

WC18.C

RFC 1945 / 2616

CLIENT CHE INVIA UNA FULL-REQUEST O RICEVE UNA FULL-RESPONSE
(CHUNKED)

\mathcal{D} = SOCKET

CONNECT(\mathcal{D} , INDIRIZZO...)

WRITE(\mathcal{D} , RICHIESTA, ...)

READ(\mathcal{D} , ...)

WS18.C

SERVER CHE RICEVE UNA FULL-REQUEST O RISPONDE CON UNA FULL-RESPONSE

\mathcal{D} = SOCKET

SET SOCKET

BIND(\mathcal{D})

LISTEN(\mathcal{D})

WHILE(1) {

\mathcal{D}_2 = ACCEPT(\mathcal{D} , ...)

READ(\mathcal{D}_2 , ...)

WRITE(\mathcal{D}_2 , RISPONDA, ...)

CLOSE(\mathcal{D}_2)

}

RICHIESTA HTTP

REQUEST LINE }
HEADERS } HEADER
CRLF
BODY

RISPONDA HTTP

STATUS LINE }
HEADERS } HEADER
CRLF
BODY

WP18.C

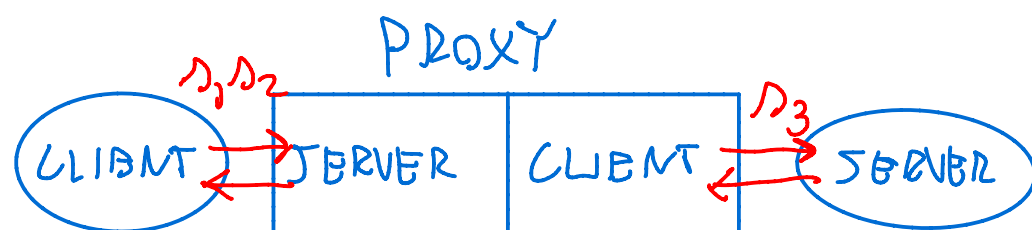
RFC1945 / 2616

PROXY CHE RICEVE UNA RICHIESTA GET DA UN CLIENT, CHE GLI DICE QUALI SERVER CONTATTA (HOSTNAME).

INVIATA LA RICHIESTA AL SERVER, LEGGE LA RISPOSTA E LA INVIATA AL CLIENT.

```
S = SOCKET  
SETSOCKOPT  
BIND(S)  
LISTEN(S)
```

PARTI
SERVER



```
WHILE(1) {  
    S2 = ACCEPT  
    FORK() {
```

```
        READ(S2, ...)
```

```
        IF != GET → WRITE(S2, "NOTIMPL")
```

```
        HP = GETHOSTBYNAME()
```

```
        S3 = SOCKET
```

```
        CONVERT(S3, ...)
```

```
        WRITE(S3, REQUEST)
```

```
        WHILE( READ(S3) )
```

```
            WRITE(S2)
```

```
        CLOSE S3
```

```
    }
```

```
}
```

PARTI
CLIENT

PARTI
SERVER

FILE WP18.C È UN CLIENT CHE SI COLLEGA A WP18.C
TRAMITE 127.0.0.1:7888 E RICHIEDE GOOGLE.COM

PING.C: CONTIENE ICMP ECHO

```
S = SOCKET
TROVA MAC()
CREA FRAME ETH+IP+ICMP ECHO
SEND TO(S, BUFFER, LUNGH BUFF, FLAG, INDIRIZZO DEST, LUNG/INDIRIZ)
FOR (m=0 → 1000) {
    RECV FROM(S, BUFFER, LUNGH BUFF, FLAG, INDIRIZZO SRC, LUNG/INDIRIZ)
    IF (ETH → TYPE = htons(0x0800)) JE ETH CONTIENE IP
    IF (IP → PROTOCOL = 1) JE IP CONTIENE ICMP
    IF (ICMP → TYPE = 0) STAMPA BUFFER() JE ICMP E ECHO RECEVUTO
}
```

{ ETH = 14 BYTE
IP = 20 BYTE
ICMP = 8 + 20 DATA = 28 BYTE

```
TROVA MAC() {
    S = SOCKET
    SEND TO(S, BUFFER, LUNGH BUFF, FLAG, INDIRIZZO DEST, LUNG/INDIRIZ)
    FOR (m=0 → 1000) {
        RECV FROM(S, BUFFER, LUNGH BUFF, FLAG, INDIRIZZO SRC, LUNG/INDIRIZ)
        IF (ETH → TYPE = htons(0x0806)) JE ETH CONTIENE ARP
        IF (ARP → OPCODE = htons(2)) JE ARP E DI RISPOSTA
        IF (IP SRC = DEST) MAC ← ARP → hsrc;
    }
}
```

{ ETH = 14 BYTE
+ ARP PACKET

TR.C: TRACE ROUTE: RICAVA IL PERCORSO CHE FANNO I PACCHETTI

SFRUTTA IL CAMPO TTL DEL PACCHETTO IP, CHE INDICA IL NUMERO DI ROUTER CHE PUO' ATTRAVERSARE PRIMA DI ESSERE SCARTATO.

VIENE INVIATO UN PACCHETTO CON TTL=1, COSI' IL ROUTER CHE LO RICEVE DA ERRORE TIME EXCEED, COSI' RICAVIAMO L'IP DEL 1° ROUTER, POI INVIAMO UN PACCHETTO CON TTL=2, POI = 3, ...

DA PING.C CAMBIA CREA IP e AGGIUNGO CICLO FOR

TCP.C:

RICEVE DAL BUFFER IL PACCHETTO ETHERNET
SE CONTIENE UN IP, SE CONTIENE UN TTL, SE IL PORTO DI DESTINAZIONE È 80
SE SYN=1