

Laboratorio reti AA 2008/2009

Dott. Matteo Roffilli

roffilli@csr.unibo.it

**Ricevimento in ufficio
dopo la lezione**

Laboratorio reti AA 2008/2009

Per esercitarvi fate SSH su:

alfa.csr.unibo.it

si-tux00.csr.unibo.it

....

si-tux15.csr.unibo.it

Eventuali variazioni di orario/giorno verranno comunicate in anticipo via mail.

Laboratorio reti AA 2008/2009

- **Marzo**
 - 5 Intro,SSH,VI/VIM,GCC base
 - 12 Richiami di C e Compilazione
 - 19 Socket e Co.
 - 26 Socket e Co. parte seconda
- **Aprile**
 - 2 Utilizzo di IPC: Client
 - 16 Client parte seconda
 - **23 Server**
 - 29 mercoledì 10:00-12:00 Server parte seconda

Provate ...

```
./client www.csr.unibo.it 80
```

```
IP: 137.204.78.100
```

```
read effettuata, risultato n=486 len=10000 nread=486 len-nread=9514
```

```
stringa ricevuta: HTTP/1.1 414 Request-URI Too Large
```

```
Date: Fri, 14 Apr 2006 09:01:06 GMT
```

```
Server: Apache/1.3.34 (Debian) mod_ssl/2.8.25 OpenSSL/0.9.8a
```

```
PHP/4.3.10-16
```

```
Connection: close
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<HTML><HEAD>
```

```
<TITLE>414 Request-URI Too Large</TITLE>
```

```
</HEAD><BODY>
```

```
<H1>Request-URI Too Large</H1>
```

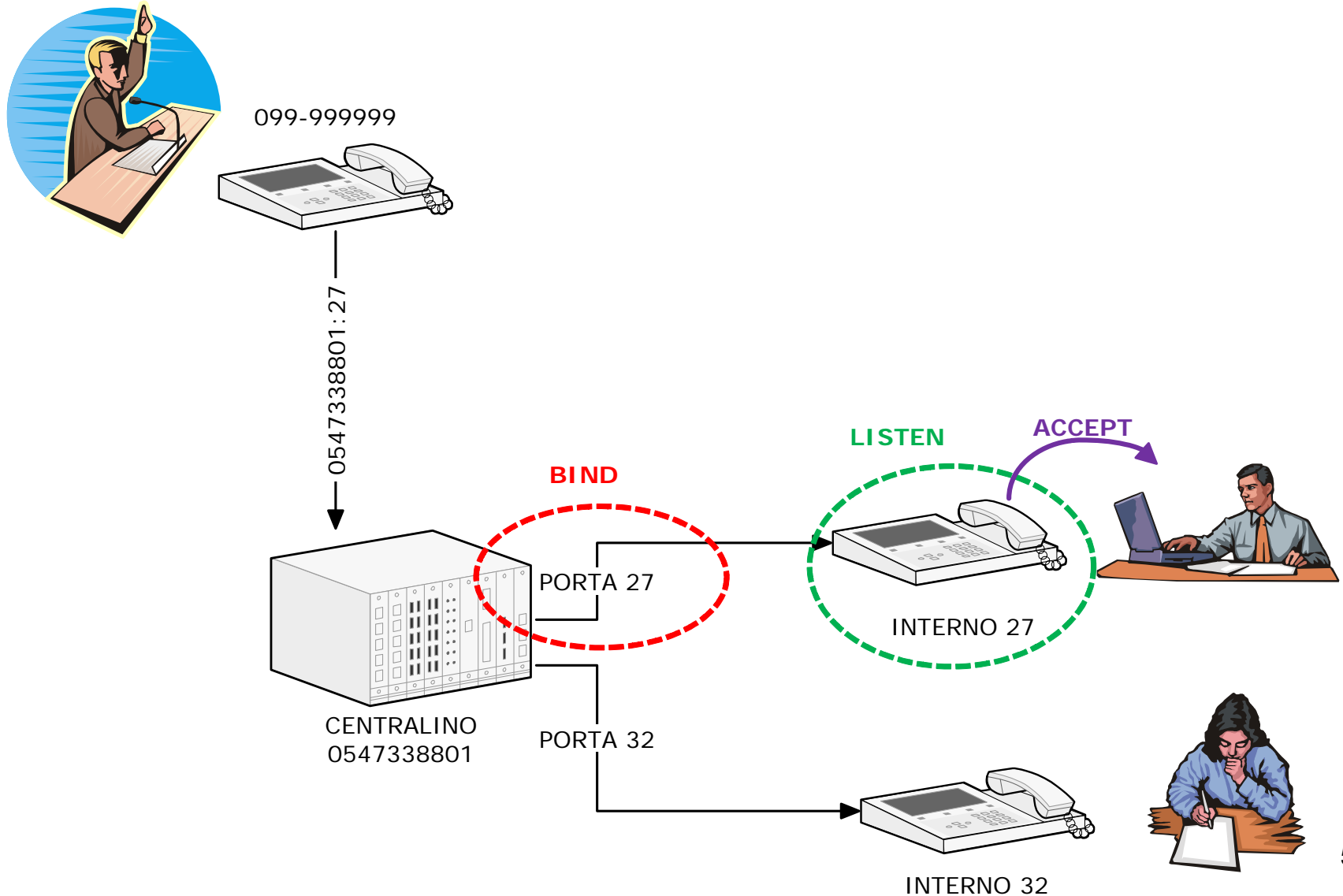
```
The requested URL's length exceeds the capacity
```

```
limit for this server.<P>
```

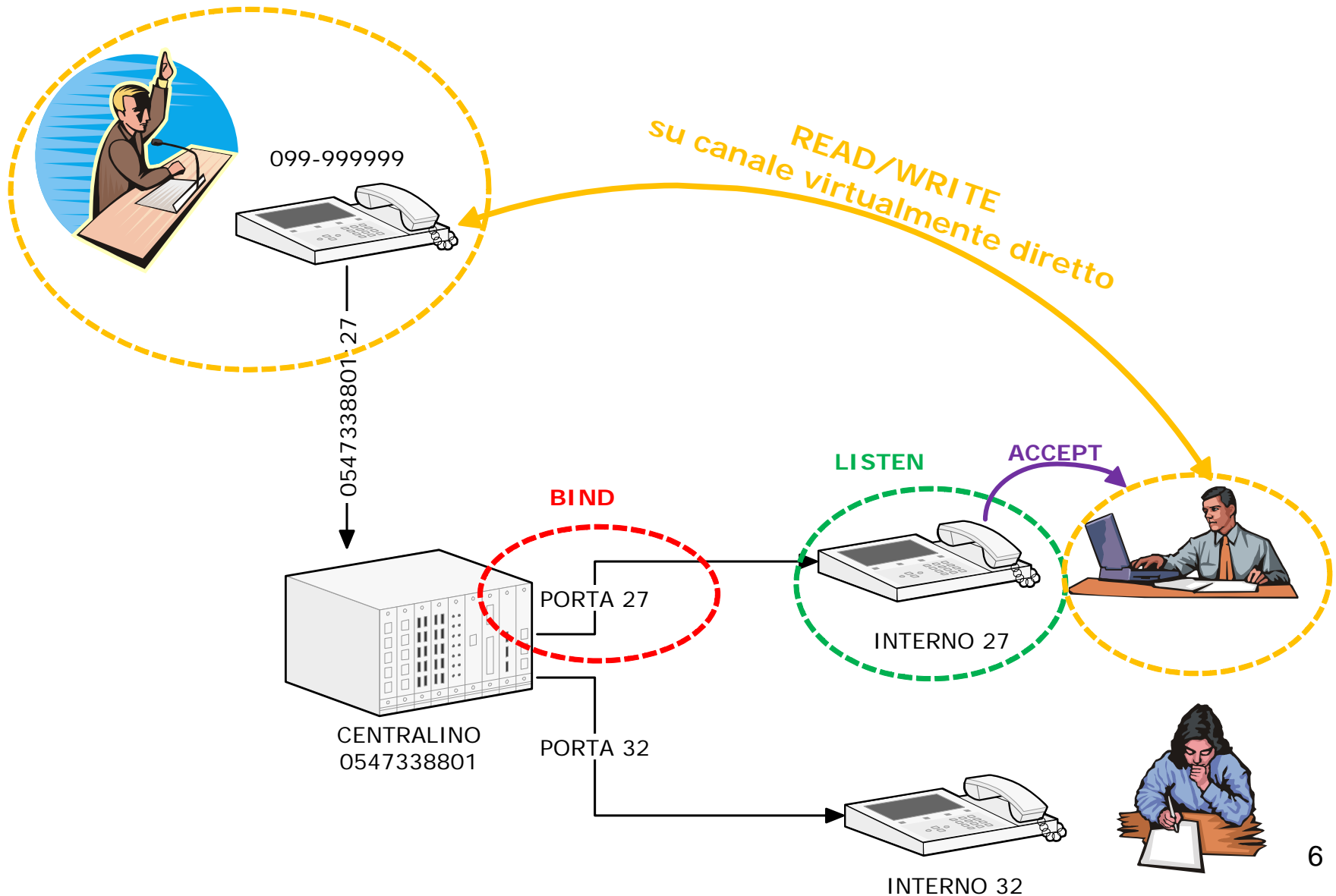
```
request failed: URI too long<P>
```

```
</BODY></HTML>
```

Analogia “telefonica”



Analogia “telefonica” 2



setsockopt

```
#include <sys/socket.h>
#include <sys/types.h>

int setsockopt(    int sock,
                  int level,
                  int optname,
                  const void* optval,
                  socklen_t optlen)
```

Imposta le opzioni di un socket.

La funzione restituisce 0 in caso di successo e -1 in caso di errore,
nel qual caso errno assumerà uno dei valori:

EBADF	il file descriptor sock non è valido.
EFAULT	l'indirizzo optval non è valido.
EINVAL	il valore di optlen non è valido.
ENOPROTOOPT	l'opzione scelta non esiste per il livello indicato.
ENOTSOCK	il file descriptor sock non corrisponde ad un socket

setsockopt

```
int setsockopt(    int sock,  
                  int level,  
                  int optname,  
                  const void* optval,  
                  socklen_t optlen)
```

sock indica il socket su cui si intende operare

level seleziona allora il protocollo su cui vuole intervenire

optname permette di scegliere su quale delle opzioni che sono definite
per quel protocollo si vuole operare:

SOL_SOCKET opzioni generiche

SOL_IP per i socket che usano IPv4

SOL_TCP per i socket che usano TCP

optval è un puntatore ad una zona di memoria che contiene i dati che
specificano il valore dell'opzione che si vuole passare al socket

optlen è la dimensione in byte dei dati presenti all'indirizzo indicato da optval

setsockopt - opzioni

Opzione	get	set	flag	Tipo	Descrizione
SO_KEEPALIVE	•	•	•	int	controlla l'attività della connessione.
SO_OOBINLINE	•	•	•	int	lascia in linea i dati <i>out-of-band</i> .
SO_RCVLOWAT	•	•	•	int	basso livello sul buffer di ricezione.
SO_SNDLOWAT	•	•		int	basso livello sul buffer di trasmissione.
SO_RCVTIMEO	•	•		timeval	timeout in ricezione.
SO_SNDTIMEO	•	•		timeval	timeout in trasmissione.
SO_BSDCOMPAT	•	•	•	int	abilita la compatibilità con BSD.
SO_PASSCRED	•	•	•	int	abilita la ricezione di credenziali.
SO_PEERCREC	•			ucred	restituisce le credenziali del processo remoto.
SO_BINDTODEVICE	•	•		char *	lega il socket ad un dispositivo.
SO_DEBUG	•	•	•	int	abilita il debugging sul socket.
SO_REUSEADDR	•	•	•	int	consente il riutilizzo di un indirizzo locale.
SO_TYPE	•			int	restituisce il tipo di socket.
SO_ACCEPTCONN	•			int	indica se il socket è in ascolto.
SO_DONTROUTE	•	•	•	int	non invia attraverso un gateway.
SO_BROADCAST	•	•	•	int	attiva o disattiva il <i>broadcast</i> .
SO_SNDBUF	•	•		int	imposta dimensione del buffer di trasmissione.
SO_RCVBUF	•	•		int	imposta dimensione del buffer di ricezione.
SO_LINGER	•	•		linger	indugia nella chiusura con dati da spedire.
SO_PRIORITY	•	•		int	imposta la priorità del socket.
SO_ERROR	•			int	riceve e cancella gli errori pendenti.

setsockopt - esempio

```
int setsockopt(  int sock,
                int level,
                int optname,
                const void* optval,
                socklen_t optlen)

/* avoid EADDRINUSE error on bind() */
int OptVal = 1;

printf ("setsockopt()\n");

ris = setsockopt(    sockfd,
                    SOL_SOCKET,
                    SO_REUSEADDR,
                    &OptVal,
                    sizeof(OptVal));

if (ris == SOCKET_ERROR) {
    printf ("setsockopt() SO_REUSEADDR failed, Err: %d\n",
            errno, strerror(errno));
    exit(EXIT_FAILURE);}
```

Programma esercitazioni

- Creare un server che risponda al client già sviluppato
- Accettare solo client provenienti da un determinato IP
- Modificare la dimensione del buffer di ricezione dello stack TCP/IP
- Rendere il server multi-processo con `fork ()` o multi-thread con `lpthread`

Esercizio

Goal:

- Creare un server che risponda al client già sviluppato

Requisiti:

1. Il server risponde con il messaggio ricevuto dal client (tipo echo)
2. Il messaggio deve essere di 100 caratteri
3. Lanciate il server su **si-tux00.csr.unibo.it** alla porta **8888** o altra.
4. Testate il client e il server.

Tempo a disposizione:

60 minuti

Traccia Soluzione

```
int main(int argc, char *argv[])
{
    /* get a stream socket */
    sockfd = socket(PF_INET, SOCK_STREAM, 0);

    /* name the socket */
    Local.sin_family =      AF_INET;

    ris = bind(sockfd, (struct sockaddr*) &Local, sizeof(Local));

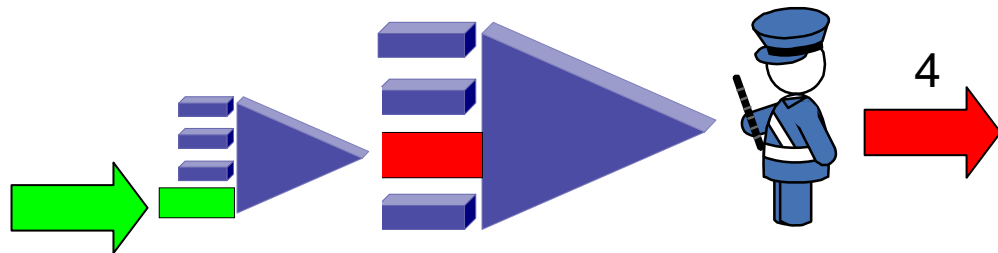
    ris = listen(sockfd, 10 );

    newsocketfd = accept(sockfd, (struct sockaddr*) &Cli, &len);

    /* wait for data */
    printf ("read()\n");

    /* scrittura */
    printf ("write()\n");

    /* chiusura */
    close(newsocketfd);
    close(sockfd);
}
```



Esercizio extra I

Goal:

- Accettare solo client provenienti da un determinato IP

Requisiti:

1. Stampare a video le informazioni del client che richiede la connessione
2. Accettare solo client con IP **137.204.72.5** (**alfa.csr.unibo.it**)

Tempo a disposizione:

30 minuti