Application and Transport Layer Standards

Muhammad Hazimi Bin Yusri (mhby1g21, Team E1),
Law Shaw Zuan (szl1c21, Team E2)
ELEC3227 Embedded Networked Systems Coursework

Application Layer (APP)

- Hostnames are E11, E12, E13 and E21, E22, E23 for Team E1 and E2.
- One LED on PC6 and one button on PA6, UART communication at Port D, RFM12B-S2 Radio Module at Port B.
    - Leftover pins can be used for optional implementation by individuals.
- "IPPLS" request is broadcast together with device own hostname and IP address whenever a node joins a network to get other devices hostname and IP address. The receiving nodes replies with APP Data containing 3 bytes hostname and 1 byte IP address it is associated with.
- Hostname-IP address table is used to resolve hostname into destination IP address for Network Layer.

Application 1: Button interaction with LED

- Only 1 byte of APP data will be used.
- Button LED ON sends a byte of 0xFF, button LED OFF sends a byte of 0x00.
- Button/LED interactions between nodes are hard coded.
    - Button on a node will send data (LED ON/OFF) to only one of its adjacent nodes.
    - E.g.: E11 button will turn on E12 LED, E12 button will turn on E13 LED etc...
    - The ending node will turn on starting node (E23 button turns on E11 LED)

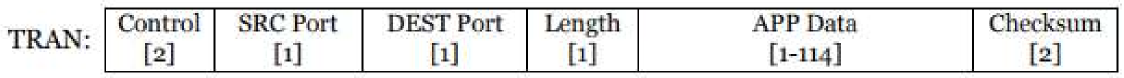
Application 2: UART as input and output for sending strings and specific LED commands

- UART for sending/receiving messages, constrained to 9 bytes or characters maximum.
    - E.g. string commands: SEND E11 <MESSAGES> (Send string messages to E11)
- UART LED commands use 9 bytes, the first left-most byte is reserved for basic ON/OFF byte functionality and the remaining bytes are for individual team use.
- UART commands to interact with LED on other nodes.
    - Example of basic commands: ON E11 (Turn on LED on node E11), OFF E23 (Turn off LED on node E23), ON E11, E22 (Turns on LED on E11, and E22)
    - Example of optional commands that uses additional parameters: ON E11 R5G9B7 (Turn on optional RGB LED with different brightness level given)
        - Clarification: UART to control LEDs will result in 0xFF source port and 0x00 LED port so it can be easily differentiated from button presses.
- The special character ("/") is used to toggle between input or output mode in UART command line interface (cli).

Interfaces:

- APP Layer provides destination port, source port and destination IP address to TRAN Layer for correct connection.
- APP Layer provides APP data with correct length to TRAN Layer for segmentation.

Transport Layer (TRAN)



TRAN:	Control [2]	SRC Port [1]	DEST Port [1]	Length [1]	APP Data [1-114]	Checksum [2]

Figure 1. Segment field structures defined by the coursework

- Using TCP (3-way handshake connection) with segment structure shown below,
  a. Control bits (2 bytes

Left Byte Assignment

(15)	(14)	(13:11)	(10)	(9)	(8)
SYN	ACK	BUF	DR	IP_flag	IPPLS_flag

** SYN and ACK for handshake, BUF for buffer remaining, DR for disconnect request, IP_flag for signifying other device IP and hostname in 4-byte APP Data, IPPLS_flag to request other devices IP and hostname information.

Right Byte Assignment

(7:4)	(3:0)
SEQ NUM	ACK NUM

b. SRC Port (1 byte)
   - 0x00 for button presses, 0xFF for UART cli inputs.

c. DEST Port (1 byte)
   - 0x00 for LEDs, 0xFF for UART cli outputs.

d. Length (1 byte)
   - Length of APP Data bytes which are not 0x00, for example during TCP initial and disconnect handshake, it would be 0 as no APP Data is sent.
   - Unused bytes of UART messages will be set to 0x00, thus will not always be 9 bytes.

e. APP Data (0 bytes OR 1 byte OR 4 bytes OR 9 bytes)
   - 0 bytes for TCP handshake 1 byte for LED with button, 4 bytes for HOSTNAME|IP ADDRESS, and 9 bytes for UART messages/commands.

f. Checksum (1 byte)
   - Even parity bit check
     - Simpler checksum is used to reduce the processing time requirement as both NET and DLL layer are already using CRC

Interfaces:

- TRAN Layer provides destination IP address to Network Layer.
- TRAN Layer provides segment to Network Layer

The absence of **Bandwidth Allocation** or **Congestion Control** is deliberate, due to the periodic and bursty nature of the data rate, to reduce microcontroller load for improved power efficiency and less overhead. In rare cases of congestion and collision, CSMA p-persistent from DLL combined with TCP connection-oriented transport should minimize its impact.

Network Layer Standards

Jade Vaux (jv1g21, Team E1), Wai Wah Tang (wwt1u21, Team E2)
ELEC3227 Embedded Networked Systems Coursework

1 Protocols

1.1 Packet Bit Allocation:

Figure 1 holistically displays the allocated functions for the available bytes in the network packet.

Control [1] (Byte 1)	Control [1] (Byte 2)	SRC Address [1]	DEST Address [1]	Length [1]	TRAN Segment [8-121]	Checksum [2]

Transmission Type/ DEST for next hop	Hops for Flooding/ ARP	IP SRC Address	IP DEST Address	Length of the TRAN Segment	TRAN Segment	CRC-16-CC ITT

Figure 1: Bit Allocation.

1.2 IP (Internet Protocol) Addressing System:

The IP address format is hierarchical and comprised of two parts: the network number, and the device number. The network number forms the three most significant bits of the address byte ranging from bits 001 to 111 as the network address 000 has been reserved for flooding purposes. The remaining five bits of the byte represent the device number on the network, providing a total of 244 addresses available (7 networks housing 32 hosts/routers each). *N.B.*: devices on different networks may share the same device number. Figure 2 displays an example of the addressing format.

0	0	1	0	0	0	0	1

Figure 2: Address Format.

1.3 First Byte of Control:

The first control byte in the packet will typically contain the IP address, as has been defined in the previous section, of the next hop. However, when the network is not involved in one-to-one transmission it may use this byte to depict a special operation (e.g. flooding data or broadcasting) occurring when all 8 bits have been set to 0.

1.4 Second Byte of Control

The second control byte is used for counting the hops for the flooding operation and the ARP. The first 4 most significant bits represent the operation type, while the remaining four lest significant bits are employed for flood control.

Bits 1-4 (Operation Type):

- ’0000’: Normal one-to-one data transmission
- ‘0001’: Flooding data
- ‘0010’: Flood HELLO packet (For new device to the network)
- ‘0011’: Regular ARP check
- ‘0100’: Update ARP table
- ‘0101’ to ‘1111’: Reserved for future use or other specific operation

Bits 5-8 (Flood Control):

- ‘0000’: No hop limit (Floods throughout the entire network/No limit)
- ‘0001’: 1 hop limit
- ‘0010’: 2 hop limit
- … up to ‘0111’: 7 hop limit
- ‘1000’ to ‘1111’: Reserved for future use

2 Interface:

ARP:

- Used to interface with the Data Link Layer.
- Runs a check returning the IP and MAC addresses in a cache which may be used as a lookup table for DLL to assign the appropriate MAC address in their Addressing Field.

Application and Transport Layer:

- The Application Layer will send the IP address of the desired destination device directly to the Network Layer so that the Network Layer does not deal with host names.
- Network layer provides routing service from source host to destination host for the application and transport layers.

3 Error Detection:

A 16-bit CRC will be used due to its simple, fast and advanced error detection capability. The parameters used are as follows:

- Polynomial: ‘0x1021’ (CRC-16-CCITT), due to its good performance in detecting common error patterns, especially in noisy environments like radio communication.
- Initial Value: ‘0xFFFF’ (All bits set to 1). This allows for immediate interaction error detection at the beginning of the data.
- For simplicity the final XOR Value is ‘0x0000’ and the CRC value will not be modified after being calculated. Additionally, reflection at input or output for this same reason.

Data link layer Standard Document – Joel & Syazwan

DLL:	Header [1]	Control [2]	Addressing [2]	Length [1]	NET Packet (or part of) [1-23]	Checksum [2]	Footer [1]

Table 1. Frame field structure

Table 1 shows the frame field structure of the Data Link Layer. A frame is encapsulated by its header and footer bytes. Within the header and footer are the 2-byte control field, 2-byte addressing field, 1-byte length field, up to 23 bytes of the NET Packet payload per frame, and the 2-byte Checksum.

Header & Footer

The header and the footer will be used as byte delimiter to indicate the start and end of the frame. The header and the footer byte used in the frame will be binary 01111110. To prevent framing errors from occurring if the byte pattern occurs in the payload, bit stuffing is used on transmission such that after five 1's occurs in the payload, a 0 is added. On the receiving end, after 5 consecutive 1's is detected, the following 0 is removed to de-stuff the data.

Control

Acknowledgment bits [0-7]	Sequence bits [8-15]

Table 2. Control field bytes allocation

Table 2 shows how the control bytes will be allocated. The sliding window protocol will be used to implement flow control, including selective repeat with a window size of 8 and a buffer size of 4 at a time. This protocol will also be used for retransmission of erroneous or missing frames using packet numbers and negative acknowledgements. The first byte (bit 0-7) represents acknowledgements (ACK) or negative acknowledgements (NAK). This field will be set to 0x00 if not acknowledging (or is sending). The second byte of the control field represents the sequence number attached to the frame used for the sliding window protocol.

Addressing

MAC Source Address [0-7]	Source Address [8-15]

Table 3. Addressing field bytes allocation

Table 3 shows the addressing field byte allocation. The first byte of the addressing field will be a hard-coded source MAC address of the sending device. The second holds the MAC address of the next hop. The next hop address required is provided by NET layer through an ARP table.

Length

Length field holds the number of frames representing one network packet (will be 1 if the network packet is less than 23 bytes or does not require splitting).

Net packet payload

The NET Packet is retrieved from network layer. It has a minimum of 1 byte and a maximum of 23 bytes. If a network packet is larger than 23 bytes, the payload will be split into multiple DLL frames and will be reassembled at the receiver. Before the splitting is done, the payload will be bit stuffed at any occurrence of the header and footer delimiter byte.

Checksum

The checksum will be generated by a 16-bit CRC generator. The generator used will be $x^{16}+x^{12}+x^{5}+1$ (0b10001000000100001). If erroneous frame is detected, the receiver will send a request for retransmission using NAK. If no error is detected, the receiver will send an ACK.

CSMA

As for collision detection, CSMA p-persistent protocol will be used. When the channel is detected to be idle, the transmitting device will transmit the data with the probability of *p*. The choice of *p* must provide the best balance between collision prevention and latency. If the channel remains idle in the subsequent time slot, the process repeats itself. However, if the channel is found to be busy, the transmitting device waits for a random back-off time, before starting again.

Interfaces

- getMACAddress - provides the MAC address of the current device.
- fromNETLayer - obtain NET Packet from NET layer.
- fromDLL - NET layer call function from DLL.
- fromPHYLayer - get messages from PHY layer.
- toPHYLayer - send messages to the PHY layer to transmit it.