Programming End-to-End Protocols with Socket API

Hui Chen

Department of Computer & Information Science

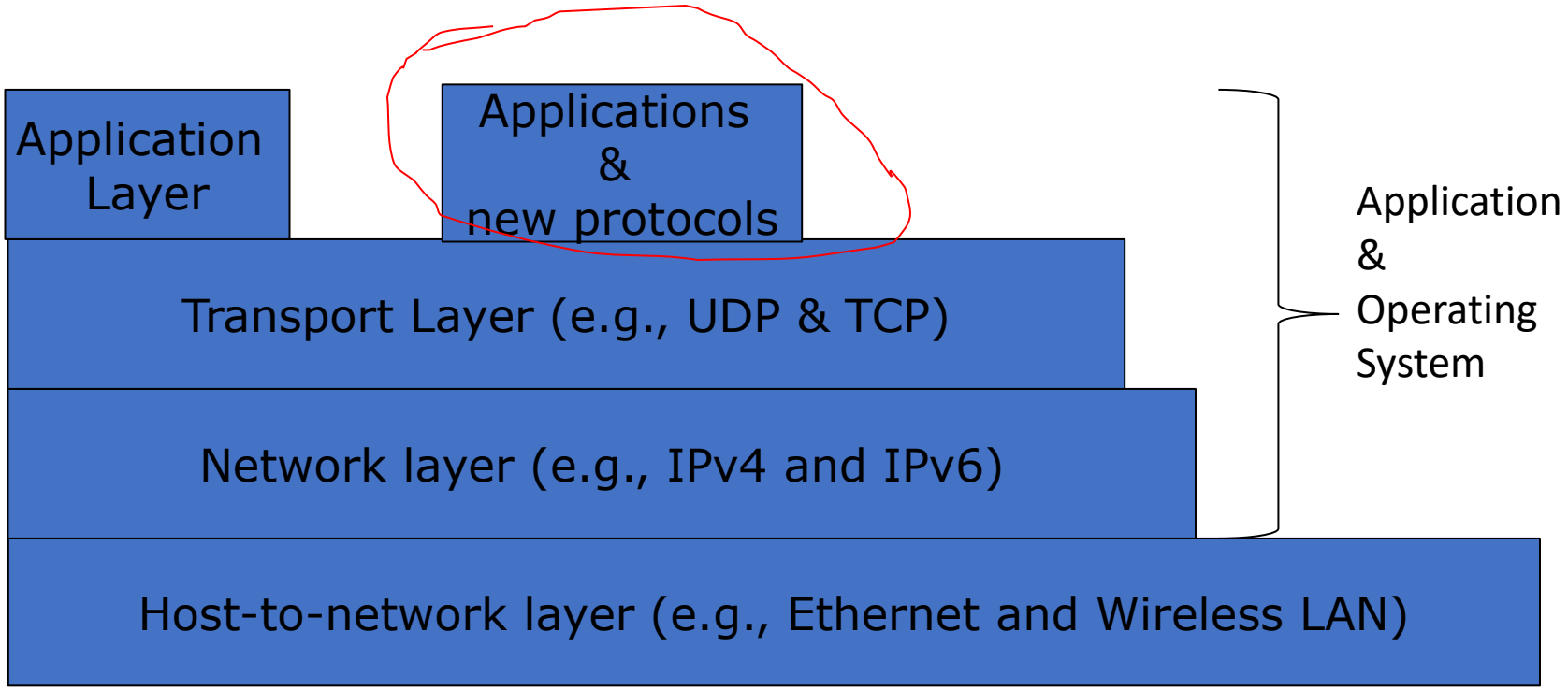
CUNY Brooklyn College

Outline

- Socket programming with Internet
  - Berkeley Sockets
    - udp(7)
    - tcp(7)
  - Example programs

Internet Protocol: Programming

- How to access functionality provided by Transport Layer Protocols and to build new applications?



Working with Ethernet using Berkeley Socket API

- Interested in two domains
  - AF_INET, see Linux manual page, ip(7)
  - AF_INET6, see Linux manual page, ipv6(7)
- Examples
  - Creating a socket
  - Sending messages
  - Receiving messages

Communication Domain

- int socket(**int domain**, int type, int protocol)
- Our interests
  - AF_INET
    - See ip(7) for more information
  - AF_INET6
    - See ipv6(7) for more information

Communication Type

- int socket(int domain, **int type**, int protocol)
- Specify a communication semantics with a communication domain
- For AF_INET and AF_INET6
  - SOCK_RAW
    - For raw IP packets (including the link level header)
  - SOCK_DGRAM
    - For UDP
  - SOCK_STREAM
    - For TCP

Protocol

- int socket(int domain, int type, **int protocol**)
- Specifies a particular protocol to be used with the socket.
- Protocol is a protocol number in network order
- If type is SOCK_RAW
  - A valid IANA IP protocol (defined in RFC 1700/RFC 3232) assigned numbers.
  - See
    - Typical location: /usr/include/netinet/in.h

Sending Messages

- ssize_t send(int sockfd, const void *buf, size_t len, int flags);
- ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *dest_addr, socklen_t addrlen);
- ssize_t sendmsg(int sockfd, const struct msghdr *msg, int flags);
- ssize_t write(int fd, const void *buf, size_t count);

Sending Messages

- For more see send(2), sendto(2), sendmsg(2), and write(2)

Sending Message

- Relationship among the system calls
  - write(fd, buf, len);

    is equivalent to

    send(sockfd, buf, len, 0);
  - send(sockfd, buf, len, flags);

    is equivalent to

    sendto(sockfd, buf, len, flags, NULL, 0);
  - write(fd, buf, len);

    is equivalent to

    sendto(sockfd, buf, len, 0, NULL, 0);

Sending Messages

- ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *dest_addr, socklen_t addrlen);
  - sockfd: the file descriptor of the sending socket
  - buf: message to send
  - len: message len
  - flags: the bitwise OR of flags or 0
  - dest_addr: the address of the target
  - addrlen: the size of the target address

Receiving Messages

- ssize_t recv(int sockfd, void *buf, size_t len, int flags);
- ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen);
- ssize_t recvmsg(int sockfd, struct msghdr *msg, int flags);
- ssize_t write(int fd, const void *buf, size_t count);

Receiving Message

- For more information, see recv(2), recvfrom(2), recvmsg(2), and read(2)

Receiving Message

- Relationship among the system calls
  - read(fd, buf, len);

    is equivalent to

    recv(sockfd, buf, len, 0);
  - recv(sockfd, buf, len, flags);

    is equivalent to

    recvfrom(sockfd, buf, len, flags, NULL, NULL);
  - read(fd, buf, len);

    is equivalent to

    recvfrom(sockfd, buf, len, 0, NULL, NULL);

Putting Together

- See sample programs

Unicast vs. Multicast/Broadcast

- TCP
  - Unicast
- UDP
  - Unicast
  - Multicast
  - Broadcast
- How?

Summary

- Application and new protocol implementation
  - Berkeley Socket APIs
    - AF_INET
    - AF_INET6