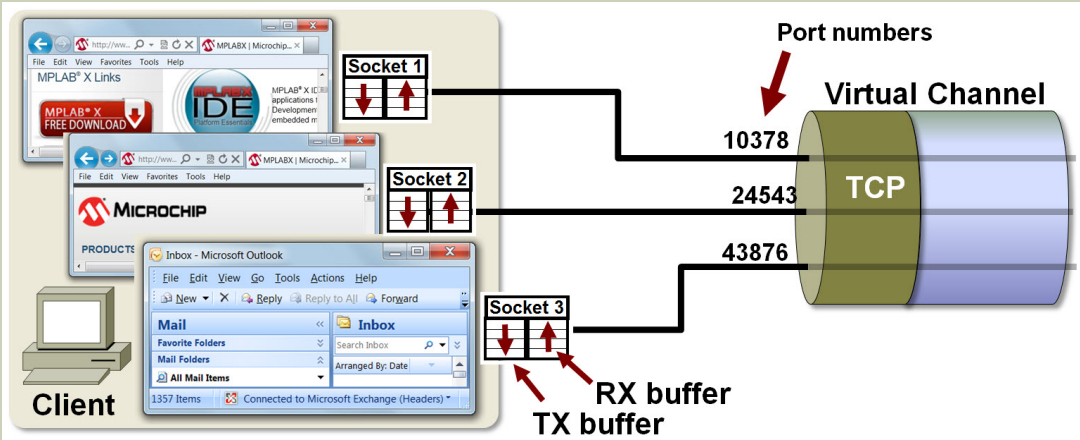
TCP PORTS

COMPS381F

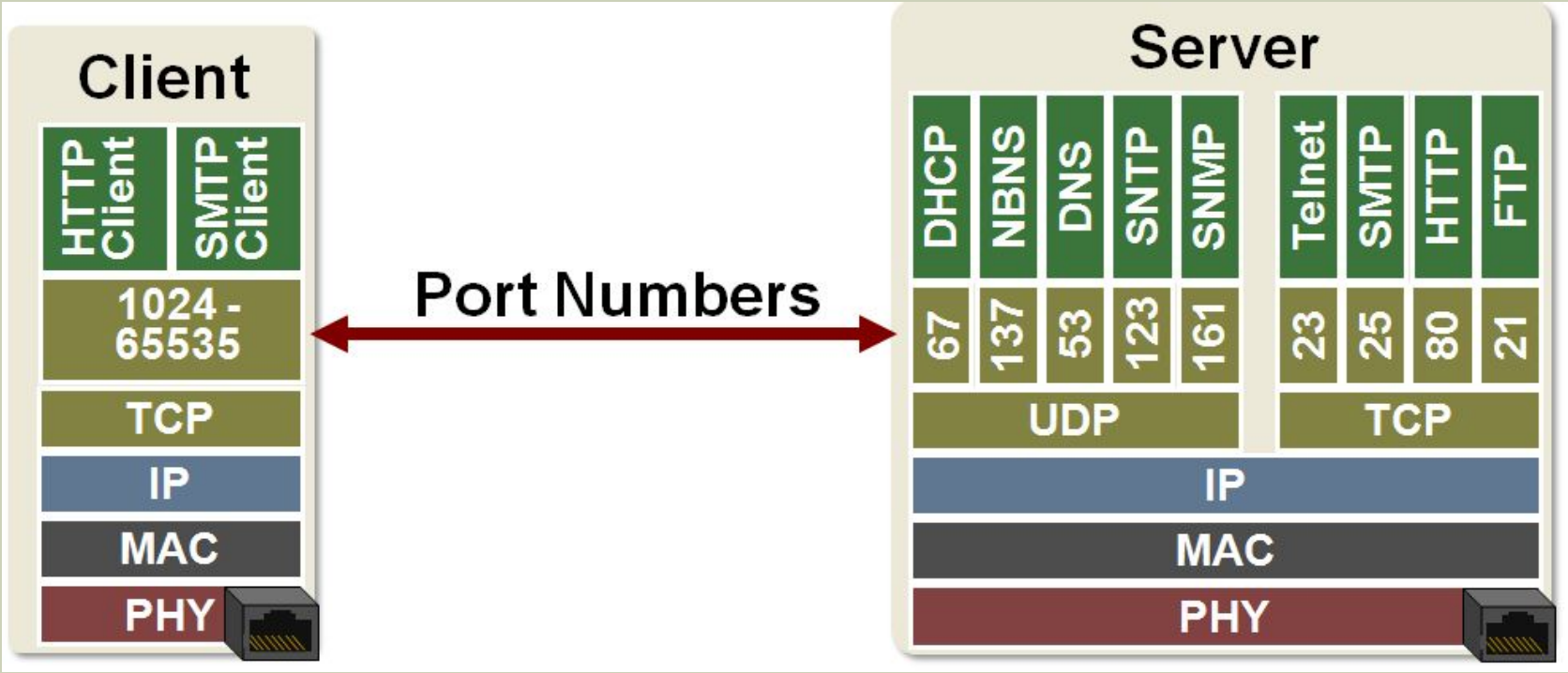
WHAT ARE TCP PORTS?

- Let's assume we have two applications running on one PC that require TCP/IP (Internet) communications.
- Assume one is a web browser and the other is an email client. Both applications send and receive packets with the same IP address, so how does the Transport layer differentiate a web browser packet from an email packet?
- The answer is **port numbers**.



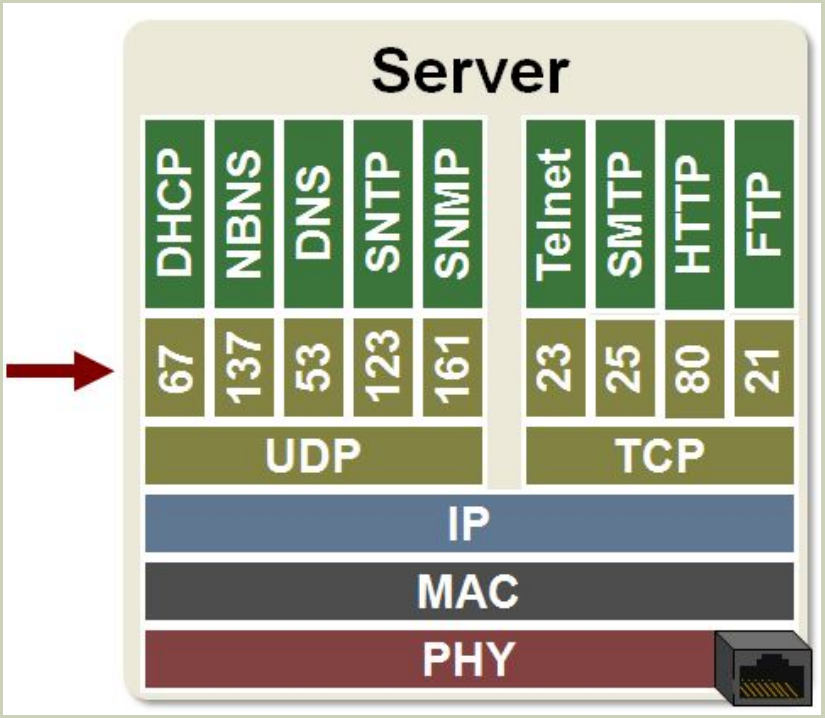
Source: http://microchip.wikidot.com/com4104:tcp-ip-ports

TCP PORT



Source: http://microchip.wikidot.com/com4104:tcp-ip-ports

TCP/IP "WELL-KNOWN" PORTS

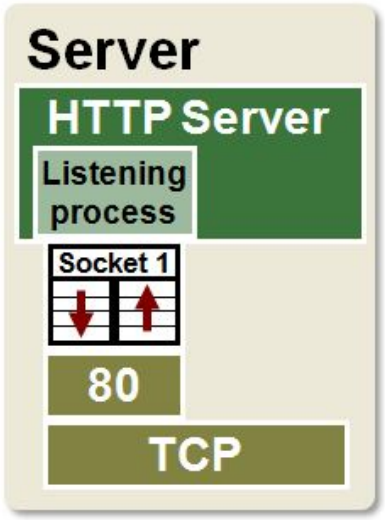


- "Well-Known" ports are port numbers that have been **reserved** for common applications, typically server applications.
- Clients know that servers will be listening for their requests at these reserved port numbers.
  - Example:
    - Web browsers (clients) connect to Web servers listening on port 80 (by default)
    - mongo connects to MongoDB servers listening on port 27107 (by default)
      - Try connect to port 28017 via a web browser...

Source: http://microchip.wikidot.com/com4104:tcp-ip-ports

SERVER CREATES *SOCKET* AND *LISTENS*

Server Sockets	Socket 1	
Transport	TCP	
Source Port	80	
Source IP Addr	192.168.1.102	
Destination Port	n/a	
Destination IP Addr	n/a	



IP Address = 192.168.1.102

Your Node.JS server listens on port 8099

Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection

CLIENT CREATES A SOCKET AND *CONNECTS*

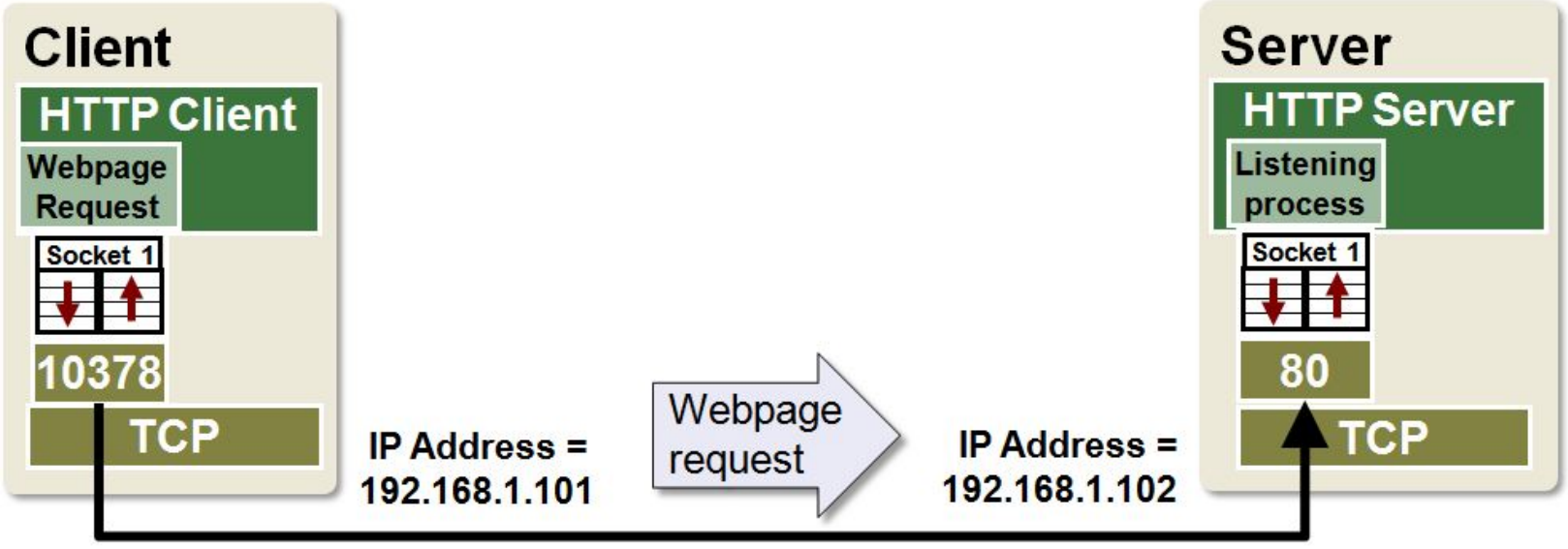
Web browser, curl and mobile apps (other types of client) connect to your Node.JS sever that listens on port 8099

Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection

TRANSPORT LAYER DELIVERS MESSAGE TO SERVER

Client Sockets	Socket 1
Transport	TCP
Source Port	10378
Source IP Addr	192.168.1.101
Destination Port	80
Destination IP Addr	192.168.1.102

Server Sockets	Socket 1	
Transport	TCP	
Source Port	80	
Source IP Addr	192.168.1.102	
Destination Port	n/a	
Destination IP Addr	n/a	

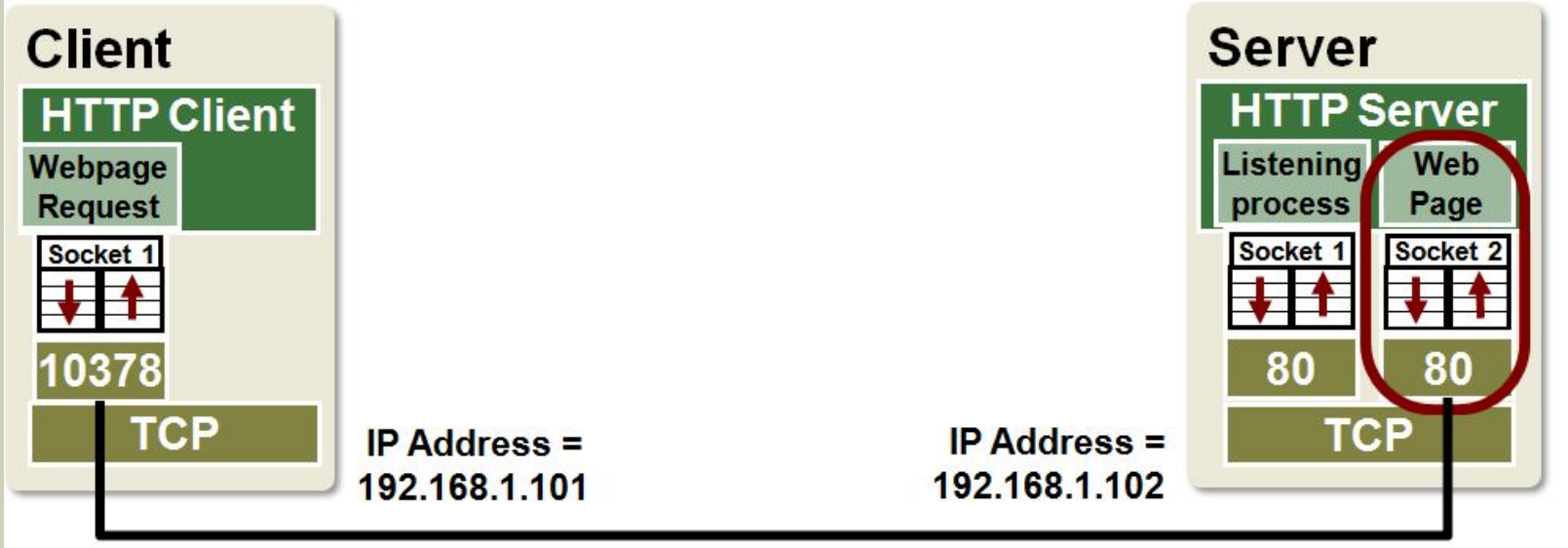


Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection

SERVER CREATES SOCKET & PROCESS

Client Sockets	Socket 1
Transport	TCP
Source Port	10378
Source IP Addr	192.168.1.101
Destination Port	80
Destination IP Addr	192.168.1.102

Server Sockets	Socket 1	Socket 2
Transport	TCP	TCP
Source Port	80	80
Source IP Addr	192.168.1.102	192.168.1.102
Destination Port	n/a	10378
Destination IP Addr	n/a	192.168.1.101

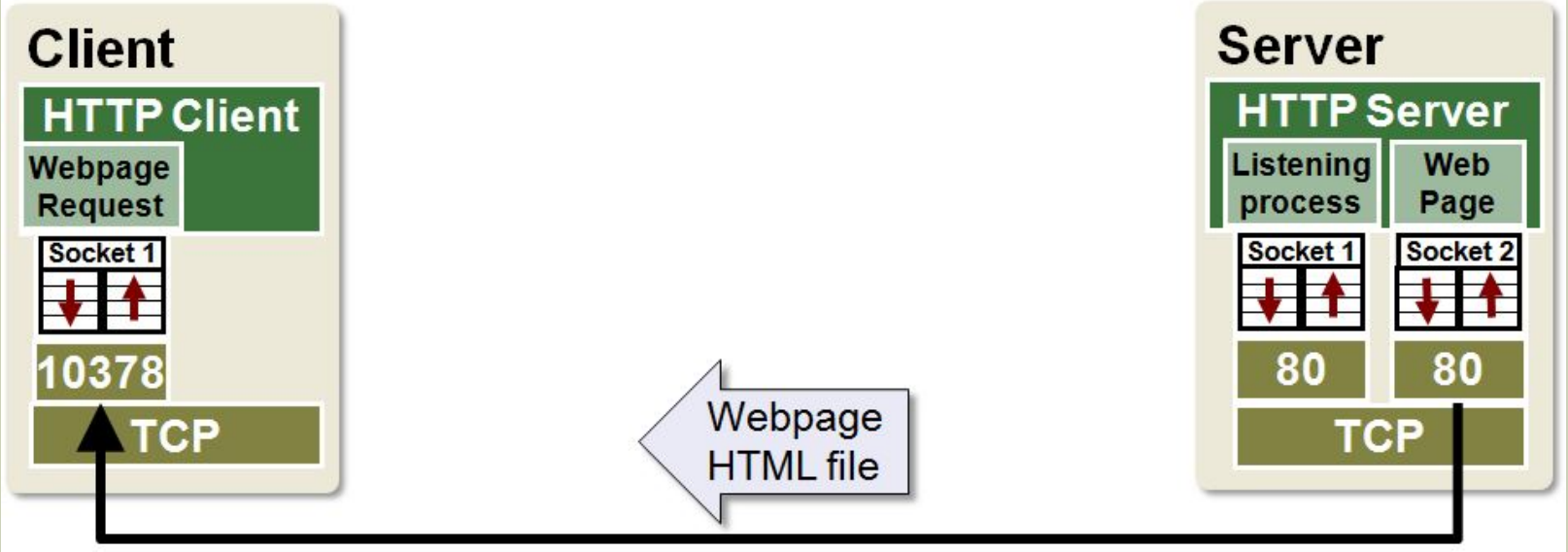


Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection

TRANSPORT LAYER DELIVERS MESSAGE TO CLIENT

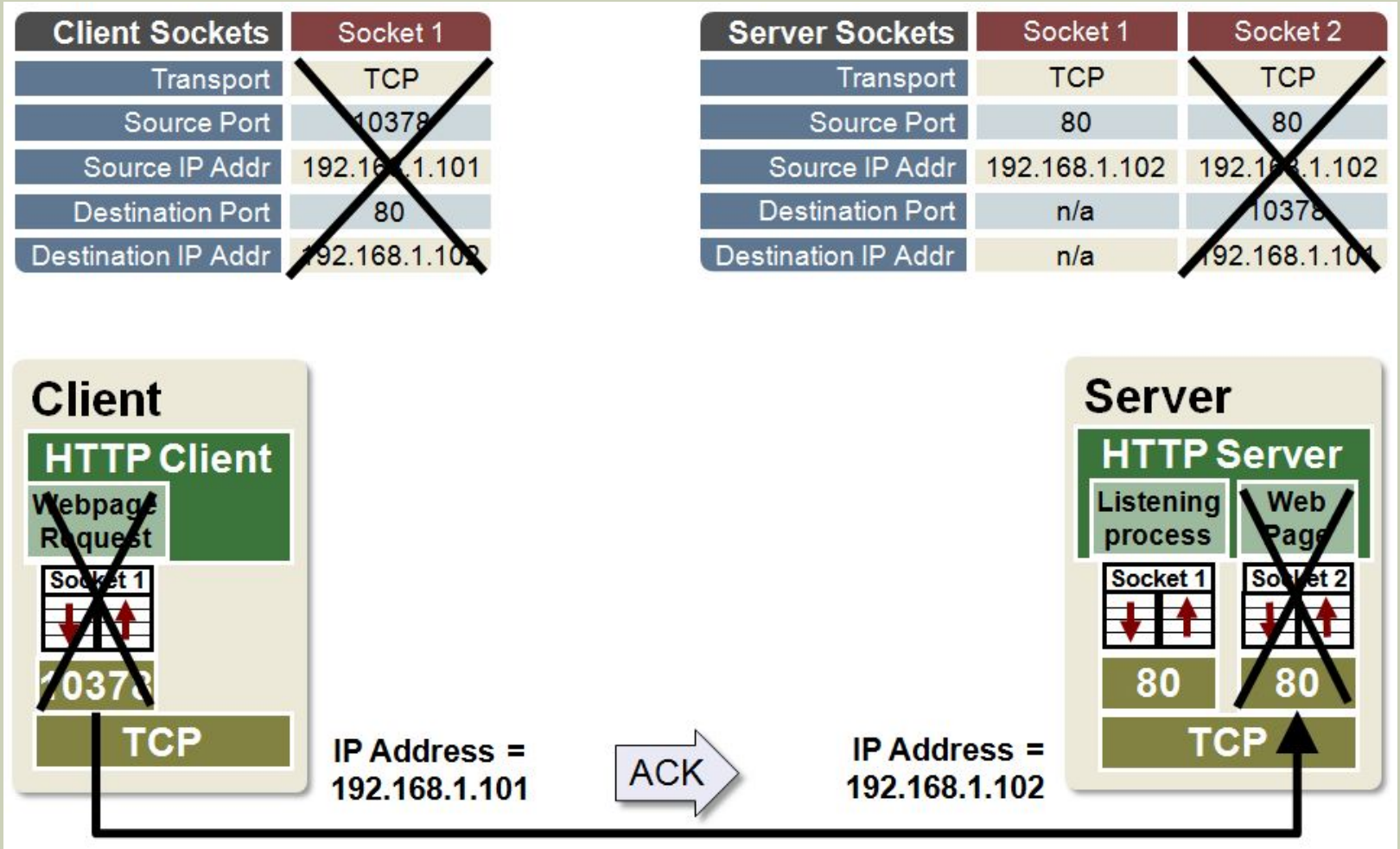
Client Sockets	Socket 1
Transport	TCP
Source Port	10378
Source IP Addr	192.168.1.101
Destination Port	80
Destination IP Addr	192.168.1.102

Server Sockets	Socket 1	Socket 2
Transport	TCP	TCP
Source Port	80	80
Source IP Addr	192.168.1.102	192.168.1.102
Destination Port	n/a	10378
Destination IP Addr	n/a	192.168.1.101



Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection

SOCKETS CLOSED



Source: http://microchip.wikidot.com/com4104:example-use-sockets-to-create-a-tcp-connection