


# Introduzione alla tecnologia Client / Server e TCP/IP

http://www.vbsimple.net/cliserv/clser_00.htm



**Difficoltà: 3 / 5**

## Introduzione alla tecnologia Client / Server

Con lo sviluppo di Internet e dei mezzi di trasferimento di dati le applicazioni che prima lavoravano su macchine locali ed utilizzavano i dati presenti sullo stesso computer in cui il programma veniva eseguito si sono innalzate di livello.

Sono, infatti, molto comuni le applicazioni che si collegano ad altri computer posti nello stesso locale, in una rete LAN. Quasi tutti i nuovi programmi prevedono la possibilità di collegarsi ad Internet ed alcuni d'essi lavorano completamente su Internet. Alcuni programmi molto più avanzati prevedono lo sfruttamento di dati presenti in un computer remoto, posto chissà dove.

Alcuni esempi di questi programmi sono le Chat, le videoconferenze, gli strumenti di amministrazione remota, i gestori di commercio elettronico, e (perche no?) i virus.

Per effettuare questo genere di operazioni di scambio di dati possono essere utilizzati i protocolli TCP/IP preesistenti oppure si possono generare nuovi protocolli personalizzati. Per utilizzare un protocollo preesistente è necessario conoscere le RFC relative al protocollo da utilizzare. Un completissimo archivio di RFC (in inglese) è presente in rete all'indirizzo http://www.faqs.org/rfcs.

Per tutte le applicazioni Client/Server sarà necessario utilizzare la libreria Winsock di Windows. Tale libreria è disponibile in due versioni: una è la **WSOCK32.DLL** da richiamare mediante l'API, e l'altra è la **MSWINSCK.OCX**, un controllo ActiveX instanziabile all'interno di Visual Basic.
Soltanto per alcuni esempi, verrà utilizzata una terza libreria, la **ICMP.DLL**, la libreria del protocollo ICMP.

## Introduzione al protocollo TCP/IP - Gli indirizzi IP

Prima di iniziare con lo sviluppo Client/Server è necessario conoscere il funzionamento del protocollo TCP/IP ed il comportamento di Windows nei confronti delle chiamate

 

TCP/IP.

Tutti i computer connessi mediante TCP/IP, il protocollo di Internet, possiedono un loro indirizzo unico, detto indirizzo IP (Internet Protocol) che si compone di quattro cifre decimali, ognuna delle quali è compresa tra 0 e 255.
Un esempio di indirizzo è **212.24.124.151**.

Quando utilizziamo Internet siamo abituati a digitare il nome di un sito in maniera non numerica. Un esempio di indirizzo Internet è **http://www.vbsimple.net/index.htm**.
In realtà tale indirizzo è un DNS, ovvero una forma alternativa per rappresentare un indirizzo IP. Nel momento in cui richiediamo un collegamento ad un sito remoto in forma di DNS, la richiesta va verso un risolutore di DNS che trasforma il sito richiesto in un indirizzo IP numerico.

Il DNS che abbiamo visto si compone di tante parti:

1. **http://**
   Indica il protocollo utilizzato per connettersi.
2. **vbsimple**
   Indica un sottodominio di nome vbsimple all'interno di una rete.
3. **virtualave**
   Rappresenta il nome dell'host che conterrà tutte le informazioni e gli eventuali sottodomini.
4. **net**
   Indica un nome di dominio e rappresenta un po' la categoria cui appartiene l'host.
5. **/index.htm**
   Rappresenta il nome di un file all'interno dell'host/sottodominio indicato.
   È un percorso logico, non strettamente fisico, poiché una determinata cartella, sebbene sembri essere contenuta all'interno di un'altra, in realtà potrebbe essere posizionata anche su un altro computer. Sarà compito del programma server effettuare il mapping tra percorsi logici e fisici.

Esistono due tipologie di indirizzi IP: quelli ***statici*** e quelli ***dinamici***. I primi sono tipici di siti web e non variano mai; i secondi sono molto comuni nelle connessioni dial-up via modem. Nel momento in cui un utente si connette ad Internet mediante un modem, gli viene assegnato dal suo *ISP* (*Internet Service Provider*) un indirizzo IP numerico. Tale indirizzo non cambierà fino a quando l'utente non si disconnetterà.

Gli indirizzi IP si possono classificare anche in due altre tipologie: esistono indirizzi *pubblici* ed indirizzi *privati*. Gli indirizzi ***pubblici*** sono univoci in tutto il mondo (non possono esisterne due uguali) e sono effettivamente raggiungibili tramite ogni connessione ad Internet. I siti web, i server di posta elettronica e tutto quanto troviamo su Internet possiede indirizzi pubblici. Anche se un sito web è inaccessibile perché protetto da qualche parola d'ordine oppure perché l'amministratore della rete ne ha disabilitato l'accesso, il suo indirizzo IP è pubblico.

Gli indirizzi ***privati***, al contrario, sono degli indirizzi virtuali che l'amministratore di una rete può utilizzare per installare una propria rete interna (LAN), senza che questi effettivamente vadano in conflitto con altri indirizzi Internet. Infatti, a differenza degli indirizzi pubblici, possono esistere due o più computer con lo stesso indirizzo IP, purché

 

essi stiano in due reti differenti o non collegate tra loro. Tuttavia, nessuno, all'esterno della rete con indirizzi privati potrà connettersi ad essa utilizzando Internet.

Gli indirizzi privati si dividono in tre classi, più una speciale.

1. **Classe A**: dall'indirizzo 10.0.0.0 al 10.255.255.255
2. **Classe B**: dall'indirizzo 172.16.0.0 al 172.31.255.255
3. **Classe C**: dall'indirizzo 192.168.0.0 al 192.168.255.255
4. **Indirizzo di loopback**: dal 127.0.0.1 al 127.255.255.254

L'indirizzo di loopback è un po' una finzione. È un indirizzo privato che possiedono **tutti** i computer che utilizzano TCP/IP. Pertanto tutte le operazioni effettuate sull'indirizzo di loopback saranno effettuate sul computer che le esegue. Il tentativo di connessione ad un indirizzo di loopback rappresenta una connessione allo stesso computer chiamante.

---

## Introduzione al protocollo TCP/IP - Le porte TCP

Affinché un computer possa comunicare con un altro mediante TCP/IP è necessario che su entrambi i computer sia disponibile un socket utilizzabile, una sorta di handle TCP/IP. L'utilizzo di un socket comporta l'apertura di una porta TCP, ovvero la messa in ascolto su un computer server che attende che i client si colleghino. Sul computer client ci sarà anche una porta TCP aperta, ma non sarà posta in ascolto come sul server, ma tramite questa porta il computer invierà i dati relativi al tentativo di connessione.

Finché la connessione non è stabilita il server assume una posizione passiva di ascolto. Sarà compito del client effettuare tutte le operazioni di preparazione, la scelta dell'indirizzo e della porta corretta e l'effettuazione della chiamata.

L'attivazione di un collegamento comporta l'allineamento del computer client al computer server. Sarà il programma server che stabilirà su quale porta si dovrà comunicare. Il client, dal lato suo, potrà aprire il suo socket su qualunque porta. Ciò che conta, infatti, è il numero di porta su cui i dati confluiscono.

Per esempio un programma server può aprire in ascolto la sua porta locale 1500. Il programma client, affinché la connessione sia stabilita, dovrà aprire un socket libero sul computer in cui il programma viene eseguito (locale) e mediante tale socket dovrà inviare una richiesta di connessione alla porta remota 1500 dell'indirizzo del computer scelto.

Stabilita la connessione avremo tale situazione:


Sebbene a connessione stabilita entrambi i computer potranno inviare dati, tuttavia il programma server non dovrebbe mai inviare dati che non sono effettivamente richiesti dal client, ma si dovrebbe limitare a reagire alle operazioni richieste dal client.

Quando si vogliono rappresentare connessioni si utilizza scrivere il DNS o il numero IP

 

del computer seguito dai due punti e dal numero della porta.
Ad esempio: 127.0.0.1:80 indica la porta 80 sul computer locale (loopback).

---

## L'utilizzo di NETSTAT

Per visualizzare tutte le porte TCP in ascolto o collegate nel computer basta aprire un prompt di MS-DOS  e digitare il comando **NETSTAT -NAP TCP**.

```
  Proto  Indirizzo locale     Indirizzo remoto    Stato
  TCP    127.0.0.1:1043       213.41.60.24:80     ESTABLISHED
  TCP    127.0.0.1:137        0.0.0.0:0           LISTENING
  TCP    127.0.0.1:1054       62.18.24.221:110    TIME_WAIT
```

L'output del comando mostra le connessioni attive, in attesa e in chiusura e per ognuna d'esse riporta il protocollo (Proto), l'indirizzo locale con la sua porta, l'indirizzo remoto con la sua porta e lo stato della connessione rispetto alla porta locale. Tale stato può essere:

- **LISTENING**: la porta è aperta in ascolto ed attende una connessione.
- **SYN_SENT**: la porta è aperta e sta inviando dati.
- **SYN_RECEIVED**: la porta è aperta e sta ricevendo dati.
- **ESTABLISHED**: la porta è aperta ed è collegata con un computer.
- **TIME_WAIT**: la connessione è terminata, tuttavia la porta attende la chiusura.
- **FIN_WAIT_1**: la porta è ancora aperta, ma si accinge a chiudersi.
- **FIN_WAIT_2**: la porta è ancora aperta, ma si accinge a chiudersi.
- **CLOSE_WAIT**: la porta attende la chiusura.
- **CLOSING**: la porta è in chiusura.
- **LAST_ACK**: la porta è in chiusura (ultimo messaggio prima di chiuderla).
- **CLOSED**: la porta è effettivamente chiusa.

Alcuni di questi stati sono molto difficili da vedere poiché la loro permanenza dura pochi decimi di secondo. Alcuni programmi di protezione, ad esempio un firewall, possono forzare alcuni di questi stati, anticipando o bloccando alcuni stati.

Una nota particolare la merita lo stato **TIME_WAIT** poiché è spesso causa di errori di funzionamento dei programmi client. Quando un programma client, utilizzando una porta **x** si connette ad un programma server sulla porta **y** ed in seguito alla fine del lavoro si disconnette (volontariamente o meno) e prova a ricollegarsi continuando ad utilizzare la porta **x** si verifica un errore.
Infatti, subito dopo la chiusura di una connessione, la porta **x** se ne va nello stato **TIME_WAIT**, risultando occupata per qualche secondo. Il tentativo di riconnettersi mediante la stessa porta **x** genererà un errore di ***"Indirizzo in uso"***.
Per risolvere questo problema è necessario attendere la completa chiusura della porta oppure cambiare la porta locale **x** con un'altra libera. In genere basta specificare la porta locale 0 per farsi assegnare un socket ad una porta libera.

Confronta anche l'Article ID Q137984 della Microsoft Knowledge Base.

Fibia FBI
4 Marzo 2001

 


Torna all'indice Client / Server

 