Networking for Operating Systems
CS 111
Operating Systems
Peter Reiher

Outline

- Introduction to networking
- Networking implications for operating systems
- Networking and distributed systems

Networking: A Brief History

- In the early 1960s, operating systems rarely had any concern with networks at all
- Today, networking is a core concern of almost all operating systems
- How did we get from there to here?

The Analog Age of Networking

- Point-to-Point connection technology
  - Lay or lease (analog) dedicated lines
    - Limited connectivity, very expensive, special purpose hardware
  - Use the (analog) telephone network
    - Limited bandwidth, intermittent connectivity, primitive modems
- Services
  - Remote device connection
    - Remote terminal (dial-in access)
    - Remote card readers and printers (for job submission)
    - Remote instrumentation (attached to phones)
  - Computer-to-computer communication

The ARPANET

- Based on a dedicated sub-network
  - Special purpose Interface Message Processors
  - Interconnected by 56KB leased lines
  - Packet switched (vs. circuit switched) communication
  - Automatic routing (negotiated among the IMPs)
- Host computers saw a digital network
  - Host-to-IMP interconnection was digital
  - Packet routing and delivery was automatic
  - Continuous connectivity between all network hosts
- First message sent in 1969
- Modest, but increasing deployment by early 1970s

Early Data Protocols

- Goals
  - Enable exploitation of networked computer resources
- Remote access protocols
  - BBN report 1822 (interconnection of a host & IMP)
  - telnet (1969, RFC #15)
  - Remote Job Submission (1971, RFC #88)
  - File Transfer Protocol (1971, RFC #114)
- Impact:
  - Got researchers working on digital networking
  - Led to development of collaborative protocols
    - mail (1972, RFC #385)
    - voice (1977, RFC #741)

Work Group Computing

- Goals:
  - Enhanced collaboration (e-mail, calendars, files)
  - Sharing expensive resources (printers, large disks)
- Peer-to-peer resource sharing
  - Machines export resources for use by the group
  - Users send requests to owners of desired resources
  - Little/no centralization of resources or services
- Impact:
  - Challenged notion of the self-contained system
  - Introduced global resource/authentication domains
- Primarily supporting single enterprise

Client/Server Computing

- Goals:
  - Cost-effective resource & service concentration
  - Centralized system management
  - Larger scale shared resource domains
- Extended peer-to-peer resource sharing
  - Discovery, configuration, authentication, etc.
- Impact:
  - Fat servers and thin clients
  - Ubiquitous standards, high interoperability
  - Assumed availability of network infrastructure
  - Major changes to OS structure and philosophy

The World Wide Web

- The technological innovations were simple
  - HTTP – anonymous file transfer with caching
  - HTML – a “mark-up” language with external links
  - www – a “Universal Resource Locator” namespace
- The implications changed the technological world
  - All systems are now fully inter-connected
  - People buy services, software is just an implementation
  - Services are provided over networks, via protocols
  - Heterogeneity (of hardware, OS, software) is a given

Distributed Computing

- A model where some or almost all of the computation occurs on multiple machines
- Becoming increasingly important
  - Why?
    - Most of the interesting resources are elsewhere
    - A single system has very limited capacity & bandwidth
    - A single system is a single point of failure
- Rejects the old model
  - Software runs on the local CPU, under the local OS
  - Some resources may be fetched over a network
- Instead, the network is the computer
  - The local CPU and OS are merely a point of access

Networking Implications for the Operating System

- Increasing amounts of activity will require networking
- Handling networking well will become ever more critical
- The operating system must be better at handling the special characteristics of networks
- Not just another peripheral device
- Instead, the key demand on future systems

Changing Paradigms

- Network connectivity becomes “a given”
  - New applications assume/exploit connectivity
  - New distributed programming paradigms emerge
  - New functionality depends on network services
- Thus, applications demand new services from the OS:
  - Location independent operations
  - Rendezvous between cooperating processes
  - WAN scale communication, synchronization
  - Support for splitting and migrating computations
  - Better virtualization services to safely share resources
  - Network performance becomes critical

The Old Networking Clients

- Most clients were basic networking applications
  - Implementations of higher level remote access protocols
    - telnet, FTP, SMTP, POP/IMAP, network printing
  - Occasionally run, to explicitly access remote systems
  - Applications specifically written to network services
- OS provided transport level services
  - TCP or UDP, IP, NIC drivers
- Little impact on OS APIs
  - OS objects were not expected to have network semantics
  - Network apps provided services, did not implement objects

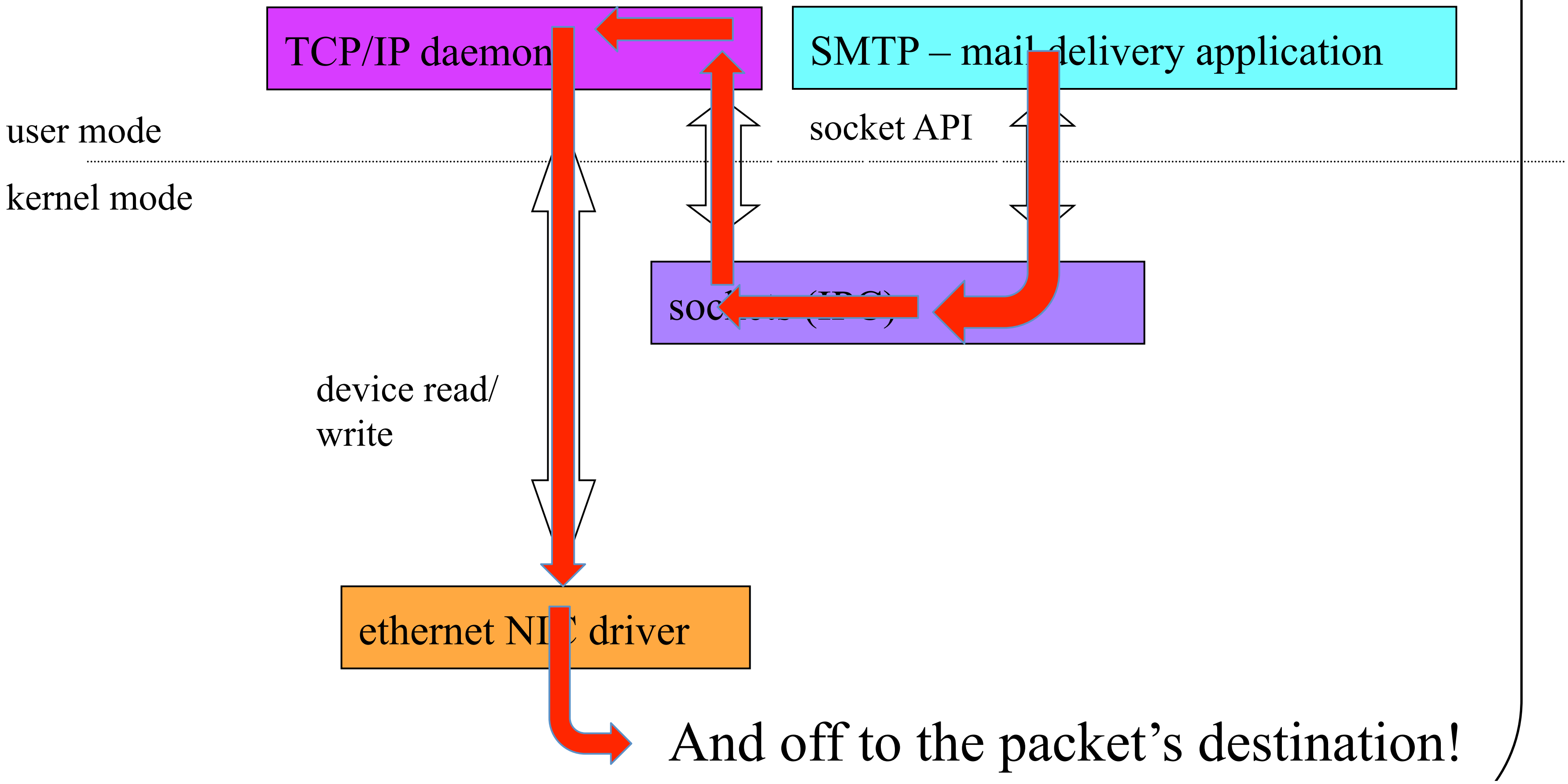
The New Networking Clients

- The OS itself is a client for network services
  - OS may depend on network services
    - netboot, DHCP, LDAP, Kerberos, etc.
  - OS-supported objects may be remote
    - Files may reside on remote file servers
    - Console device may be a remote X11 client
    - A cooperating process might be on another machine
- Implementations must become part of the OS
  - For both performance and security reasons
- Local resources may acquire new semantics
  - Remote objects may behave differently than local

The Old Implementations

- Network protocol implemented in user-mode daemon
  - Daemon talks to network through device driver
- Client requests
  - Sent to daemon through IPC port
  - Daemon formats messages, sends them to driver
- Incoming packets
  - Daemon reads from driver and interprets them
  - Unpacks data, forward to client through IPC port
- Advantages – user mode code is easily changed
- Disadvantages – lack of generality, poor performance, weak security

User-Mode Protocol Implementations



TCP/IP daemon

SMTP – mail delivery application

user mode

kernel mode

socket API

sockets (IPC)

device read/ write

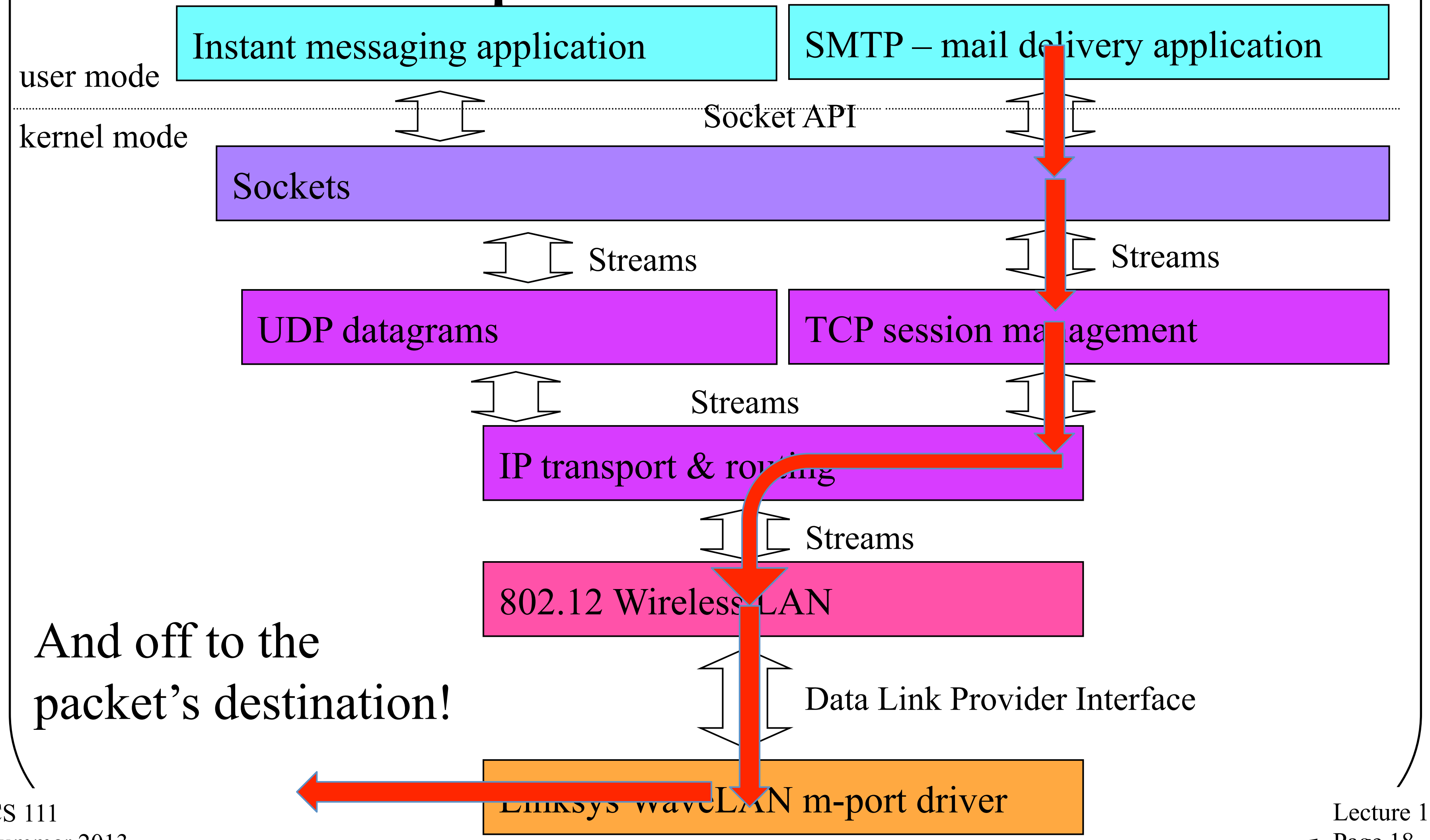
ethernet NIC driver

And off to the packet's destination!

The New Implementations

- Basic protocols implemented as OS modules
  - Each protocol implemented in its own module
  - Protocol layering implemented with module plumbing
  - Layering and interconnections are configurable
- User-mode clients attach via IPC-ports
  - Which may map directly to internal networking plumbing
- Advantages
  - Modularity (enables more general layering)
  - Performance (less overhead from entering/leaving kernel)
  - Security (most networking functionality inside the kernel)
- A disadvantage – larger, more complex OS

In-Kernel Protocol Implementations



user mode

kernel mode

Instant messaging application

SMTP – mail delivery application

Socket API

Sockets

Streams

Streams

UDP datagrams

TCP session management

Streams

IP transport & routing

Streams

802.12 Wireless LAN

Data Link Provider Interface

Linksys WaveLAN m-port driver

And off to the packet's destination!

IPC Implications

- IPC used to be occasionally used for pipes
  - Now it is used for all types of services
    - Demanding richer semantics, and better performance
- Used to interconnect local processes
  - Now it interconnects agents all over the world
    - Need naming service to register & find partners
    - Must interoperate with other OSes IPC mechanisms
- Used to be simple and fast inside the OS
  - We can no longer depend on shared memory
  - We must be prepared for new modes of failure

Improving Our OS Plumbing

- Protocol stack performance becomes critical
  - To support file access, network servers
- High performance plumbing: UNIX Streams
  - General bi-directional in-kernel communications
    - Can interconnect any two modules in kernel
    - Can be created automatically or manually
  - Message based communication
    - Put (to stream head) and service (queued messages)
    - Accessible via read/write/putmsg/getmsg system calls

Network Protocol Performance

- Layered implementation is flexible and modular
  - But all those layers add overhead
    - Calls, context switches and queuing between layers
    - Potential data recopy at boundary of each layer
  - Protocol stack plumbing must also be high performance
    - High bandwidth, low overhead
- Copies can be avoided by clever data structures
  - Messages can be assembled from multiple buffers
    - Pass buffer pointers rather than copying messages
    - Network adaptor drivers support scatter/gather
- Increasingly more of the protocol stack is in the NIC

Implications of Networking for Operating Systems

- Centralized system management
- Centralized services and servers
- The end of “self-contained” systems
- A new view of architecture
- Performance, scalability, and availability
- The rise of middleware

Centralized System Management

- For all computers in one local network, manage them as a single type of resource
  - Ensure consistent service configuration
  - Eliminate problems with mis-configured clients
- Have all management done across the network
  - To a large extent, in an automated fashion
  - E.g., automatically apply software upgrades to all machines at one time
- Possibly from one central machine
  - For high scale, maybe more distributed

Benefits of Central Management

- Zero client-side administration
  - Plug in a new client, and it should just work
    - Since everything it needs to get going will be automatically delivered over the network
  - Reduced (per client) costs of support
    - Since all management info is centralized, rarely have to manually examine a client machine
- Uniform & ubiquitous computer services
  - All data and services available from all clients
  - Global authentication and resource domain
- Security benefits
  - All important security patches get applied with certainty
  - Individual users can’t screw up their machine’s security

Dangers of Central Management

- Screw-ups become ubiquitous
- Loss of local autonomy for users
- Administrators gain extreme power
  - So you’d better be sure they’re trustworthy and competent
- Security disadvantages
  - All machines are arbitrarily reconfigurable from remote sites
  - Encourages monocultures, which are susceptible to malware

Centralized Services and Servers

- Networking encourages tendency to move services from all machines to one machine
  - E.g. file servers, web servers, authentication servers
- Other machines can access and use the services remotely
  - So they don't need local versions
  - Or perhaps only simplified local versions

Benefits of Service Centralization

- Quality and reliability of service
  - "Guaranteed" to be up 24x7
  - Performance monitored, software kept up-to-date
  - Regular back-ups taken
- Price performance
  - Powerful servers amortized over many clients
- Ease of use
  - No need to install and configure per client services
  - Services are available from any client
- Allows thinner, cheaper clients
  - Or allows existing clients to devote resources to their users

Dangers of Centralized Services

- Forces reliance on networking
  - Which is “almost always” available, but . . .
  - Makes network congestion more likely
- Makes per-user customization harder
  - Sometimes that’s a good thing, though
- From a security perspective, one big fat target
  - As opposed to lots of little skinny targets
  - But automation of attacks makes this less important
- Can lead to huge privacy breaches

The End of Self Contained Systems

- Years ago, each computer was nearly totally self-sufficient
- Maybe you got some data from some other machine
- Or used some specialized hardware on one machine
- Or shared a printer over the network
- But your computer could do almost all of what you wanted to do, on its own

Now Vital Services Provided Over the Network

- Authentication
  - Active Directory, LDAP, Kerberos, …
- Configuration and control
  - Active Directory, LDAP, DHCP, CIM/WBEM, SNMP, …
- External data services
  - CIFS, NFS, Andrew, Amazon S3, …
- Remote devices
  - X11, web user interfaces, network printers
- Even power management, bootstrap, installation
  - vPro, PXE boot, bootp, live CDs, automatic s/w updates

Benefits of Losing Self-Sufficiency

- Remote specialized servers often do the job better
- Your machine doesn't need to pay the costs of doing the work itself
- Advantages of centralized administration
- Generally possible if any networking available
  - And, for modern use, relatively little is possible when networking isn't available, anyway

Dangers of Losing Self Sufficiency

- Your device is a brick without connectivity
- Your security depends on the security of many others
- Worse, your privacy is dependent on a bunch of service providers
  - In many cases, their business model is using your information . . .
- Harder, maybe impossible, to customize services to your needs