Computer and Communication Networks (Project)

Summer 2011
Introduction
RKN assignment 1

- Implementation of a “PING”
  - ARP
  - ICMP
  - IP packets
  - Network sniffing

- Tools
  - Wireshark
  - JPCAP
  - Java Network Stack
Development Environment

- Install Eclipse for Java development (or any other IDE or editor of your liking)
  [http://eclipse.org](http://eclipse.org)
  - Most prominent open source IDE
  - Can be used for many different languages
  - Has a good tools support

- Install Java Development Kit version 1.6
  [http://java.sun.com](http://java.sun.com)
  - Latest version of the Java programming language plus tools
Development Tools

- Install Wireshark for packet inspection
  
  http://www.wireshark.org

  - Helps you verifying your development
  - Captures packets from almost all devices
  - Has a good GUI for packet visualization

- Install Jpcap for packet capturing under Java
  

  - Is based on WinPcap (already included in Wireshark)
  - Is a packet capture library which can be used in Java
Java Network Stack

- Can be downloaded from IAIK homepage
  [http://www.iaik.tugraz.at/content/teaching/bachelor_courses/rechner_und_kommunikationsnetze/downloads/](http://www.iaik.tugraz.at/content/teaching/bachelor_courses/rechner_und_kommunikationsnetze/downloads/)

- Is a Java framework which uses Jpcap as capture library as basis for the network stack implementation

- We will use only same basic classes for
  - network initialization, packet structures, basic processing and common tools

- The rest will be implemented by you !!!
First Assignment

- Allow other hosts (any under specific circumstances also the local host) to ping your network stack

- What do we need for that purpose?
  - Receive and send raw packets and not only data (content) as available through the Socket API
  - Assign an IP address and MAC address to the network stack
  - Allow others to obtain the physical address of your network stack
  - Receive ping requests, process them and send a reply back to the initiator
Receiving and sending of RAW packets

- Is already available in the Java Network Stack

- **TestNetwork** initializes and starts the „network stack“
  - Loads *properties* from a config file which specifies the *IP* and *MAC address* as well as some implementation issues
  - It is possible to specify a *filter option* in the configuration in order to not capture and forward all packets to the Java network stack
  - Initializes a *link layer* which is realized through Jpcap
  - Can be stopped (exited) through a keystroke
Receiving and sending of RAW packets

- JPcapLinkLayer is the abstraction of the link layer which has one receiver and one sender class

- The sender is **JPcapPhysicalSender**
  - The **receiver** captures all the packets and stores the ones intended for the local network stack (IP/MAC) in a buffer

- The receiver is **JPcapReceiver**
  - The **sender** receives input from the internet layer and puts them on the link
Receiving and sending of RAW packets

- **DefaultInternetLayer** implements the so called internet (TCP/IP) or network layer (OSI)

- Obtains *EthernetPackets* from the receive buffer as soon as they arrive in its *run* method

- Sends packets which have been put into the send buffer

- This is the starting point for your implementation
  - Get familiar with the structure and the usage of the framework
Address Resolution Protocol

- Is a network protocol for determining a network host's *link layer or hardware address* when only its Internet Layer (IP) or Network Layer address is known.

- Is a Link layer protocol since it is transmitted as *data portion* of an *ethernet* packet.

- Defined in IETF RFC 826 „*An Ethernet Address Resolution Protocol*“
Paket mit beliebiger Ziel-IP-Adresse soll gesendet werden

Ermittlung der Host-IP-Adresse und Schnittstelle in der Routingtabelle

- IP-Paket mit Ziel-IP-Adresse erzeugen und senden
  - nein
    - Ethernet-Schnittstelle?
      - ja
        - Ermittlung der MAC-Adresse und des Datums in der ARP-Tabelle
          - nein
            - Warten auf Eintrag in der ARP-Tabelle
              - optional
            - gefunden und nicht abgelautet?
              - nein
                - ARP-Anforderung (Broadcast) mit Host-IP-Adresse
                  - Warten auf Antwort
                    - nein
                      - ARP-Antwort mit Host-IP-Adresse und MAC-Adresse
                        - Host-IP-Adresse, Datum und MAC-Adresse in ARP-Tabelle eintragen
      - nein
        - (optional)

Routingtabelle (Schnittstelle, Host)
- 192.168.2.4 (lokal, Host 192.168.2.4)
- 127.0.0.1 (lokal, Host 192.168.2.3)
- 192.168.2.x (eth0, Ziel-IP-Adresse)
- x.x.x (eth0, Host 192.168.2.99)

ARP-Tabelle (MAC, Datum)
- 192.168.2.1 (08-00-20-0A-00-03, 8:23)
- 192.168.2.4 (08-00-20-10-20-00, 8:23)
- 192.168.2.99 (08-00-20-10-20-00, 8:23)
- 192.168.2.223 (08-00-20-10-20-00, 8:23)
Internet Control Message Protocol

- One of the core protocols of the Internet Protocol Suite. It is mainly used by networked computers operating system to send *error messages*
- Every machine that forwards an IP datagram has to decrement the *time to live* (TTL) field of the IP header by one; if the TTL reaches 0, an ICMP “*Time to live exceeded in transit*” message is sent to the source of the datagram.
- *Relies* on IP to perform its task.
  - ICMP header starts after bit 160 of the IP header
  - Usually doesn’t transport data
  - Each ICMP message is encapsulated within a single IP datagram and is therefore unreliable
Internet Control Message Protocol

ICMP Message Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Code/Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Echo Reply</td>
</tr>
<tr>
<td>3</td>
<td>Destination Unreachable</td>
</tr>
<tr>
<td>0</td>
<td>Net Unreachable</td>
</tr>
<tr>
<td>1</td>
<td>Host Unreachable</td>
</tr>
<tr>
<td>2</td>
<td>Protocol Unreachable</td>
</tr>
<tr>
<td>3</td>
<td>Port Unreachable</td>
</tr>
<tr>
<td>4</td>
<td>Fragmentation required, and DF set</td>
</tr>
<tr>
<td>5</td>
<td>Source Route Failed</td>
</tr>
<tr>
<td>6</td>
<td>Destination Network Unknown</td>
</tr>
<tr>
<td>7</td>
<td>Destination Host Unknown</td>
</tr>
<tr>
<td>8</td>
<td>Source Host Isolated</td>
</tr>
<tr>
<td>9</td>
<td>Network Administratively Prohibited</td>
</tr>
<tr>
<td>10</td>
<td>Host Administratively Prohibited</td>
</tr>
<tr>
<td>11</td>
<td>Network Unreachable for TOS</td>
</tr>
<tr>
<td>12</td>
<td>Host Unreachable for TOS</td>
</tr>
<tr>
<td>13</td>
<td>Communication Administratively Prohibited</td>
</tr>
<tr>
<td>4</td>
<td>Source Quench</td>
</tr>
<tr>
<td>5</td>
<td>Redirect</td>
</tr>
<tr>
<td>0</td>
<td>Redirect Datagram for the Network</td>
</tr>
<tr>
<td>1</td>
<td>Redirect Datagram for the Host</td>
</tr>
<tr>
<td>2</td>
<td>Redirect Datagram for the TOS &amp; Network</td>
</tr>
<tr>
<td>3</td>
<td>Redirect Datagram for the TOS &amp; Host</td>
</tr>
<tr>
<td>8</td>
<td>Echo</td>
</tr>
<tr>
<td>9</td>
<td>Router Advertisement</td>
</tr>
<tr>
<td>10</td>
<td>Router Selection</td>
</tr>
</tbody>
</table>

Checksum

Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.
How does a **PING** work

- `ping 10.27.152.172`
- Tries to send a ICMP „**echo request**“ message to the host with IP `10.27.152.172`
- First it is verified if the IP belongs to the local link or to another network (requires a router)
- For local addresses there should be an ARP entry in the ARP table
  - If not send an ARP **request** into the LAN
- The remote host answers with an ARP **reply** and its **physical address**
- Thereafter, send an ICMP „**echo request**“ and wait for an „**echo reply**“
Next steps

- Build groups of up to 4 students (if you are not able to find others use newsgroup)
- Send me a mail containing the members of your group
- Download the JNS and Jpcap from our homepage or directly use the Debian Virtual Appliance which has everything set up.
- Start to use the framework and try to understand how it works and implement the first assignment
2 Course assistants

- David Monichi and Christian Hartinger
- Both will have a tutorial and are there for questions about the assignment
- David has his tutorial Monday, 14.03, 13:00-14:30
- Christian has his tutorial Tuesday, 15.03 11:00-12:30
- Attend at this tutorials if you have direct questions and for more information about the exercise
- Use the newsgroup if you have questions thereafter