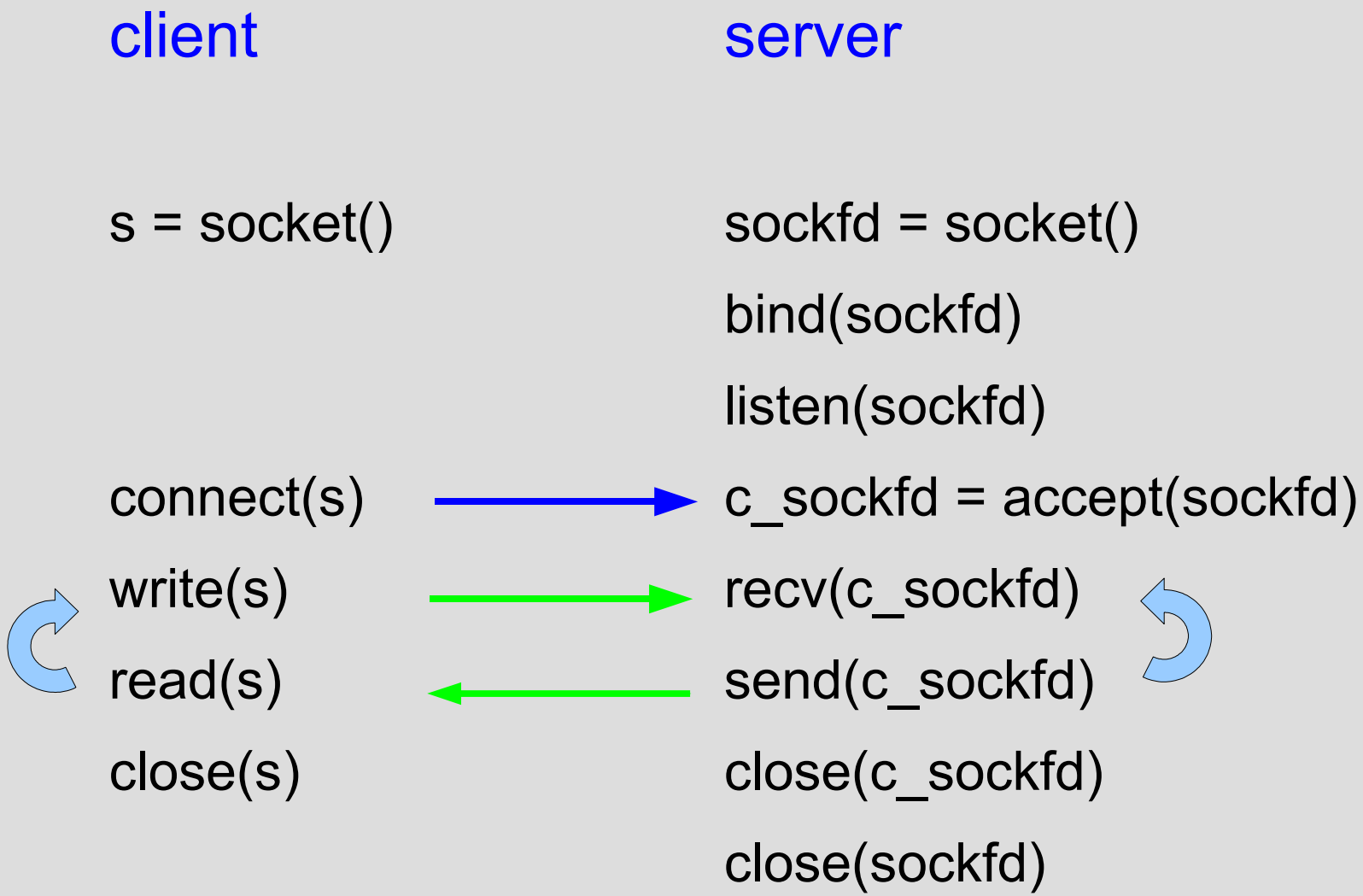
# X36PKO
# TCP/IP rozhraní socketů

# TCP - C

- sockfd = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP));
- bind(sockfd, &my_addr, sizeof(my_addr));
- listen(sockfd, 5);
- connect(s, (struct sockaddr *)&sin, sizeof(sin));
- rem_addr_length = sizeof(rem_addr);
- c_sockfd = accept(sockfd, &rem_addr, &rem_addr_length));
- mlen = recv(c_sockfd, buf, BUFFSIZE, 0);
- send(c_sockfd, buf, mlen, 0);
- close(c_sockfd);
- close(sockfd);

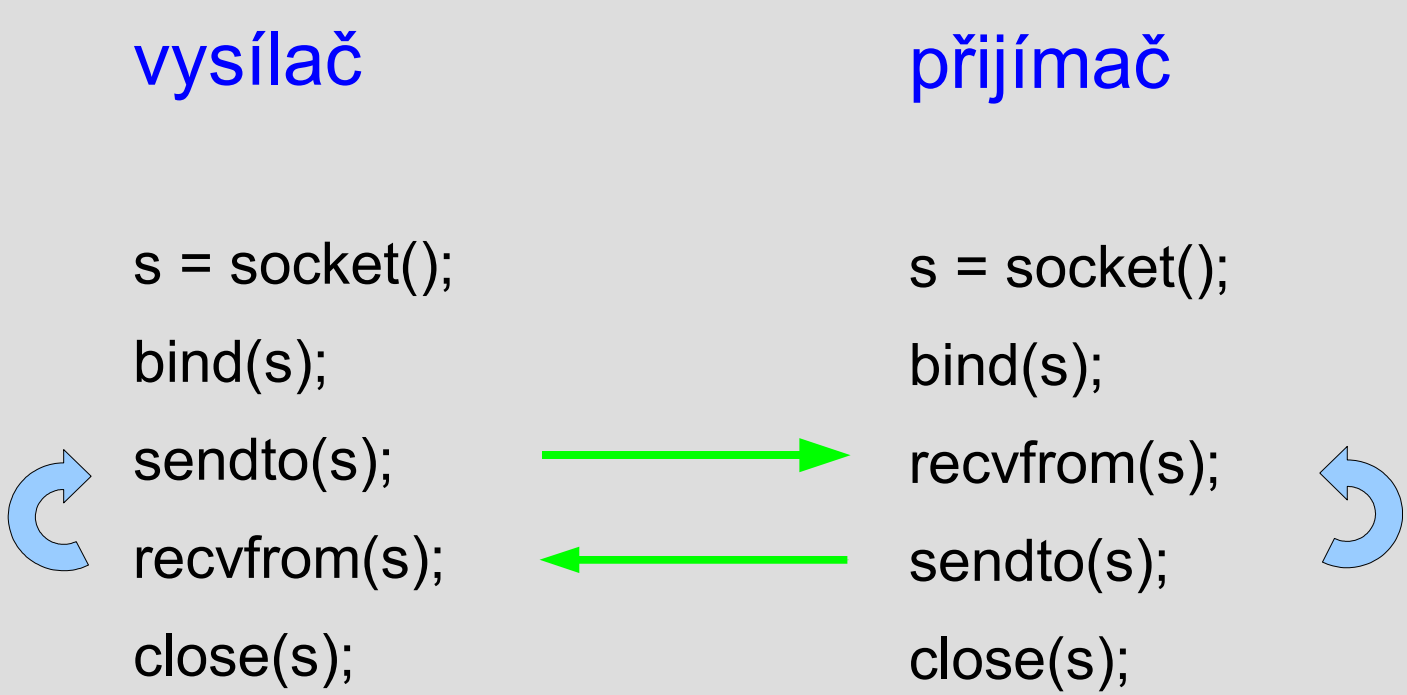
# TCP – C


client
server
s = socket()
sockfd = socket()
bind(sockfd)
listen(sockfd)
connect(s)
c_sockfd = accept(sockfd)
write(s)
recv(c_sockfd)
read(s)
send(c_sockfd)
close(s)
close(c_sockfd)
close(sockfd)

# UDP – C

- s = socket(PF_INET, SOCK_DGRAM, 0);
- bind(s, &my_addr, sizeof(my_addr);
- len = sendto(s, sbuf, len, 0, &rem_addr, sizeof(rem_addr));
- rem_addr_length = sizeof(rem_addr);
- len = recvfrom(s, rbuf, sizeof(rbuf), 0, &rem_addr,&rem_addr_length);
- close(s);

# UDP – C


vysílač
přijímač
s = socket();
bind(s);
sendto(s);
recvfrom(s);
close(s);
s = socket();
bind(s);
recvfrom(s);
sendto(s);
close(s);

# TCP – Java

- java.net.Socket
  - konstruktory
    - Socket()
    - Socket(String, int)
    - Socket(InetAddress, int)
    - ...
  - metody
    - close()
    - getInetAddress()
    - getPort()
    - getLocalPort()
    - getInputStream()
    - getOutputStream()
    - set*()
    - toString()

# TCP – Java 2

- java.net.ServerSocket
  - konstruktory
    - ServerSocket()
    - ServerSocket(int)
    - …
  - metody
    - accept()
    - bind(SocketAddress)
    - getInetAddress()
    - getLocalPort()
    - toString()
    - ...

# TCP – Java 3

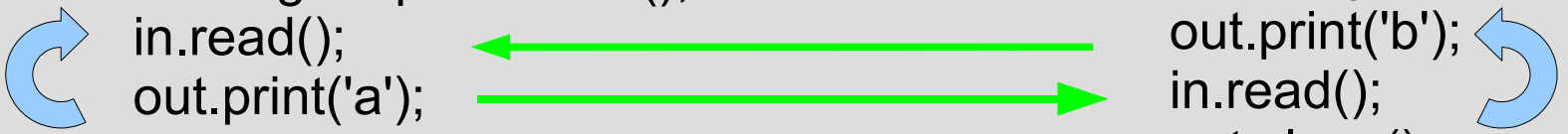
## klient

```
s = new Socket(sName, pNum);

out = s.getOutputStream();
in = s.getInputStream();
in.read();
out.print('a');
out.close();
in.close();
s.close();
```

## server

```
ss = new ServerSocket(1313);
cs = ss.accept();
out = cs.getOutputStream();
in = cs.getInputStream();
out.print('b');
in.read();
out.close();
in.close();
cs.close();
ss.close();
```



# UDP – Java

- java.net.DatagramSocket
  - kostruktory
    - DatagramSocket()
    - DatagramSocket(int)
  - metody
    - void close()
    - void bind(SocketAddress)
    - int getLocalPort()
    - InetAddress getLocalAddress()
    - void receive(DatagramPacket)
    - void send(DatagramPacket)
    - void setSoTimeout(int)
    - ...

# UDP – Java 2

- java.net.DatagramPacket
  - konstruktory
    - DatagramPacket(byte[], int)
    - DatagramPacket(byte[], int, InetAddress, int)
  - metody
    - InetAddress getAddress()
    - int getPort()
    - byte[] getData()
    - int getLength()
    - void setData(byte[])
    - void setLength(int)
    - ...

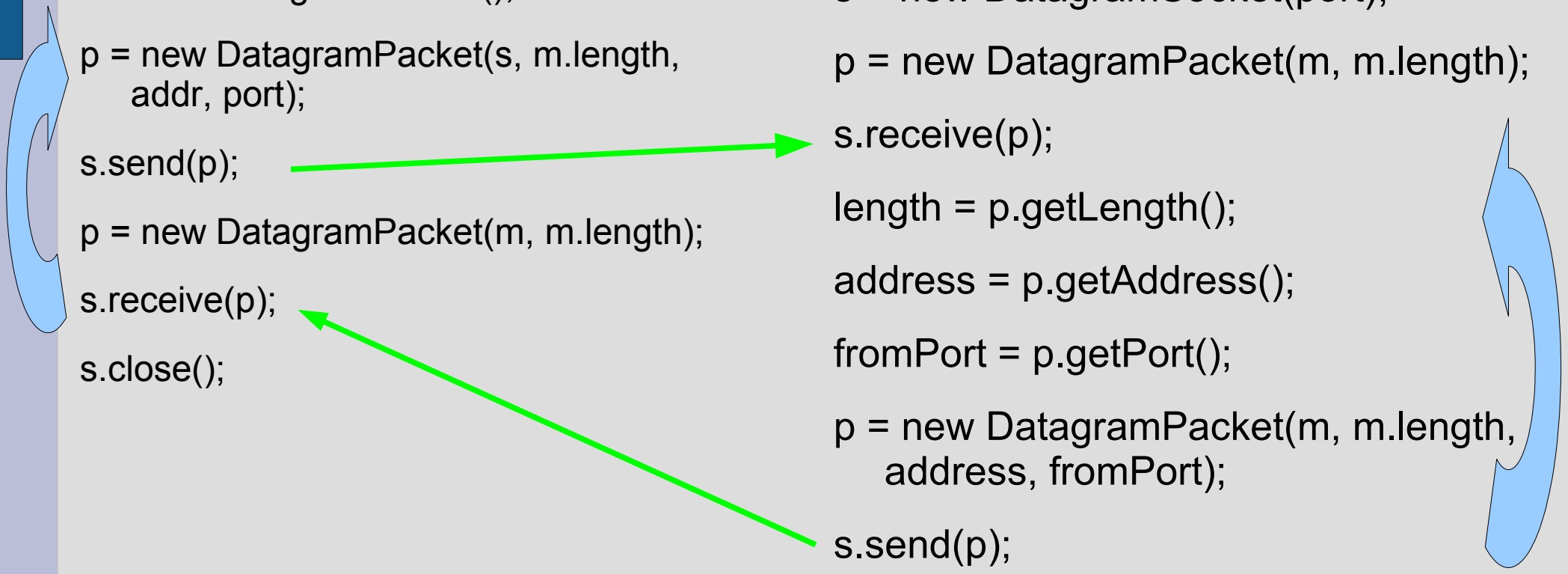
# UDP – Java 3

## vysílač

```
s = new DatagramSocket();
p = new DatagramPacket(s, m.length,
    addr, port);
s.send(p);
p = new DatagramPacket(m, m.length);
s.receive(p);
s.close();
```

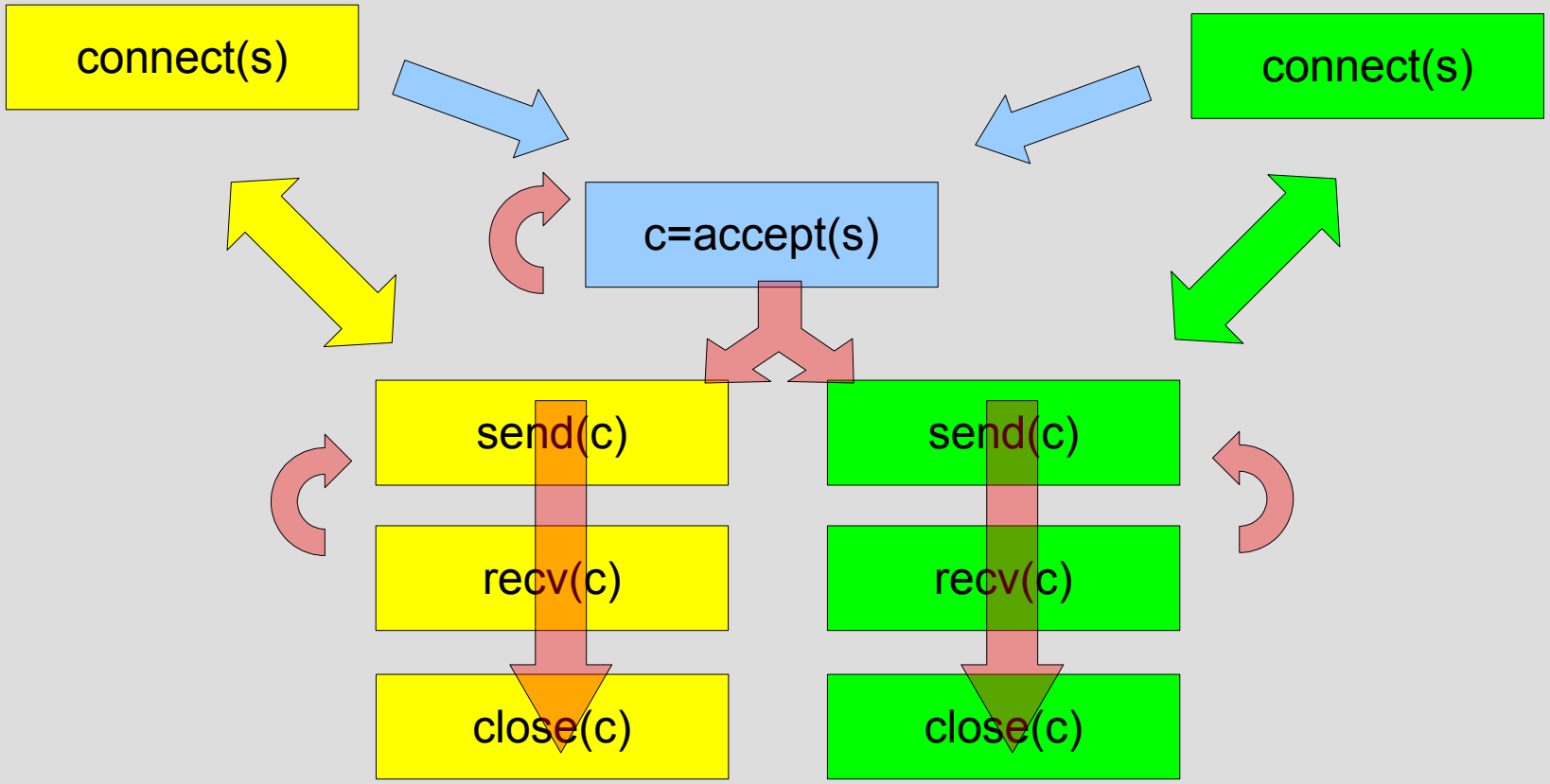
## přijímač

```
s = new DatagramSocket(port);
p = new DatagramPacket(m, m.length);
s.receive(p);
length = p.getLength();
address = p.getAddress();
fromPort = p.getPort();
p = new DatagramPacket(m, m.length,
    address, fromPort);
s.send(p);
s.close();
```



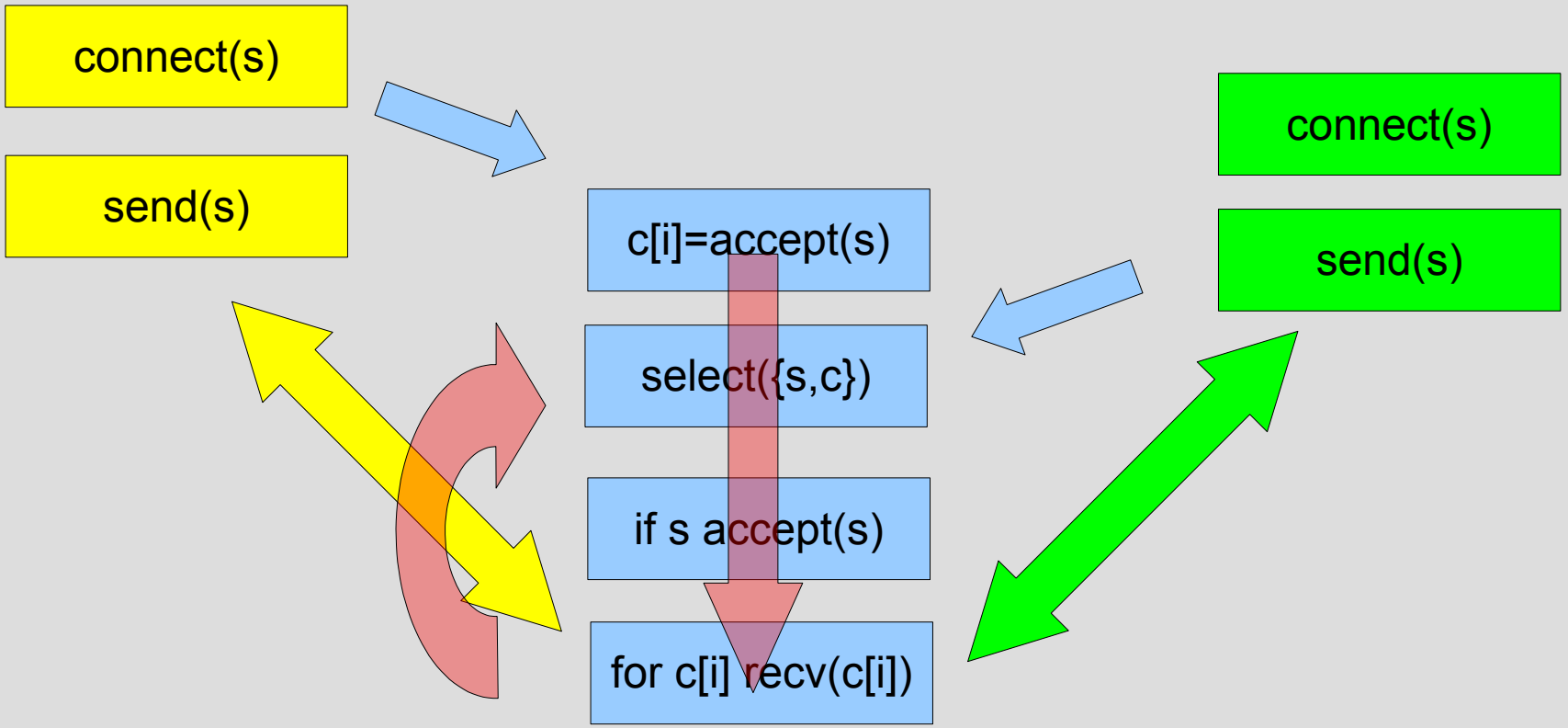
# TCP – souběžné zpracování

- několik vláken (Apache)



# TCP – souběžné zpracování 2

- jedno vlákno (Boa)



# Select()

- java.nio.channels
  - selector.select()
  - http://www.javaworld.com/javaworld/jw-04-2003/jw-0411-select.html

- int select(int nfds, fd_set *rfds, fd_set *wfds,fd_set *exfds, struct timeval *timeout);

- void FD_CLR(int fd, fd_set *set);
- int  FD_ISSET(int fd, fd_set *set);
- void FD_SET(int fd, fd_set *set);
- void FD_ZERO(fd_set *set);

a nebo úplně jinak ...