A Performance Evaluation Framework for Wireless Mesh Routing Protocols

Diplom Thesis - Final Talk

Harald Schiöberg
harald@net.t-labs.tu-berlin.de

Technische Universtität Berlin
Deutsche Telekom Laboratories

26.02.2007

Outline

1 Introduction

Introduction

What are meshes, and why are they interesting? What are the open research topics?

2 Layer 2.5

3 Design of the Framework

4 Implementation

5 Wifi Introduction

What is a mesh?

One definition per researcher, I will stick with this:

Multihop wireless network with multiple possible paths

802.11 distribution system is no mesh

Uses omnidirectional radios

Multiple neighbors per node, typically densely meshed, often 802.11 ad-hoc mode on one frequency

All nodes are equal and might communicate with every other node

Unlike 802.11 managed mode or cell phones

Options (incomplete): Mobility, central management, auto configuration, ad hoc network (don't confuse with ad-hoc mode), point-to-point connections, heterogeneous links

Why do we do mesh research?

Meshes are cool:

- For the market: cheap, ubiquitous broadband access
- For research: Clear, controlled scope gives research opportunities with little interoperability issues

What's so special with meshes?

Transport

Interacting control loops, unknown loss reasons, QoS, different line conditions

Routing

Low usable bandwidth, very dynamic topology, no fixed link properties

MAC

Spatial reuse, power management, collision avoidance, error correction

What do we want?

Testbed for all kind of wireless experiments:

- User behaviour
- Applications
- Transport layer interactions
- Routing protocols
- MAC layer evaluation
- Heterogeneous physical interfaces

Examples for routing protocols: Table based

Ad-hoc on demand distance vector – AODV

1. If you want to transmit packet: flood route request.
2. If you receive route request: answer by routing along rreq path or continue flooding

Optimized link-state routing – OLSR

1. Do link-state routing using periodic LSAs
2. Select Multipoint Relays from your neighbors, so that all 2-hop neighbors are reachable. Designate them in the LSA.
3. Only MPRs forward your LSA.

Examples for routing protocols: Other

Ant based routing

1. Send out ants, ants carry "smell" or "pheromone" of origin
2. If you want to forward packet: random next hop weighted by smell on link

Location based

1. Distribute location of nodes
2. Forward in direction of node

E.g. used in vehicular networks: "Ask any car two kilometers ahead for road condition"

Examples for other things

Transport layer interactions

Mesh gateway-gateway retransmission? Multipath routing?

Use lower layer data

Link quality metrics (Signal strength...), MIMO antenna ratios, ...

Cooperate with uplink routers

What can be done when the uplink router cooperate with the mesh?

Outline

1 Introduction

2 Layer 2.5

Layer 2.5

What is Layer 2.5, and why should we use it? Aren't there enough unused layers?

3 Design of the Framework

4 Implementation

5 Wifi Introduction

IP Routing?

IP addressing interacts with transport layer

Changing IP addresses leads to problems with the transport layer ⇒ Mobile IP ?

Address auto-configuration is difficult with IPv4

Central instances like DHCP-servers are not reachable without routing.

Sophisticated addressing schemes might be necessary

e.g. location based routing

MAC Layer Routing?

Technology dependant

Fixed to one MAC layer technology, open drivers required

Non-extensible headers in IEEE 802 protocols

IEEE 802 protocols have no optional headers, and no loop detection

Jobs of the Layers in the IP Stack

Transport

- Enhance underlaying network (reliability ...) on end-to-end base
- Addressing by ports *plus* IP address

IP

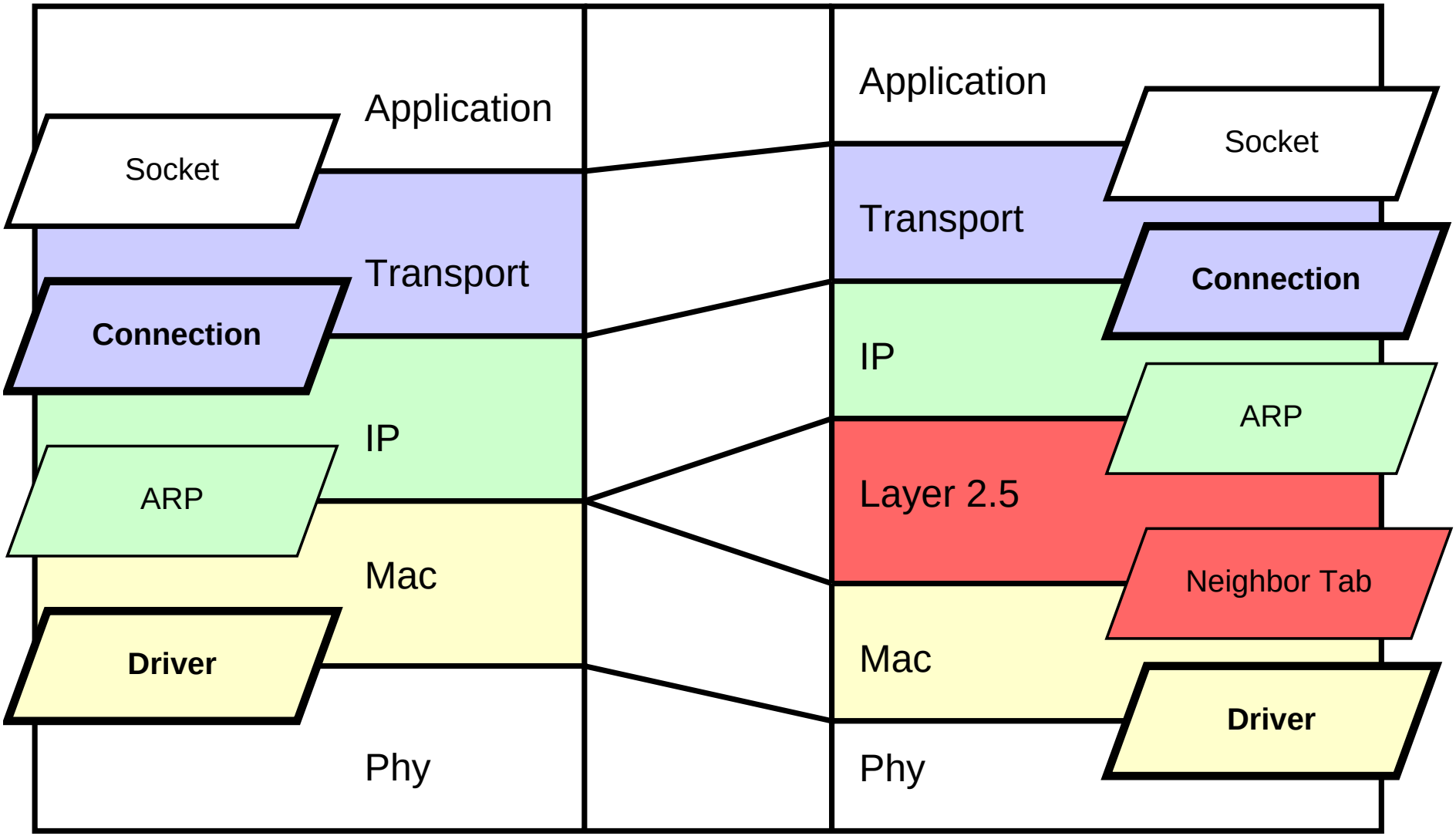
- Worldwide routing
- Packet addressing by network

Link

- Medium access, collision handling
- Frame addressing by station on the same medium

Meshes are **one** network on **multiple** link layer domains ?!

Layer 2.5 Routing !



Tasks of the Layers - revisited

IP

IP routes packet between the different networks, the mesh is one network

Layer 2.5

Layer 2.5 cares for packet handling in the mesh.
for instance: routing, retransmissions, encryption ....

Link Layer

The link layer cares for packet transmission between the individual hops in the mesh

Outline

1 Introduction

2 Layer 2.5

3 Design of the Framework

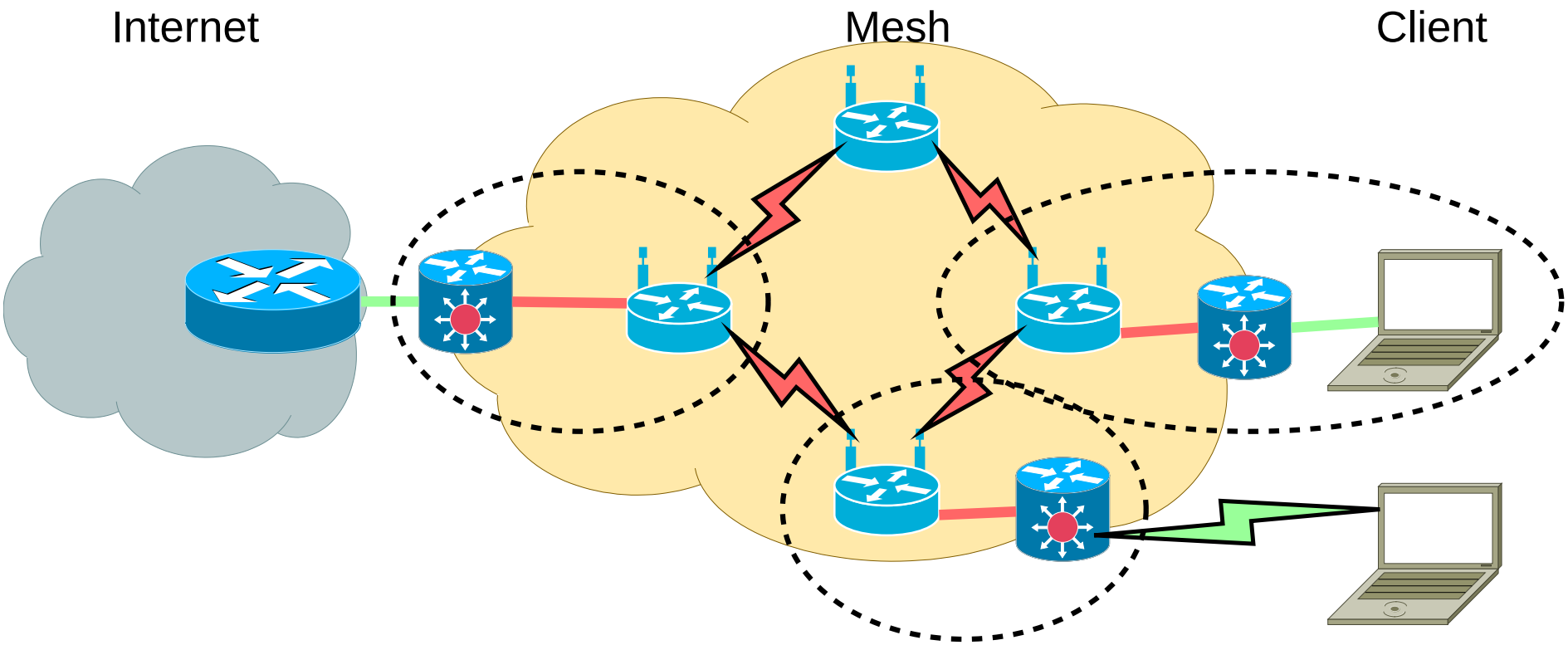
Design of the Framework

What must a framework look like to cope with all that?

4 Implementation

5 Wifi Introduction

Components of a mesh



Packet handling

Programmable packet handler

The packet handling is the core task of a routing framework

- Fully programmable forwarding
- Support for all kinds of routing schemes
- Easy prototyping and highly efficient code support
- Cooperation with IP
- Access to MAC Layer information
- Libraries for standard tasks: addressing, routing tables, queues...

Network management

Prototypes have bugs

No control net $\Rightarrow$ Routing bugs loose the nodes

Software must reach the nodes

Different experiments require different software configurations.

- Hardcoded failover mechanism preserves connectivity
- Software distribution framework
- Reservation system and configuration database

Instrumentation

No measurements without instrumentation

Measurement must not influence device behaviour
(No dedicated network, small CPUs, little storage)

- Local storage of measurement data, local preprocessing and filtering
- Use devices with disk and/or control net access for traces
- Provide analysis tools for Layer 2.5 data (e.g. tcpdump plugins...)

Outline

1 Introduction

2 Layer 2.5

3 Design of the Framework

4 Implementation

Implementation

Bugs anyone?

5 Wifi Introduction

Operating System (too much text for one slide, not worth two)

Open Environment

Almost all parts of the system may need adoption

General Purpose OS

Requirements for instrumentation, measurement, management requires a general purpose OS, despite higher resource usage.

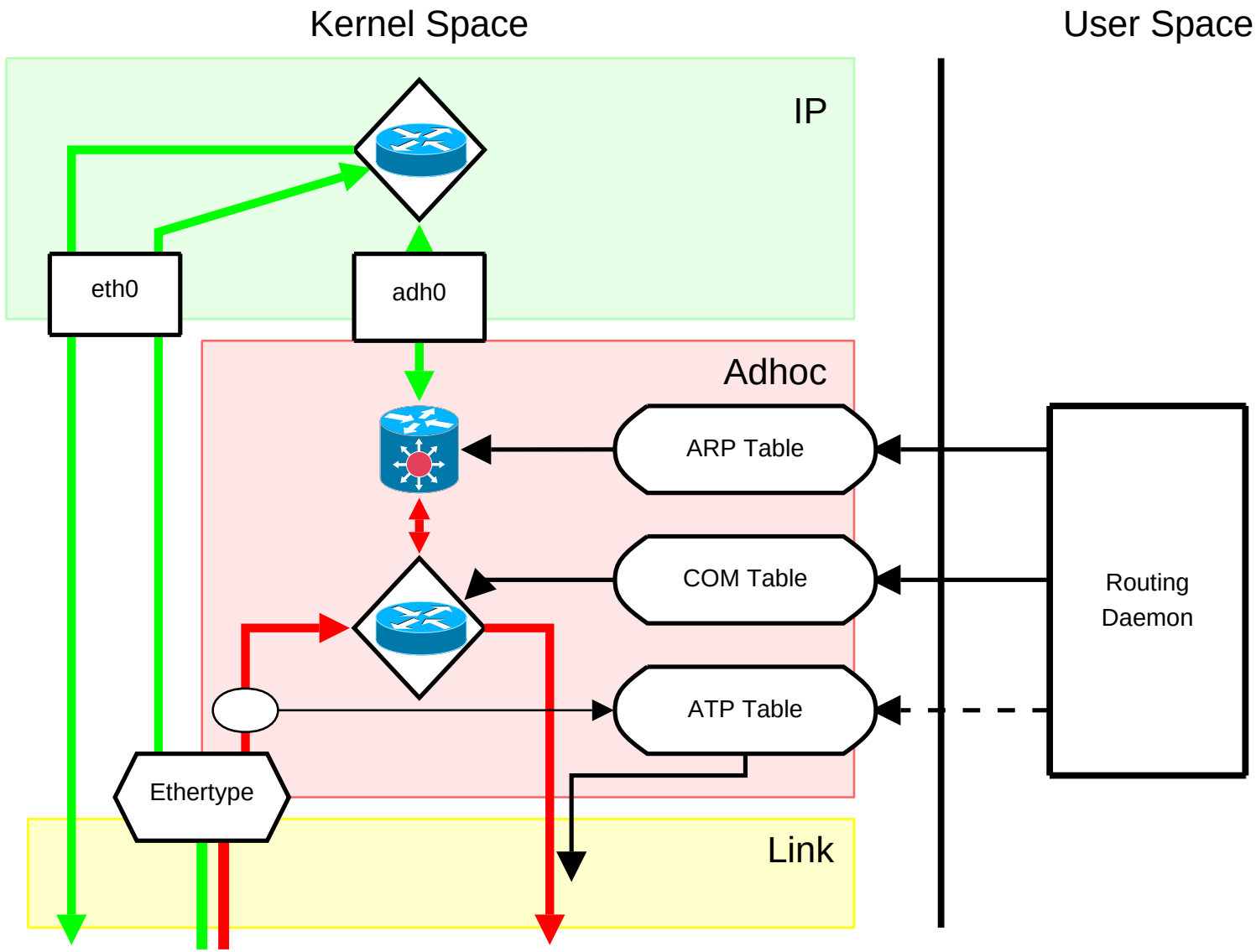
Hardware Support

Different CPUs, different network devices

Embedded Linux: OpenWRT

- Fairly small (Minimum: 4 MB non-volatile storage, 8 MB Ram)
- Dedicated to lightweight routers (originally developed for Linksys WRT54 Series)
- Good hardware support

Ana4 Architecture



Ana4 Features

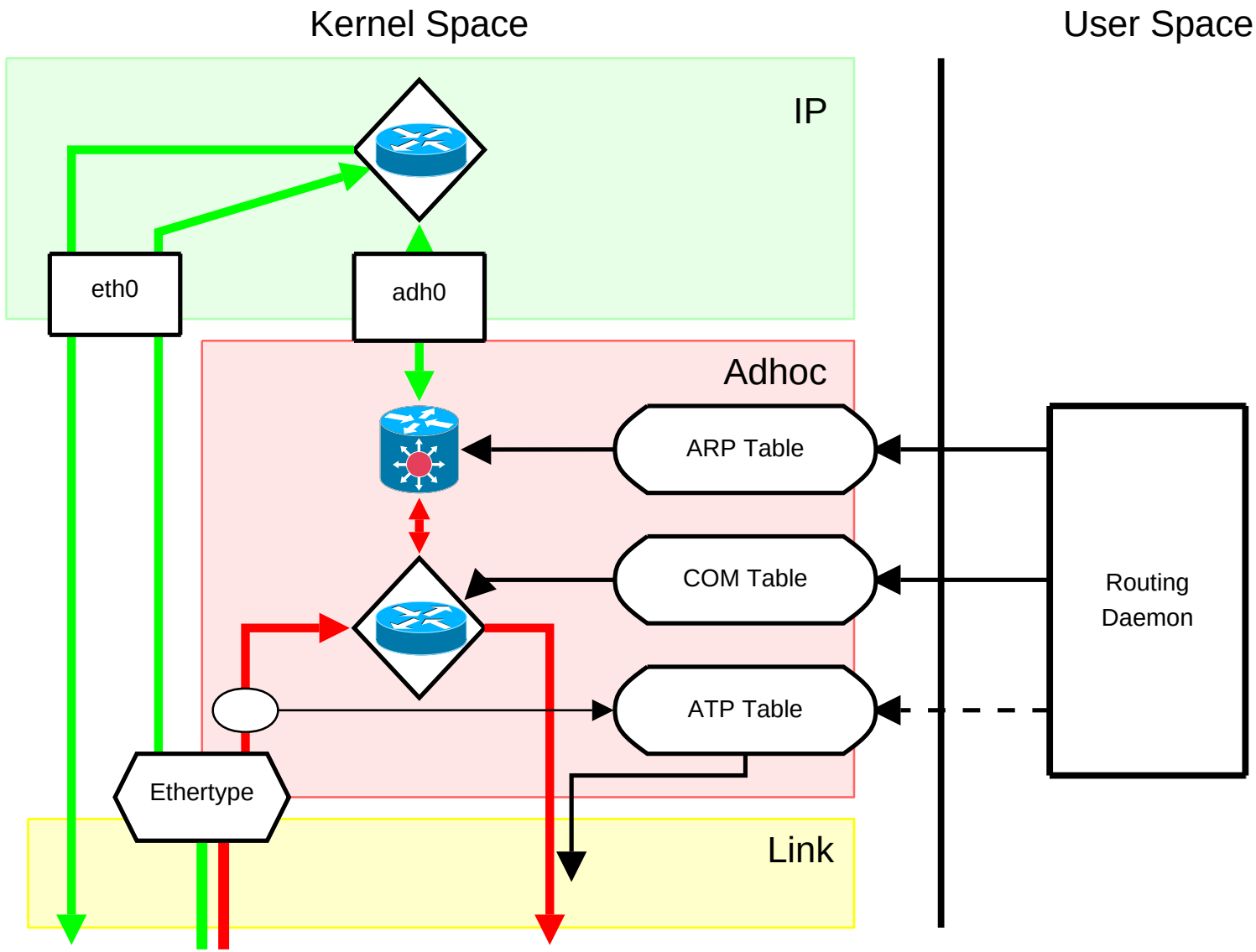
Pro: Complete Layer 2.5 router

Gateway, integration with IP, packet format, addressing ...

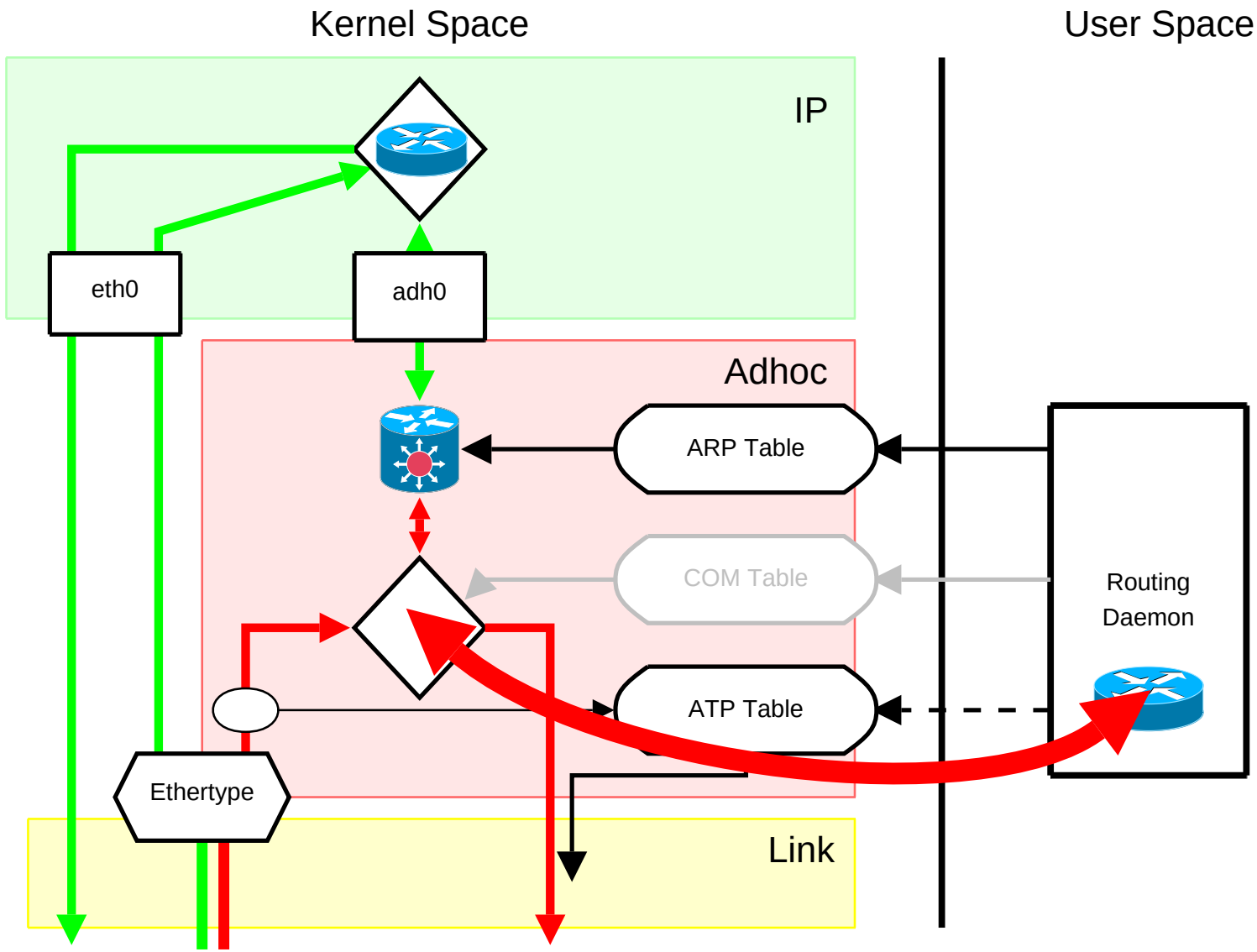
Con: Packet handling is not flexible enough

Only conventional routing table based forwarding

Ana4 Architecture



Ana4 Userspace Router



Ana4 Userspace Features

Packet headers in userspace

All packet headers are transfered to userspace for processing

Efficiency vs. Prototyping

Fast C API with Perl bindings

Ana4 Userspace Problems

Global lock in Ana4

Ana4's implementation holds one giant lock during packet processing, transfer to userspace must release lock

$\Rightarrow$ stability issues

Ana4 development ceased

Ana4 is no longer actively maintained, has no active user base

$\Rightarrow$ no bug fixes, no support

Outlook: The Click Router

Concept: Replace Linux packet handling

- Large library of router components (packet classifiers, queues, packet generators, tables, ARP handlers, IPSec authenticator...) written in C++
- Components are put together at runtime by script
- Runs in (patched) Linux & FreeBSD Kernel, as pcap application and in NS-2

Pro:

Good userbase, active development (4-10 mails/day), very flexible packet handling, lots of platforms

Con:

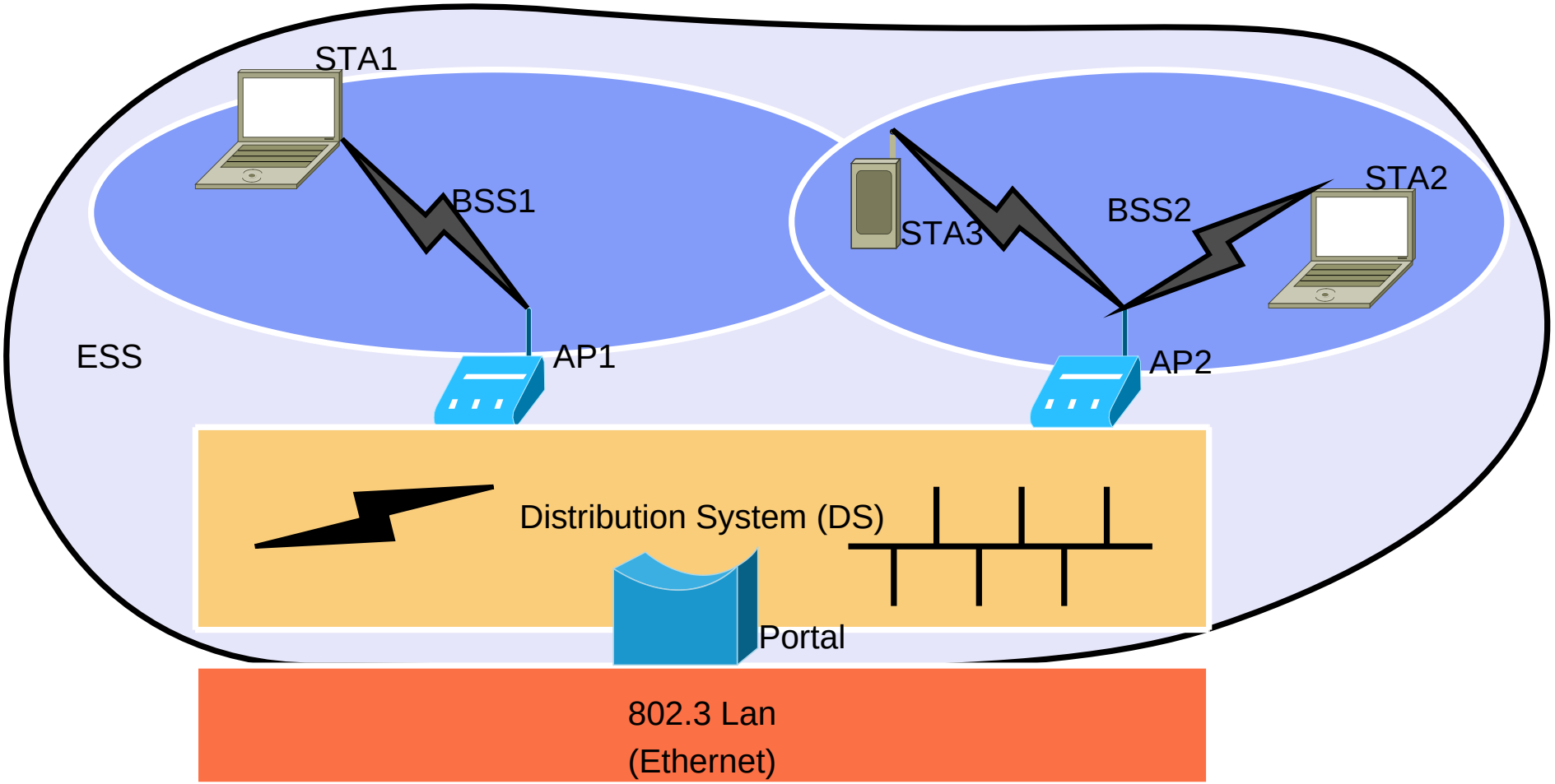
No native Layer 2.5 support, but most elements are there

That's it

Anybody who dares to ask

Questions?

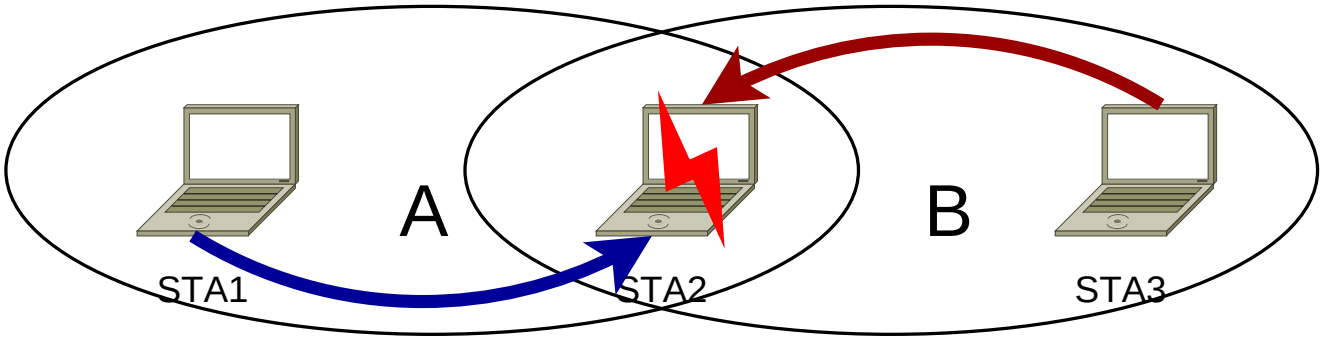
802.11 Components



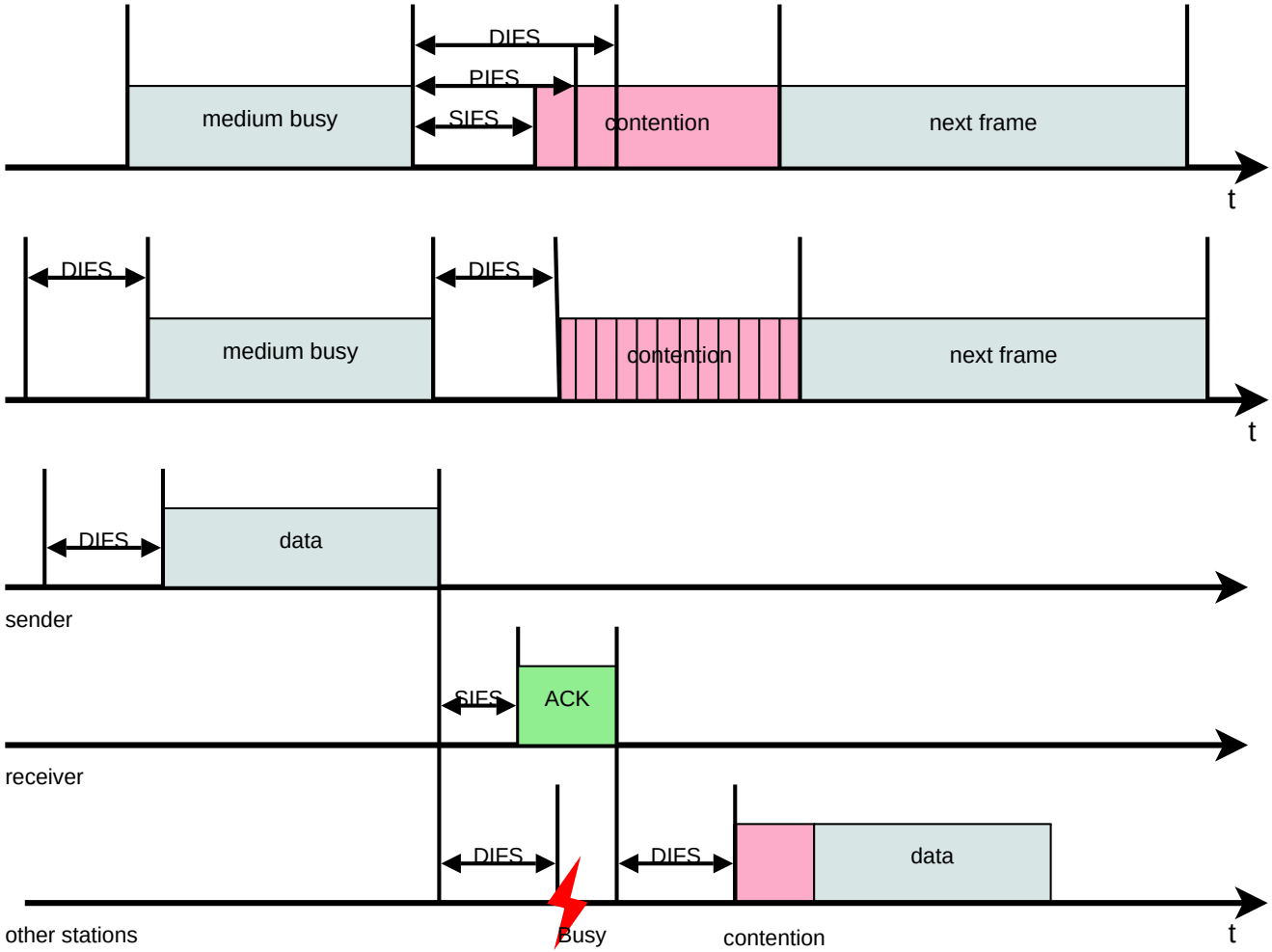
802.11 CSMA/CA Part 1

No collision detection

- Radios can either transmit or receive, no duplex operation with one radio ⇒ No collision detection
- Two stations may be out of range, but their transmission may collide (Hidden terminal problem)



802.11 CSMA/CA Part 2



802.11 CSMA/CA Part 3