COMP 7003
Assignment 2
Design

Gursidh Sandhu
A01319563
Oct 3,2024

Purpose

This assignment captures packets from the network and manually processes those packets to display all fields for the following protocols:

- IPv4
- TCP
- UDP
- ARP

Data Types

Arguments

Purpose: To hold the unparsed command-line argument information

Field	Type	Description
program_name	string	The name of the program
interface	string	The interface used for that certain OS to capture packets

Settings

Purpose: To hold the settings the program needs to run.

Field	Type	Description
program_name	string	The name of the program
interface	string	The interface used for that certain OS to capture packets

Context

Purpose: To hold the arguments, settings, and exit information

Field	Type	Description
arguments	Arguments	The command line arguments
settings	Settings	The parsed command line arguments

Functions

Function	Description
hex_to_decimal	Convert hexadecimal value to decimal
hex_to_ip	Convert hexadecimal value to ip address format
hex_to_hardware	Convert hexadecimal value to hardware address format
hex_to_binary_with_spaces	Convert hexadecimal value to binary with spaces every 4 bits
check_bit	Check if a binary bit is 0 or 1
get_first_three_bits	Get the first 3 bits from string of bits
parse_ethernet_header	Parse the beginning ethernet header section of network packet
parse_ipv_packet	Parse the ipv4 packet and print out each field
parse_tcp_packet	Parse the tcp packet and print out each field
parse_udp_packet	Parse the udp packet and print out each field
parse_arp_packet	Parse the arp packet and print out each field
packet_callback	Function to handle each captured packet
capture_packets	Function to capture packets on a specified interface using filters
__main__	Main method that starts flow of program

States

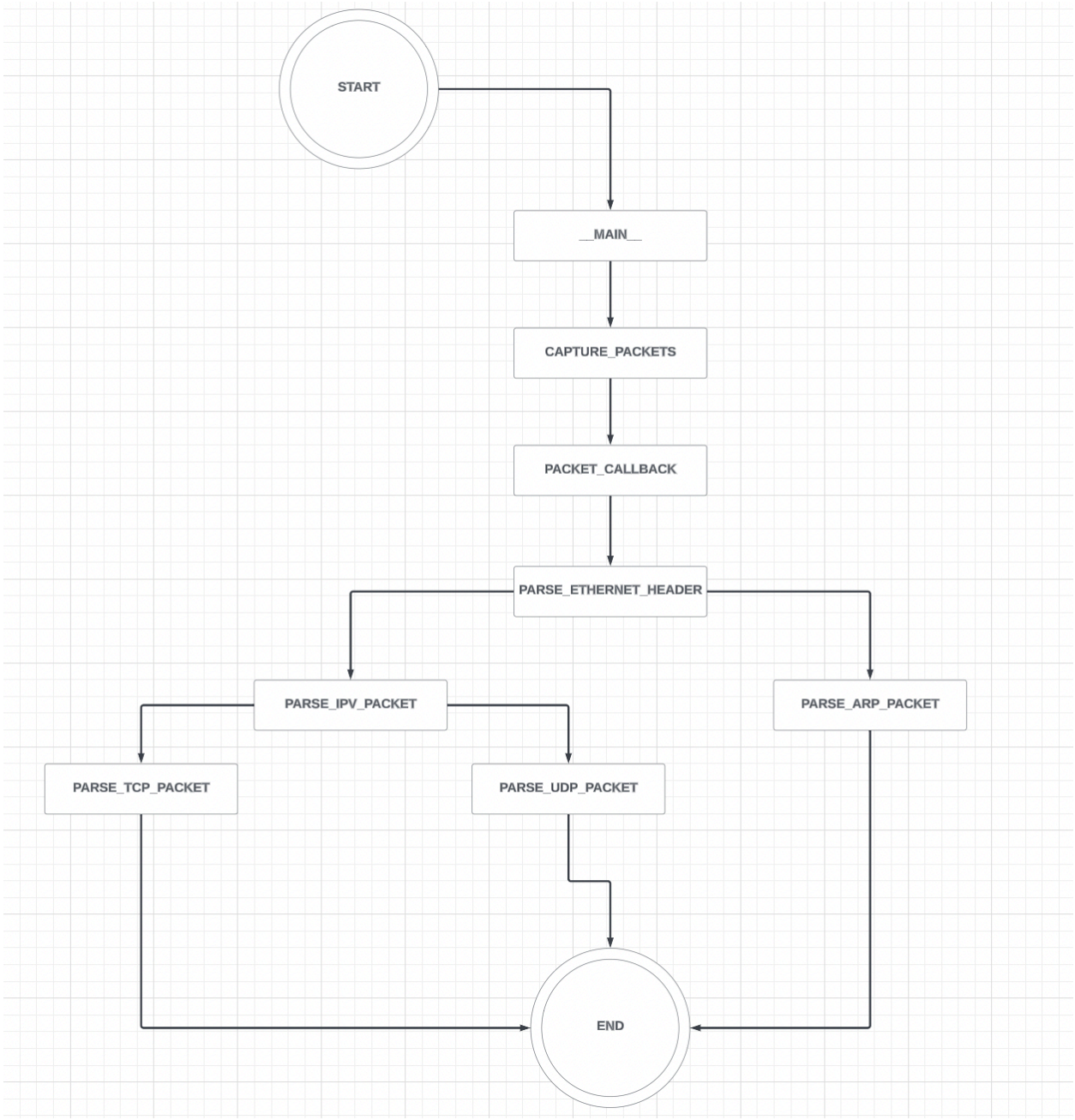
State	Description
__MAIN__	Main method that starts flow of program
CAPTURE_PACKETS	Function to capture packets on a specified interface using filters
PACKET_CALLBACK	Function to handle each captured packet
PARSE_ETHERNET_HEADER	Parse the beginning ethernet header section of network packet
PARSE_IPV_PACKET	Parse the ipv4 packet and print out each field

PARSE_ARP_PACKET	Parse the arp packet and print out each field
PARSE_TCP_PACKET	Parse the tcp packet and print out each field
PARSE_UDP_PACKET	Parse the udp packet and print out each field

State Table

From State	To State	Function
START	__MAIN__	-
__MAIN__	CAPTURE_PACKETS	capture_packets
CAPTURE_PACKETS	PACKET_CALLBACK	packet_callback (within sniff)
PACKET_CALLBACK	PARSE_ETHERNET_HEADER	parse_ethernet_header
PARSE_ETHERNET_HEADER	PARSE_IPV_PACKET	parse_ipv_packet
PARSE_ETHERNET_HEADER	PARSE_ARP_PACKET	parse_arp_packet
PARSE_IPV_PACKET	PARSE_TCP_PACKET	parse_tcp_packet
PARSE_IPV_PACKET	PARSE_UDP_PACKET	parse_udp_packet
PARSE_ARP_PACKET	EXIT	-
PARSE_TCP_PACKET	EXIT	-
PARSE_UDP_PACKET	EXIT	-

State Transition Diagram



Pseudocode

hex_to_decimal

Parameters

Parameter	Type	Description
hex_string	string	A string of hex bits

Return

Value	Reason
decimal_value	The decimal version of converted hex bits

Pseudo Code

```
convert hex_string to decimal_value
return decimal_value
```

hex_to_ip

Parameters

Parameter	Type	Description
hex_string	string	A string of hex bits

Return

Value	Reason
ip_value	The hex bits converted into an ip address

Pseudo Code

```
split hex_string into pairs
set ip_value = Convert each pair to decimal
return ip_value
```

hex_to_hardware

Parameters

Parameter	Type	Description
hex_string	string	A string of hex bits

Return

Value	Reason
hardware_value	The hex bits converted into a hardware address

Pseudo Code

```
set hardware_value = split hex_string into pairs and add :
return hardware_value
```

hex_to_binary_with_spaces

Parameters

Parameter	Type	Description
hex_value	string	One hex bit

Return

Value	Reason
spaced_binary	The hex bit converted into individual bits in sets of 4 bits each

Pseudo Code

```
convert hex_value into binary bits
set spaced_binary = binary bits split into 4s
return spaced_binary
```

check_bit

Parameters

Parameter	Type	Description
bit	integer	Binary bit

Return

Value	Reason
bit	The bit is 1 or 0

Pseudo Code

```
check if bit is 0 or 1
return bit
```

get_first_three_bits

Parameters

Parameter	Type	Description
binary_value	string	String of binary bits

Return

Value	Reason
first_three_bits	The first three bits from the binary string

Pseudo Code

```
set first_three_bits = extract first three from binary_value
return first_three_bits
```

parse_ethernet_header

Parameters

Parameter	Type	Description
hex_data	string	The entire packet contents in hex form

Return

No returns.

Pseudo Code

```
extract ether_type from hex_data
If ether_type == 0800
     parse_ipv_packet
Else if ether_type == 0806
     parse_arp_packet
Else
     Print error
```

parse_ipv_packet

Parameters

Parameter	Type	Description
hex_data	string	The entire packet contents in hex form

Return

No returns.

Pseudo Code

```
Extract each field from hex_data
Check for options and data based on header_length
```

```
Print each field using appropriate conversion method
Extract protocol field
If protocol == 6
     Parse_tcp_packet
Else if protocol == 11
     Parse_udp_packet
Else
     Print error
```

parse_tcp_packet

Parameters

Parameter	Type	Description
hex_data	string	The entire packet contents in hex form
packet_endpoint	integer	The place in the hex_data where the ipv4 contents finished

Return

No returns.

Pseudo Code

```
Extract each field from hex_data
Check for options and data based on header_length
Print each field using appropriate conversion method
```

parse_udp_packet

Parameters

Parameter	Type	Description
hex_data	string	The entire packet contents in hex form
packet_endpoint	integer	The place in the hex_data where the ipv4 contents finished

Return

No returns.

Pseudo Code

```
Extract each field from hex_data
```

```
Print each field using appropriate conversion method
```

parse_arp_packet

Parameters

Parameter	Type	Description
hex_data	string	The entire packet contents in hex form

Return

No returns.

Pseudo Code

```
Extract each field from hex_data
Print each field using appropriate conversion method
```

packet_callback

Parameters

Parameter	Type	Description
packet	integer	The entire packet contents as bytes

Return

No returns.

Pseudo Code

```
Convert packet into hex_data
Call parse_ethernet_header
```

capture_packets

Parameters

Parameter	Type	Description

interface	string	The interface used for that certain OS to capture packets
capture_filter	string	The packet type to be captured
packet_count	integer	Number of times to capture packet

Return

No returns.

Pseudo Code

```
Call sniff using parameters
```

__main__

Parameters

No parameters.

Return

No returns.

Pseudo Code

```
Extract arguments from command line
Call capture_packets once each for ip+tcp,ip+udp and arp
```