© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

MiND: Mind Networked Device Architecture for Attention-Gated Ambient Assisted Living Systems

Anandarup Mukherjee, *Student Member, IEEE,* Sudip Misra, *Senior Member, IEEE,* and Abhay Atrish

***Abstract*—This paper proposes and evaluates an online architecture for networked Brain-Computer Interfaced (BCI) smart home systems for enabling Ambient Assisted Living (AAL). Our work uses multiple portable, low cost, single channel EEG (Electroencephalograph) systems to achieve mind networked devices (MiND) for home appliance control aiming to enable better accessibility for the infirm and people with mobility disability. We tested our approach on real-life hardware towards the control of devices such as lights, fans, and room temperature conditioning systems and emulated the behavior of the networked hardware for a more substantial number of users on conceptualized scenarios, which will be encountered in implementing this architecture. The NS2 simulation results of the large-scale implementation show promising results and give a precise idea about various essential features such as routing protocols, routing configuration, mode of transmission, attenuation losses, and delays incurred during in-house operation of this architecture.**

***Index Terms*—Ambient Assisted Living systems, Brain Computer Interface, Smart Homes, Bluetooth, NS2.**

I. Introduction

THIS This paper envisions and implements an AAL enabling system and its architecture using low cost, over-the-counter, and hassle-free BCI devices. The BCI device used in this work has a single electrode, which is placed on the forehead to acquire single channel EEG signals. The elderly and people with mobility disabilities often have to rely on others for tasks such as switching on/off lights, fans and television sets. The proposed system and architecture enables the natural control of household appliances and systems without actually being in the line of sight of the target devices themselves.

The architecture uses commercially available, low-cost, single-channel EEG device(s) [1] acting as the BCI interface to the system. This BCI device interfaces to an Electronic Control Terminal (ECT), which is primarily an embedded systems-based processor, utilizing Bluetooth. This arrangement enables the BCI device to act upon actuators controlled by the **E**lectronic **C**ontrol **T**erminal (ECT). The ECT is further distinguishable as ECT-1 or ECT-M, with their respective SELECT-1 and SELECT-2 algorithms. These algorithms help the users in selecting appropriate devices to actuate. SELECT is an acronym for **SELE**ct from **C**ontrol **T**erminal; 1 or M denotes the number of EEG headset users, which can simultaneously connect to the ECT. Another algorithm, the SILO (**SI**gnal **LO**ck), ensures the signal lock of the headset and the ECT module under various conditions and situations. At the system level, each of these ECTs controls a series of relay switches, which are electronically turned on or off based on inputs from a low-power embedded wireless Bluetooth enabled EEG sensor unit (the headset). This method allows the users to control various household appliances – lights, fans, television, and air-conditioning. The user can switch on the state of the connected devices by adjusting and increasing their attention level above an adjustable threshold, which is also termed as attention gating. Blinking lights are provided to act as a visual indicator of the current status of devices to the users. The various components of the proposed model are shown in Fig. 1.

Anandarup Mukherjee and Sudip Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India e-mail: (anandarupmukherjee@ieee.org).

Abhay Atrish is with the Department of Computer Sc. and Engineering, National Institute of Technology, Uttarakhand, India.

A. Use Case: Home Appliance Control

Jane and Jack are in their eighties. Jack has arthritis, which makes regular movements painful for him. Jane's movements are also restricted due to her weak and deteriorating eyesight. Jane needs to switch off the air conditioning of their apartment but is disinclined to move because her weak eyesight does not allow her to see or perceive her surroundings too well in the semi-darkness. Suddenly she realizes that she does not have to move to achieve this trivial task, which seems herculean to her at that time. She puts on her BCI headset and powers it on. Once the ECT near her bed detects her headset, it automatically connects to it and starts polling the relays controlling the actuator devices, glowing a single Light Emitting Diode (LED) for some predefined seconds, corresponding to the selected device. Jane increases her attention level above the set threshold when the LED corresponding to the room's temperature controller adjusts. The ECT stops polling and checks if Jane can maintain her average attention level above the threshold for a few seconds. It detects consistent above-threshold attention level from Jane, confirming that she wants the air conditioning to be turned off and consequently does so. She has to repeat the same process to turn the device back on. In the event, Jack wants to turn on the television while Jane is still in the process of negotiating with the ECT for the air conditioning, he too can connect to the system, and in a manner similar to Jane's, he too can select the television and turn it on, without moving from his place. The ECT-M detects two simultaneous headset transmissions in the room, switches on the additional Bluetooth receiver and performs in the same manner as it would have in case of a single user.

B. Motivation

BCI is one of the most challenging and promising domains for enabling AAL, and is being used for controlling home

for personal use only

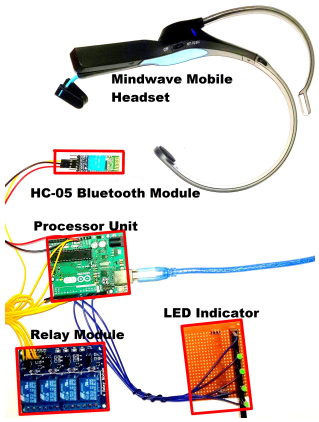


Figure 1. The lab prototype of our ECT-1 system showing the connected as well as individual modules for achieving the ECT-1 architecture.

environments, among many other applications [2]. The training of the subjects and the complexity of the system can also be considered as a drawback of these systems. Work on home appliance control is mainly in a virtual environment [3]. None of these works addresses the problem of networked BCI interfaces working simultaneously on a fixed number of actuating devices such as – sudden contradictory commands to the same ECT due to multiple users and interference in Headset-ECT radio connectivity due to neighboring radios, to name a few.

II. Related Work

Activity detection tasks such as fall detection, during the care of infirm and physically disabled, is yet another popular domain of AAL. Various types of falls and their signal signatures are identified in work by Bruno *et al.* [4]. The use of a smartphone-centric platform for assisting and monitoring subjects with physical limitations is demonstrated by Bisio *et al.* [5]. Points of interest (POI) based context-aware method of assisted navigation for the cognitively impaired is demonstrated by Hervas *et al.* [6]. These context-based systems increase efficacy, acceptance, adaptability, and overall performance of human-system interaction applications. Similarly, intention recognition systems incorporated into an intelligent agent-based architecture provide sensor-filtering rules and evaluate complex activities for enabling assistive living in smart homes [2].

Controlling the state of the immediate environment using network actuated devices, guided by environmentally-aware sensors is demonstrated by Broy and Schmidt [7]. Similar objectives have been undertaken by Lin *et al.* [8]. Works such as evaluation of distributed signal processing approaches for Wireless EEG Sensor Networks (WESN) by Bertrand [9] demonstrate the need and usefulness of networked BCI approaches in AAL. Works on similar lines include those by Faller *et al.* [10], who propose an approach for auto-calibrating smart home device control systems and internet access by using a hybrid BCI, and McMullen *et al.* [11], who demonstrate a hybrid approach to BMI by making use of both EEG and eye tracking, which is adapted for controlling a robotic prosthetic limb. Additional methods for controlling devices, such as wheelchair and household appliances using human thoughts are described by Kawanabe [12]. Additional approaches for BCI-based device control are described in works by Waytowich and Krusienski [13], and others [14], [15].

Synthesis: The works pursued on BCI for AAL do not take into a count, the need for making the access control of their systems inclusive to accommodate a broader section of the society including those with mobility and speech disabilities. Besides addressing this challenge, the algorithms for the proposed system are capable of including multiple BCI units in a networked fashion to work upon the same set of physically actuable target devices.

III. System Setup & Parameter Analysis

A. The EEG Headset

We use a readily available single-channel and hassle-free EEG headset for acquiring the crucial signals from the users, which is then integrated to our ECT system. The commercially available device is used in the transmission of single-channel EEG information on the human brain for thought-based applications. It has a sensor touching the forehead, and contact and reference points located on the ear pad. An onboard chip processes all of the data and sends this data to the interfaced software and applications as a bit-stream. It calculates the raw brainwave values $(\alpha, \beta, \gamma, \delta)$ as well as the derived values of Attention and Meditation. Attention signifies the act of intense concentration or mental focus, whereas Meditation is associated with calmness or mental relaxation. Parameters such as *signal quality*, *EEG power* and *mind wandering levels* can be extracted from the device. The packet structure is shown in Fig. 2. The transmitted packets contain a header, payload, and a checksum part. Data is transmitted serially

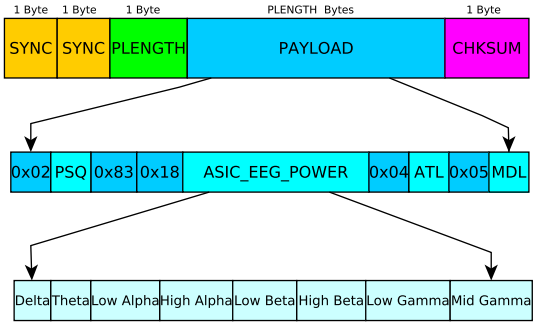


Figure 2. The transmitted packet structure denoting the relative byte position of each component of the recorded EEG signal.

through Bluetooth. The header marks the start of a new packet, and the Checksum indicates its end. Packet lengths vary from a few bytes to very large ones. Standard valid packets, also called "Big packets", lie between $32 - 173$ bytes of size. Depacketization starts with checking the condition for the two synchronization bytes while marking the arrival of a new data packet. The Data Payload follows next. The size of the packet

for personal use only

is contained in the PLENGTH header byte. It is transmitted as key-value pairs. Each pair first contains the code (called Extended CODE or EXCODE) for the respective data field, followed by the data fields, which may be of single or multiple bytes. The EXCODEs for *Poor Signal Quality*, *Attention* and *Meditation* are 0×02, 0×04 and 0×05, respectively. The calculated checksum must match the data checksum byte of the packet indicating strong data integrity.

B. System Functions and Controls

The proposed system can be visualized as one having M-inputs and 1-output. Multiple headsets trying to connect to the same output MCU, in order to control the devices connected to it are shown in Fig. 3 and has the following parameters,

Input Parameters: Accuracy of the device for sending *Big Packets* (A_i), *Payload Data* (D_i), *Time of sending* (T_i) and *Packet Integrity* (I_i)

Checking Parameters: *Signal Strength* (S_i), and *Battery Power of Headset* (B_i)

Output Parameters: *Actuation Command* (Y_i)

Mathematically, we represent the model's input parameters as $H_i = f(A_i, D_i, I_i)$. The error checking term is represented by:

$$E_i = f(S_i, B_i) = \begin{cases} 1 & S_i, B_i > (S, B)_{Thrs} \\ 0 & S_i, B_i \leq (S, B)_{Thrs} \end{cases} \quad (1)$$

The processing stage is represented by $U_i = f(T_i, A_i, D_i)$ and $G_i = f(I_i)$. Finally, the output is represented as $Y_i = (U_i \times e_i) \times G_i$, which can be represented as,

$$Y_i = \begin{cases} f(T_i, A_i, D_i) & E_i \times I_i = 1 \\ 0 & E_i \times I_i = 0 \end{cases} \quad (2)$$

The input from the headset, H_i is fed to the processor for analysis. The additional check parameter E_i consists of S_i and B_i (Equation 1). It gives an output of 0 or 1 depending on the state of these signals corresponding to their minimum threshold values. If the signal quality is permissible, the processor analyzes the packets. Otherwise, the packets are discarded. Similar to E_i, a packet integrity check is also used. It is denoted by I_i and gives output as either 0 or 1, depending on the goodness of the packet. If the processed signals pass the integrity check, only then it is allowed to actuate the devices as denoted by Equation 2.

C. The Single User Electronic Control Terminal (ECT-1) Setup

The hardware implementation of ECT-1 is done using an Arduino UNO MCU board, an HC-05 Bluetooth module, a Neurosky Mindwave Mobile headset, a ULN2003 relay driver, along with LEDs (as device indicators). The HC-05 module is configured to a baud-rate of $57,600$ to match the headset's standard baud-rate and hard-paired to the headset using AT commands. This configuration enables the headset to connect to ECT-1, whenever it is in range automatically. The Bluetooth module is interfaced to the MCU for standard serial communication, and outputs are interfaced to a ULN2003 relay-driver, which drives four household devices. The device indicator is made using an array of LEDs. Upon successful connection, the status LED on the Bluetooth module starts to blink at a slower rate, and the headset begins transmission of data. The packets received over the Bluetooth channel contain information such as *Poor Signal Quality*, *Attention* and *Meditation* e-sense values, amongst others. We de-packetize the incoming packets using the MCU, which is configured to extract just the attention levels of the user from the whole packet. Each of the four devices is polled, one at a time for three sets of readings, after which the next device is polled. LEDs indicate the device being polled. For each poll, three e-sense values are taken per second and averaged. Finally, the averaged reading is compared to a threshold of 70%. If the average reading is found to be higher than the threshold, the device being polled undergoes a state transition (OFF to ON or vice-versa). If the calculated value is less than the threshold, then no change in state occurs, and the next device is polled, and the process continues. The SELECT-1 and the SILO algorithms handle the connection requests and decision-making processes regarding device actuation.

D. The Multiple User Electronic Control Terminal (ECT-M) Setup

Similar to ECT-1, the hardware implementation of the ECT-M is done using an Arduino UNO MCU board, 4 HC-05 Bluetooth modules, 4 Neurosky Mindwave Mobile headsets, a ULN2003 relay-driver along with LEDs (as indicators). The Bluetooth modules are configured as before to match the headset's standard baud-rate and hard-paired to all the nearby headsets so that any of the paired headsets can connect with any of the modules, whenever they are in range. All four Bluetooth modules are interfaced to a single MCU board for standard serial communication. The output side is identical to the circuit used in the ECT-1, as previously described. Whenever a headset comes in the range of the awake Bluetooth module, automatic pairing occurs, and communication between them starts. In every iteration of the Arduino program, all four Bluetooth modules are scanned sequentially for incoming data. The data payloads are stored in designated memory locations for the different ports, and each of the extracted payload values are calculated and executed using the SELECT-M algorithm for the actuating the devices. Multiple Bluetooth modules are activated at designated intervals of time to check for multiple incoming connections. If multiple headsets connect to the MCU at the same time, the SELECT-M algorithm handles the requests accordingly.

E. Packet Capture and Analysis

The data transmission from the headset to ECT is intercepted and analyzed using Wireshark packet analyzer, to get the packet details and study which protocols are involved in real-time. Approximately, $55,000$ packets are captured for analysis. As not all parts of the Bluetooth protocol stack are used for serial communication, only selected portions of the Bluetooth stack take part in the data exchange process. The controller stack protocols of the Bluetooth stack are first

for personal use only

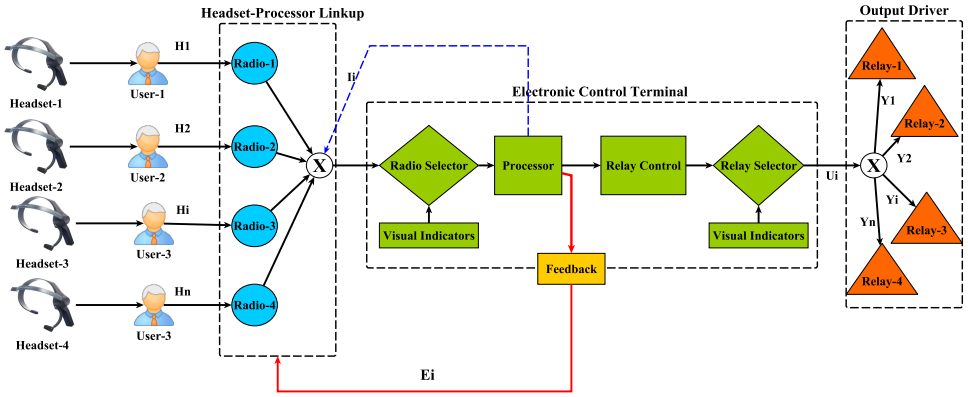


Figure 3. The Control model of ECT architecture, denoting multiple users in a room connected to an ECT unit. The ECT unit controls several relays, which control the state of the room's electrical appliances.

involved in the set-up of communication, primarily based on the $HCI - H4$ controller interface. An HCI command is issued using the "Create Connection" command. It is followed by the $HCI - Event$ packets for acknowledgment of the request, initiation of supported features, and channel creation. In the host stack, $L2CAP$ is invoked for data transfer and "Connection Request", "Connection-Response", "Configure-Request", "Information-Response" and a periodic acknowledgement of number of packets. The HCI commands to read the "RSSI Information", "Link Quality" and "Transmit Power Level" are also issued before the initiation of data transfer. $Service - Search - Attribute$ requests and responses are handled by SDP. Periodic checks are made to ensure successful channel set-up before any information is transferred. $RFCOMM$ is involved in setting up the virtual COM ports for serial data transfer.

Algorithm 1 The **SILO** Algorithm

Inputs: $UserList[t]$, $SignalStrength[t]$, $BatteryLevel[t]$, $PktContent[]$
Output: $Pkt[a_1, a_2, a_3, ...a_n]$
Parameters: $MWSigQuality$=$PktContent[a_i]$, $Attn = PktContent[a_j]$, $Thresh \leftarrow 70$
if $(SigStrength(t+1) < SigStrength(t))$ **then**
  if $(BatteryLevel(t+1) < BatteryLevel(t))$ **then**
    while $(Attn > Thresh)$ **do**
      Maintain Connection
    Start Polling

Finally, SPP begins data transfer after all the connection parameters are successfully set-up. Each SPP packet is encapsulated within an $RFCOMM$ packet which, in turn, is encapsulated in a $L2CAP$ packet. Along with data transfer, the HCI commands in the UIH frames and the $HCI-Event$ packets exchange acknowledgment information once a data packet is transmitted. "RSSI Information", "Link Quality" and "Transmit Power Level" are also constantly kept track of to ensure least data loss probability. The SPP packets primarily contain the data payloads – mostly $93 - 94$ bytes long. They contain the data transmitted by the headset. It is received as a stream of information transmitted serially and is read by the

Algorithm 2 The **SELECT-1** Algorithm

Inputs: $UserList[1, 2, ..n]$, $DevIndex[1, 2, ..k]$, $PktContent[a_1, a_2, ..a_i]$, $Attn$
▷ %UserList[] contains the indexed addresses of all paired headsets in range, whose signal strength are sorted in descending order%
▷ %DevIndex[] contains the indexed locations of all actuating device controllers (relays) on the ECT%
▷ %PktContent[] contains the indexed locations of all eSense values%
Output: $DevState$
Initialize Parameters:
  $Thresh \leftarrow 70$
  $DevState \leftarrow 0$
  $T_{tick} \leftarrow 5$
  $Attn \leftarrow PktContent[a_i]$
  $PktIntegrity \leftarrow PktContent[a_i + n]$
while $(SILO)$ **do**
  while $(PktIntegrity)$ **do**
    $DevIndex \leftarrow 1$
    while $(Attn > Thresh$ & $DevIndex[i] < K)$ **do**
      for $(T = 0; T < T_{tick}; T++)$ **do**
        ▷ %T: timer from 0 to T_{tick} seconds%
        if $(Attn > Thresh)$ **then**
          $!(DevState)$
          ▷ %invert device state%
          Goto 12
      $DevIndex++$
Return $DevState$

Arduino Serial ports. The MCU does the task of depacketizing the stream of serial data to filter only the packets containing the data of interest – viz. the ones containing the "Attention" e-sense. Their packet length headers identify them. Usually packets of length 32 bytes or more (termed as "Big Packets") contain the required fields. They are read by the Arduino MCU, and the required data fields are extracted from them for further processing on retrieving the actuation parameters. Fig. 2 shows the structure of the captured packet.

F. Technology Enabling Algorithms

The methods in Algorithms 1, 2 and 3 are implemented in this architecture. SILO (Algorithm 1) scans for available headsets in range, checks their signal parameters, and connects

for personal use only

to the most promising one(s). If certain extracted parameters exceed a preset threshold, a connection is established between the headset and the ECT unit. When connected, SELECT (Algorithms 2 and 3) checks the packet integrity and the status of the relays connected to the ECT unit. Parallel to this procedure, attention levels from the headset packets are extracted and analyzed to check whether the user's attention level exceeds a certain threshold for a predefined interval of time. In case it does, the ECT unit toggles the state of the relay connected to it, which in turn, switches on or off, the device connected to it. The SELECT-M algorithm deals with the selection of users, and the selection of devices to actuate based on visual indicators and timers as discussed in Section I of this paper.

Algorithm 3 The **SELECT-M** Algorithm

Inputs: $UserList[1, 2, ..n]$, $DevIndex[1, 2, ..k]$, $PktContent[a_1, a_2, ..a_i]$, $PortList[1, 2, ..M]$, $M \leftarrow$ No. of Radio Receivers on Hardware, $k \leftarrow$ No. of relay switches on Hardware
▷ %PortList[] contains the COM port locations of all Bluetooth modules, connected to the ECT%
Output: $DevState[DevIndex[i]...DevIndex[k]]$
Initialize Parameters:
    $Thresh \leftarrow 70$
    $DevState[DevIndex[1]..DevIndex[k]] \leftarrow 0$
    $T_{tick} \leftarrow 5$
    $Attn[] \leftarrow PktContent[a_i]$
    $AttnM[] \leftarrow [Attn[1]..Attn[M]]$
    $PktIntegrity[] \leftarrow PktContent[a_{1+k}, ..a_{i+k}]$
while $(SILO)$ **do**
    while $(PktIntegrity[1]toPktIntegrity[M])$ **do**
        $[DevIndex[1]toDevIndex[M]] \leftarrow 1$
        while $(AttnM[1]toAttnM[M] > Thresh)$ **do**
            $i \leftarrow 1$
            for $(T = 0; T < T_{tick}; T++)$ **do**
                ▷ %T: timer from 0 to T_{tick} seconds%
                if $(Attn[i] > Thresh)$ **then**
                    $!(DevState[DevIndex[i]])$
                    ▷ %invert device state%
                    Goto 19
            $i++$
        $DevIndex++$
Return $DevState[DevIndex[1]..DevIndex[M]]$

IV. Simulation

The data from ECT-1 hardware is captured to analyze and extract information regarding packet transmission statistics. Implementing multiple ECT-1 and/or ECT-M architectures to analyze the performance of these devices and their effects on one another, is not feasible at lab-scale. Hence, real-time data captured from the ECT hardware setup scenario is used as crucial input for simulating the proposed architecture on a large-scale platform using the NS-2 simulator. The original architecture is simulated to validate the data acquired. Table I lists the simulation parameters used. A few well-known routing protocols – NOAH [16] and DSDV [17], AODV [18] and DSR [19] – are applied to check the performance of this architecture. The effect of types of packets – synchronous (SYNC) and asynchronous (ASYNC) – are also studied.

Table I
Simulation Parameters

Parameter	Value
Region (in meter)	$5 \times 5, 10 \times 10, 20 \times 20$
Packet Size	94 bytes
Payload	80 bytes
Packet Type	DH5
Time between packets	0.8 sec
Packet loss mode	Enabled
Protocols	NOAH [16], DSDV [17], AODV [18], DSR [19]
Scheduling Algorithm	DRR
Data Rate	0.0072 Mbps
Radio Range	10 m
Radio Signal	2.4 GHz
Simulation Time	600 sec

Post-validation tasks involve simulation of possible scenarios, which may arise during real-life implementations of our ECT-M architecture. Two types of room-based scenarios are considered – hospitals and retirement homes, as our targeted application area is geriatrics and care for the physically infirm. In the first case, a subject is allotted a room, whereas, in the second, multiple subjects are allotted the same room. Only one controller is present in each room. These rooms are surrounded by similar types of rooms having similar set-ups in them. As the architecture is designed for indoor use only and relies on Bluetooth for communication between the headsets and the ECTs, various Ad-hoc routing protocols are considered for simulation. High power radios, capable of communication over long ranges are not desirable as they cause interference to neighboring nodes, cannot be restricted within rooms and may capture nearby ECTs, not in their planned area of operation using conventional setup.

Table II
Table of Symbols

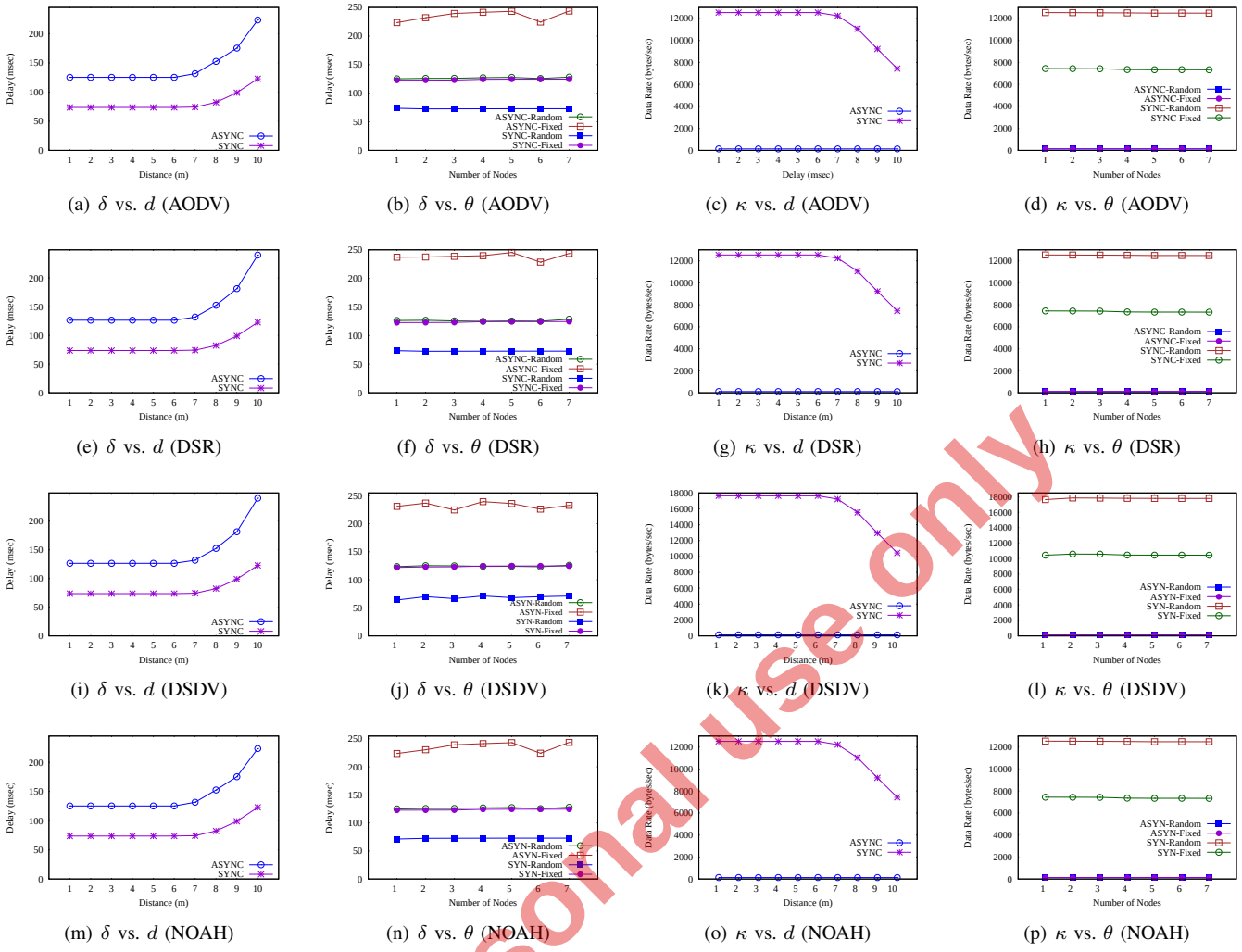
Symbol	Description
ρ_{max}	Maximum possible flow
ρ_{recd_j}	Received flow
θ	Network Size
Θ	Room Size
μ	Step Size
β_i	Unit i Bytes
$\delta_i(1)$	Unit delay at i^{th} instance
C_H	Hop Count
$PkL_{(hx)}$	Number of Packets lost at last hop
$Pk_{(hx)}$	Number of Packets at last hop
P	Power
d	Distance

A. Performance Metrics

We use different performance metrics to analyze the performance of the ECT nodes. Table II lists the meanings of the symbols used.

- *Data Rate* (κ): It is defined as the total data received by destination node in bytes in unit time. It is denoted as $\kappa = \frac{\sum_{j=0}^{\rho_{max}} \rho_{recd_j} \sum_{i=0}^{\theta} \beta_i}{\mu.\theta}$
- *Delay* (δ): It is defined as the time lapse between the packet generation time and its arrival time at destination. It is denoted as $\delta = \frac{\sum_{j=0}^{\rho_{max}} \rho_{recd_j} \sum_{i=0}^{\theta} \delta_i(1)}{C_H}$

for personal use only



(a) δ vs. d (AODV) (b) δ vs. θ (AODV) (c) κ vs. d (AODV) (d) κ vs. θ (AODV)

(e) δ vs. d (DSR) (f) δ vs. θ (DSR) (g) κ vs. d (DSR) (h) κ vs. θ (DSR)

(i) δ vs. d (DSDV) (j) δ vs. θ (DSDV) (k) κ vs. d (DSDV) (l) κ vs. θ (DSDV)

(m) δ vs. d (NOAH) (n) δ vs. θ (NOAH) (o) κ vs. d (NOAH) (p) κ vs. θ (NOAH)

Figure 4. Simulation Results showing the effect of various parameters on delay (δ) and data-rate (κ) of the ECT-M architecture. This also serves as a guide for selecting the appropriate protocol and node configuration for implementing the architecture in real-time.

For personal use only

- *Packet Delivery Ratio* (η): It is the ratio of the number of data packets received at the destination to the number of data packets sent from the source. η is represented as $\eta = 1 - \sum_{i=0}^{C_H} \frac{PkL_{hx}}{Pk_{fwx}}$

In all the previously mentioned simulation scenarios, the packet delivery ratio is observed to be close to 100%, if the source and destination nodes are in the range of each other, and zero, if they were not. There is no packet loss when connection persisted, because the signal strength is nearly constant in the whole range of nodes, which practically decreases with distance. The attenuation (α) in dB is represented as $\alpha_{dB} = 10.log.\frac{P_{source}}{P_{Destination}}$. Table I summarizes the attenuation in radio signal power due to the presence of various obstructions.

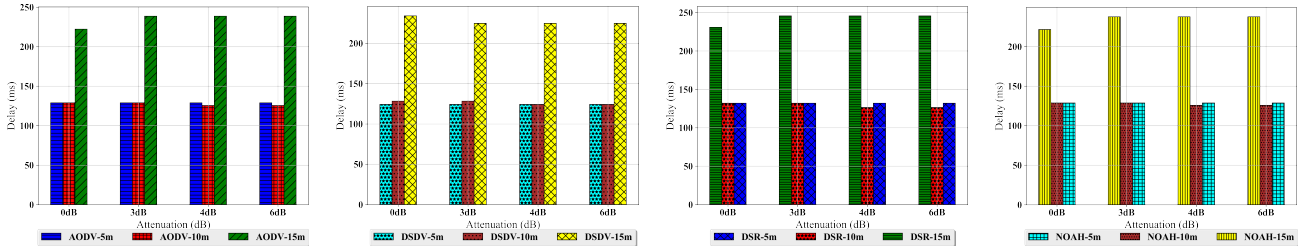
V. Results

This section discusses the simulation outputs with respect to two parameters – *Delay* and *Data-rate* – for the proposed implementation scenarios using the ECT architectures and algorithms.

A. Delay

The effect on delay (δ) due to distance, network size and interference of neighboring headsets or ECT (nodes) is examined in this section. For ease of simulation, and for the sake of symmetry, a square room having an ECT is considered. Eight other rooms of similar type surround the room. Three square rooms of dimensions $5 \times 5\ m^2$, $10 \times 10\ m^2$ and $20 \times 20\ m^2$ are considered. The position of the subject and the controller in all these rooms are kept the same and fixed. This scenario is further subdivided by two different factors, i.e., the size of the room and type of room wall. Walls are known to cause attenuation to wireless radio signals, depending on the material of the wall.

1) Effect of Distance: In this scenario, the headset sends data to the ECT, and then its behavior is examined, when the radial distance between the ECTs are varied from 1 to 10 m. Based on the observations, the plots for the delay concerning distance (δ vs. d) for AODV, DSR, DSDV, and NOAH are shown in Figs. 4(a), 4(e), 4(i) and 4(m). It is clear from the plots that, for all four protocols, δ is constant till 6m and



(a) AODV (b) DSDV (c) DSR (d) NOAH

Figure 5. Effect of attenuation α_{dB} with respect to changes in room sizes for AODV, DSDV, DSR and NOAH protocols for the ECT-M architecture selection.

starts increasing beyond it. The δ values in ASYNC mode are higher than in SYNC mode. For seamless functioning and extension of the communication range between a headset and the ECT, an ad-hoc network strategy can be used. The estimated d values should be $\leq 6m$ in the SYNC mode of packet transmission, using these ad-hoc strategies.

2) Effect of Network Size: The number of nodes in the same ECT network is gradually increased, and its effects were observed. This experiment is the variation between δ and θ. The plots in Figs. 4(b), 4(f), 4(j) and 4(n) show that for all the four protocols mentioned above, ASYNC mode with fixed node configuration, contributes to higher δ values. This rise in δ signifies that an increased delay will be incurred in the ASYNC-fixed configuration. Application of ad-hoc strategies for communication range extension gives the optimum results, with least δ values for the SYNC mode of operation with random node placement.

3) Effect of Attenuation: The effect of different wall types on the resulting delay is studied by mapping the different wall materials to their attenuation (in dB) values, and is a comparision between δ and α_{dB}. These mapped attenuation values are then studied for their contribution to delay, for both ASYNC and SYNC modes of operation, using AODV, DSR, DSDV as well as NOAH protocols. Figs. 5(a), 5(b), 5(c) and 5(d) show that the δ values, while using AODV, DSR and NOAH, remain same for the rooms with dimensions 5×5 m^2 and 10×10 m^2, but increases for the 20×20 m^2 room, upto the 3dB mark. The δ value falls for the 10×10 m^2 room beyond the 3dB mark. Using DSDV, the 5×5 m^2 room shows constant δ values at all α_{dB} levels, but falling δ values for the 10×10 m^2 and 20×20 m^2 room.

4) Effect of Room Size and Node Activity: This section studies the effects of node activity (idle/active), along with room size, on δ values, and can be summarized as a comparision between δ and Θ. Plots in Figs. 6(a), 6(b), 6(c) and 6(d) show that, for active nodes, the δ values are constant for room sizes of 5×5 m^2 and 10×10 m^2 but increases beyond it. The δ values for the idle nodes decreases as the room size changes from 5×5 m^2 to 10×10 m^2, and then, increases for 20×20 m^2. This behavior is seen for all four protocols.

B. Data-rate

The effect of distance and network size on data-rate(κ) is examined in this section. The parameters of the setup are same as the previous sub-section.

1) Effect of Distance: This parameter can be summed up as a variation between κ and d. The plots in Figs. 4(c), 4(g), 4(k) and 4(o) show the changes in κ with respect to d for both ASYNC and SYNC modes of operation using AODV, DSR, DSDV and NOAH respectively. The data rates in the ASYNC modes of operation remain mostly constant throughout, whereas, for the SYNC mode of operation, it starts reducing after 6m. DSDV achieves much higher data-rate for SYNC mode of operation, as compared to the other three protocols and their configurations.

2) Effect of Network Size: This parameter can be represented as a comparison between κ vs. θ. The plots in Figs. 4(d), 4(h), 4(l) and 4(p) show that higher κ values are achieved for SYNC mode of operation than for the ASYNC mode. Even higher κ values can be attained if the nodes are placed randomly as compared to when they are given a fixed arrangement. The data rate for the SYNC-random configuration in DSDV is significantly higher than the other three protocols.

VI. Conclusion

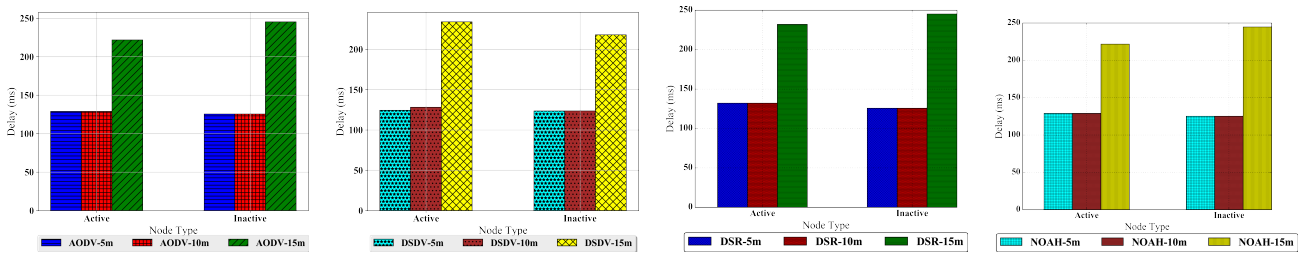
The proposed AAL enabling architecture for people with a mobility disability and the infirm is a step towards assisting this section of the society with user-friendly technologies. The ECT-1 and ECT-M hardware architectures are designed to be user-friendly, less complicated, and cheaper than other AAL-based technologies present in the market. The unique feature of this architecture is its networking ability and collision resolution algorithm, which has not been attempted with these kinds of devices, yet. The performance metrics of simulating the architecture as mentioned earlier, against scenarios which are most likely to occur during its *ex-situ* implementation, show that the performance is entirely satisfactory and does not affect the control of actuating devices adversely.

In the future, we plan on deploying the proposed architecture for controlling devices and machinery, such as computers and vehicles, as well as implementing non-verbal communication techniques such as text messages, and the prospect of using opportunistic methods to control far-away and off-grid devices will also be explored.

References

[1] Neurosky mindwave mobile. [Online]. Available: http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

for personal use only



(a) AODV (b) DSDV (c) DSR (d) NOAH

Figure 6. Changes in delay δ for various room sizes and state of the node for AODV, DSDV, DSR and NOAH protocols for the ECT-M architecture simulation.

[2] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From Activity Recognition to Intention Recognition for Assisted Living Within Smart Homes," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 368–379, June 2017.

[3] C. Holzner, C. Guger, G. Edlinger, C. Gronegress, and M. Slater, "Virtual Smart Home Controlled by Thoughts," in *18^{th} IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09*, June 2009, pp. 236–239.

[4] B. Ando, S. Baglio, C. Lombardo, V. Marletta, E. Pergolizzi, and A. Pistorio, "An event polarized paradigm for ADL detection in AAL context," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014*, May 2014, pp. 1079–1082.

[5] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "Smartphone-centric ambient assisted living platform for patients suffering from co-morbidities monitoring," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 34–41, January 2015.

[6] R. Hervas, J. Bravo, and J. Fontecha, "An Assistive Navigation System Based on Augmented Reality and Context Awareness for People With Mild Cognitive Impairments," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 368–374, Jan 2014.

[7] M. Broy and A. Schmidt, "Challenges in Engineering Cyber-Physical Systems," *Computer*, vol. 47, no. 2, pp. 70–72, Feb 2014.

[8] C.-T. Lin, B.-S. Lin, F.-C. Lin, and C.-J. Chang, "Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking," *IEEE Systems Journal*, vol. 8, no. 2, pp. 363–370, 2014.

[9] A. Bertrand, "Distributed Signal Processing for Wireless EEG Sensor Networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, Nov 2015.

[10] J. Faller, S. Torrellas, F. Miralles, C. Holzner, C. Kapeller, C. Guger, J. Bund, G. Muller-Putz, and R. Scherer, "Prototype of an auto-calibrating, context-aware, hybrid brain-computer interface," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2012*, Aug 2012, pp. 1827–1830.

[11] D. McMullen, G. Hotson, K. Katyal, B. Wester, M. Fifer, T. McGee, A. Harris, M. Johannes, R. Vogelstein, A. Ravitz, W. Anderson, N. Thakor, and N. Crone, "Demonstration of a Semi-Autonomous Hybrid Brain-Machine Interface Using Human Intracranial EEG, Eye Tracking, and Computer Vision to Control a Robotic Upper Limb Prosthetic," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 784–796, 2014.

[12] M. Kawanabe, "Robust feature construction against non-stationarity for EEG brain-machine interface," in *International Winter Workshop on Brain-Computer Interface (BCI), 2014*, Feb 2014, pp. 1–4.

[13] N. R. Waytowich and D. J. Krusienski, "Multiclass Steady-State Visual Evoked Potential Frequency Evaluation Using Chirp-Modulated Stimuli," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 593–600, Aug 2016.

[14] M. Simic, M. Tariq, and P. Trivailo, "EEG-Based BCI Control Schemes for Lower-Limb Assistive-Robots," *Frontiers in Human Neuroscience*, vol. 12, p. 312, 2018.

[15] M. Almonacid, J. Ibarrola, and J.-M. Cano-Izquierdo, "Voting strategy to enhance multimodel eeg-based classifier systems for motor imagery BCI," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1082–1088, 2016.

[16] J. Widmer. No ad-hoc routing agent (NOAH). [Online]. Available: http://icapeople.epfl.ch/widmer/uwb/ns-2/noah

[17] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM computer communication review*, vol. 24, no. 4. ACM, 1994, pp. 234–244.

[18] S. R. Das, E. M. Belding-Royer, and C. E. Perkins, "Ad hoc on-demand distance vector (aodv) routing," 2003.

[19] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*. Springer, 1996, pp. 153–181.

For personal use only