OPEN ACCESS
ISSN: 2249-6645

International
Journal
Of Modern Engineering Research (IJMER)

Adaptive Binding for Wireless Networking: Design and Implementation

N.L.Aravinda[1], G.Divya[2]
[1]Asst.Prof,ECE Dept, CMR Engineering College, JNTUH, India
Email: aravindanl@gmail.com
[2]Asst.Prof,ECE DEPT, CMR Engineering College, JNTUH, India
Email: dha.grs@gmail.com

ABSTRACT:- *while more and more wireless devices are phased in, binding relationship among them may become involved as wireless devices often operate in a self-organizing fashion. Though convenient, self-organization might fall short of user's expectation of flexibility sometime, leaving the user less room for binding adjustment. As a remedy, we propose a software- hardware codesign and implementation that enable the user to adapt network operations to certain needs anytime without foreknowledge of low-level details. Binding manipulation is done straightway with simple mouse clicking and dragging on icons on our graphical user interface. This study demonstrates a practical platform over which to inspire home automation or traffic management applications.*

I. INTRODUCTION

In the presence of a variety of wireless devices, a user may be foreign to manage wireless networks to customized needs. One major reason is that wireless nodes often operate in a form of automated binding, which is generally set upon startup of these devices themselves due to internal machinery. Although such self-organizing functionality benefits the user for bypassing involved device configurations, binding resolu- tion might fall short of user's expectation of flexibility in that the binding relationships, once established, leave the user less room for intervention and adjustment.

To see current binding issues, consider two computer hosts in the neighborhood, each with a same brand of wireless mice and keyboards. When powering up the hosts, a user can hardly predict which peripherals are connected to either host. A possible consequence is that both mice are bound to one host, while keyboards to the other. Such binding relationships will inconvenience the user unless these devices are reconfigured properly. Another common occurrence in using electrical appliances takes place when a wall switch is utilized to control a fixed set of ceiling lamps in a conference room. One can often find it difficult to keep suitable combinations of lamps on or off for groups of audience when making a presentation. These examples imply that an automatic or preset binding may not cater well for everyday needs.

This study covers a design and implementation of a free binding mechanism for wireless networking. Further, as far as electronics appliances are concerned, many commodities consume some power in standby mode even when turned off yet remaining plugged. Given a multitude of such devices, the accumulated standby power consumption over long hours is becoming significant. For this, we incorporate our binding mechanism in a set of reconfigurable wireless outlets that prevent plugged-in appliances originally left on standby from wasting energy by total power cut. In our architecture the wireless outlet is still plugged into the normal electricity power supply system of civil utilities but can be governed by a wireless switch whether to block the current sourced from power plants. The remainder of this paper is organized as follows. The next section gives a brief background on this study. Section 3 elaborates on our developed platform, including system architecture and circuit layouts. Example applications are given in Section 4. Lastly Section 5 concludes this work.

II. BACKGROUND

This work covers an IEEE 802.15.4-based protocol suite named JenNet implementing ZigBee circuit design, and prototyping. On top of the protocol suite are various applications. Beneath applications is the JenNet stack level, namely the network layer that handles network addressing and routing by invoking IEEE 802.15.4 MAC (Medium Access Control) operations, whose tasks include: 1) starting the network,2) adding/removing devices to/from the network, 3) routing messages to destinations, and 4) enforcing secure message transfer. The bottom level comprises IEEE 802.15.4 protocols, a low-cost and low-power solution.

2.1. Topology and Routing

We consider a tree topology of wireless nodes as illustrated in Fig. 1(a). The topology is rooted at the coordinator, the most capable device which may bridge to other networks. The coordinator serves to select the frequency channel in use by the network, start the network, allow nodes to join the network, and provide message routing. In tree topology each node communicates only with its parent and children (if any.)

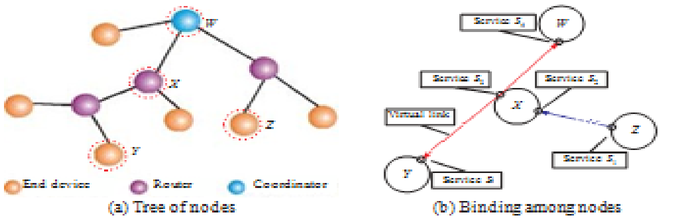


Fig. 1. Tree topology versus service binding. (a) A message is routed from the source node up the tree to the nearest common ancestor and then down the tree to the destination node. (b) Binding associates a service on one node with a service on another node.

Any parent node acts as a local coordinator for its children. Note that a router is an intermediary to relay messages from one node to another.

As for routing, a message is first passed from the sending node (e.g., an end device) to its parent. If addressed to a child of this parent, the message is sent directly to the destination node. Otherwise, the message is passed up the tree to the next parent, which then decides whether the message should be passed down to one of its children or up to its own parent. Message propagation continues in this fashion by routers and the coordinator maintaining route information.

2.2. Service Binding

A *service* indicates a high-level feature or function of the node whereby certain data can be exchanged among nodes. A service allows a node to determine with which other nodes to communicate meaningfully. For instance, a heating control node may be interested in nodes with a temperature sensor (one service) or a switch (another service). In this regard, binding is exercised to establish virtual links among a service on a node and one or more services on other nodes. Fig. 1(b) shows example bindings between services on different nodes W , X , Y , and Z . Binding establishment requires a node to discover services available on which nodes. Bindings are stored in the binding table on each source node. An entry of the binding table relates a pair of endpoints. One endpoint indicates the source node and service from which data is produced.The other identifies the remote node and service to receive the data. Binding relationships among nodes may remain even though the network has altered physical connectivities.

2.3. Previous Work

While dynamic binding in object oriented programming refers to determining which method to invoke at runtime, there is as yet little literature addressing dynamic binding in IEEE 802.15.4.These previous work, however, tackled the problem with programming paradigms. In comparison, this study is concerned with more than programming but also hardware implementation, so as to reflect a software-hardware codesign. Our adaptive binding mechanism is applicable to residential energy management. This is achieved by our software-hardware codesign. Current product work fine with bindings, i.e., a remote control is designated to govern a fixed set of outlets or sockets for powering appliances. In contrast, our implementation allows of flexible binding adjustment. Further, an elaborate design was realized in our wireless outlets that sense the load current of each socket, learning whether the plugged-in appliance is left on standby. If so, the socket is shut off electricity provision. In comparison, our wireless outlet is bi-directional, capable of reporting electricity usage statistics to a console periodically for smart metering. Besides, in our architecture the threshold standby power consumption is learned in a self-reliant manner, as opposed to being done semi-automatically.

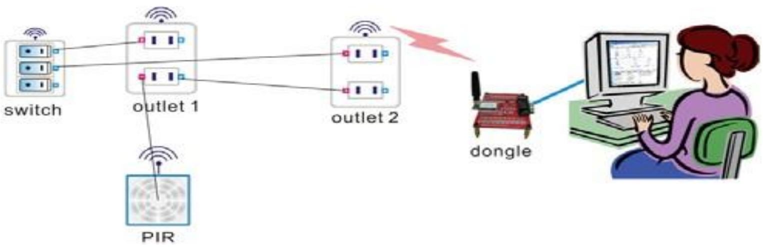
III. A USER-FRIENDLY PLATFORM

3.1. Software Consideration

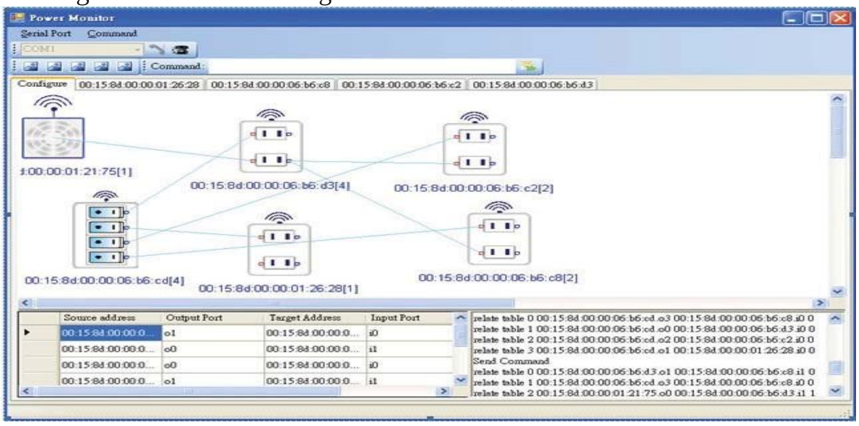
Fig.2 (a) depicts how bindings are reconfigured in our architecture. All the managed devices within radio range are symbolized as separate icons on a GUI console. The console, an own-developed software system, displays each such device in an automated manner. Bindings are manipulated with simple mouse clicking and dragging on these icons. Fig. 2(b) shows an example scenario, where each line connecting two icons signifies a binding between two physical devices. A binding relation is created by selecting a point with a mouse click on the chosen icon and drawing a line to another point by dragging the mouse cursor. Conversely, an existing binding can be removed straightway by mouse operations as well.

The soft console is paired with a dongle (implementation of the coordinator) to detect wireless nodes in range. The dongle reports detected results to the console for proper icon display. The console retains system-wide knowledge of bindings among all the nodes. The console is also charged with sending binding information via the dongle to the intended devices. Each receiving site will keep such information in storage and then act as programmed accordingly. In the reverse direction, the dongle is meanwhile tasked to collect real time statistics of interest like electricity usage from each of wireless devices. The collected information is interpreted and graphed on the console along with time.

1) Binding Maintenance: Network formation is carried out by JenNet itself, forming a tree topology. Provided that a node N joins the network, the neighbor table and the routing table on its parent node change accordingly. Changes of the



(a) Binding reconfiguration is done through a soft console.



(b) A snapshot of our soft console

Fig. 2. Binding relationships can be reset at user's discretion through a GUI console. An icon on the console represents a specific device shown in Fig. 5

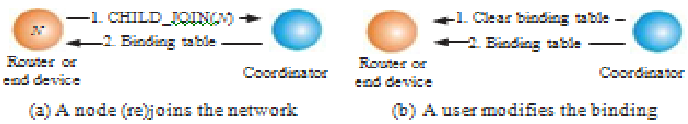


Fig.3. Two cases of binding table updates.

tables will, in turn, be propagated hop by hop toward the coordinator, causing nodes on the path to update routing tables correspondingly. On detecting the new node (upon receipt of a CHILD_JOIN(N) message), the coordinator looks up N 's MAC address in its binding table locally. If N cannot be found, a new icon for N pops out on the soft console. If N is found, however, N is viewed to rejoin the network due to recovery from intermittent link outage. Then the coordinator additionally sends binding entries about N to the node, as shown in Fig. 3(a). This causes updates to N's binding table for bookkeeping the most recent bindings that might have altered during N 's leave.A user is also able to resettle virtual links between devices over the console. In the event of binding changes, the console renders user defined new relations into updates to binding table entries and lets the dongle disseminate such updates to involved nodes. This is accomplished by the console instructing each of the involved nodes to clear and reset its binding table to the received contents, as outlined in Fig. 3(b). In our design, the binding table was purged before being renewed, so as to avoid update complexity.

2) Data Transfer: Data is sent from one endpoint to another, as per the binding table. Every entry of the binding table relates a pair of endpoints or data handling. Taking Fig. 2(b) as an example, if an endpoint P on a wireless outlet is bound to other two, specifying P as the source will automatically assume the destination

endpoints on the relevant sites. Data transfer in our architecture proceeds by exploiting JenNet functionality. That is, when data is generated on an endpoint, the source node or the first-hop router consults the binding table to see for which node(s) this data is destined. The data is then packed into a JenNet message for trans- mission or multiple messages, each for an intended endpoint, for data duplication. Subsequently the first-hop router checks whether the destination node of the message exists in its neighbor table. If so, the message is sent directly. If not, the message is passed up the tree to the next parent according to JenNet routing machinery.

3.2. Hardware Consideration

We have dynamic binding notions incorporated in hardware modules. Let us target an environment where appliances are on or off under power control per binding requirements. This subsection will first describe the basic communication module, core to other modules.

1) Basic Communication Module: The basic communication module is designated to communicate over the air for message exchanges and over an RS-232 interface for issuing I/O commands. As shown in part of Fig. 4 and Fig. 5(a), the module consists of a JN5139 chip as a building block and up to 20 basic I/O blocks, with the UART (universal asynchronous receiver/transmitter) pin used in conjunction with a standard RS-232 port. The RS-232 port is connected to an external device for debugging that can bridge to other network entities. Inside a basic I/O block, the digital I/O pin DIOn can be wired to either a touch switch or LED indicator and to DIOn on JN5139, n = 0, 1, 2, ••• , 20 [4].

2) AC Control and Sensor Unit: This unit is to sense the voltage signal level of alternating current (AC) and realize a wireless controlled node (slave device) under service binding. Fig. 4 shows this unit of four components:

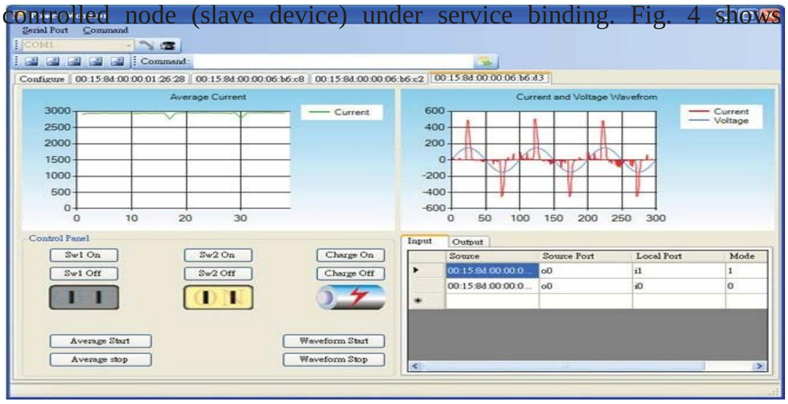


Fig. 4. Block diagram of our AC control and sensor unit.

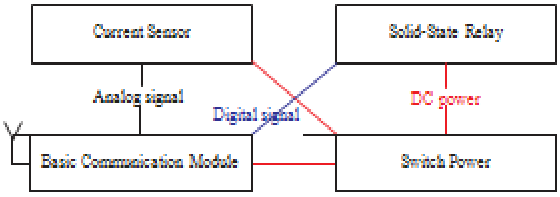


Fig. 5. Our prototype adopts the Jennic JN5139 microcontroller and operates in the unlicensed 2.4 GHz band

- Switch power: Powering other three components (see below);
- Current sensor: Sensing currents through the AC socket, converting the sensed current to voltage signals (with a Hall component [6]), and sending the converted data to the basic communication module (recall Section III-B1);
- Solid-state relay (SSR): Keeping the power at the AC socket on or off;
- Basic communication module: Receiving instructions from a remote site, quantifying analog signals from the current sensor (with the JN5139-embedded analog-to- digital converter), driving the SSR for power control, and sending the quantified measurements to the coordinator over the wireless medium.

Apart from above, our AC control and sensor unit distinguishes noise signals from normal load current that powers a working device. By doing so, we learn that a device plugged in which socket is operating normally or tell which socket is not defacto active for a dormant device. This is of utility to determine which sockets can shut off power supply to the plugged-in apparatus for energy-saving purpose.

3)Prototype: Resulting from several stages of circuitry soldering tryouts, Fig. 5 shows our hardware implementation—a basic communication module, wireless outlet, wireless switch, and wireless PIR (passive

infrared) module. In particular, the basic communication module functions as the dongle and a constituent block of wireless outlets and switches. Our wireless outlet mounts two sockets for powering home appliances. A rechargeable battery inside is fitted out to power the circuitry on the wireless outlet in a self-reliant manner.
The wireless PIR module is to detect whether a user or moving object has appeared in vicinity. If so, the module can be programmed to issue commands over the wireless medium to its bound outlets to restore power to the plugged- in appliances thereon. It is apropos, for example, to install the wireless PIR module at the entrance of a residence that instructs its bound outlets to power up electric appliances ahead just before the house owner arrives home.

Fig. 6. The console operates not only for monitoring electricity usage dynamics but also as soft switches to control wireless outlets.

IV. APPLICATION SCENARIOS

This section exemplifies use cases of our developed plat- form. The first is pertinent to binding management, while the second features traffic management.

4.1 Free Binding

In most cases, a switch is preset to control a fixed set of electrical apparatus, for example, ceiling lamps and so forth. It is often prohibitive to tailor the binding relationships to such various needs as allowing for different combinations of appliances based on the time of day, especially when electric wires running inside the building require re-disposing. In a broader sense, there have been demands for a main device collaborating with other equipments. For instance, one computer host with a group of peripherals and a TV set with a home cinema system have similar concerns that binding relationships might vary because accessories are phased in and out over time. Our platform makes use of wireless technology in lieu of physical cabling. The platform serves a user to adapt bindings to customized needs whenever necessary, as outlined in Fig. 2(b). The soft console of our platform allows the user to reconfigure bindings with simple mouse operations and then orders the dongle to notify the involved devices of new settings. In that figure, a switch is made trigger multiple wireless outlets that power appliances, revealing an 1-to-many relationship. In the meantime, more than one switch can be bound to a single outlet (many-to-1 relationship.)
Besides, Fig. 6 gives an example screenshot, showing that our console enables the user to switch on or off a selected socket, through soft buttons on the Control Panel, to activate or inactivate its power provision. Resulting from a mouse click on some MAC address tab on the console, the screenshot exhibits quadrantal areas with respect to the selected device. The Average Current subarea indicates the mean load (power output) on the chosen device for a long run. The Current and Voltage Waveform subarea plots momentary current and voltage readings. The Binding Status subarea tabulates from and to which endpoint the device is bound. The Control Panel contains 5 pairs of buttons for starting or stopping a socket, the built-in rechargeable battery, showing average currents, and displaying current and voltage waveforms, respectively. Therefore, our console can also be employed as a monitor for remote surveillance.

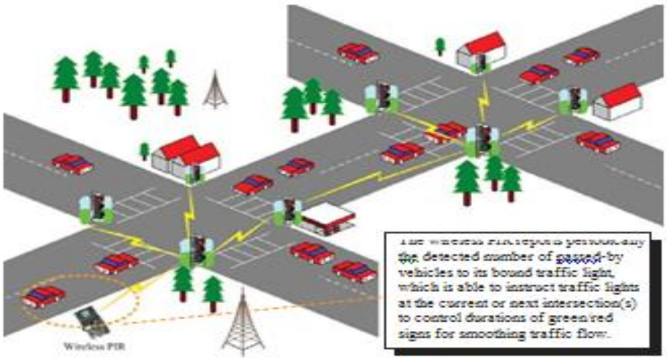


Fig. 7.With assistance from the wireless PIR, traffic lights at successive intersections are bound collectively to resolve green/red light timing lengths

4.2 Online Traffic Light Control

Next consider an urban area as shown in Fig. 7. Supposing that traffic lights are outfitted with our basic communication modules onboard (Fig. 5), such as to operate as instructed over the air to control the duration of green or red lights. Green lights can be kept long enough, allowing most vehicles to pass through intersections. In addition to resolving the duration of green/red lights in line with traffic dynamics, a group of traffic lights at successive intersections may need to work in chorus someway. Vehicle density per length of road will determine how many successive intersections to be grouped and thereby how binding among traffic lights to be formed. Higher vehicle density means a higher likelihood of congestion, implying the need of involving a smaller group of traffic lights because vehicle speed is probably slow. In this situation, it does not make significant difference to tightly synchronize farther traffic lights, say at next two intersections. As roadway conditions are likely to change over time, we can achieve intelligent traffic light control by tailoring binding relationships among the group of traffic lights whenever necessary. When the group grows or shrinks in reaction to traffic conditions, our mechanism applies. For this, our wireless PIR can be deployed at roadside before an intersection entry zone. The PIR reports the detected number of passed by vehicles to its bound traffic light, which then exchanges such statistics with other traffic lights locally and deduces proper green and red light timing lengths. Timing lengths for mainline traffic flows may have slightly higher priority over traffic in the perpendicular direction. The deduced timing lengths are sent to other traffic lights of the same group for subsequent reference. Note that the group tends to change based on the time of day. As reasoned, our developed mechanism gets fielded where bindings among devices require modification in traffic management scenarios.

V. CONCLUSION

As wireless technology simplifies the deployment of un- tethered devices, such strengths are well suited to applications in various contexts. Provided a plurality of wireless devices, however, a user might encounter difficulty in inter-relating them for better performance or adaptiveness. This necessitates binding management. This study came up with a user-friendly platform for wireless applications in terms of managing off-the-shelf appliances. Consumer appliances do not need to change machinery. The interface was approached by a software-hardware codesign and implementation.

This study developed a platform in support of efficient interactions between users and their environs. On our plat- form, bindings among devices were manipulated through a GUI. Our development has leveraged current protocol fabrics, ensuring backward and forward compatibility with legacy technology. We brought in a value-added service to the wire- less network, such as to make it better manageable. We also demonstrated application that prevented dormant appliances from wasting energy out of proper bindings. We practiced an object-oriented modular methodology to provide an integrated software with hardware solution to iconify geographically dispersed network entities of concern. Our development lends itself to home automation and traffic management as well.

REFERENCES

[1] M.Erol-Kantarci, and H. T. Mouftah, “Wireless Sensor Networks for Cost Efficient Residential Energy Management in the Smart Grid”, *IEEE Trans. Smart Grid*, Vol. 2, No. 2, June 2011, pp. 314–325.

[2] J. Han, H. Lee, and K.-R. Park, “Remote-Controllable and Energy- Saving Room Architecture based on ZigBee Communication” *IEEE Trans. Consumer Electronics*, Vol. 55, No. 1, Feb. 2009, pp. 264–268.

[3] IEEE Std 802.15.4, “Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)”, Institute of Electrical and Electronics Engineers, Sep. 2006.

[4] Jennic, Data Sheet: JN5139-001 and JN5139-Z01, JN-DS-JN5139, v1.6, 2008.
[5] Jennic, JenNet Stack User Guide, JN-UG-3041, Revision 2.0, Sep. 2010. [6] R. S. Popovic´, Hall Effect Devices (2nd Ed.), CRC Press, 2004.
[6] G. Song, F. Ding, W. Zhang, and A. Song, "A Wireless Power Outlet System for Smart Homes", *IEEE Trans. Consumer Electronics*, Vol. 54, No. 4, Nov. 2008, pp. 1688–1692.