In order to connect to each other, routing protocols at the network layer aim to find a node having the specific identification(ID) while service discovery at the application layer aims to reveal node ID providing services. The purpose is the same for both protocols, which is finding the specific node and connecting with it. However, the difference is how to find it; either by the nodes ID or service descriptions. Only one ID should be assigned to each entity but services can be provided by multiple providers. Most service discovery protocols integrated with well-known routing schemes such as AODV(Ad-hoc On-demand Distance Vector), DSR(Dynamic Source Routing), and ZRP(Zone Routing Protocol). They embedded and extracted service information in/from management packets like route request, reply and update. It can reduce to disperse lots of overhead packets compared to the case that both routing and service discovery protocol are adopted separately.

3. HARMS-based Service Discovery Protocol

3.1. Hierachical address-based query approaches

In HARMS model, all of the agents, sensors, machines and humans produce outputs as physical actions and processed information which affect or are considerably related to the environment surrounding itself. Therefore, when we find a service provider, we have no alternative but to ruminate about its location. For example, if we want to print something, it implies that someone is supposed to pick up printed papers at the printer. When finding a washing machine, its involved that we need to go there with laundry. Even if services incur outputs either in remote areas or in cyber spaces, it can be set to remote location and URI(Uniform Resource Identifiers), also known as URL(Uniform Resource Locater), respectively. So, a service query includes region information which can be either nearest a client or assign the specific point. Postal address, in the process of global standardization, is one of widely-used methods of identifying locations and suitable to HARMS model because of its human-like, worldwide use and no build-up cost. Also, GPS(Global Position System) is an internationally adopted manner for locating objects, but it both cannot measure its position in the indoor environment or between tall buildings and doesn't seem human-like because it uses numerical digits as longitude and latitude. Additionally, it must require for all things to have a GPS receiver, which is not fit to HARMS model.

Also, we have DNS which is a huge hierarchical system and used in the whole world in order to translate human-like expression, domain name, into numeric information, IP address. One of top-level domains is ccTLD(country code Top-Level Domain) which is two letter both established for countries or territories and can be mapped to the country of postal address. The ccTLDs are handled by root DNS. At the second level, it can be either domain name or a delimiter which identifies the administrative owner relative to its domain. It can be mapped on a state or city and the rest of domain name represents other remaining address. We defined a service domain, generated from the postal address, and LSR(Local Service Resolver) which registers and withdraws its service domain under LSR's management. LSR can be embedded in AP, router or gateway and our scheme assumed that DNS can store service domains and LSR's IP address because DNS is not designed for only converting the host name into IP address. The standard describes that

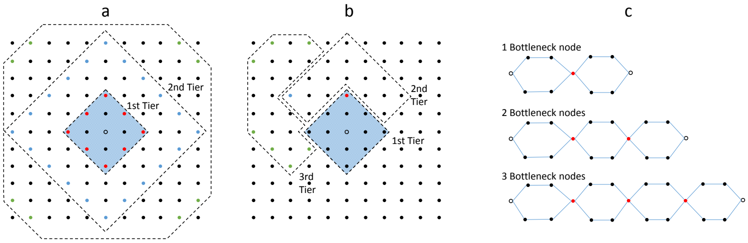

Fig. 2. Grid topology (a) requesting all boundary nodes; (b) requesting one among boundary nodes; (c) Bottleneck topology.

“the DNS is a general (if somewhat limited) hierarchical database, and can store almost any kind of data, for almost any purpose”[5 6]. In addition, DNS is distributed database and hierarchical structure similar to the postal address. It affords our scheme because the domain can have 127 sub-domains and the length of each label and all is equal or less than 63 and 253 characters, respectively.

For registering its own services, as shown in Fig. 1 (a), a service provider requests for LSR to register service descriptions composing of service title, using protocol and details. If LSR has already registered its service domain, it just sends acknowledgment back and if not, it registers its service domain, not service descriptions. When a client wants a service as shown in Fig. 1 (b), the first thing is to get IP address of LSRs related to intended regions. Similar to the case of finding IP address of host name, if there are no cached data, it starts sending a query to root DNS and receives IP address list through a couple of requests. Original DNS returns one IP per host name but converting service domains provides multiple IPs, which can be supported by current message type, 2 bytes ANCount. A client receiving LSR list send a service title to them and waits for receiving service provider IPs and using protocols. After this, agents or programs at the application layer can connect them based on their protocol type.

3.2. Exponentially increased search area

Without connecting to Internet, we cannot get LSRs IP address from DNS and IP routing is also not working because both there are no routers/servers and the hierarchical routing strategy is not useful. Some approaches provided ways to elect special nodes acting like a router by establishing a multi-cast channel able to listen only to registered ones. The most cases in their researches is good performance if the network topology is stable or some nodes have more computing power or resources than others. However, if considering that ad-hoc means off-the-cuff and temporally nature, the specific circumstance is not closed to HARMS model. In general cases, a broadcast manner is one of the most powerful technique to find something, but it both consumes high network cost and causes a big problem, known as broadcast storm problem[7] leading to less reachability. In this chapter, therefore, we propose an exponential increase broadcast manner without both GPS and election rules in order to reduce the number of broadcasting messages but it does not diminish a chance of success to fine services or nodes in the intended area.

For limiting scope to propagate broadcast message, there are three approaches: hop count, distance and time to live. First is giving max-hop count which is decreased by one when re-broadcasting. It does not consider what time broadcasting stops and how long time it reaches to nodes but is very simple because of just decreased-by-one. Second is to calculate the propagated radius based on the signal power level of receiving message. Most wireless modem chips provide the quantized RSSI(Received Signal Strength Indicator) so we can easily get the distance but its vulnerable to multipass, obstacles and interferences. Last is to specify the time until re-broadcasting and suitable to time sensitive applications. Our scheme in HARMS model uses the modified hop counts because of simple and general manner.

Generally, we have no information about the network size and the distance between a finder and found one, and hence first broadcast a query to small region, 1st tier shown in Fig. 2 (a) and (b), and then gradually enlarge the propagation area, 2nd and 3rd tier. After 2nd tier, prior broadcast regions are overlapped so to cope with overlapping region, we defines two fields in the message: original node ID and its sequence number. A node with zero hop count does not re-broadcast and notify that I am in the last hop, named a boundary node. Then, a finder decides whether to enlarge the search area or not. In the former cases shown in Fig. 2 (b), it requests again for subset with inconsistent routes to re-broadcast by unicast. Fig. 2 (a) takes the shortest time while it has no effect of reducing overlap. To prevent reversely propagating, nodes re-broadcast only when both original node ID and sequence number are not in broadcast

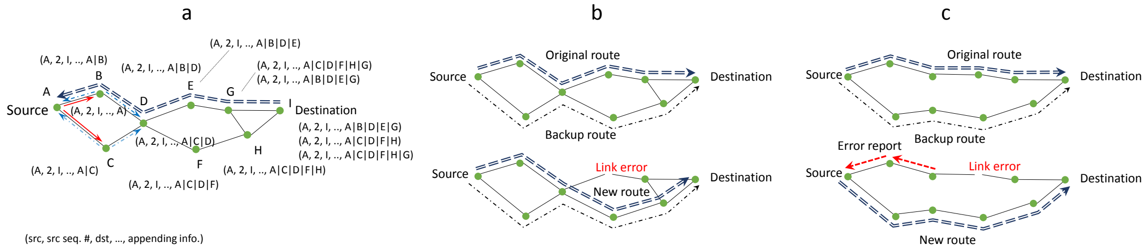

Fig. 3. (a) Discovery procdures; (b) Partial route recovery; (c) Full route recovery.

history. When a discovery message reaches to a service provider, it sends a reply through boundary node and based on the reply, the route establishment is basically same as AODV(Ad-hoc On-demand Distance Vector). However, the destination replies all requests receiving through different routes in order to set up multiroutes. Additionally, because all control messages such as hello, request and reply accumulate their past information such as routes and one-hop neighbors, nodes can use it when a request matches in cache. If nodes have a route to destination, they dont rebroadcast it and directly send it by unicast.

3.3. Robust link errors

When a wireless link is broken, the node recognizing it searches another route in own routing table and if it exists as shown in Fig. 3 (b), a new route is set up without reporting route change to the source, named partial route recovery, and otherwise, error message is propagated to the source. If the intermediate node has another route, the error propagation stops and it sets up new route. Fig. 3 (c) describes that intermediate nodes are sequentially connected with no other branches. In this case, the source receives the error message and based on stored multiroutes, it can establish a whole different route compared to the original route, named full route recovery. If there are no chances to recover a route, the source initiates a new route discovery. If there is an opportunity to fix a route, we can get benefit from avoidable broadcasts.

4. Performance evaluation

We have developed our mobile nodes using the discrete event-driven network simulator, OMNET++ 4.6[8]. Intended networks, called map, contains 121(11x11) nodes where each node can directly communicate with four cardinal points; other 4 neighbors is out of transmission range. A mobile nodes adopts IEEE 802.11 module supporting CSMA/CD, as a wireless interface, with default parameters in INET[9]. Two performance metrics are observed: overhead and success probability. The former is the number of all broadcast messages used to find a service provider and the latter is calculated by the number of successes discovering intended services over the number of attempts.

The simulation result from three situations shown in Fig. 3 and fours schemes for service discovery: ours, random delay, drop, and drop & random delay. Fig. 4 (a) shows that a finder is located in top and left position, (1, 1), and a service provider is at diagonal position. All schemes except ours are almost not affected by the distance between a finder and a service provider in terms of overheads. The bigger the topology size is, the performance is worse. However, our scheme depends on only the location of a service provider. The drop manner certainly reduces the number of re-broadcasting but in most cases, it can find a service provider with less than the probability of 0.5. The random delay technique provides higher success probability and overhead than those of drop manner. The combination of both drop and random delay is good choice at this topology but they are still producing lots of overheads. If we assume that each position is the same probability for locating a service provider, our scheme provides almost equal performance with a half of overhead compared to combination strategies. When a finder is at the center of the topology, shown in Fig. 4 (b), results have same pattern as previous experiment.

Fig. 4 (c) shows the case that some bottleneck nodes exist between a finder and a service provider, which implies that bottleneck nodes is more important position in message flows. If they suffer collision, a finder cannot have choices to discover services because of no route which is a more practical situation than grid topology. The random delay

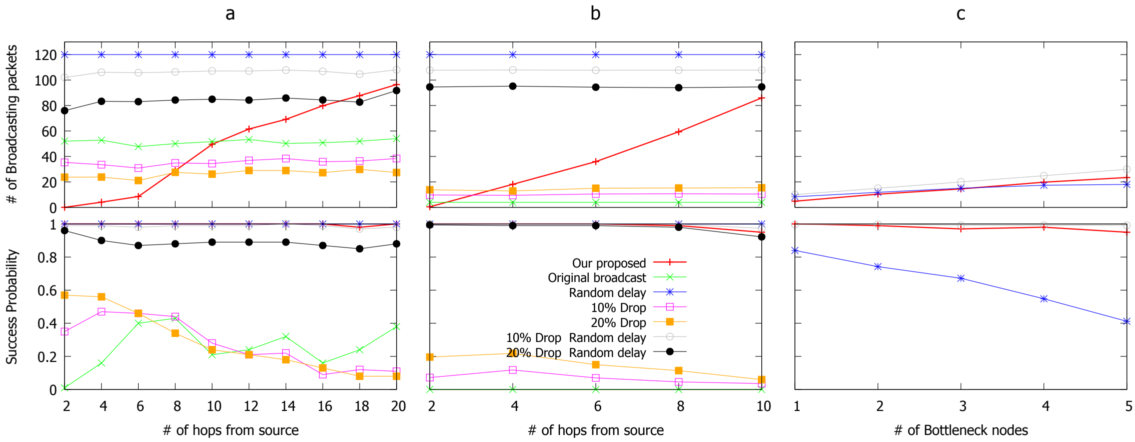

Fig. 4. Grid topology (a) with top-left source; (b) with center source; (c) Bottleneck topology.

has worse reachability because even if channel acquisitions are scattered, both it's small duration and the collision probability gradually increases when a message passes through bottleneck nodes. Our scheme and random delay with 10% drop show similar performance in terms of success probability and overhead: the former is slightly less overhead and the letter is little higher success probability.

5. Conclusion

IoT with robots has difficulty to find somethings because there are huge amount of things connected to each other. On a wide view, things including robots can communicate either over Internet or not. In the former case, we proposed service discovery using postal address and DNS, which combines two systems widely used over the world. In the letter case, we proposed exponentially increased search area efficiently to find out a service provider and simulated with 3 situations and 7 different manners. The dominant factor of our scheme is the distance from intended things while others are the network size. If considering the service area robots can provide, we believed our way is more practical. In future works, we need to resolve the authentication of service domain registration and consider time domain.

References

1. National Intelligence Council, Disruptive Civil Technologies: Six Technologies With Potential Impacts on US Interests Out to 2025. 2014. URL: `http://fas.org/irp/nic/disruptive.pdf`.
2. Matson, E.T., Min, B.C.. M2m infrastructure to integrate humans, agents and robots into collectives. In: *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*. IEEE; 2011, p. 1–6.
3. Ververidis, C.N., Polyzos, G.C.. Service discovery for mobile ad hoc networks: a survey of issues and techniques. *Communications Surveys & Tutorials, IEEE* 2008;**10**(3):30–45.
4. Sivavakeesar, S., Gonzalez, O.F., Pavlou, G.. Service discovery strategies in ubiquitous communication environments. *Communications Magazine, IEEE* 2006;**44**(9):106–113.
5. Elz, R., Bush, R.. Clarifications to the DNS Specification. RFC 2181 (Proposed Standard); 1997. URL: `http://www.ietf.org/rfc/rfc2181.txt`.
6. Cheshire, S., Krochmal, M.. DNS-Based Service Discovery. RFC 6763 (Proposed Standard); 2013. URL: `http://www.ietf.org/rfc/rfc6763.txt`.
7. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.. The broadcast storm problem in a mobile ad hoc network. *Wireless networks* 2002;**8**(2-3):153–167.
8. OMNET++, Discrete Event Simulator. 2014. URL: `http://omnetpp.org/`.
9. INET, An OMNeT++ model suite for wired, wireless and mobile networks. 2014. URL: `http://inet.omnetpp.org/`.