5-12-2005

Service discovery in wireless ad-hoc control networks

Shengrong Bu
University of Wollongong

F. Naghdy
University of Wollongong, fazel@uow.edu.au

Service discovery in wireless ad-hoc control networks

Abstract

A new concept in distributed control systems called Wireless Ad-hoc Control Networks (WACNets) is developed. WACNets is formed by a collection of nodes with the ability to sense, actuate and control. The network does not have a fixed structure, but evolves and self organises itself according to the control requirements of the system. The service discovery developed for WACNets is reported. A review of the existing Service Discovery Protocol (SDPs) including Jini, Salutation, Universal Plug and Play (UPnP), and Bluetooth technology is carried out. An overview of WACNets is provided. The service discovery protocol developed for WACNets is introduced and its characteristics are described. Some results are provided.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Bu, S. & Naghdy, F., Service discovery in wireless ad-hoc control networks, Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 5-8 December 2005, 157-162. Copyright IEEE 2005.

Service Discovery in Wireless Ad-hoc Control Networks

Shengrong Bu, Fazel Naghdy
School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong, Wollongong, Australia, 2522
Email: sb91@uow.edu.au, fazel@uow.edu.au

Abstract

A new concept in distributed control systems called Wireless Ad-hoc Control Networks (WACNets) is developed. WACNets is formed by a collection of nodes with the ability to sense, actuate and control. The network does not have a fixed structure, but evolves and self organises itself according to the control requirements of the system. The service discovery developed for WACNets is reported. A review of the existing Service Discovery Protocol (SDPs) including Jini, Salutation, Universal Plug and Play (UPnP), and Bluetooth technology is carried out. An overview of WACNets is provided. The service discovery protocol developed for WACNets is introduced and its characteristics are described. Some results are provided.

1. INTRODUCTION

With the advent of mobile computing and wireless communication, wireless ad-hoc networks are becoming feasible and attractive. Such systems represent peer to peer wireless communication among intelligent devices without any fixed infrastructure. The peers form a network of various devices including sensors, actuators, appliances, PDAs, laptops, etc. With the increasing number of services, automatic service discovery plays an essential role in ad hoc communications, where no fixed infrastructure is present but the nodes themselves form the network.

This paper reports a service discovery protocol developed for Wireless Ad-hoc Control Networks, a concept being developed by the authors for distributed control. WACNets is formed by a collection of nodes with the ability to sense, actuate and control. The network does not have a fixed structure, but evolves and self organises itself according to the control requirements of the system.

WACNet explores a framework for organic, evolutionary and scalable method of integrating a large number of intelligent and heterogeneous nodes. Each node consists of sensing and/or actuation, local intelligence and control, data processing and communication components.

The paper is organized into six sections including the introduction. Section 2 provides a background on service discovery protocols. It highlights the characteristics of the most commonly used protocols in industry. Section 3 presents a general overview of WACNets. Section 4 focuses on the Service Discovery Protocol for WACNets. Section 5 specifically describes the implementation of WACNet SDP. Section 6 describes some results obtained from validation of WACNet SDP. Section 7 draws some conclusion and outlines future work possible.

2. BACKGROUND

Typically, service discovery involves clients, lookup or directory servers and service providers. Service Discovery Protocol (SDP) enables applications, network devices, and services to advertise their capabilities to find other applications, network devices, and services to complete specified tasks [1]. Service registration and lookup are very important components for most Service Discovery Protocols (SDPs). Currently, the most popular SDPs are SLP (Service Location Protocol), Jini, Salutation, UPnP (Universal Plug and Play), and Bluetooth SDP [2].

SLP [2], developed by the IETF SvrLoc working group, aims to be a vendor-independent standard. It is designed for TCP/IP networks and is scalable up to large enterprise networks. The SLP architecture consists of three main components: User Agents (UA), Service Agents (SA) and Directory Agents (DA), which is not mandatory. In SLP, the failure of the DA will lead to information loss and breakdown in service discovery.

Jini [2], developed by Sun Microsystems, is an extension of the programming language Java. It addresses the issue of how devices connect with each other in order to form a simple ad hoc network and how these devices provide services to other devices in the network. The fact that Jini is tightly tied to the programming language Java makes it dependent on the programming environment.

The Salutation [2] architecture is developed by the Salutation Consortium. Service discovery in Salutation is defined at a higher layer, and the transport layer is not specified. Salutation is also independent of the network technology and the programming language, and may run over multiple infrastructures. The Salutation architecture consists of Salutation Managers (SLMs) that have the functionality of service brokers. Services register their capabilities with an SLM, and clients query the SLM when they need a service. After discovering a desired service, clients are able to request the utilization of the service through the SLM.

UPnP [2] is being developed by an industry consortium, founded and lead by Microsoft. Its usage is proposed for small office or home computer networks, where different devices are able to automatically discover and utilize each other. The main drawback of UPnP is that it depends on a

0-7803-9399-6/05/$20.00 © 2005 IEEE

specific network technology (TCP/UDP and IP). Also, an UPnP entity needs heavy resources to allow support for GENA and SOAP web servers, and XML parsing [2]. It is purely peer-to-peer architecture and has potential to increase the network traffic due to extensive use of multicast messaging.

Bluetooth SDP [3] is based on the Piano platform by Motorola and has been modified to suit the dynamic nature of an ad hoc network. SDP is used to locate services provided by or available via a Bluetooth device. Bluetooth SDP requires an SDP server running on any device that is capable of providing services. The server maintains a set of service records that contain a list of attributes. A Bluetooth device intending to use a service is called an SDP client.

The SDPs like SLP, Jini and UPnP are designed with traditional wired network, which have quite different characteristics from wireless ad-hoc networks. The Bluetooth SDP is based on connecting to a specific node and querying about its available services. When the network becomes larger, the protocol of querying every single node in the network about the possibly unavailable service wastes both time and resources. Various kinds of Service Discovery Protocols should therefore be developed for self-organizing ad-hoc networks.

3. WACNETS OVERVIEW

The configuration and type of self-organizing ad-hoc network varies based on the characteristics and number of nodes in the network and the applications of networks. In this paper, Wireless ad-hoc Control Network is introduced because of its importance in application.

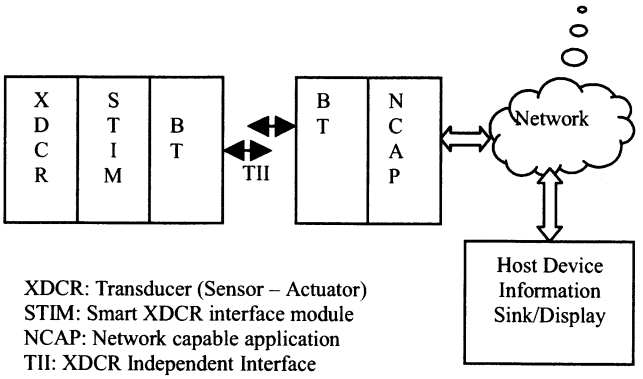
Wireless ad-hoc Control Networks (WACNets) are designed for distributed and remote monitoring and control. Such systems represent the next stage in the evolution of distributed control and monitoring. Recent advances in mobile computing, wireless communications, MEMS-based sensor technology, low-powered analogue and digital electronics, and low-power RF design have created opportunities for the introduction of WACNets.

WACNet explores a framework for organic, evolutionary and scalable method of integrating a large number of intelligent and heterogeneous nodes. Each node consists of sensing and/or actuation, local intelligence and control, data processing and communication components. The size, number, density, capabilities and location-dependency of such nodes will be determined by the specific application for which the nodes are employed. Ideally they are expected to be low-cost, low-power, multi-functional and small in size.

True self-organization of the nodes and collaboration between them is a necessity in WACNet. This is determined based on the task expected to be carried out by a cluster of nodes and the interaction they should perform with their environment and other nodes to achieve that task.

In the proposed system, IEEE 1451 standards are employed because of their ability to provide self-identification, self-testing and adaptive calibration [4]. A combination of IEEE 1451 compliant smart sensor and Bluetooth standard implements several advanced features such as plug-n-play while maintaining minimum hardware overhead at the transducer nodes.

In the proposed system, STIM uses a Bluetooth to communicate with an NCAP module, which is connected to the network via a wired module. In this kind of network, the monitoring station can exist far away from the Plant (or at any point within the Internet.) The structure of wireless implementation of IEEE1451 in WACNet is shown in Fig. 1.



XDCR: Transducer (Sensor – Actuator)
STIM: Smart XDCR interface module
NCAP: Network capable application
TII: XDCR Independent Interface
BT: TX/RX Bluetooth transmitter/ receiver

Fig. 1. The Combining Model of IEEE 1451 and Bluetooth Standard

As illustrated, a STIM communicates with an NCAP module over a Bluetooth link. The TII protocol, which defines a form of communication between the STIM and NCAP, is implemented via Bluetooth. NCAP nodes are connected to the local monitor and /or control system. An operator can monitor or control the whole system in real time through the local monitor system on demand. In order to achieve wide coverage for wireless STIM units over the space of a factory area, NCAP units could be distributed across an Ethernet backbone.

According to IEEE 1451.2 standard, in each STIM node, TEDS consists of one Meta-TED that includes common information for all the transducers and several channel-TEDSs with information about each transducer [5]. The parameters of Meta TEDS and Channel-TEDS are redefined to satisfy the need of WACNets and provided in [6].

The configuration and type of WACNets varies based on the number of STIMs and NCAPs present in the system. Accordingly, the structure of WACNets can be classified into three different categories of one-layer cluster model, two-layer cluster model and multi-layer cluster model.

The one-layer cluster model is the direct communication model as illustrated by an example in Fig. 2. In this model, the nodes organize themselves into local clusters. Each cluster which consists of up to seven STIM slave nodes and one NCAP master node is called a Piconet. NCAP receives data from STIMs in the cluster, performs data fusion, and transmits the aggregated data to its own cluster or other neighbouring cluster replacing the data.

In the example shown, Piconet 1 can perform a localized control function. In the piconet, the NCAP reports the local sensing and process information and control status to a remote Monitor when it is required. In Piconet 2, the NCAP

receives sensory data from STIM nodes, and then condenses them before further transmission. Based on the broadcast data received from Piconet 2, NCAP in Piconet 3 activate the actuator in the piconet.

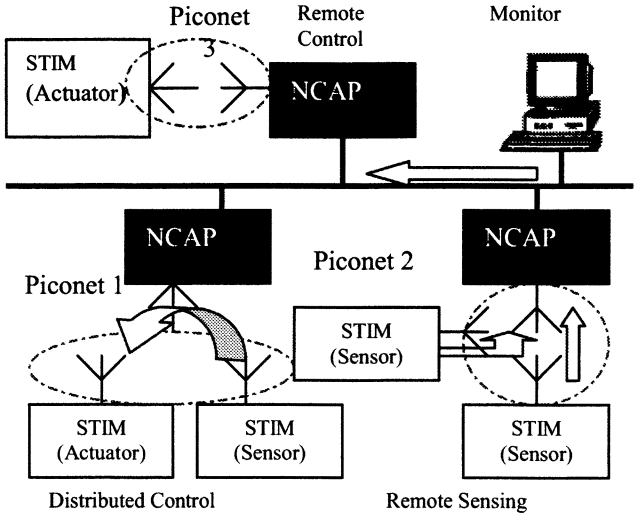


Fig.2. One-Layer Cluster of WACNets

The second type of WACNet can be employed in a situation where the number of STIM nodes are much larger that the NCAP. In this mode, the STIMs organize themselves into local clusters and communication with NCAPs via the STIM master nodes.

In the third type of WACNets, STIM nodes might be distributed widely over a large area, and therefore some nodes may not fall in the communication range of NCAPs as illustrated in Fig.3. In this example, Piconet 8 cannot directly communicate with the NCAPs.

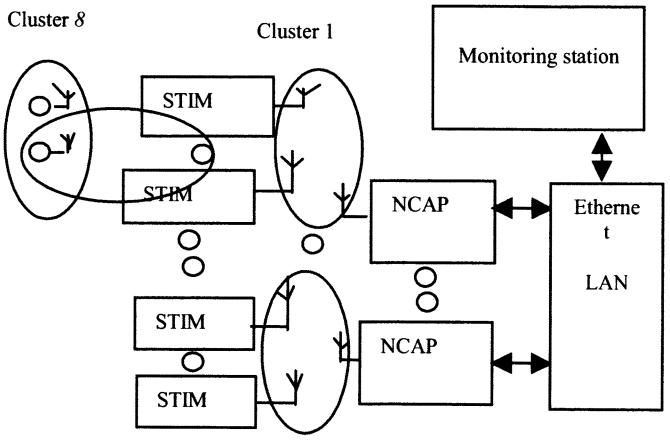


Fig. 3. Piconet 8 has no direct access to NCAPs

In the WACNet, transmission (E_{Tx}) and receiving energy costs (E_{Rx}) in each node are calculated as follows [7]:

$$E_{Tx}(k,d) = E_{elec}k + \varepsilon_{amp}kd^{\lambda} \quad (1)$$

$$E_{Rx}(k) = E_{elec}k \quad (2)$$

where k is the length of the message in bits, d the distance between transmitter and receiver node and λ the path-loss exponent ($\lambda \geq 2$), a factor that depends on the RF environment, and is generally between 2 and 4 for indoor environments, E_{elec} is the energy being dissipated to run the transmitter or receiver circuitry and ε_{amp} is the energy dissipation of the transmission amplifier.

According to equation 1, transmission of data through several short intermediate hops of data across different nodes is more energy efficient than using a long hop. Because of these advantages, the clustering and multi-hop routing algorithm employed in this work employ data aggregation methods to greatly reduce energy dissipation.

In this model, there are more a large number of STIMs in the WACNet. Neighbouring nodes with common or complimentary functions form clusters. The clusters satisfy different requirements. For example, in a building, light sensors need to update information back to monitor every several minutes. However, the temperature information does not change as fast. Hence, the real time constraint on light sensors is tighter than temperature sensors.

Another important factor in the operation of WACNet and the connectivity of the nodes is the amount of power available in the node for its communication and operation. Based on these factors, an index called Device Grade (*DG*) is defined which determines the quality of the connectivity of the nodes, clusters and NCAPs in WACNet. This index is defined by

$$DG = w_p * DeviceLeftPower + w_t * Class\,\mathrm{Re}alTime - w_r * Dis\tan ceToNCAP \quad (3)$$

In this relationship, $w_r, w_p,$ *and* w_t represent the weights for distance between the STIM master node and NCAP node, Power left in the master device, and class of real-time requirement for the cluster, respectively. The class of real-time is associated with the priority of the cluster in WACNet. In this equation, $w_r < w_p < w_t$, which indicates that the class of real-time requirement of tasks in the node is the most important parameter compared to the power left in the device, and distance to the NCAP.

For the multiple clusters in the vicinity of an NCAP, the cluster with higher *DG* has higher priority to connect to the NCAP than others. Other clusters might be forced to connect to the NCAP via other clusters. Hence, the whole network system forms a hierarchical structure of the *DG*s.

4. Architecture of Service Discovery Protocol for WACNets

Service Discovery Protocol has been found essential for WACNets in implementing automatic discovery of networked devices and remote control of one device by another. In the proposed system, the architecture of WACNet service discovery is a hybrid of peer-to-peer and client-server

architecture. In WACNet, master nodes are considered as server-like devices that reduce communication cost as service advertisement and search messages are addressed to master nodes instead of broadcast to the entire WACNet.

In the proposed system, master nodes and bridge nodes (namely, the nodes connected to more than one piconet) are aware of their piconet members. The slave node first sends a request to the master node to discover service in the same piconet. If the requested service is available in the piconet, the master node will send back a response to the slave node. Otherwise, the master node will forward the inquiry to the next bridge node.

A message from a slave is sent to the master node via the physical layer. This will require the transfer of the message from the higher layers to the physical layer in the salve and vice-versa in the master, as illustrated in fig. 4. The "lower layers" referred to in this diagram are lower layers embedded in the Bluetooth Module.

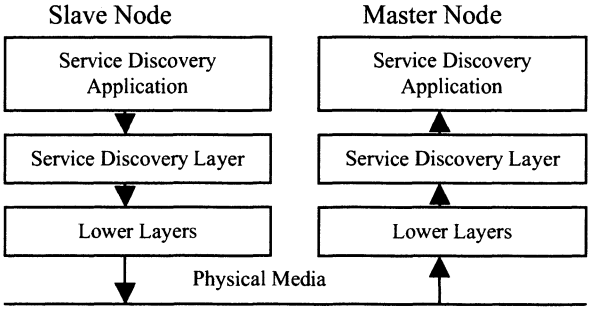


Fig.4 Protocol Model of WACNets

Every Service Discovery Protocol Data Unit (SDP_PDU), generated by the service discovery layer consists of a PDU header followed by PDU_specific parameters. The header contains three fields: a PDU_ID, a transaction ID and a Parameter Length. PDU_specific parameter varies according to PDU_ID.

5. IMPLEMENTATION OF WACNET SDP

The WACNet SDP includes two important parts: service registration and service search, which are described in the following subsections.

A. Service Registration

Service registration is an important component of WACNets SDP. It enables the nodes or users to look for a node with a particular service, and for the WACNet to offer plug and play capability for minimum data transmission. This subsection specifically describes the implementation of service registration of WACNets.

Initially, each node in the WACNet decides whether to become a master and each node has a probability p ($0<p<1$) of becoming a master. Therefore, the possibility of n masters in the network with N nodes is determined by equation 4.

$$P_p(n \mid N) = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n} \qquad (4)$$

The probability P_p reaches its maximum when $\frac{dP_p}{dp} = 0$.

In a WACNet with N nodes, n is chosen as $\frac{N}{8}$ to minimize the number of the piconets (In each piconet, there is one master and up to 7 active slave nodes.). Hence, the probability p of each node to become a master is $\frac{1}{8}$. This decision is made by each node choosing a random integral number between 0 and 7. If the number is 0, the node becomes a master, otherwise, a slave node.

A node which decides to be a slave carries out an Inquiry process to find the nearby master nodes. It then tries to connect to one of the found nodes in the identified order, by sending a *Create_Connection* packet to the node. If the node eventually responds with a *Connection_Complete* Event packet (with the status indicating successful establishment of connection.), the connection is implemented.

In the proposed system, only neighbouring nodes with common or complimentary functions (namely having same Group IDs defined in TEDS) can form clusters of nodes. Hence, the slave node sends a *JoinClusterRequest* message with all Group IDs it has in the Channel_TEDSs to the connected master, requesting permission to join the cluster. The master receiving this message will compare the received Group IDs with all the Group IDs it owns. If there is no equal Group ID, the master will disconnect this STIM node. Following that, the STIM node will connect to the next neighbouring master, and will send it another *JoinClusterRequest* message. The process continues until it connects to one of the masters with the same Group ID. Following that, the master will reply a *ServiceRegistrationRequest* message to the slave node, which then sends a *serviceRequestReply* message containing the service information the slave can provide back to the master. If the slave node cannot find a master with the same Group ID, it will finally become a master node.

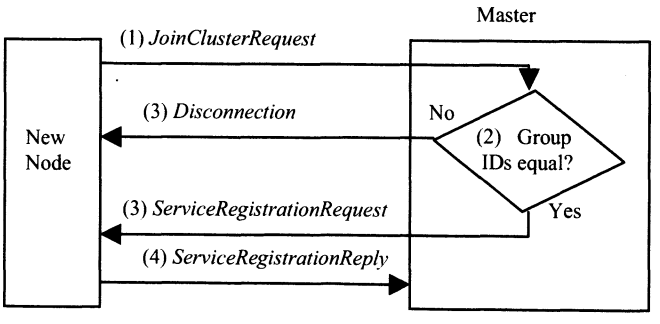


Fig. 5 the Service Operations to Join a Cluster

The whole process, as illustrated in Fig. 5, assists in the implementation of the Plug and Play feature. The numbers in the brackets shown in the diagram above denote the order in which the packet is transferred. All PDU specific parameters for the SDP_PDUs in this paper are defined in table 1.

Table 1 definition of the PDU specific parameters

PDU_ID Description	PDU specific parameters [6] [3]
SDP_JoinClusterRequest	Connection_Handle, Group IDs
SDP_ServiceRegistrationRequest	Connection_Handle
SDP_ServiceRegistrationReply	Connection Handle, Number of Implemented Channel, Each channel characteristics (Channel No., Transducer Priority, Group ID, Channel Type Key)
SDP_ ServiceRegistrationPermissionRequest	Connection_Handle, MAC_Address
SDP_ ServiceRegistrationRequest	Connection_Handle
SDP_ ServiceRegistrationReplyforNCAP	Connection Handle, Transducer Priorities and Channel Type Keys
SDP_ServiceSearchRequest	Connection_Handle, MAC_Address, Channel Type Key
SDP_SearchSearchReply	Connection_Handle, piconet ID,
SDP_ReadTEDSRequest	Connection_Handle, piconet ID,
SDP_ReadTEDSRespond	Connection_Handle, Cluster ID, piconet ID, The Technical Parameters

After the cluster formation stage, each master node has a table (Table 2) which lists the connection information of all the slave nodes in the cluster.

Table 2. Connection Information of all slave nodes in the cluster

MAC Address	Piconet ID	Connection Handle	Role of the Slave
6 Bytes	3 Bits	12 Bits	2 bits

The parameters of this table are defined as follows:

- MAC Address: address of all of the slave nodes in the cluster.
- Piconet ID: When a slave node connects to the master, the master will provide it with a Piconet which is stored in the table. The first connecting device would be allotted a piconet ID of 1, the second one would be allotted a piconet ID of 2, and so on.
- Role of the Slave Node: The role of slave node is assigned by the master as 0: Normal slave; 1: Backup slave; and 2: Bridge slave. Bridge slave is deployed in connection with different piconets and holds the master's MAC addresses of these piconets.

After the process of service registration, every master node forms a service record for all slaves in the cluster, as illustrated in table 3. The bridge nodes acting as a relay between piconets also are supplied with this service record table, so the efficiency of inter-piconet communication can be significantly improved.

Table 3 Service Record for one slave node in the cluster [6]

Piconet ID	Number of Implemented Channels(*N*)	characteristics of Channel 1	...	characteristics of Channel *N*
3 Bits	1 Byte	4 Bytes	...	4 Bytes

- Each Channel Characteristics: it describes the main characteristics of each channel, as defined in table 4.

Table 4. Definition of Each Channel Characteristics

Characteristics of Channel [6]			
Channel No.	Transducer Priority	Group ID	Channel Type Key
1 Byte	1 Byte	1 Byte	1 Byte

In each cluster, the service record in the master node will also be backed up in the backup node. The master will then send updates to the Backup devices whenever a change occurs in the service record. This process is illustrated in Fig. 6. The numbers in the brackets shown in the figure denote the order in which the packet is transferred.

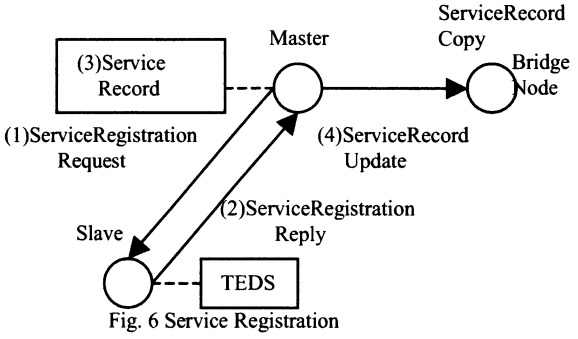


Fig. 6 Service Registration

In the WACNets, all the clusters with common or complimentary functions are considered as one group, in which all the nodes have at least the same group ID in the Channel_TEDSs, which can represent the whole group. Each group connects with one NCAP node.

In WACNets, after a cluster is formatted, the master node then sends the connected NCAP node a *ServiceRegistrationPermissionRequest* message to require a permission of the NCAP node to register the services of all the nodes in the cluster into it. If NCAP finds the information of this cluster in its service table, it will send a *ServiceRegistrationPermissionDeny* message back to the master. Otherwise, the NCAP node will reply with a *ServiceRegistrationRequest* message to the master. After the master has received this message, it will send a *ServiceRegistrationReplyforNCAP* to the master node. The whole process is illustrated in Fig. 7.

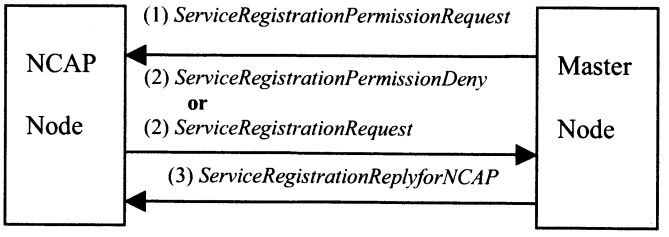


Fig.7 Service Registration between Master nodes and NCAP nodes

Each NCAP node has a service record, which lists the characteristics of all the clusters connected with this NCAP node. In the service record, each cluster is described as in the Table 5.

Table 5. Description of Each Cluster in NCAP [6]

Description of a Cluster		
Field #	Description	Field Length
General Description of Each Cluster		
1	Cluster No.	5 bits
2	The Number of the Slaves	3 bits
Service Abilities for the Cluster		
3	Channel Type Key 1	1 Byte
	Transducer Priority 1	1 Byte
	....	
	Channel Type Key N	1 Byte
	Transducer Priority N	1 Byte

B. Service Search

When a user looks for a node with particular service ability, the monitoring station will send a *ServiceQuest* message with Channel Type Key to the corresponding NCAP node, and the NCAP will check its service record table. If a match occurs, the NCAP node will send a *ServiceSearchRequest* message to the corresponding master node first searched. The master node will send a *ReadTEDSRequest* to the corresponding slave node. After receiving this message, the slave node replies with a *ReadTEDSReply* message to the master node. On receipt of this message, the master node sends a *ServiceSearchReply* message to NCAP nodes. This is eventually sent to the Monitor, via which, the user will get the required information. The process is shown in the fig. 8.

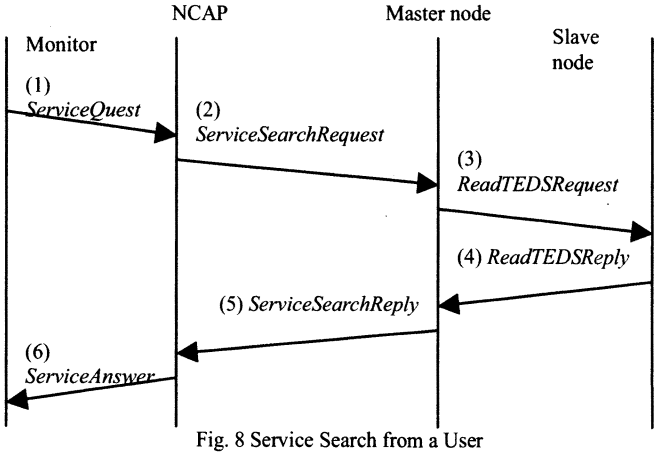


Fig. 8 Service Search from a User

To discover a node with particular service ability, the slave node will first send a *ServiceRequest* message to the corresponding master node, and the master node will check its service record table. If a match occurs, the master node will send a *ServiceReply* message back to the slave node. If the requested service is not available in the piconet, this *ServiceRequest* message will be forwarded to the bridge node in the same piconet to check whether the requested service is available in the neighbouring piconet (within the scatternet) or not.

In the proposed system, the bridge node can respond with information about services of all the piconets connected to it, without questioning the master. This would significantly improve the efficiency of the inter-piconet communication in the network.

6. TEST RESULTS

In the test-bed, there are two STIM nodes, one NCAP node and one PC deployed as the Monitor. The first experiment was undertaken for service registration using the test-bed, where two STIM nodes were set as master nodes in the experiment. After the process of service registration, the service table shown in Table 6 was formed in the NCAP node. This table lists the characteristics of these two STIM nodes.

Table 6 the Result of Service Record in NCAP Node [6]

Cluster No.	The Number of the Slaves	Channel Type Key	Transducer Priority
00001	000	00000001	00000010
00002	000	00000001	00000010

In the above table, the Channel Type Key 00000001 and Transducer Priority 00000010 mean that transducer attached to each STIM is a temperature sensor and real-time requirement is medium respectively. This table also shows that two master nodes without salve nodes have registered their services successfully. The corresponding information is also transmitted and shown on the applet in the monitor.

7. CONCLUSION

The concept of WACNets as the next stage in the development of distributed control networks was introduced. A review of service discovery protocols, as an important feature in self-organizing ad hoc networks was carried out. The result of validation of the service protocol was presented. The current test-bed consists of 3 nodes only. In the second generation of test-bed, a larger number of nodes will be included. The SDP will be further validated on the new test-bed which will represent more complex scenarios.

REFERENCES

[1] F. Mattern, "State of the Art and Future Trends in Distributed Systems and Ubiquitous Computing," *Vontobel TeKnoBase*, August 2000.

[2] http://www.cs.uno.edu/~golden/Stuff/37_richard_g.pdf.

[3] Specification for the Bluetooth System, http://www.bluetooth.com.

[4] Johnson, Robert N. "IEEE-1451.2 Update", http://www.sensorsmag.com/articles/0100/17/main.shtml

[5] IEEE 1451.2 standard, http://ieeexplore.ieee.org.ezproxy.uow.edu.au:2048/xpl/standards.jsp?findtitle=1451.2&letter=1451.2.

[6] S. Bu, F. Naghdy, "Wireless ad hoc control networks", *3rd IEEE Conference on Industrial Informatics*, August, 2005

[7] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *33rd Hawaii International Conference on System Sciences*, Vol. 08, No. 8, P.8020.