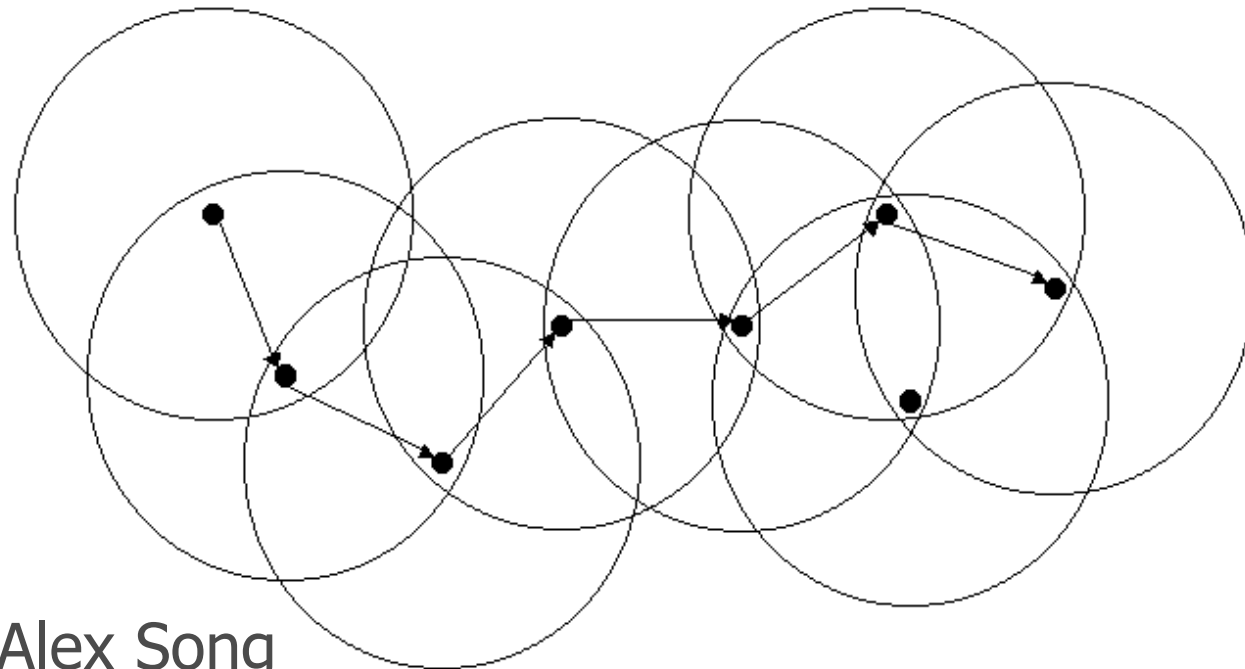


# *picoNet II*

Ad Hoc Multihop Wireless Networking  
for Mobile Devices



Name: Alex Song

Supervisor: Dr Mark Schulz

Web Page: <http://student.uq.edu.au/~s369677/>

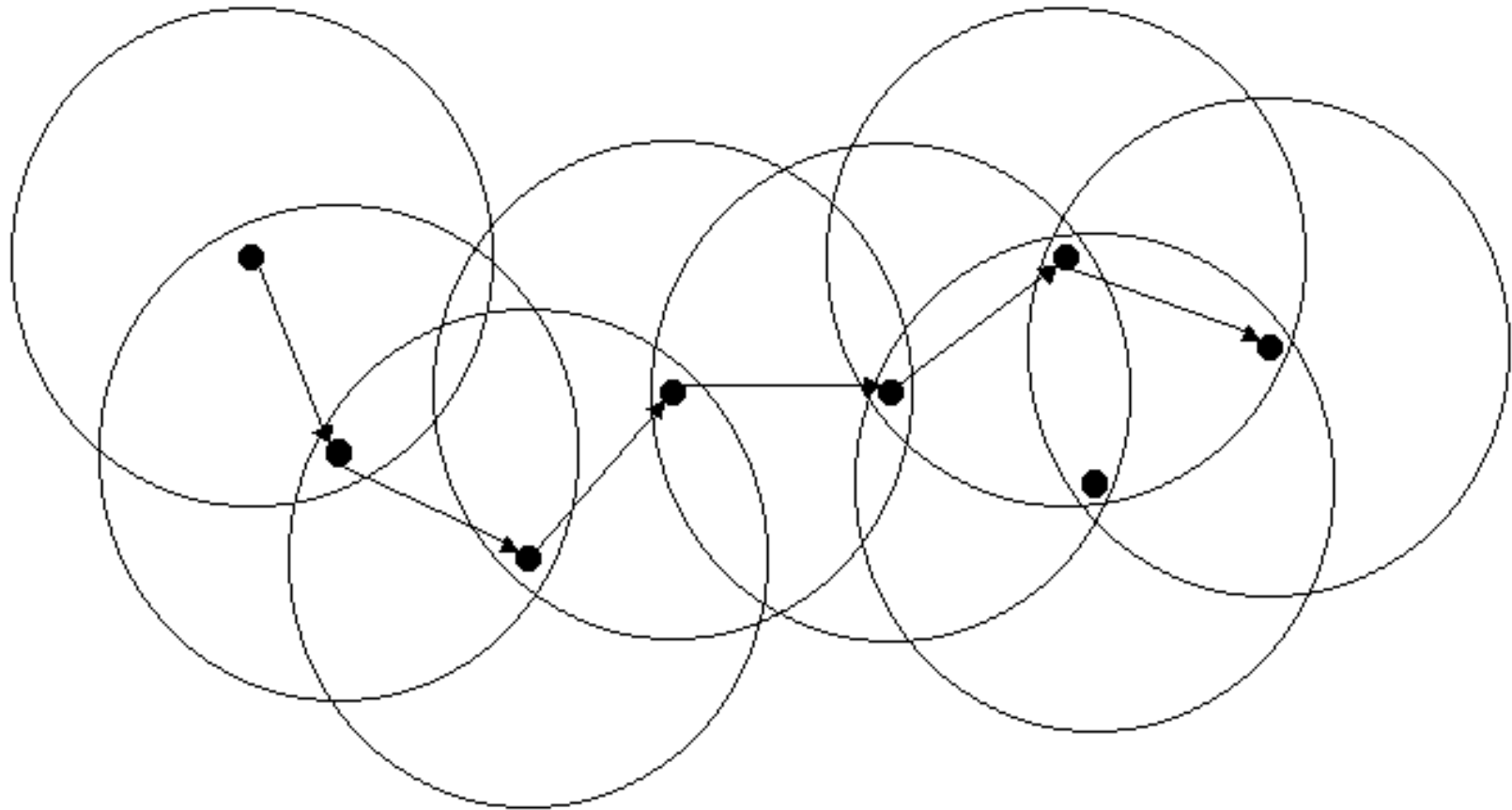
# Seminar Outline

- ◆ Describe the thesis and the need for it.
- ◆ Compare the features of current commercial products and my thesis.
- ◆ Technical details of my thesis.
- ◆ Current status of the thesis project.

# Thesis description

- ◆ To implement an ad hoc multihop routing protocol for IPv4 which will allow a dynamic network of mobile and handheld devices to be self-organizing and self-configuring.

# Thesis description

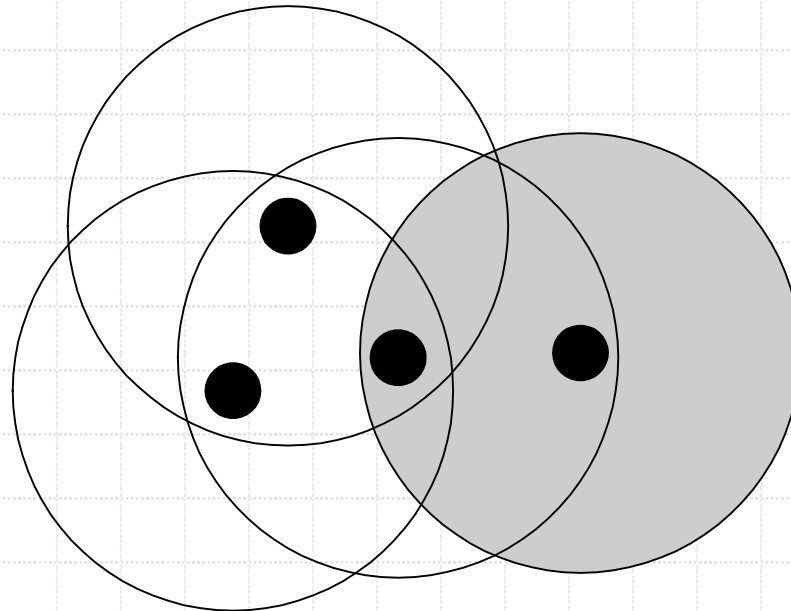


# Applications for thesis project

- ◆ Search and rescue missions.
- ◆ Conferences and meetings.
- ◆ Any application where there is no pre-existing networking infrastructure.
- ◆ Can also form a wireless extension to existing networks.

# Commercial Products

- ◆ IEEE 802.11 – Wireless LAN
- ◆ Bluetooth (version 1.1)
- ◆ Both technologies can only single hop in ad hoc mode.



# Piconet II features

- ◆ Provide multihop capabilities to existing hardware and software platforms eg. Linux/x86 and Linux/arm/iPAQ.
- ◆ Compatibility with standard IP networks.
- ◆ Transparent to existing TCP/IP applications.

# TCP/IP Network Model

Application (Telnet, FTP, DNS)

Transport (TCP, UDP)

Internet (IP)

Host-to-network (802.11, Bluetooth)



# Routing Protocol

- ◆ Dynamic Source Routing Protocol.(DSR)
- ◆ Internet draft:  
draft-ietf-manet-dsr-05.txt
- ◆ Current draft does not support packet fragmentation.
- ◆ Plan to extend draft and add support for packet fragmentation.

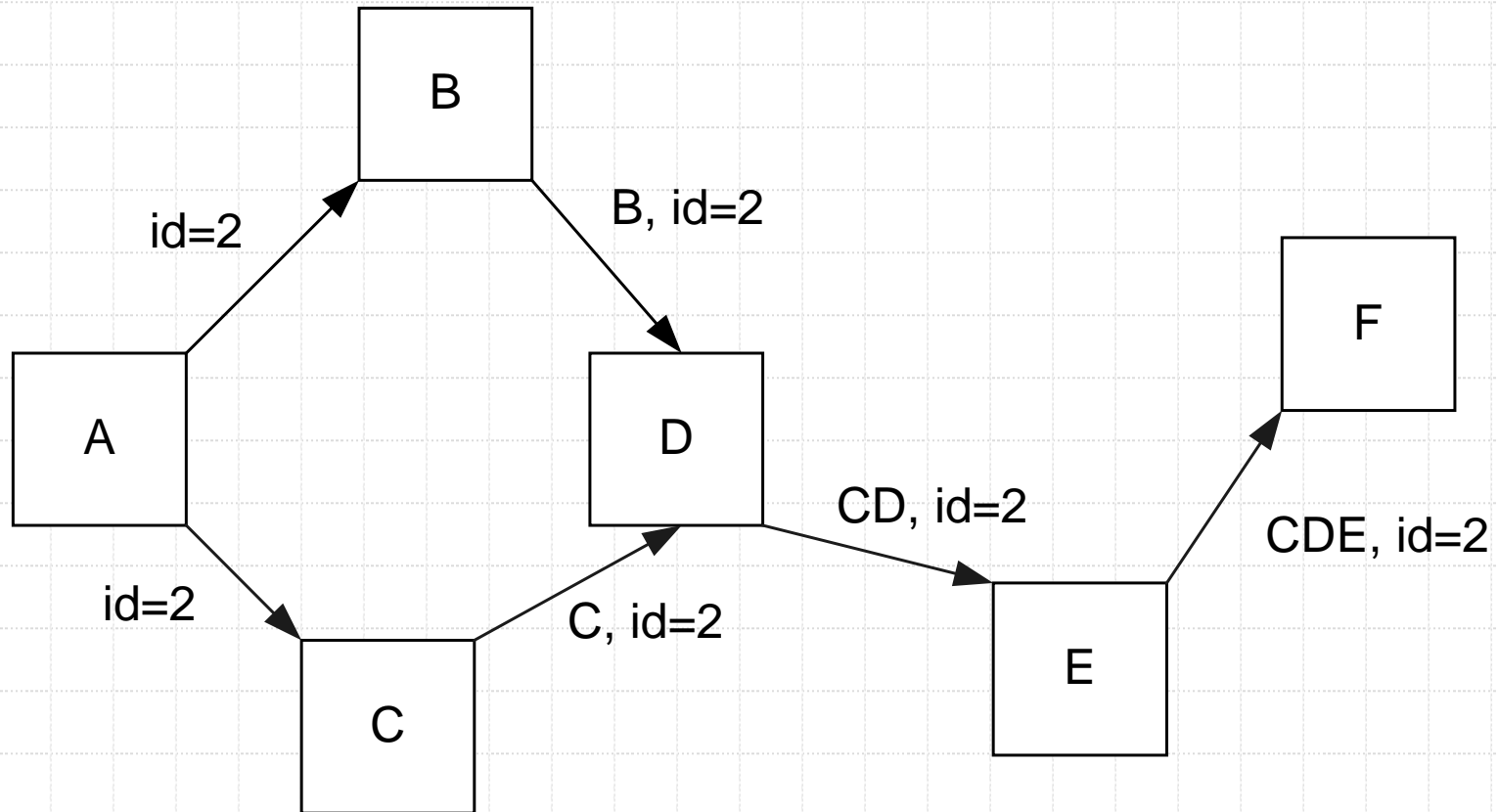
# Why Dynamic Source Routing?

- ◆ Not a complex protocol.
- ◆ Low packet overhead.
- ◆ Routes efficiently.
- ◆ Does not require promiscuous mode and supports uni-directional links.

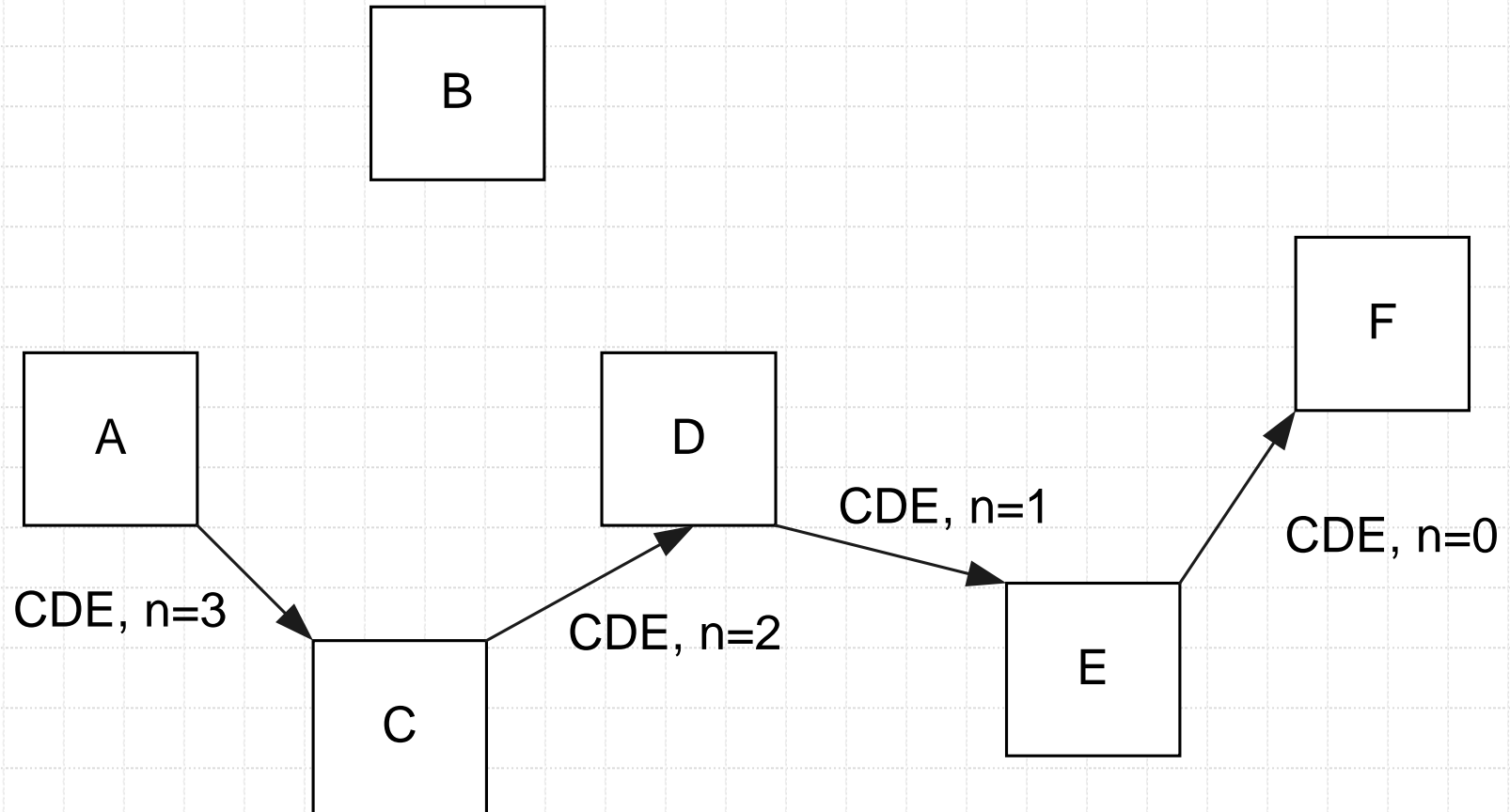
# Dynamic Source Routing

- ◆ Route Discovery.
- ◆ Packet Forwarding.
- ◆ Route Maintenance.

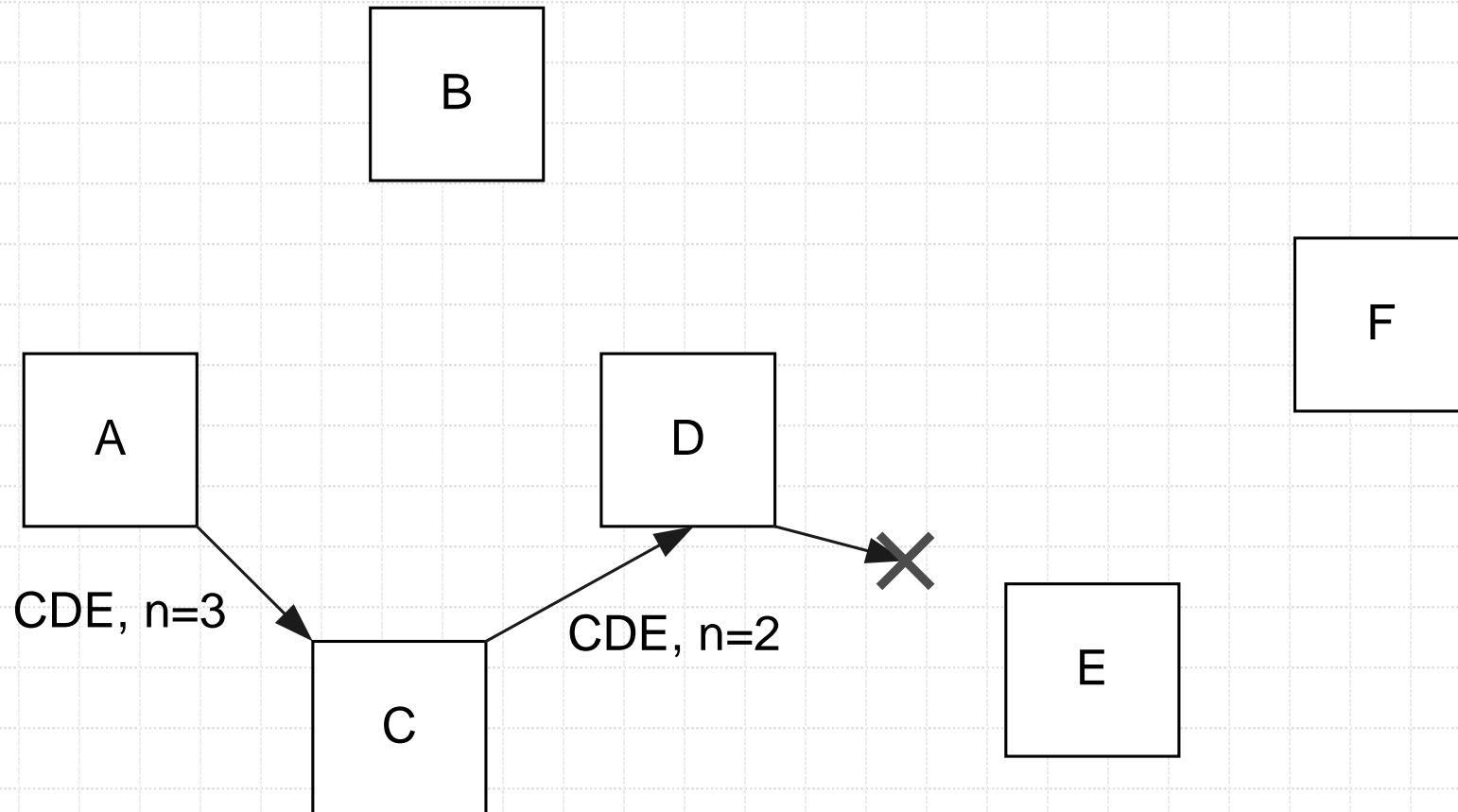
# Route Discovery



# Packet Forwarding



# Route Maintenance

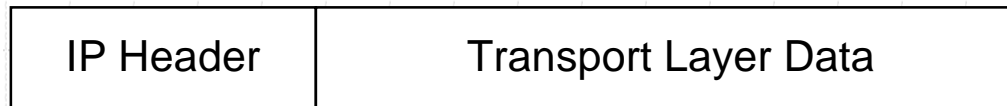


# DSR on IPv4

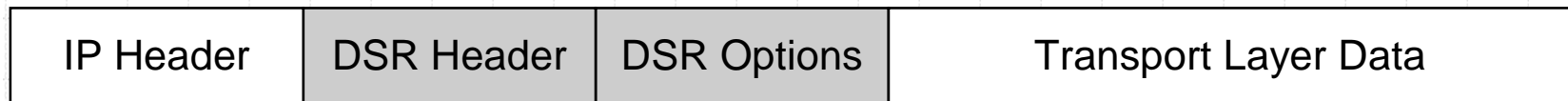
- ◆ IP options field (40 bytes) too small.
- ◆ DSR protocol uses a special header which can be included in any IP packet.
- ◆ The DSR header format is similar to the options format of IPv6.

# DSR Header Format

Standard IP packet



Standard IP packet with DSR options





# DSR Header Format

## IPv4 Header

Version	IHL	Type of Service	Total Length		
Identification			D	M	Fragment Offset
			F	F	
Time To Live	Protocol		Header Checksum		
Source Address					
Destination Address					

## DSR Header

Next Header	Reserved	Payload Length
-------------	----------	----------------

# DSR Options

- ◆ Route Request Option.
- ◆ Route Reply Option.
- ◆ Route Error Option.
- ◆ Acknowledgement Request Option.
- ◆ Acknowledgement Reply Option.
- ◆ Source Route Option.
- ◆ Pad1 and PadN Options.

# DSR Options Format

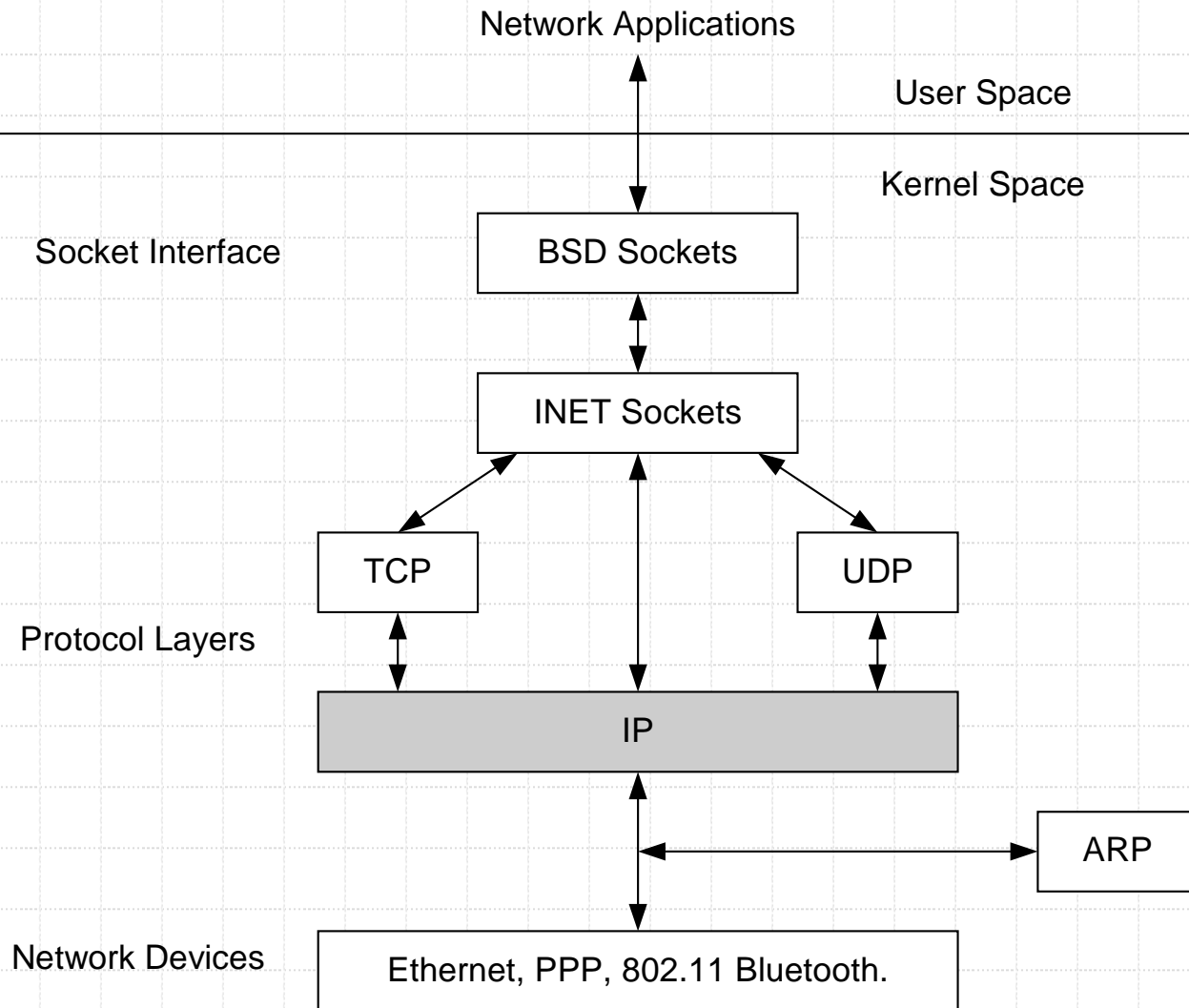
## Route Request Option

Option Type	Opt Data Len	Identification
Target Address		
Address[1]		
...		
Address[n]		

# Why Implement on Linux ?

- ◆ Open Source
  - access to existing TCP/IP code.
- ◆ Available on multiple platforms
  - x86, arm, sparc, powerpc etc.
- ◆ Allows kernel code to be written as modules, which can be inserted and removed at anytime.

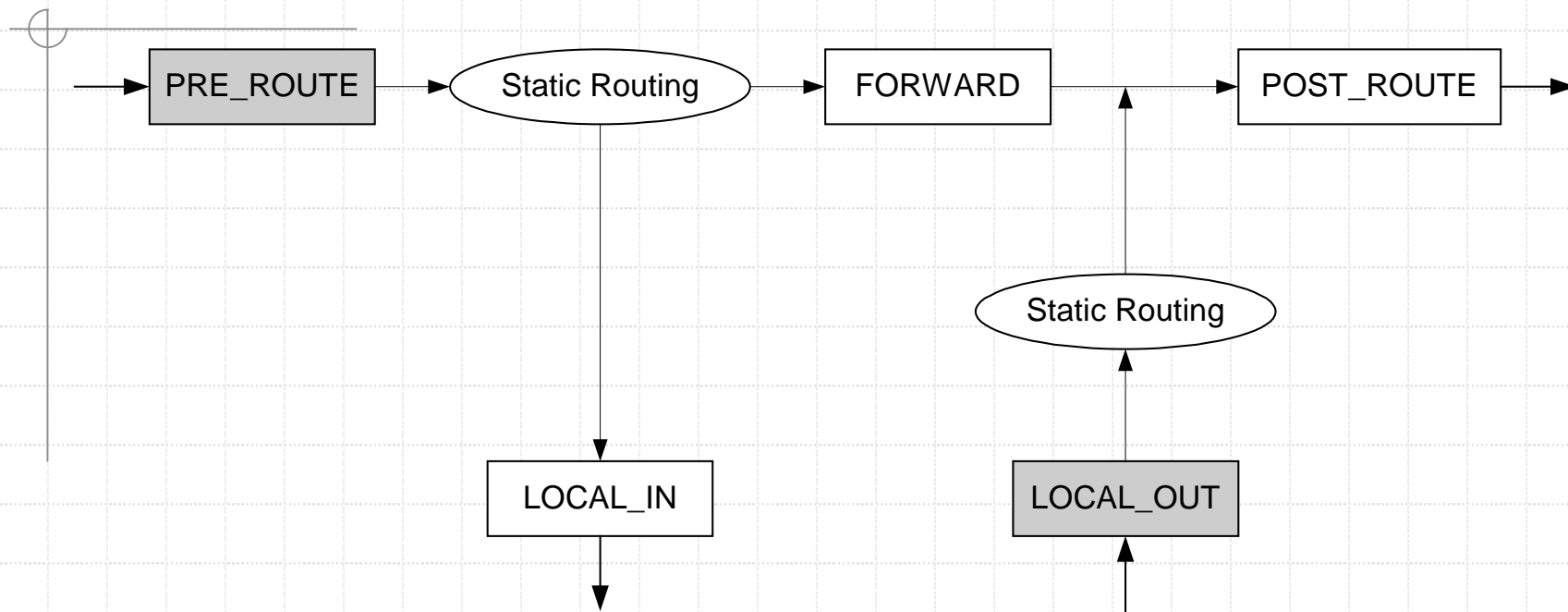
# Linux TCP/IP Network Layers



# Linux Netfilter

- ◆ A framework that allows kernel modules to observe and modify packets as they pass through the IP stack.
- ◆ New architecture, available in Linux 2.4
- ◆ Used for packet filtering, packet mangling and network address translation (NAT).

# Netfilter Architecture



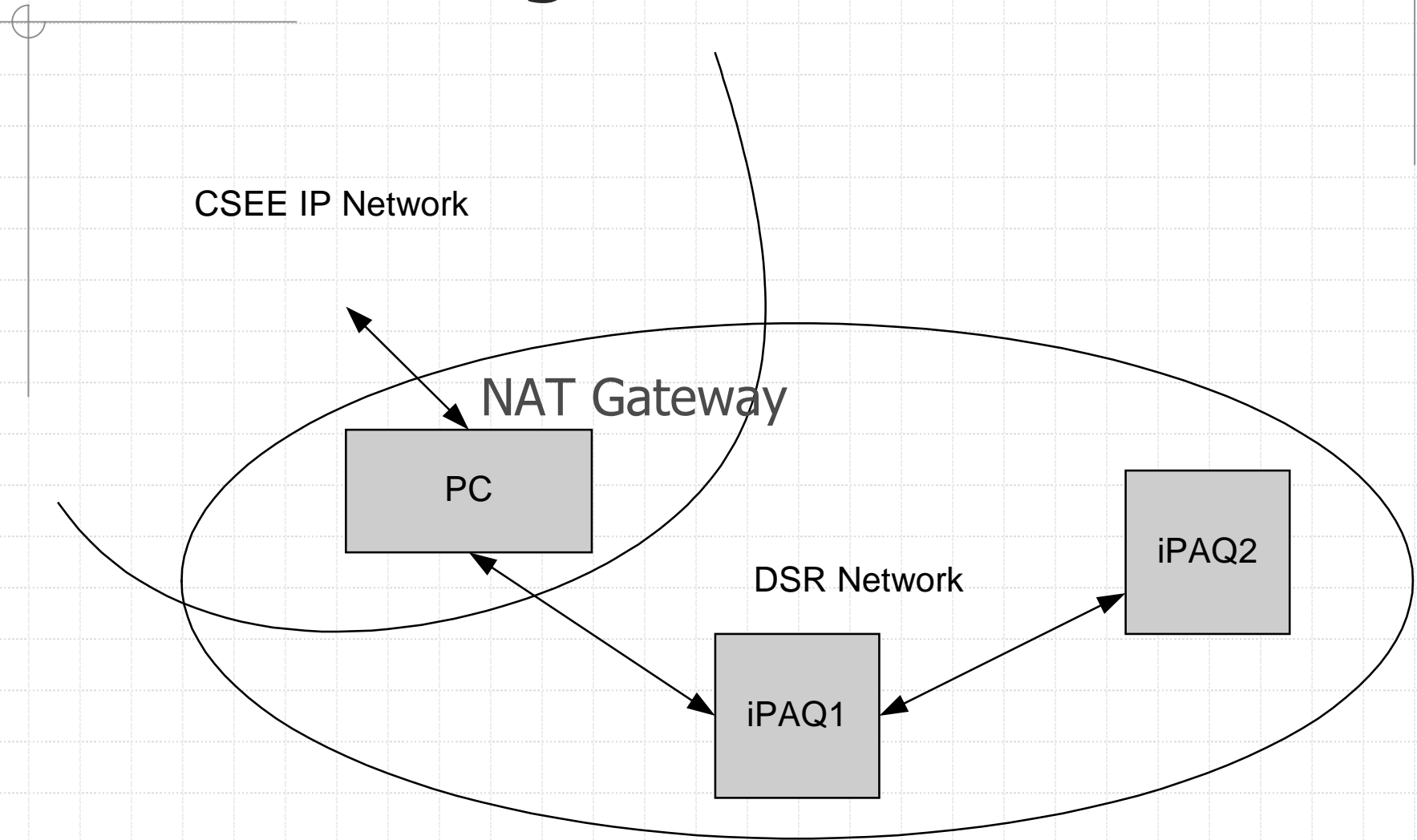
- 5 Hooks defined for IPv4.
- functions register to listen at specific hooks.
- functions are called in order of priority when a packet arrives at a hook.

# Current Progress

- ◆ Written a kernel module.
- ◆ Packet forwarding implemented with static routes.
- ◆ Can forward ICMP and all transport protocols like TCP and UDP.
- ◆ The DSR network can access standard IP networks via a PC gateway running NAT.



# Current Progress



# Features to be implemented

- ◆ Route discovery.
- ◆ Route maintenance.
- ◆ Packet fragmentation.
- ◆ Performance and packet statistics.



Any Questions ?