UDP Sockets

Giảng viên: TS. Trần Hải Anh

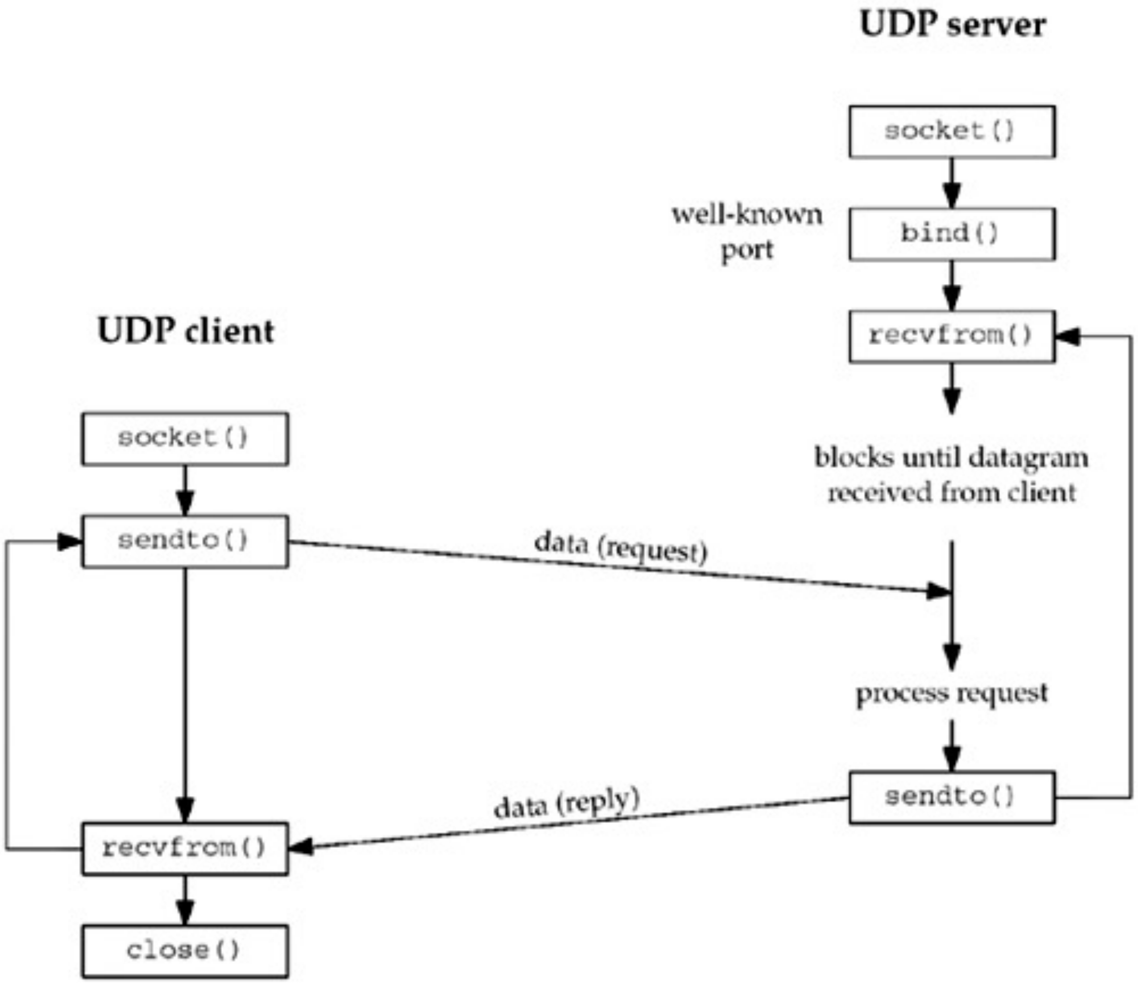
Bộ môn Truyền Thông & Mạng máy tính

Khoa CNTT & TT

UDP Protocol

- connectionless, unreliable, datagram protocol
- Some popular applications are built using UDP: DNS, NFS, and SNMP, ...

Intro


UDP server
socket()
well-known port
bind()
recvfrom()
blocks until datagram received from client
UDP client
socket()
sendto()
data (request)
process request
sendto()
data (reply)
recvfrom()
close()

recvfrom and *sendto* Functions

```
#include <sys/socket.h>
ssize_t recvfrom(int sockfd, void *buff, size_t
nbytes, int flags, struct sockaddr *from,
socklen_t *addrlen);
ssize_t sendto(int sockfd, const void *buff,
size_t nbytes, int flags, const struct sockaddr
*to, socklen_t addrlen);
```

- Both return: number of bytes read or written if OK, –1 on error
- Writing a datagram of length 0 is acceptable.

Chương trình ví dụ:

udp-client.c

udp-server.c

Verifying Received Response with `memcmp()`

- UDP is a connectionless protocol →we need a mechanism to verify the sender
- →Use the function `memcmp()`

Chương trình ví dụ: **mem-check-client.c**

UDP with `connect()`

- UDP is a connectionless protocol
  →the connect() function might seem unnecessary
- However, a few useful purposes:
  - Simplified Sending and Receiving
    - When you call connect() on a UDP socket, the system "associates" the socket with the specified remote address. After that, you can use send() and recv() instead of sendto() and recvfrom().
  - Enforcing Remote Peer
    - the socket is "locked" to the specified remote IP address and port.
  - Error Handling (ICMP Feedback)
  - Potential performance improvements

Chương trình ví dụ: **udp-connect-client.c**