TCP/IP Le risposte alle domande più frequenti (parte 3)

http://www.vbsimple.net/news/news_04_4.htm

Torna all'indice | Parte 1 | Parte 2 | Parte 3

Quali sono i protocolli di applicazione del TCP/IP più comuni?

Programmazione TCP/IP

Quali sono i protocolli di applicazione del TCP/IP più comuni?

1. DHCP

   Il Dynamic Host Configuration Protocol (**DHCP** - Protocollo di configurazione host dinamica) permette agli indirizzi IP di essere allocati in una maniera come-necessario. Lo schema convenzionzionale di allocare un indirizzo IP fisso permanente ad ogni host è uno spreco di indirizzi in situazioni in cui sono attivi soltanto un numero relativamente piccolo di computer per volta. Il DHCP permette ad un host di ricevere un indirizzo IP da un insieme (pool) di indirizzi IP; quando

l'indirizzo non è più necessario sarà riciclato e reso disponibile per l'uso da un altro host. Il DHCP permette ad un host di recuperare varie informazioni di configurazione assieme all'acquisizione dell'indirizzo IP.

Il DHCP dipende dall'UDP per trasportare pacchetti tra sistemi client/server.

Il DHCP è definito nelle RFC 2131 e 2132. Un'implementazione largamente usata di DHCP può essere scaricata da <http://www.isc.org/dhcp.html>.

2. DNS

   Il Domain Name System (**DNS** - Sistema di nomi di dominio) provvede una traduzione dinamica su richiesta tra nomi in forma leggibile (come www.pizzahut.com) e l'indirizzo numerico utilizzato dall'IP (come 192.112.170.243). Le informazioni basilari sulle operazioni DNS sono definite nelle RFC 1034, 1101, 1876, 1982 e 2065.

   DNS utilizza entrambi UDP e TCP. Il primo viene utilizzato per trasportare le richieste e le risposte semplici, ma dipende dal TCP per garantire che siano consegnate sulla rete grandi quantità di dati (ad esempio i trasferimenti di configurazione di intere zone) nel metodo corretto ed ordinato.

   I DNS standard sono discussi nel newsgroup comp.protocols.dns.std. Un'implementazione largamente utilizzata di DNS chiamata BIND (Berkeley Internet Name Domain) è discussa nel newsgroup comp.protocols.dns.bind ed il software stesso BIND può essere scaricato da <http://www.isc.org/bind.html>. Le operazioni e le politiche di DNS sono discusse nel newsgroup comp.protocols.tcp-ip.domains.

3. FTP

   Il File Transfer Protocol (**FTP** - Protocollo di trasferimento di files) provvede un meccanismo per spostare file di dati tra sistemi. In aggiunta alle operazioni fondamentali PUT e GET, l'FTP provvede un piccolo numero di semplificazioni per gestione dei files e autenticazione degli utenti. Il protocollo è definito nella RFC 959.

   L'FTP dipende dal TCP per garantire la consegna corretta ed ordinata di dati nella rete.

4. HTTP

   L'Hyper Text Transfer Protocol (**HTTP** - Procollo di trasferimento di ipertesti) è il protocollo utilizzato per spostare pagine Web in un internet. La versione 1.0 dell'HTTP è definita nella RFC 1945. La versione 1.1 ha un utilizzo molto più efficiente del TCP ed è definito nella RFC 2068.

   L'HTTP dipende dal TCP per garantire la consegna corretta ed ordinata dei dati nella rete.

5. IMAP

   L'Interactive Mail Access Protocol (**IMAP** - Protocollo di accesso alla posta interattivo) permette ai client di manipolare i messaggi email e le caselle che risiedono su alcune macchine server. La versione attuale dell'IMAP è la Versione 4, generalmente definita IMAP4. Essa è descritta nella RFC 2060. L'IMAP non provvede alcuna maniera per inviare email; i programmi che utilizzano IMAP per leggere la posta, usano generalmente SMTP per inviare messaggi. L'IMAP é molto più complesso e potente dell'altro protocollo di lettura della posta, ampiamente utilizzato, POP.

   L'IMAP dipende dal TCP per garantire la consegna corretta ed ordinata dei dati nella rete.

   L'IMAP è discusso nel newsgroup comp.mail.imap.

6. NFS

   Il Network File System (**NFS** - Filesystem di rete) permette ai files memorizzati in una macchina (server) di essere acceduti dalle altre macchine (clients) come se fossero presenti nel sistema client. L'NFS è definito come astrazione di Remote Procedure Call (**RPC** - Chiamata a procedura remota) che a sua volta formatta i suoi pacchetti come un'eXternal Data Representation (**XDR** - Rappresentazione di dati esterna) indipendente dal processore.

   La versione 2 dell'NFS è definito nella RFC 1094 e la versione 3 è definita nella RFC 1813. Il meccanismo RPC spesso utilizzato con NFS, ONC/RPC, è definito nella RFC 1831. Le convenzioni XDR utilizzate dall'ONC/RFC sono definite nella RFC 1832. Il meccanismo di binding ONC/RPC (un minimo servizio di directory che permette ai clients RPC di conversare con i server RPC) è definito nella RFC 1833.

   NFS può essere utilizzato su qualunque sistema di trasporto, ma molto spesso viene utilizzato con UDP. Ma poiché UDP non garantisce la consegna e l'ordine dei pacchetti, quando viene utilizzato per trasportare NFS l'implementazione RPC deve provvedere una sua maniera di garanzia di correttezza. Quando l'NFS viene utilizzato su TCP, lo strato RPC può dipendere dal TCP per provvedere questo tipo di correzioni.

   L'NFS è discusso nel newsgroup comp.protocols.nfs.

7. NNTP

   Il Network News Transfer Protocol (**NNTP** - Protocollo di trasferimento di notizie sulla rete) viene utilizzato per propagare post (invii) di notizie in rete (inclusi i post su Usenet) tra sistemi. È definito nella RFC 977. Il formato dei messaggi delle news è definito nella RFC 1036.

   L'NNTP dipende dal TCP per garantire la consegna corretta ed ordinata dei dati nella

rete.

Il protocollo NNTP è discusso nel newsgroup news.software.nntp.
Un'implementazione largamente utilizzata di NNTP chiamata INN (InterNet News) può essere scaricata da <http://www.isc.org/inn.html>.

8. NTP

   Il Network Time Protocol (**NTP** - Protocollo di tempo in rete) viene utilizzato per sincronizzare l'orario degli orologi tra vari sistemi di computer. La versione attuale dell'NTP è la Versione 3, definita nella RFC 1305. Le versioni precedenti (2 e 1 rispettivamente) del protocollo sono definite nelle RFC 1119 e 1059. David Mills mantiene un'implementazione disponibile al pubblico di server e client NTP assieme ad una comprensiva raccola di documentazioni NTP sul web all'indirizzo <http://www.eecis.udel.edu/%7Entp/>.

   L'NTP dipende dall'UDP per trasportare i pacchetti tra i server e i clients.

   L'NTP è discusso nel newsgroup comp.protocols.time.ntp.

9. POP

   Il Post Office Protocol (**POP** - Protocollo di ufficio postale) permette ai clients di leggere e cancellare le email da una casella postale che risiede su un altra macchina server. La versione attuale del POP è la Versione 3, definita generalmente POP3. È descritto nella RFC 1939. Il POP non provvede alcuna maniera per inviare email; i programmi client che utilizzano POP per leggere la posta in genere sfruttano SMTP per inviare i messaggi. POP è più semplice e meno potente dell'altro protocollo di lettura della posta largamenete utilizzato, IMAP.

   Il POP dipende dal TCP per garantire la consegna corretta ed ordinata dei dati nella rete.

   POP non ha un suo newsgroup dedicato. Viene a volte discusso nei newsgroup dei client specifici nella gerarchia comp.mail.*.

10. Rlogin

    Il Remote Login (rlogin - Collegamento remoto) provvede un terminale di rete oppure una possibilità di collegamento remoto. Rlogin è simile a Telnet ma aggiunge un paio di caratteristiche che lo rendono un po' più conveniente di Telnet.

    Rlogin è uno dei cosiddetti comandi-r Berkeley (dove la "r" sta per remoto), una famiglia di comandi creato all'UC Berkeley durante lo sviluppo di BSD Unix per provvedere accesso a sistemi remoti in maniere più convenienti dei comandi originali TCP/IP.

    La convenienza più ovvia di rlogin, come gli altri comandi-r, è che esamina un file .rhosts (pronunciato "dot ar hosts") presente sul server per autenticare i

collegamenti basati sull'indirizzo del client. Il file .rhosts può essere costruito per permette accesso remoto senza richiedere l'inserimento di una password. Se utilizzata impropriamente, questa caratteristica può essere un problema di sicurezza, ma se utilizzata correttamente può migliorare la sicurezza perché non richiede che sia inviata una password sulla rete, che potrebbe essere letta da uno sniffer di pacchetti.

I comandi-r dipendono dal TCP per garantire la consegna corretta ed ordinata dei dati nella rete.

11. Rsh

    Il Remote Shell (**rsh** - Console remota) è un comando-r che provvede l'esecuzione remota di comandi arbitrari. Esso permette di eseguire un comando sul server senza doverst connettere al server. Ancor più importante, permette di inserire dati nei comandi remoti e ricevere l'output del comando senza dover tenere i dati in files temporanei sul server.

    Come gli altri comandi-r Berkeley, rsh utilizza il file .rhosts sul server per autenticare gli accessi basandosi sull'indirizzo del client.

    In alcuni sistemi non BSD il comando Remote Shell è chiamato **remsh** poiché al tempo in quei sistemi il comando di nome **rsh** era utilizzato per l'applicazione "restricted shell", un'interprete di riga di comando intesa a migliorare la sicurezza impedendo agli utenti di eseguire certe attività.

    Nei sistemi Unix gran parte del lavoro di rsh è gestito dalla funzione di libreria rcmd, così, se stai scrivendo che necessità di funzioni simili ad rsh, potresti utilizzare tale funzione. Tuttavia, poiché il protocollo rsh richiede che i client utilizzino una porta privilegiata, sarai in grado di utilizzare rcmd se il tuo programma viene eseguito con i permetti di superuser (amministratore). Ecco perché l'eseguibile rsh è impostato con i permessi di superuser nelle macchine Unix.

    Se il tuo programma non verrà utilizzato come root (amministratore) potresti invece utilizzare la funzione rexec, che non utilizza il file .rhosts sul server. Invece essa richiede al client di inserire una password di accesso che sarà trasmessa in maniera non criptata sulla rete.

12. SMTP

    Il Simple Mail Transfer Protocol (**SMTP** - Protocollo di trasferimento di posta semplice) viene utilizzato per consegnare email da un sistema ad un altro. Le basi dell'SMTP sono definite nella RFC 821 ed il formato dei messaggi di posta Internet è descritto nella RFC 822.

    L'SMTP dipende dal TCP per garantire la consegna corretta ed ordinata dei dati nella rete.

    Un'implementazione largamente utilizzata di SMTP chiamata **sendmail** può essere

scaricata da <http://www.sendmail.org/>. Altre implementazioni SMTP open-source includono **qmail** (disponibile su <http://www.qmail.org/>), **postfix** (disponibile su <http://ftp.planix.com/pub/Smail/>), **exim** (disponibile su <http://www.exim.org/>) e **smtpd** (disponibile all'indirizzo <http://www.obtuse.com/smtpd.html>).

13. SNMP

    Il Simple Network Management Protocol (**SNMP** - Protocollo di gestione di rete semplice) provvede un metodo di controllo e gestione di sistemi su una rete. SNMP definisce un metodo di ricezione delle domande (i primitivi GET e GET-NEXT) e i comandi (il primitivo SET) da una stazione client di gestione ad un agente server eseguito nel sistema di destinazione, e la raccolta di risposte e notifiche di eventi non richiesti (il primitivo TRAP).

    La Versione 1 dell'SNMP è definita nelle RFC 1098 e 1157. La Versione 2 dell'SNMP è definita nelle RFC 1441, 1445, 1446, 1447 e 1901 fino alla 1909. Le varie cose che possono essere controllate e gestite dall'SNMP sono assieme chiamate Management Information Base (**MIB**) e sono definite in dozzine di RFC aggiuntive.

    L'SNMP invia il traffico tramite UDP a causa della sua relativa semplicità e basso impiego.

    L'SNMP è discusso nel newsgroup comp.protocols.snmp.

14. Ssh

    Il Secure Shell (**ssh** - Console Sicura) provvede un collegamento remoto e caratteristiche di esecuzione simili a quelle dei comandi-r rsh e rlogin, ma ssh cripta i dati che sono scambiati all'interno della rete. La criptatura può proteggere informazioni importanti e non è raro che per ragioni di sicurezza gli amministratori disabilitino i servizi di rsh e telnet in favore di ssh.
    Il protocollo SSH utilizzato dal comando ssh è utilizzato anche per effettuare applicazioni di trasferimento file sicure che possono essre utilizzate alternativamente a FTP per i dati importanti.

    Informazioni complete su ssh e sul suo protocollo SSH possono essere trovate all'indirizzo <http://www.ssh.fi/>.

15. Telnet

    Telnet provvede un terminale di rete o capacità di collegamento remoto. Il server Telnet accetta dati dai client telnet e li invia al sistema operativo in modo che i caratteri ricevuti sono trattati nella stessa maniera in cui sarebbero trattati se fossero stati premuti i tasti sulla tastiera del terminale. Le risposte generate dal sistema operativo server sono ritornate al client Telnet e visualizzate.

    Il protocollo Telnet provvede la possibilità di negoziare molte tipologie di comportamenti di terminale (echo locale e remoto, modo linea oppure carattere, e altri) tra il client e il server. Il protocollo di base Telnet è definito nelle RFC 818 e

854 ed i meccanismi di negoziazione delle opzioni sono descritti nelle RFC 855.

Opzioni Telnet specifiche, notizie implementative e trucchetti del protocollo sono definiti in svariate dozzine di RFC a partire dal 1971. Come si comprende da questa presentazione Telnet è un protocollo largamente utilizzato.

Telnet dipende dal TCP per garantire la consegna corretta ed ordinata dei dati tra client e server.

16. X Window System

    L'X Window System (X11R6 - Sistema a finestre X - ultima versione) permette ai programmi client di controllare in altre macchine il video grafico, la tastiera e il mouse nei loro video-terminali X.

    X dipende dal TCP per garantire la consegna corretta ed ordinata nella rete.

    L'X Window System è discusso nel newsgroup comp.winsows.x.

Programmazione TCP/IP

1. Cosa sono i socket?

   Un socket è un'astrazione che rappresenta un punto di comunicazione finale. Molte applicazioni che utilizzano TCP e UDP lo fanno creando un socket del tipo appropriato ed in seguito eseguono una serie di operazioni su quel socket. Le operazioni che possono essere eseguite su un socket includono operazioni di controllo (come l'associazione di un numero di porta al socket, l'inizializzazione o l'accettazione nel socket, oppure la distruzione del socket), operazioni di trasferimento (come la scrittura dei dati attraverso il socket in un'altra applicazione, oppure la lettura di dati da un'applicazione attraverso il socket) ed operazioni di stato (come il ritrovamento dell'indirizzo IP associato al socket).

   L'insieme completo di operazioni che possono essere eseguite su un socket costituisce il **Sockets API** (Application Programming Interface). Sei sei interessato o stai scrivendo un programma che utilizza TCP/IP probabilmente ti sarà necessario ed utile utilizzare e conoscere il **Sockets API**. Una lista di FAQ riguardo la programmazione dei socket è disponibile sul web all'indirizzo <http://www.ibrado.com/sock-faq/>, oppure al suo mirror inglese <http://kipper.york.ac.uk/%7Evic/sock-faq/> o via FTP anonimo su <ftp://rtfm.mit.edu/pub/usenet/news.answers/unix-faq/socket>.

2. Come posso rilevare che l'altro capo di una connessione TCP ha generato un errore?

   Il rilevamento di errori nel sistema su TCP/IP è difficile. TCP non richiede nessuna trasmissione su connessione se l'applicazione non sta inviando niente, e molti media su cui il TCP/IP è utilizzato (come Ethernet) non provvedono un modo pratico per rilevare se un particolare host è funzionante. Se un server non riceve nulla dal client, potrebbe significare che il client non ha niente da dire, alcune reti tra il client ed il

server sono down, il client può essere down, le interfacce tra client e server possono essere scollegate, oppure il client ha generato un errore. I problemi di rete sono spesso temporanei (un thin Ethernet può sembrare down quando qualcuno aggiunge un collegamento alla catena, scollegando temporaneamente gli altri, e possono passare alcuni minuti prima che le postazioni si ripristinino) e le connessioni TCP non dovrebbero essere scollegate per queste motivazioni.

Il Keepalive è una caratteristica del socket API e richiede che sia inviato periodicamente un pacchetto vuoto su una connessione vuota (idle); questo dovrebbe informare che il sistema remoto è ancora attivo, inviare un reset quando il sistema è riavviato oppure produrre un timeout quando il computer è down. Questo normalmente non è inviato fino a che la connessione non è rimasta vuota (in idle) per un paio d'ore. Lo scopo non è quello di rilevare un problema immediatamente, ma di non mantenere risorse inutili allocate per sempre.

Se sono richieste caratteristiche di rilevazione di problemi remoti più rapide, questo dovrebbe essere implementato nel protocollo di applicazione. Non esistono meccanismi standard, ma un esempio è quello di richiedere ai client di inviare un messaggio **no-op** ogni minuto o due. Un protocollo di esempio che utilizza questo è l'X Display Manager Control Protocol (**XDMCP** - Protocollo di controllo della gestione del display X), parte dell'X Window System Versione 11; il server XDM gestisce una sessione mandando periodicamente un comando Sync al video del server, che dovrebbe inviare una risposta all'applicazione oppure resettare la sessione se non riceve risposta.

3. Può essere configurato il TCP keepalive timeout?

   Questo varia da un sistema operativo all'altro; esiste un programma che funziona su molti Unix (ma non Linux o Solaris), chiamato netconfig che permette di effettuare quest'operazione. È disponibile via FTP anonimo all'indirizzo <ftp://cs.ucsd.edu:/pub/csl/Netconfig/netconfig2.2.tar.Z>.

4. Ci sono strumenti di programmazione di reti orientati ad oggetti

   Si. Uno di questi sistemi è ADAPTIVE Communication Environment (**ACE**). Il file README per ACE è disponibile nel web all'indirizzo <http://www.cs.wustl.edu/%7Eschmidt/ACE.html>. Tutto il software e la documentazione è disponibile sia via FTP anonimo che web.

   ACE è disponibile via FTP anonimo su <ftp://ics.uci.edu/gnu/C++_wrappers.tar.Z>. Questo è un archivio TAR compresso di circa 500KB di grandezza. Questa versione contiene il codice sorgente, la documentazione, gli esempi per librerie di collegamento C++.

Documento originale: TCP/IP Frequently Asked Question
Traduzione di Fibia FBI
10 Marzo 2001

Torna all'indice degli Articoli

