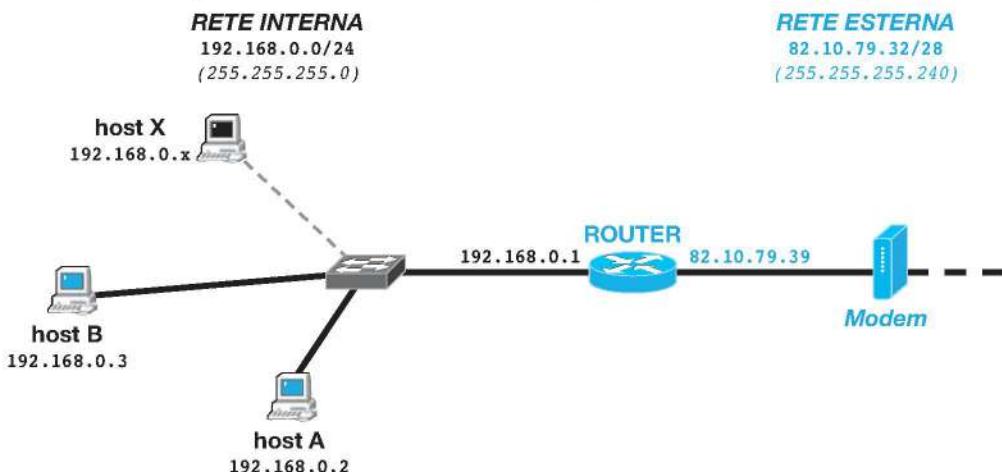


# A6

## Internetworking

Dopo aver analizzato i principali protocolli di livello 7 Applicazione operanti all'interno delle reti private, è necessario analizzare le tecniche utilizzate per dotare le reti private del supporto per la connessione alla rete pubblica, intesa quasi sempre come Internet.

Come si è visto nel Volume 2 (*Sistemi Reti, vol. 2, Sez. B, Il livello 3: Rete*), la connessione di reti differenti avviene tramite il dispositivo di livello 3 denominato **router**: per connettere una rete privata a una rete pubblica, pertanto, è necessario un router, ovvero un apparato di rete che possegga almeno due interfacce, una sulla rete privata e una sulla rete pubblica. In molti casi (per esempio per reti casalinghe) il router può essere direttamente un computer avente una delle sue interfacce di rete connessa a un modem (per esempio un modem ADSL) per l'accesso alla rete pubblica.



Per tutta una serie di ragioni, però, il solo router non è sufficiente a garantire un internetworking efficace tra rete interna ed esterna.

Prima di tutto rimane la questione della scarsità degli indirizzi IP: i molti host di una rete privata non possono, da tempo, essere configurati con indirizzi IP pubblici, nemmeno se ottenuti dinamicamente. È necessario che una rete privata si doti di un meccanismo per concedere agli host interni di usufruire dei servizi resi dalla rete pubblica (per esempio **NAT**).

In secondo luogo è estremamente serio il problema della sicurezza: sulla rete pubblica operano milioni di utenze, tra le quali anche quelle ansiose di intercettare, violare o accedere ai dati contenuti nelle reti private che si affacciano al dominio pubblico. È necessario dotare le reti private di sistemi in grado di assicurare la rete da intrusioni indesiderate pur concedendo l'accesso alle utenze legittime (per esempio **firewall**).

Infine il traffico di una rete privata verso la rete pubblica può essere anche molto sostenuto e variegato; il bitrate di una rete privata è senz'altro superiore al bitrate di una rete pubblica e lo stesso canale di connessione è utilizzato continuamente in contemporanea da numerose utenze. È spesso necessario dotare una rete privata di meccanismi per ottimizzare e controllare il traffico verso e dalla rete esterna (per esempio **proxy**).

## 1 NAT

L'indirizzamento IP degli host di una rete tramite *indirizzi privati* sottrae tutti gli host della rete, di fatto, dalla partecipazione alla rete pubblica: quegli indirizzi non sono univoci su scala mondiale e non appartengono all'indirizzamento TCP/IP di Internet.

Viene così definita una tecnica per cui un processo operante su un router è in grado di sostituire gli indirizzi IP nei pacchetti che giungono da una interfaccia (per esempio su una rete privata) in indirizzi IP differenti e poi inoltrati su un'altra interfaccia (per esempio, su una rete pubblica).

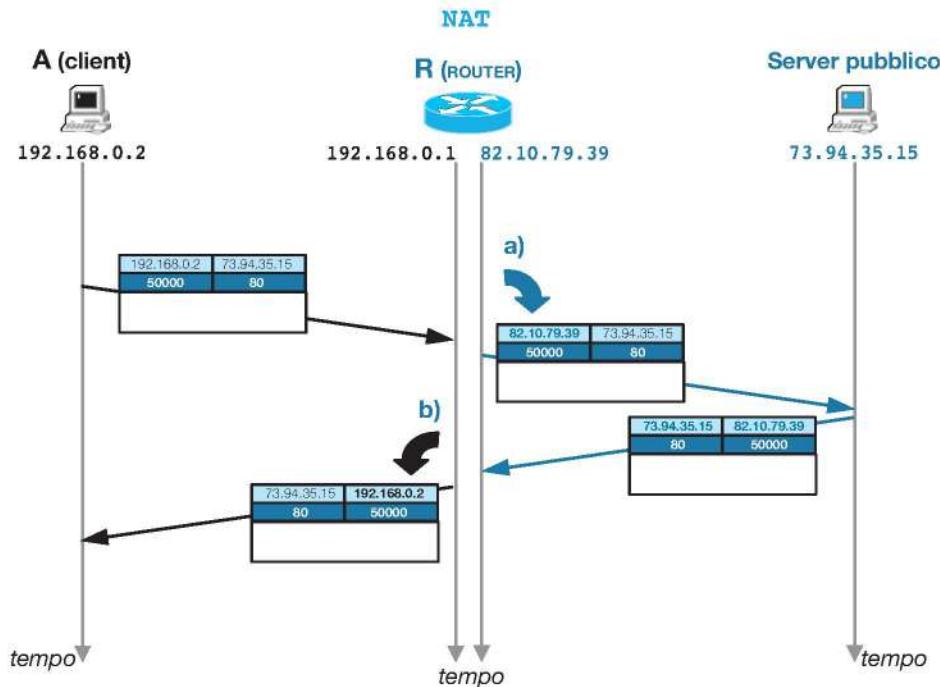
Questa tecnica è detta **NAT** (*Network Address Translation*, RFC 1631).

Non si tratta di un protocollo, ma di una programma, o meglio, di un processo.

Ora, se un host con un indirizzo IP privato tentasse una connessione TCP/IP verso un host con indirizzo IP pubblico attraversando un router provvisto di NAT, il router potrebbe:

- cambiare l'IP sorgente** (privato) con il proprio IP (pubblico), inoltrare il pacchetto sulla rete pubblica, attendere le risposte e, su queste,
- sostituire** nei pacchetti ricevuti l'indirizzo IP destinazione (proprio) con l'indirizzo IP (privato) dell'host e inoltrarlo a esso sulla rete privata.

In questo modo l'host privato ottiene il servizio di rete pubblica:



### ICS

Una forma rudimentale di NAT (source NAT) può essere realizzata «nativamente» anche sui sistemi operativi non server di Windows (per esempio Windows 98 e successivi). Questa implementazione di source NAT è detta **ICS** (*Internet Connection Sharing*).

Se per esempio una piccola rete privata con qualche host su uno switch, ne prevede uno con una connessione a un modem ADSL per Internet, questa connessione può essere «nattata», ovvero condivisa con altri host. Sull'host connesso a Internet è sufficiente impostare una «spunta» nella cartellina «Condivisione» delle proprietà dell'interfaccia di rete che sta offrendo la connessione pubblica.

Gli altri host della LAN dovranno impostare come *gateway di default* l'indirizzo IP (privato) dell'host che ha condiviso la connessione pubblica.

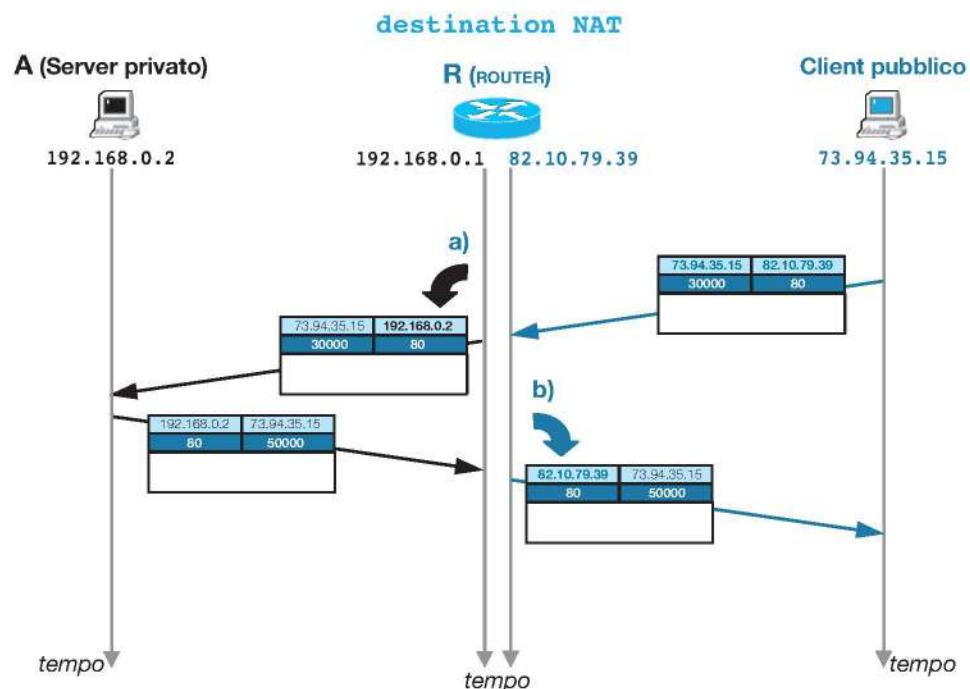
Ora, se tutti i client della rete privata intendono usufruire di NAT, il router sarebbe costretto a mapparne le richieste (per ricordarne le restituzioni) basandosi sul numero di porta effimera utilizzata nei pacchetti client (nel disegno, 50000).

Infatti l'ipotesi più sfavorevole è che due o più client vogliano connettersi sulla stessa porta di un server pubblico (per esempio Google): al router non rimarrebbe che «ricordare» le connessioni in entrata in base alla porta TCP sorgente (la porta effimera), l'unico parametro identificativo del client. A rigor di logica, anche le porte effimeri potrebbero coincidere (sono infatti decise casualmente in locale ai client).

Un NAT effettivo, quindi, mappa le connessioni sulle porte effimeri, e prende il nome di **PAT** (*Port Address Translation*, oppure **IP masquerading**, oppure ancora **NAPT**, *Network Address and Port Translation*). In questo caso la porta effimera (porta TCP sorgente) viene effettivamente modificata dal server NAT in modo che sia univoca e che possa contraddistinguere i pacchetti di ritorno per la riconsegna all'host originario.

Naturalmente è possibile che un router effettui l'operazione opposta, ovvero mappi le connessioni provenienti dall'esterno e dirette a un suo indirizzo pubblico verso l'indirizzo privato di un host interno alla rete privata.

In questo caso si ottiene che un host privato fornisce un servizio alla rete pubblica (per esempio un server Web).



In questo caso si parla di **destination NAT**, duale al NAT/PAT (che spesso, per questo motivo, è anche chiamato *source NAT*).

A volte l'host privato che fornisce il servizio pubblico è detto «host pubblicato».

## 2 Sicurezza NAT

Il NAT, come processo complessivo, induce alcune considerazioni:

- obbliga i router a mantenere grandi tabelle di correlazione, quando invece i router non hanno questo scopo. Ciò significa che i router che realizzano NAT sono sottoposti ad alte sollecitazioni prestazionali.
- La modifica dell'indirizzo IP nell'header IP implica il ricalcolo della checksum IP (e TCP/UDP), aumentando l'onere di elaborazione da parte dei router.
- Qualche protocollo negozia gli indirizzi IP o le porte TCP/UDP: NAT, effettuando una sostituzione «alla cieca», non è in grado di gestire questi protocolli (dovrebbe infatti analizzare il contenuto dei pacchetti e calcolare quali sostituzioni fare).

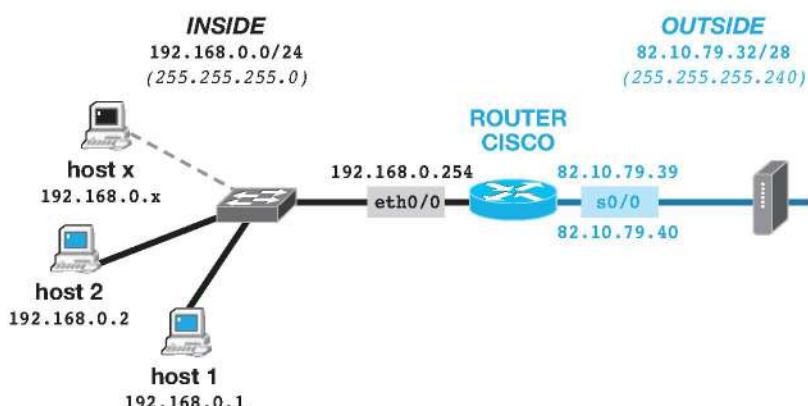
In generale il servizio di NAT offre qualche attributo di sicurezza. In fondo gli host privati non si presentano mai con il loro reale indirizzo IP sulla rete pubblica e, almeno in teoria, il source NAT impedisce «nativamente» a macchine pubbliche di raggiungere le macchine private della rete. Inoltre, pur rappresentando un «*single point of failure*» (un aspetto critico perché il malfunzionamento di un solo elemento della rete determina il malfunzionamento di tutti gli elementi della rete), il server NAT, se ben configurato, riflette automaticamente la sua configurazione di sicurezza a tutti gli host interni senza che questi debbano preoccuparsene espressamente.

### ESEMPIO

#### NAT con router CISCO

Nella terminologia CISCO la rete interna è denominata *Inside*, mentre la rete esterna *Outside*.

Si nota che il router ha una configurazione **multihomed** single link, ovvero su un'interfaccia sono configurati più indirizzi IP (in questo caso due indirizzi IP pubblici).



Prima di tutto vanno configurati i ruoli delle interfacce del router:

```
Router(config)#interface ethernet 0/0
Router(config-if)#ip nat inside
Router(config)#interface serial 0/0
Router(config-if)#ip nat outside
```

Quindi si può passare a configurare una delle tre modalità di NAT disponibili sui router CISCO:

```
Router(config)#ip nat inside source static 192.168.0.1 82.10.79.39
```

Questa configurazione è detta **Static NAT**.

Il comando IOS può essere interpretato come: esponi l'indirizzo interno (inside, 192.168.0.1) tramite l'indirizzo esterno 82.10.79.39.

Si tratta di un **destination NAT**. L'utilizzo tipico è di esporre un server interno in modo che sia accessibile da Internet.

```
Router(config)#access-list 15 permit 192.168.0.0 0.0.0.255
Router(config)#ip nat pool IPEXT 82.10.79.39, 82.10.79.40
Router(config)#ip nat inside source list 15 pool IPEXT
```

Questa configurazione è detta **Dynamic NAT**.

Il primo comando IOS definisce una ACL (Access Control List) di identificativo 15 composta da tutti gli host privati a cui è concesso un permesso; quindi il secondo comando imposta un gruppo (pool) di due indirizzi IP pubblici, infine l'ultimo comando imposta un **source NAT** in modo tale che i vari host privati si presentino sulla rete pubblica con uno di quei due indirizzi IP pubblici.

```
Router(config)#access-list 15 permit 192.168.0.0 0.0.0.255
Router(config)#ip nat inside source list 15 interface serial 0/0 overload
```

Questa configurazione è detta **Overloading NAT**.

Dopo avere impostato l'ACL di tutti gli host privati, il secondo comando consente agli host privati di presentarsi sulla rete pubblica tramite un unico indirizzo IP pubblico impostato sull'interfaccia esterna (*in questo caso il router non deve essere multihomed*). Anche in questo caso si tratta di un **source NAT**.

### 3 Firewall

Anche in questo caso siamo di fronte a un processo, e non a un protocollo applicativo.

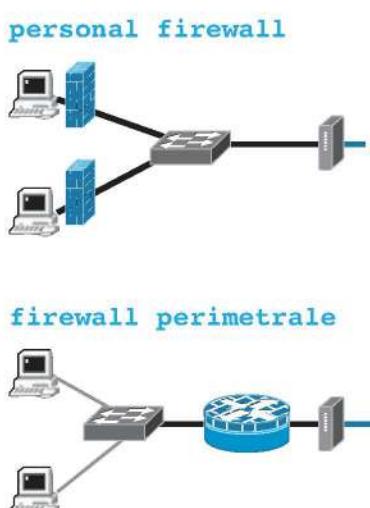
Per una rete il termine **firewall** indica un processo attivo su un sistema in grado di controllare, a vari livelli, il contenuto dei pacchetti in entrata e in uscita su una o più interfacce di rete al fine di impedire il transito di pacchetti non autorizzati. L'uso di un sistema firewall ha lo scopo di proteggere una rete o una parte di rete considerata sicura da possibili intrusioni più o meno volontarie e/o dannose che possono compromettere la sicurezza o la funzionalità della rete da proteggere.

I sistemi firewall possono essere semplicemente programmi software (**personal firewall**), ovvero programmi da installare su uno o più host della rete sicura. Il programma ha lo scopo di proteggere l'host e controllerà il traffico sull'unica scheda di rete dell'host.

Il firewall più propriamente detto, invece, è anche noto come **firewall perimetrale**, dato che deve essere installato su un apparato router che ha accesso alla rete pubblica.

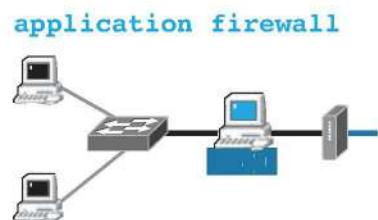
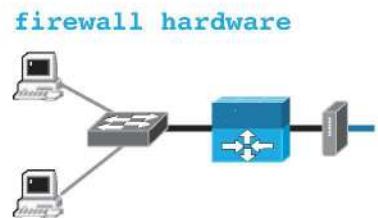
In questo caso il processo firewall opera su almeno due interfacce di rete del router, una affacciata sulla rete sicura (normalmente una rete privata detta anche *trust*), l'altra connessa a una rete insicura, tipicamente Internet.

In altri casi il sistema firewall perimetrale è costituito da un apparato hardware con almeno due interfacce di rete sulle quali il processo effettua il controllo (**firewall hardware**). Anche in questo caso



una delle due interfacce è posizionata su una rete *trust* e l'altra sulla rete pubblica. Ora l'apparato firewall perimetrale di tipo hardware svolge anche la funzione di router e può sostituirlo come apparato di interconnessione tra una rete privata e una rete pubblica.

Infine un firewall perimetrale può essere una vera e propria applicazione di livello 7 (**application layer firewall**) che opera su una macchina server dotata di più interfacce di rete (macchina server agente da router), magari integrato con applicazioni di rete specifiche che effettuano anche il controllo delle connessioni e l'ottimizzazione del traffico (cfr. **Proxy** più oltre).



### 3.1 Livelli di controllo

Di norma si tende a distinguere i processi di firewalls in alcune grandi categorie in base al tipo di ispezione che esercitano:

- firewall **packet filter**, funzionalità minima che controlla il traffico a livello di indirizzamento IP e ispeziona del flag SYN di TCP per intercettare tentativi di connessione. Livello indispensabile per bloccare o permettere il traffico tra subnet e/o reti IP pubbliche;
- firewall **stateful inspection**, in grado di valutare la qualità delle connessioni TCP tramite il controllo approfondito dell'header dei pacchetti TCP, ovvero analisi fino a livello 4 di Trasporto. Questa funzionalità consente di intercettare traffico TCP che apparentemente appartiene a una connessione ma in realtà costituisce un tentativo di intrusione;
- firewall **deep inspection**, in grado di analizzare il traffico dei pacchetti fino a livello 7 Applicazione, analizzando il corretto uso dei protocolli applicativi e ispezionandone anche il contenuto. In base a un glossario su database, questo tipo di processo riesce a intercettare le firme dei virus o degli applicativi d'intrusione più diffusi. Inoltre è in grado di bloccare il traffico per determinati siti o in base a un elenco di parole chiave.

Queste funzionalità richiedono una cospicua capacità prestazionale, pertanto un processo firewall di questo tipo è presente, solitamente, su apparati dedicati o firewall realizzati a livello applicativo su host dedicati (*proxy server*).

### 3.2 ACL

**ACL** (*Access Control List*) o *lista di controllo degli accessi* è un concetto legato alla sicurezza informatica, ed esprime la lista dei permessi d'uso legati a un determinato oggetto.

L'ACL è impiegata nei sistemi operativi (per esempio, per esprimere i permessi di accesso al file system) e nelle reti, sottoforma di regole da valutare in modo tale che un processo possa poi prendere una decisione coerente. Ogni singolo criterio di una ACL è detto **ACE** (*Access Control Entry*) o *regola*.

I firewall controllano il traffico circolante sulle interfacce in base ad ACL impostate dall'amministratore di rete.

#### Netshell

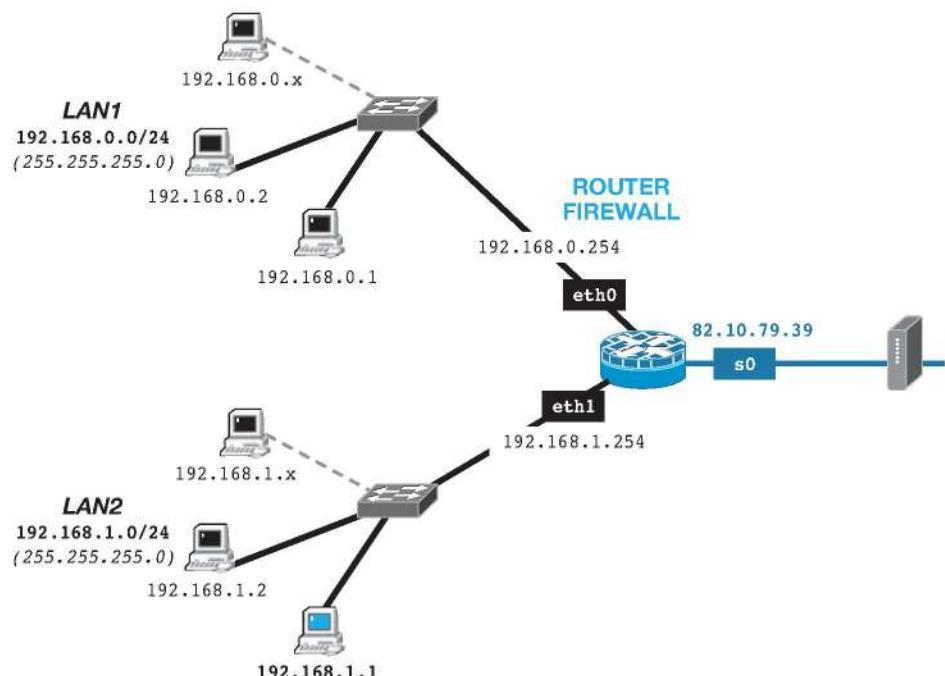
Sui sistemi operativi Windows è disponibile un'utile applicazione di rete denominata **netshell**, invocabile su linea di comando con il nome **netsh**. *Netsh* offre un ambiente interattivo a contesto che consente di visualizzare numerosissime informazioni sulle configurazioni di rete dell'host su cui è eseguito.

Inoltre *netsh* è in grado di interpretare script di configurazione contenuti in file di testo. Sui sistemi Windows Server e Windows Vista e successivi, è possibile visualizzare le ACL del personal firewall fornito dal sistema operativo con il comando:

```
netsh advfirewall firewall  
show rule name=all.
```

## Firewall e ACL

Data una rete con due subnet private e un firewall perimetrale, elencare una ACL per il firewall/router in modo tale che una subnet non sia raggiungibile da Internet e nell'altra un solo host sia raggiungibile da Internet con il protocollo HTTP (porta TCP 80):



**ESEMPIO**

interfaccia IN	interfaccia OUT	IP source	IP dest	Proto	source Port	dest Port	Azione
s0	eth0	*	*	*	*	*	deny
s0	eth1	*	192.168.1.1	TCP	*	80	permit
*	*	*	*	*	*	*	deny

L'uso dell'asterisco (\*) indica «qualsiasi valore» per quel campo; nel gergo dei firewall si usa anche il termine **any**.

Si nota che l'ultima regola dell'ACL (regola di default) blocca qualsiasi altro traffico sulle interfacce del firewall/router. Ciò significa, per esempio, che le due subnet non sono comunicanti, e anche che nessun host di questa rete privata può accedere a Internet.

Affinché l'host 192.168.1.1 possa essere raggiunto da Internet, il firewall/router deve realizzare anche la funzione di NAT per esporre l'host sulla rete pubblica (*destination NAT*). Nel gergo delle reti si dice anche che l'host 192.168.1.1 è **nattato** (o anche *pubblicato*) su Internet.

## 4 Sicurezza firewall

Il firewall è un servizio progettato per fornire sicurezza a una rete. A volte l'apparato su cui gira un processo di firewall, apparato che è quindi esposto alla rete pubblica, è detto **bastion host**. Queste macchine devono essere particolarmente attrezzate per respingere i tentativi di intrusione, quindi devono possedere un sistema operativo sicuro (per esempio, sempre aggiornato rispetto alle *patch* rilasciate dal produttore), con pochissimi account d'accesso e un numero minimo di processi in esecuzione.

Tra i tentativi di intrusione che un firewall può tentare espressamente di contrastare si ricorda l'**IP spoofing**, quella tecnica per cui si tenta di accedere ai servizi di una rete simulando un'identità dei pacchetti diversa da quella reale. Ciò si ottiene inviando pacchetti TCP/IP con l'indirizzo IP sorgente modificato (falsificato).

Obiettivo di vari tipi di pirateria informatica è il **DoS (Denial of Service)**, ovvero un'attività intrusiva il cui scopo è mettere in difficoltà un servizio di rete sottoponendolo a stress eccessivo o sfruttandone alcune caratteristiche malfunzionanti.

Una tipica attività di DoS attuata verso i firewall riguarda l'uso distorto del programma *ping* (**IP smurfing**). Il software «distorto» invia una grande quantità di pacchetti ICMP *Echo Request*, tutti con IP sorgente modificato (*spoof*) e indirizzo destinazione un IP broadcast diretto. Se il router/firewall non è impostato correttamente per ignorare pacchetti ICMP con destinazione IP broadcast, inoltra il pacchetto a tutte le sue interfacce e la richiesta ICMP si diffonde velocemente sia sulla rete privata sia sulla rete pubblica su cui si affaccia il router. Siccome la destinazione è broadcast, molti host risponderanno alla richiesta inviando pacchetti ICMP *Echo Reply* all'indirizzo IP mittente (falsificato), costringendolo a una attività stressante.

Per questa ragione i firewall hanno spesso una ACL che gli impone di non inoltrare pacchetti ICMP con IP destinatario un indirizzo di broadcast.

### Windows smurf

Sui sistemi operativi Windows è stata risolta la vulnerabilità alle attività di IP smurfing a partire dall'aggiornamento *XP Service Pack 3*.

### ACL antispoofing e antismurf

Se dalla rete pubblica giungono pacchetti TCP/IP provenienti da reti private, ciò significa che probabilmente si è di fronte a una intrusione IP Spoofing, dato che è impossibile che dalla rete pubblica giungano pacchetti originati da reti private.

Riferendosi alla rete riportata nell'Esempio precedente:

interfaccia IN	interfaccia OUT	IP source	IP dest	Proto	source Port	dest Port	Azione
s0	eth0, eth1	192.168.0.0/16	*	*	*	*	deny
s0	eth0, eth1	172.16.0.0/16	*	*	*	*	deny
s0	eth0, eth1	10.0.0.0/8	*	*	*	*	deny

L'ACL implementata per sistemi router con firewall CISCO potrebbe essere:

```
Router(config)#access-list 105 deny ip 10.0.0.0 0.255.255.255 any
Router(config)#access-list 105 deny ip 172.16.0.0 0.15.255.255 any
Router(config)#access-list 105 deny ip 192.168.0.0 0.0.255.255 any
Router(config)#interface s0
Router(config)#ip access-group 105 in
```

L'ACL realizzata invece con iptables/netfilter su un sistema Linux:

```
iptables -A FORWARD -i s0 -s 192.168.0.0/16 -j DROP
iptables -A FORWARD -i s0 -s 172.16.0.0/16 -j DROP
iptables -A FORWARD -i s0 -s 10.0.0.0/8 -j DROP
```

Sui router CISCO per evitare lo smurf si impone al router/firewall di scartare i pacchetti ICMP con IP destinazione un indirizzo broadcast con un comando ad hoc (normalmente attivo di default):

```
Router(config)#no ip directed-broadcast
```

Per Linux invece si può scrivere l'ACL:

```
iptables -A FORWARD -i eth0 -p icmp -icmp-type ping -d 255.255.255.255 -j DROP
```

## 5 Proxy

Quando il processo di firewallsing è svolto a livello 7 Applicazione ([Application Layer firewall](#)), molto spesso ci si trova di fronte ad apparati che integrano varie funzionalità di rete, come:

- il *routing* (tra rete privata e rete pubblica);
- l'*esposizione di client privati* su rete pubblica (analogamente a NAT);
- il rilevamento e il *controllo degli accessi*.

In particolare, quando un'attività di routing viene realizzata a livello 7 Applicazione, il sistema non viene più denominato router, ma [gateway](#)<sup>(NB)</sup>.

Queste, sostanzialmente, sono le caratteristiche di un servizio complesso denominato [proxy](#) (*proxy server*), normalmente realizzato tramite applicazioni utente il cui uso è riservato agli amministratori di rete.

Come per NAT e firewall, la funzionalità proxy non è un protocollo di livello 7 Applicazione, ma un processo applicativo. Esso deve essere avviato su una piattaforma hardware ad alte prestazioni (per esempio un computer server) per poter fornire tutti i servizi promessi. Le prestazioni dei firewall applicativi si misurano in [PPS](#) (*Pacchetti Per Secondo*, da 100 000 a 1 milione, 2012).

NB. Il termine [gateway](#), in effetti, trova la sua accezione originale proprio in questa definizione.

L'uso del termine [gateway](#) come sinonimo di [router](#) (per esempio, *gateway di default*) ingenera a volte qualche ambiguità sull'uso del termine.

L'application firewall agente su un proxy, oltre alle consuete attività di controllo dei pacchetti e delle connessioni (*packet filter* e *stateful inspection*), è in grado di approfondire l'analisi del contenuto dei pacchetti (*deep inspection*) individuando comunicazioni sospette in base alle stringhe contenute nei dati o a particolari patterns impostabili (per esempio, firme di riconoscimento per i virus informatici). La funzionalità di livello 7 consente a questi sistemi di analizzare e controllare i contenuti nei principali protocolli applicativi utente, come per esempio il protocollo per il Web (HTTP), la posta elettronica (POP3 e IMAP) e il trasferimento file (FTP). Con semplici impostazioni di management, i proxy possono anche bloccare interi siti Web ritenuti non idonei o aggiornare i propri criteri di controllo importando periodicamente database appositi di criteri alfanumerici (per esempio, *black list*).

Per far accedere gli host privati alla rete pubblica il proxy usa un modo molto differente da NAT. Per usare un servizio di proxy gli host privati devono essere configurati espressamente per usufruire dei servizi su rete pubblica, mentre NAT è completamente *trasparente* agli host privati (nessun management sull'host).

Un client proxy (un host della rete privata) deve essere configurato in modo tale da raggiungere la macchina server proxy, attraverso un protocollo di livello 7 (quello che l'host vuole utilizzare su rete pubblica) e un numero di porta TCP attraverso la quale comunicare con il proxy server.

Ora il client può effettuare le richieste di accesso alla rete pubblica (tipicamente con connessioni TCP/IP) tramite proxy.

Tale richiesta giunge al processo proxy che si connette all'host della rete pubblica con la propria identità pubblica (IP pubblico del proxy) ed evade la richiesta per conto del client privato. Il processo proxy colloquia con l'host pubblico al posto del client privato e reindirizza le risposte dell'host pubblico all'host privato che quindi può usufruire del servizio della rete pubblica. Nello schema seguente è mostrato un frammento dell'uso di un proxy HTTP da parte di un host della rete privata. L'applicazione client (un browser) deve essere configurata per effettuare una richiesta *modificata* al sistema proxy, invece di cercare di parlare con la destinazione finale. Le richieste «modificate» sono espressamente previste dal protocollo, in questo caso HTTP<sup>(NB)</sup>.

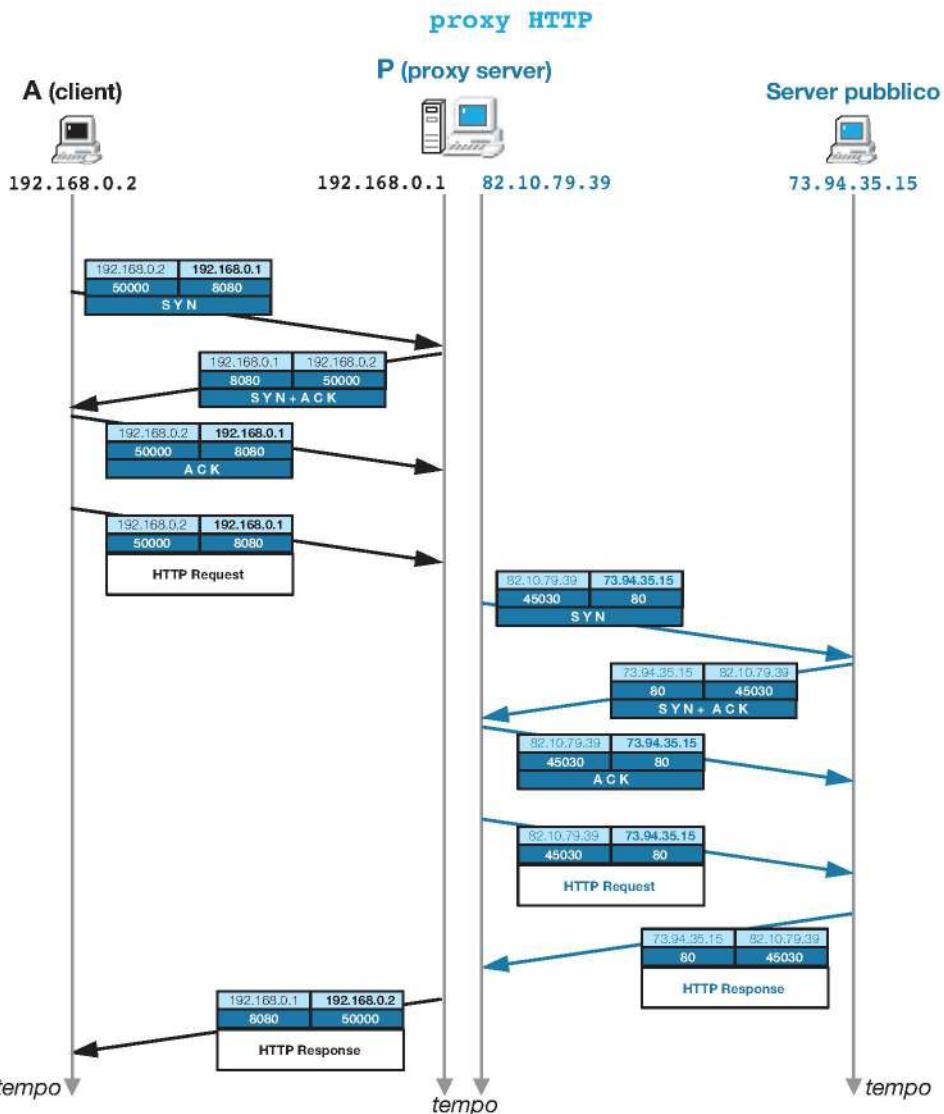
NB. Se la richiesta del client fosse stata fatta senza proxy, l'indirizzo IP del server sarebbe stato immesso direttamente nei pacchetti IP della connessione TCP (previa consultazione del DNS) e la richiesta della pagina in HTTP sarebbe stata:

```
GET /wiki/Pagina_principale HTTP/1.1
User-Agent: Mozilla/5.0
Host: it.wikipedia.org
```

Se invece la richiesta di un client viene fatta con un browser impostato per utilizzare un proxy server HTTP, la richiesta sarebbe stata:

```
GET http://it.wikipedia.org/wiki/Pagina_principale HTTP/1.1
User-Agent: Mozilla/5.0
Host: it.wikipedia.org
```

La differenza fondamentale tra le due richieste è la presenza di un URL totalmente qualificato (FQDN) nella prima riga della richiesta effettuata tramite proxy ([http://it.wikipedia.org/wiki/Pagina\\_principale](http://it.wikipedia.org/wiki/Pagina_principale)), che indica al proxy dove collegarsi al posto dell'utente (e quindi come reperire l'indirizzo IP del server Web pubblico a cui connettersi).



## 5.1 Reverse proxy

Esattamente come il DNAT (*destination NAT*) consentiva a host privati di fornire servizi pubblici pur non disponendo di un indirizzo IP pubblico, anche per i processi proxy esiste il servizio di **reverse proxy** che consente a host pubblici di utilizzare i servizi implementati sugli host di una rete privata (per esempio, un server Web).

Reverse proxy funziona esattamente come proxy, sebbene in verso opposto: un host pubblico richiede un servizio di rete connettendosi all'indirizzo pubblico del proxy server, quindi il reverse proxy crea una nuova connessione verso l'host privato che supporta effettivamente quel servizio duplicando la richiesta e rispondendo all'host pubblico con le risposte ricevute dall'host privato. Anche in questo caso deve essere fatta un'opportuna configurazione sull'host privato che offre il servizio pubblico.

Naturalmente il reverse proxy è in grado di ispezionare i pacchetti, controllare la bontà delle connessioni ed effettuare il logging degli accessi delle connessioni ricevute dall'esterno.

Inoltre il servizio di cache offerto dal reverse proxy risulta estremamente efficiente: dopotutto le richieste pubbliche all'host privato tendono a essere ripetitive (per esempio, la pagina principale del sito Web).

In generale l'approccio al routing/firewalling tramite proxy consente numerosi vantaggi rispetto all'approccio base router/firewall/NAT, come per esempio un controllo del traffico molto più accurato (deep inspection, black list, blocco dei siti, ecc.).

I processi proxy, per loro natura, forniscono un efficace servizio di **cache**, dato che il traffico richiesto dai client viene memorizzato sul server proxy e può essere restituito immediatamente nel caso una richiesta successiva fosse identica (per il traffico Web, si pensi a tanti client privati che accedono allo stesso motore di ricerca pubblico...).

D'altra parte un processo proxy è molto oneroso da un punto di vista computazionale per la macchina che lo ospita e richiede speciali configurazioni da effettuare sul lato client da parte degli utenti (non è trasparente). Inoltre un proxy server non sempre supporta ogni tipo di protocollo applicativo e, per ognuno, necessita di un processo specifico.

## 6 Sicurezza proxy

Un processo proxy offre nativamente un importante supporto per la sicurezza: il controllo completo dell'accesso alla rete pubblica a vari livelli, prima di tutto a livello utente.

Quando un server proxy riceve una richiesta di servizio può attivare un processo di autenticazione nei confronti dell'host o dell'utente che ha effettuato la richiesta (cosa impossibile per NAT, dato che è completamente trasparente). L'autenticazione può essere agganciata ai database di gestione degli oggetti operanti sul sistema di rete (per esempio, LDAP di *Active Directory* per reti Microsoft o Linux/Samba) e possono essere impostate numerose politiche di controllo e autenticazione in base alle esigenze della rete.

Un servizio proxy offre anche un completo sistema di *logging* (traccia temporale degli accessi memorizzata su disco) che consente di espletare le varie incombenze legislative che sono richieste agli amministratori di rete, consentendo le più svariate attività di **audit**.

Per i sistemi operativi di tipo Windows Server l'applicativo proxy più utilizzato è *Microsoft Forefront Threat Management Gateway* (**Forefront TMG**, ex *MS ISA Server*), mentre per i sistemi Linux l'applicativo proxy classico è **Squid**.

### Audit

Gli **audit** sono attività di valutazione e/o di controllo attivate su precise basi di analisi come raccolte dati sistematiche, informazioni standardizzate, protocolli di riferimento.

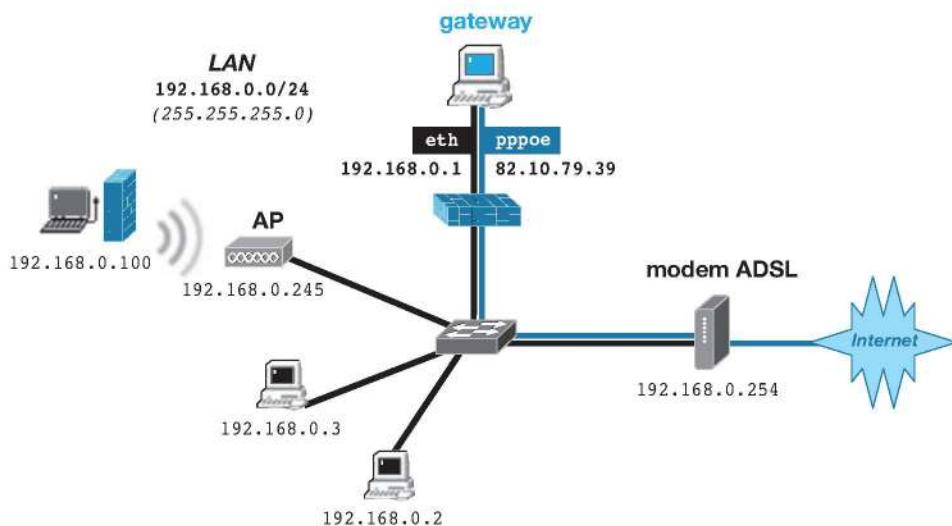
Lo scopo di un audit è verificare l'efficienza o la correttezza dei servizi resi da un'organizzazione, sia a scopo interno (per migliorare il servizio), sia a scopo esterno (per rilevare infrazioni o violazioni normative). Le attività di audit informatico spesso riguardano i servizi di sicurezza attivati nei sistemi informatici a livello di singole macchine (per esempio, sistemi server) o delle reti in cui operano (accessi, utenti, applicazioni, dati).

Alla luce dei vari processi descritti che consentono un internetworking efficiente e (relativamente) protetto tra reti TCP/IP private e pubbliche, sono stati proposti nel tempo vari modelli di configurazione per le reti private affinché possano organizzare la disposizione ottimale degli apparati di interconnessione e di controllo.

A partire da una rete privata di carattere domestico, con poche macchine e una connessione pubblica ADSL residenziale fino a reti private di grandi dimensioni con connessioni pubbliche tramite interfacce ad alta velocità, la disposizione e l'organizzazione di router, firewall e proxy risulta fondamentale.

## 1 Reti residenziali

Un primo schema per una piccola rete residenziale prevede un modem ADSL, un host gateway (router), un apparato AP (Access Point) e alcuni host paritari (Personal Computer e/o laptop):



In questo schema un host server denominato **gateway** (non necessariamente equipaggiato con un sistema operativo server) attraverso un'unica interfaccia fisica (di tipo Ethernet) possiede due indirizzi IP (*multihomed single link*): uno assegnato per la rete privata, l'altro ottenuto dal DHCP di un ISP (indirizzo IP pubblico dinamico) presso il quale è stato effettuato un contratto di accesso a Internet.

### Configurare gli apparati

Gli apparati di rete come switch, modem ADSL, router, Access Point ecc. sempre più spesso sono configurabili, dopo l'acquisto, tramite il browser di un PC, digitando nella barra degli indirizzi uno speciale indirizzo IP riportato sulla confezione. All'interno del firmware dell'apparato (spesso una distribuzione Linux *embedded*) è attivo un **server Web** che risponde alle richieste effettuate all'indirizzo IP di cui sopra. La prima pagina proposta è la richiesta di un login di sicurezza (tipo *user name/password*) a cui rispondere con i dati riportati anch'essi sulla confezione. Questi tre valori (indirizzo IP, user name e password) sono i dati da conservare per poter agire sulla configurazione dell'apparato anche in futuro.

La connessione pubblica proveniente dal modem ADSL raggiunge il gateway attraverso lo strato fisico 802.3 (Ethernet) che veicola un livello 2 punto-punto di tipo PPP (cfr. *Sistemi e reti, vol. 1, Protocollo PPP*).

Questa forma di tunneling è detta **PPPoE** (*PPP over Ethernet*, RFC 2516). Pur condividendo il livello fisico, il tunnel PPPoE è a tutti gli effetti un canale dedicato che accede via modem al router pubblico dell'ISP con l'indirizzo IP pubblico assegnato (nello schema 82.10.79.39).

La connessione pubblica va fatta quindi specificando una coppia *username/password* e superando lo schema *sfida/risposta* utilizzato da PPP.

Il gateway ora offre la connessione pubblica agli host privati della rete LAN domestica tramite NAT (su Windows con ICS; su Linux con netfilter/iptables).

Come si può notare, anche il modem e l'AP possiedono un indirizzo IP privato, attraverso i quali si possono raggiungere le relative pagine Web di configurazione.

Il resolver DNS pubblico è attivo sul gateway, che avrà ricevuto in DHCP dall'ISP un paio di indirizzi pubblici di altrettanti server DNS pubblici; gli host della rete privata, invece, dovranno indicare come *gateway di default* e DNS l'indirizzo IP del gateway della rete privata (nello schema precedente 192.168.0.1).

In una rete del genere si può attivare un server DHCP per esempio sull'AP; il gateway, infatti può possedere un server DHCP solo se equipaggiato da un sistema Microsoft Server (o, in alternativa, da un sistema Linux su cui attivare il servizio DHCP).

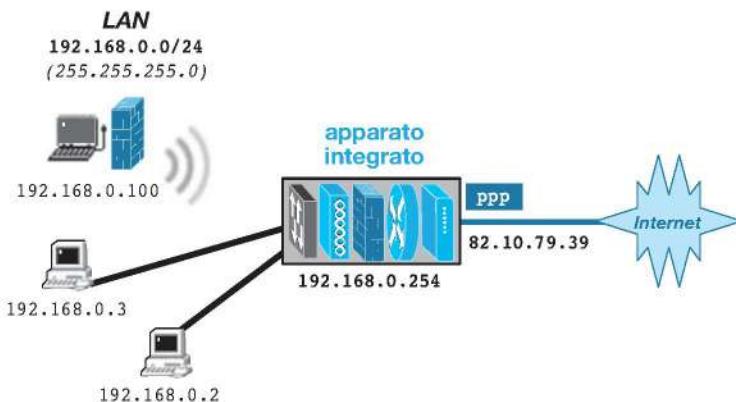
Le configurazioni degli host Wi-Fi sono analoghe alle configurazioni degli host cablati.

Per organizzare una rete e i suoi servizi in modo *workgroup* (o Linux Samba) è opportuno definire su ogni host sempre le stesse coppie *username/password*.

In una rete di questo tipo spesso è inutile usare un servizio proxy, e quindi la protezione dalle intrusioni può essere delegata a un *personal firewall* installato sul gateway (o integrato al sistema operativo): gli host cablati erediteranno la sicurezza garantita dal personal firewall.

Per gli host mobili invece è necessario configurare un proprio personal firewall allo scopo di garantire la mobilità protetta anche per altre reti su cui il dispositivo mobile potrà connettersi.

Un secondo schema di rete residenziale è quello che prevede un unico apparato integrato switch/router/firewall/modem/AP.



## DDNS

Le reti che non dispongono di un **indirizzo IP pubblico statico**, ovvero quelle che ottengono con il DHCP dell'ISP un **indirizzo IP pubblico dinamico** a ogni connessione (per esempio, le reti residenziali classiche) non hanno teoricamente la possibilità di fornire servizi sulla rete Internet (per esempio, pubblicare un server Web, cioè un sito personale).

**DDNS** (*Dynamic DNS*) è una serie di tecniche (descritte anche in RFC 2136) che consente di associare il proprio indirizzo IP dinamico, di volta in volta, a un nome di dominio fisso e registrato sul DNS pubblico. Ora, attraverso quel nome di dominio, anche un host con indirizzo IP dinamico può essere raggiunto dalla rete pubblica a prescindere dal valore dell'indirizzo IP posseduto.

DDNS può essere concesso da fornitori di servizi su Internet, sia gratuitamente sia a pagamento (per esempio, <http://www.no-ip.com>), fornendo uno speciale programma da mandare in esecuzione sulla macchina su cui effettuare l'*hosting* (programma che informerà il servizio DDNS dei cambiamenti di IP) o mantenendosi collegati a determinate pagine Web.

Alcuni router forniscono DDNS all'interno del proprio firmware, previa configurazione.

Questi apparati sono molto diffusi nel mercato di consumo (nel quale sono conosciuti con il nome generico di «**router ADSL**») perché tendono a ridurre al minimo le operazioni di configurazione (management) e, nello stesso tempo, a offrire numerosi servizi aggiuntivi come multiconnessione cablata (switch e DHCP server), connessione Wi-Fi (Access Point), controllo perimetrale (firewall integrato) e multiconnessione alla rete pubblica (NAT, routing e modem ADSL).

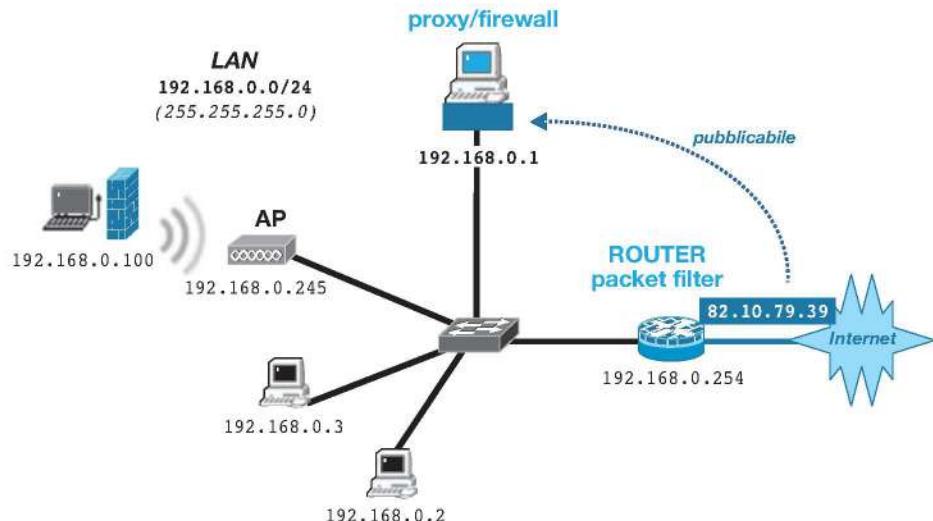
In questo caso tutti gli host connessi all'apparato sono paritari e ottengono i servizi in maniera centralizzata dall'apparato. L'apparato possiede un proprio sistema operativo minimale dotato di interfaccia di configurazione tramite un Web server minimale raggiungibile con un indirizzo IP che deve risiedere nello schema di indirizzamento della rete privata a cui parteciperanno gli host.

Tra le poche configurazioni da impostare, la chiave di sicurezza dell'apparato Wi-Fi e i dati della connessione al fornitore del servizio (ISP), normalmente con la coppia *user name/password* rilasciata al momento della sottoscrizione del contratto. I servizi preconfigurati di DHCP, NAT e firewall faranno il resto.

## 2 Reti single-homed e dual-homed

In questo caso il modello può essere adatto per reti aziendali di piccole dimensioni provviste di un servizio di controllo degli accessi (proxy/firewall installato su *bastion host*).

Il primo caso è detto *single-home bastion host* e prevede un apparato (proxy o firewall) con una sola interfaccia di rete:



I servizi «pubblicabili» su Internet sono contenuti sulla macchina proxy/firewall tramite *reverse proxy* (o *destination NAT*). Il router protegge la rete privata a livello 3 (*packet filter*) e instrada tutti i pacchetti in entrata dalla rete pubblica (**incoming**) sul proxy, che così può effettuarne il controllo anche a livello 7 (*application layer filtering*).

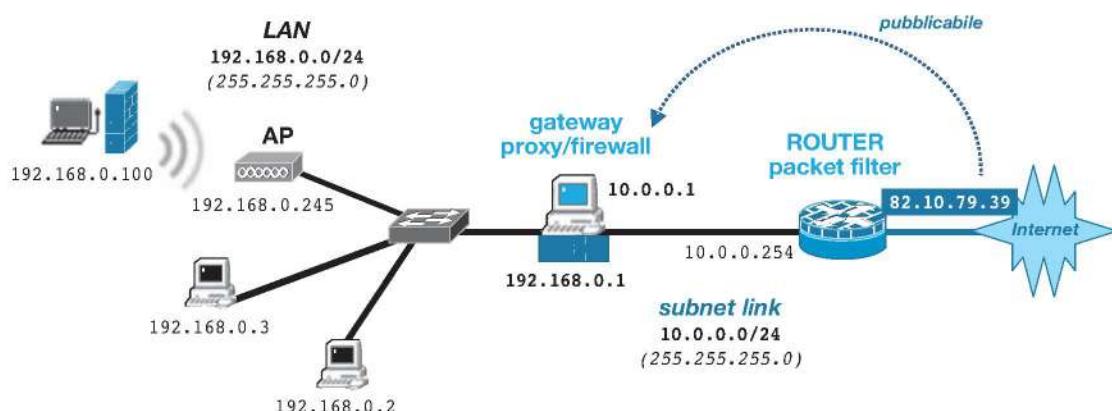
Il proxy, a sua volta, può concedere i servizi pubblici agli host privati (**out-**

**going**), che devono avere proprio l'indirizzo IP del proxy come *gateway di default*. In questo modo può effettuare anche il controllo degli accessi.

In alternativa gli host privati possono uscire verso la rete pubblica semplicemente con un *source NAT* offerto dal router.

Il punto debole di questo schema è che consente a pacchetti pubblici (*incoming*) di transitare fisicamente sulla rete privata, benché controllati dalle ACL sul router.

Una variante di questa configurazione prevede un apparato con due interfacce di rete (gateway) per obbligare tutto il traffico uscente ed entrante (*incoming* e *outgoing*) della rete pubblica a transitare su entrambi i dispositivi di controllo. Questa condizione è considerata un attributo di sicurezza soprattutto se i due firewall sono di marche (commerciali) differenti: pacchetti indesiderati potrebbero oltrepassare un firewall ma non l'altro. Questa variante è anche detta *dual-home bastion host*, e prevede l'uso di una subnet di link:



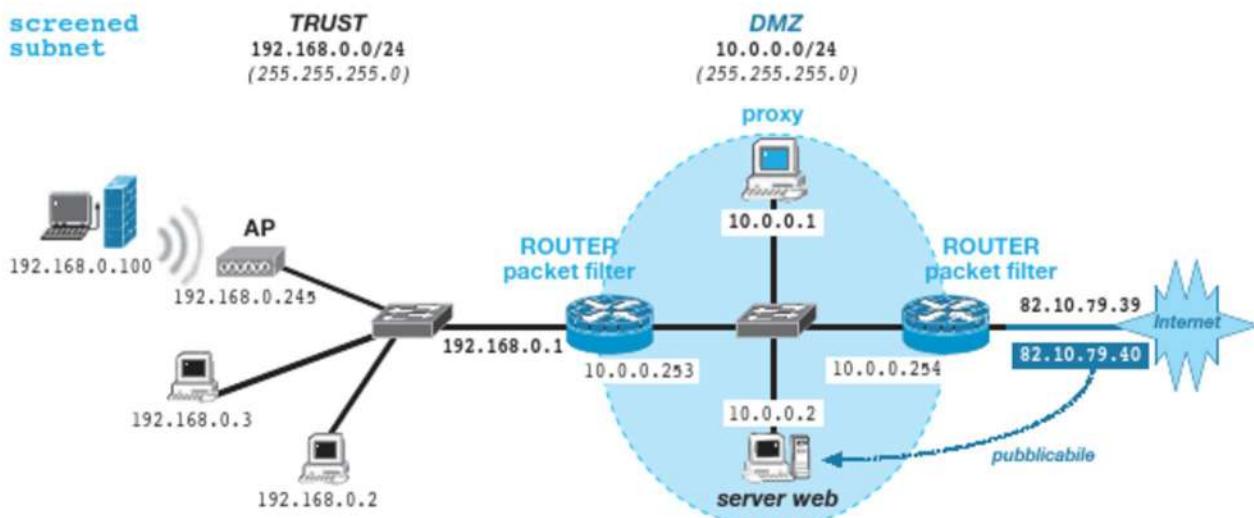
I servizi pubblicati su Internet dovranno risiedere sulla macchina proxy (*reverse proxy*). Rispetto alla soluzione *single-home*, la *dual home* non consente a pacchetti pubblici di circolare sulla rete privata.

Configurazioni come queste (*single-home* e *dual-home bastion host*) non sono ideali per fornire servizi pubblici tramite host privati (tramite *destination NAT* o *reverse proxy*) dato che il traffico «pubblico» *incoming* dovrebbe per forza di cose circolare sulla LAN privata (caso *single-home*), mentre nel caso dello schema *dual-home* un solo server pubblico è a volte insufficiente per un buon servizio.

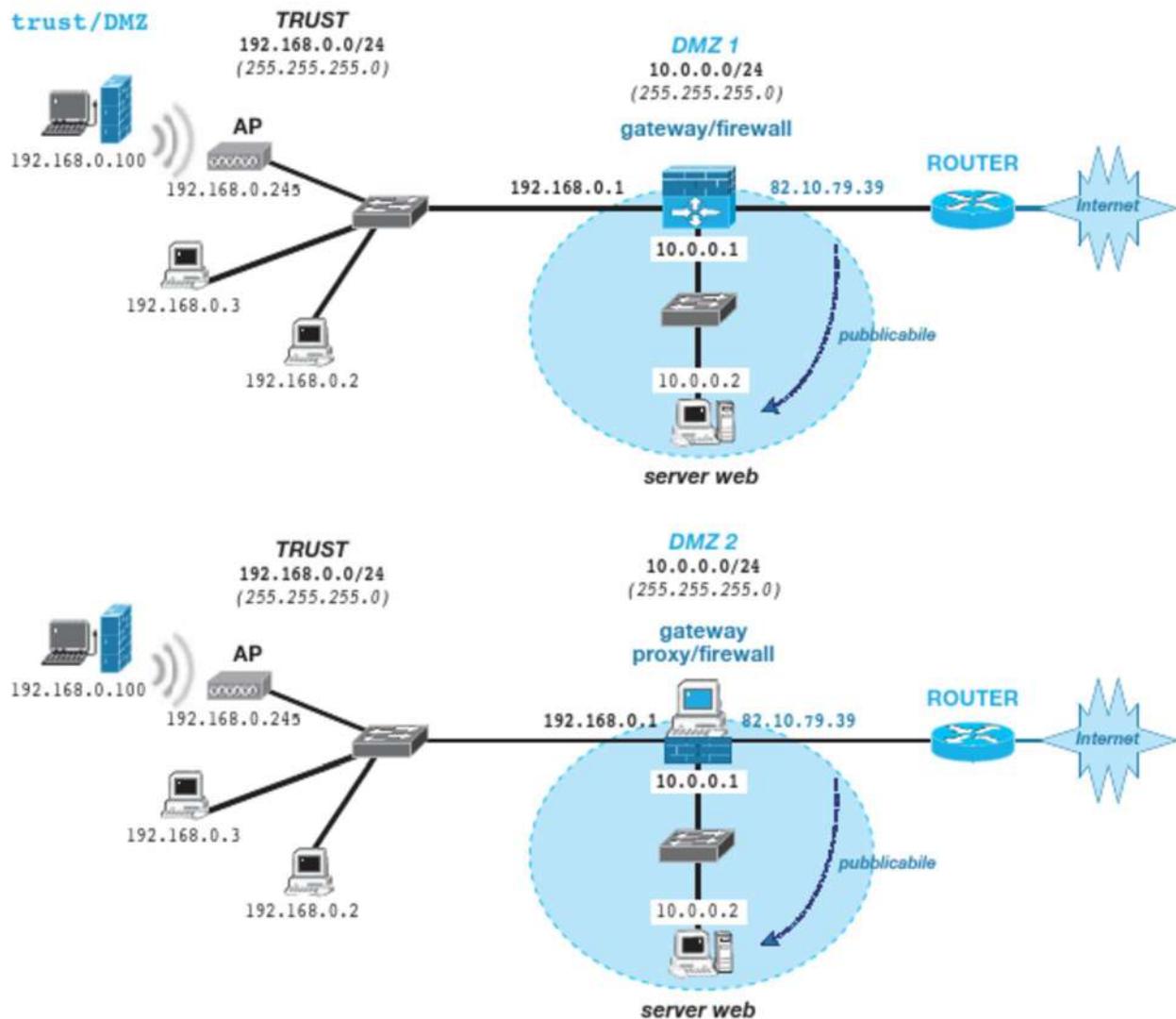
### 3 Reti trust/DMZ

La configurazione *dual-home* prevede una subnet di link. Sfruttando in modo specifico questa segmentazione si ottengono due sottoreti dedicate. L'una, considerata «sicura» e detta **trust**, contenente tutti gli host privati con solo traffico pubblico di *outgoing* (per esempio, la navigazione Web); l'altra, considerata «insicura» e detta **DMZ** (*DeMilitarized Zone*), contenente le macchine che devono esporre servizi pubblici e su cui, quindi, circola traffico *incoming* potenzialmente vulnerabile.

Questo primo schema di rete **trust/DMZ** risponde al nome di **screened subnet**:



In questo caso il traffico *outgoing* (host privati che usufruiscono di servizi pubblici) può essere inoltrato direttamente attraverso i due router (per esempio con *source NAT*) oppure transitare anche attraverso il proxy/firewall per un'analisi più profonda dei pacchetti e il controllo degli accessi.



Il traffico *incoming* (proveniente dalla rete pubblica e destinato a server privati tramite *destination NAT* o *reverse proxy*) è confinato all'interno della subnet DMZ dopo aver passato il controllo del router perimetrale ed essere stato bloccato dal router interno: la subnet trust non è toccata da questo tipo di traffico. Lo schema in alto alla pagina precedente (*screened subnet*) realizza sia il confinamento del traffico *incoming*, sia l'attraversamento di due dispositivi di controllo (possibilmente di marche differenti).

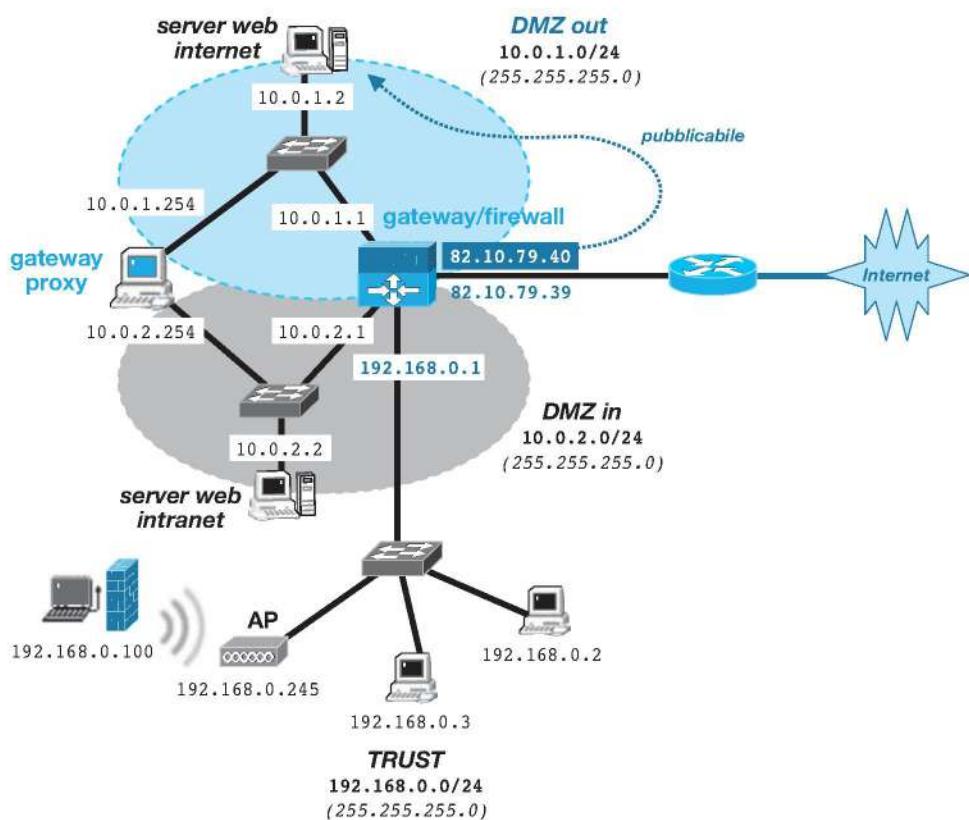
I due schemi in basso alla pagina precedente (*DMZ 1* e *DMZ 2*), del tutto analoghi tra loro, mostrano il modello *trust/DMZ* realizzato attraverso un gateway/firewall hardware con 3 interfacce, e lo stesso schema realizzato con un gateway proxy/firewall anch'esso con 3 interfacce.

In tutte le configurazioni *trust/DMZ* è bene che sull'interfaccia del router siano attestati più indirizzi IP pubblici (**multihomed**), per esempio uno per l'accesso verso Internet degli host privati della subnet trust (traffico *outgoing*) e uno per ogni servizio pubblico realizzato dai server della subnet DMZ (traffico *incoming*).

L'unico punto di contatto tra la rete trust e DMZ riguarda il traffico interno diretto al server Web sulla subnet DMZ (per esempio, servizi di intranet, posta elettronica o altro).

## 4 Rete modello Microsoft

L'idea che sta alla base di questo modello suggerito da Microsoft in alcuni documenti tecnici suggerisce la creazione di due subnet DMZ, una dedicata al traffico *incoming* (*DMZ out*, traffico potenzialmente pericoloso), l'altra dedicata al traffico *outgoing* (*DMZ in*, traffico relativamente sicuro).



Come nel caso classico, il traffico *incoming* rimane confinato nella *DMZ out*, ma il traffico interno non raggiunge direttamente i server sulla *DMZ out* come nel modello precedente, ma la *DMZ in* su cui sono replicati i server dei servizi.

Per uscire verso Internet i pacchetti degli host privati devono dapprima entrare nel firewall (su *DMZ in*), passare attraverso il gateway/proxy e ripassare dal firewall (su *DMZ out*) per poi uscire sulla rete pubblica.

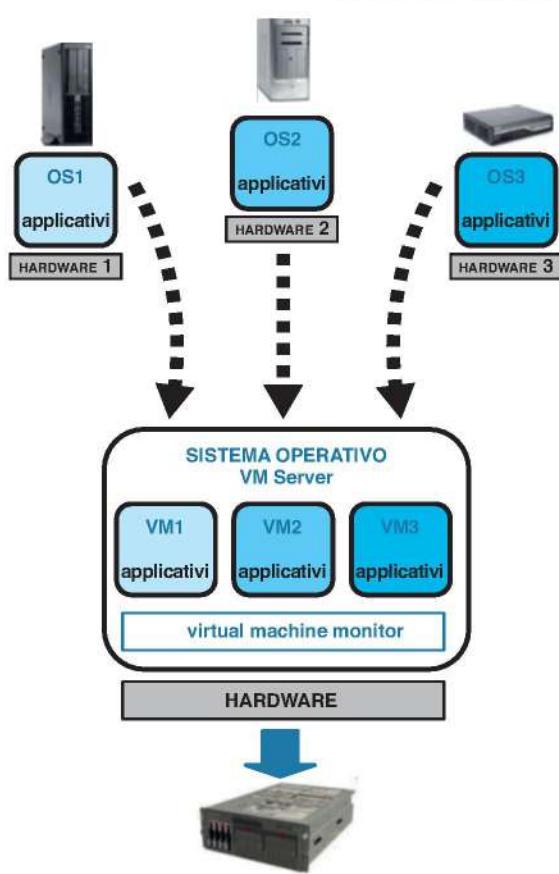
Quantunque relativamente complesso, lo schema garantisce il fatto di poter specificare ACL dedicate al solo traffico della rete trust sull'interfaccia che nello schema della pagina precedente riporta l'indirizzo IP 192.168.0.1, operando un controllo raffinato del traffico interno (e *outgoing*) di una rete di dimensioni medio-grandi. Il proxy in posizione strategica (sia su traffico *incoming* che *outgoing*) è in grado di pubblicare servizi sia pubblici (*DMZ out*), sia privati (*DMZ in*) con gli applicativi tipici operanti su reti Microsoft (per esempio, *Microsoft Shared Point*).

## 5 Virtualizzazione

Come già introdotto nel corso precedente (cfr. *Sistemi e reti, vol. 2, Switch virtuali*), sempre più spesso la gestione di una rete locale si avvale di tecniche di virtualizzazione dell'hardware. Allora si trattava di inserire in un solo sistema hardware (per esempio una macchina server ad alte prestazioni) varie macchine virtuali autonome (**VM**, *Virtual Machine*), costituite ognuna di un sistema operativo proprio, con propri e specifici applicativi server i cui servizi venivano offerti alle macchine della rete attraverso switch virtuali. Tra gli altri benefici, si poteva installare un sistema operativo «*legacy*» (obsoleto, ma ancora in uso con i relativi applicativi) su un sistema hardware più efficiente di quello originale, conferendo al servizio prestazioni migliorate rispetto alla piattaforma originale.

La centralizzazione dell'hardware per vari sistemi di rete offre una gestione semplificata del sistema complessivo: un solo sistema hardware è più facilmente gestibile di molti sistemi hardware ed è tendenzialmente meno costoso. Per esempio, la memoria principale (RAM) viene gestita in maniera più efficiente e senza le duplicazioni che la gestione di più sistemi fisici comporta. La gestione di pochi ma ampi ed efficienti hard disk è sicuramente migliore della gestione di tanti hard disk, magari di diverso formato, taglio e velocità. Il costo dell'alimentazione di una sola macchina, benché più potente, è inferiore alla somma dei costi delle alimentazioni di più macchine, ecc.

Inoltre le funzionalità avanzate di una singola macchina ad alte prestazioni vengono distribuite a tutte le macchine virtuali, come per esempio la tol-



leranza ai guasti degli hard disk (sistemi RAID), le funzionalità avanzate dei più recenti microprocessori, le prestazioni delle schede di rete più sofisticate, ecc.

Anche la manutenzione di un solo hardware è nettamente più semplice della manutenzione di molti hardware, magari diversi tra loro e di case costruttrici differenti: la gestione dei driver di periferica, da sempre, è un cruccio per i sistemisti. I servizi generali, come per esempio l'effettuazione periodica della copia dei dati (backup) può essere gestita in modo centralizzato su tutti i sistemi virtualizzati senza dover prevedere, per ogni singolo sistema, una politica dedicata.

I benefici delle tecniche di virtualizzazione, però, non finiscono qui.

Una macchina virtuale (che ospita un sistema operativo e tutte le sue configurazioni e applicazioni), essendo del tutto scorrelata dall'hardware su cui è installata, può essere completamente salvata su un disco di copia (**snapshot**), sia i dati sia soprattutto il sistema operativo con tutte le sue configurazioni. Questa opzione è fondamentale: effettuando il backup periodico di una macchina virtuale, essa potrà essere recuperata completamente in seguito a un guasto o a una errata configurazione, oppure spostata interamente su un nuovo hardware virtualizzato senza alcuna necessità di reinstallazione e riconfigurazione del sistema operativo e dei servizi.

Infine, i sistemi virtualizzati si rendono particolarmente efficienti per quanto riguarda la loro gestione remota: tramite un software di sistema l'accesso al sistema virtualizzato da remoto consente la gestione di tutti i server in esso contenuti.

È così che molti *datacenter* (il cuore di una LAN che ospita le macchine server), normalmente organizzati con più macchine server fisiche, decidono sempre più spesso la migrazione verso sistemi virtualizzati.

## Emulazione

La **virtualizzazione** (dell'hardware) e l'**emulazione** (del software) sono tecniche simili ma non equivalenti.

Nel primo caso abbiamo un sistema operativo (**host**) dedicato a ospitare altri sistemi operativi (**guest**) e, tramite uno strato software specializzato (**VMM Virtual Machine Monitor** o **hypervisor**) l'host simula gli strati hardware dei vari guest attraverso il proprio.

Nel secondo caso (**emulazione**), un sistema operativo «standard» ospita un software specializzato (l'emulatore) che è in grado di tradurre ogni singola istruzione assembly dei programmi di un sistema operativo ospite (il guest) nelle istruzioni macchina del sistema operativo ospitante (l'host). Tra i programmi di emulazione si ricorda **MAME (Multiple Arcade Machine Emulator)**, in grado di emulare i sistemi operativi che ospitavano i vecchi giochi delle macchine da bar e delle sale giochi. Altri emulatori di rilievo sono **QEMU**, **DOSBox**, **VMware Workstation**, **VirtualBox** e **MS Virtual PC**.

## ESEMPIO

### Virtualizzazione con VMware

La società **VMware Inc.** è una delle società leader del mercato del software di virtualizzazione.

Il prodotto di punta, ovviamente, è il sistema operativo **host** che ospita le macchine virtuali (Vmware ESX Server e Vmware ESXi vSphere) all'interno delle quali sono installati i vari sistemi operativi **guest**.

L'host server ESXi di VMware è un pacchetto di qualche centinaio di MB disponibile anche in versione gratuita: dal sito omonimo si scarica un file ISO da masterizzare direttamente (**VMware vSphere Hypervisor ESXi** o **server ESXi**).

L'installazione del sistema server *ESXi* su una macchina fisica è estremamente veloce e avviene tramite riavvio e boot con il cd masterizzato.

Una volta installato e riavviato il sistema, bisogna configurare i parametri TCP/IP dell'interfaccia di rete fisica (impostazione dell'indirizzo IP e del server DNS), in modo che il sistema server *ESXi* sia gestibile da remoto con un applicativo dedicato denominato **vSphere Client**.

Ora su un pc della rete si installa *vSphere Client* e ci si connette al server *ESXi* tramite l'indirizzo IP configurato precedentemente.

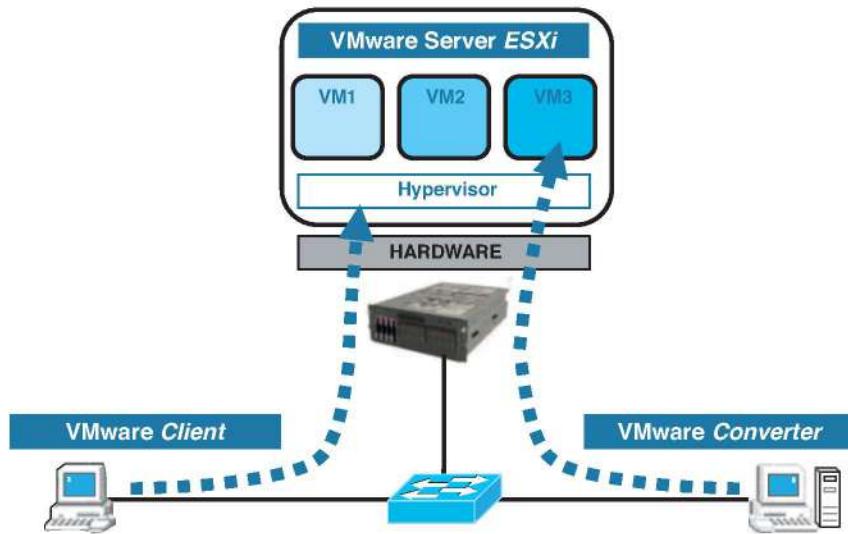
Con *vSphere Client* vanno definite le singole macchine virtuali, specificando quanta CPU fisica dovranno utilizzare, quanta memoria centrale e quanto disco, a partire dalle caratteristiche della macchina fisica. Una volta creata, la macchina virtuale rappresenta un vero e proprio Personal Computer autonomo: lo si può accendere (e spegnere) con comandi software.

Ovviamente all'accensione (software) di una VM «vuota», lo schermo mostrerà l'impossibilità di proseguire, almeno fino a quando non si sarà installato un sistema operativo sulla VM, esattamente come si farebbe con una ➤

macchina fisica (per esempio tramite il boot da DVD equipaggiato con una versione di installazione del sistema operativo desiderato).

Un modo molto interessante di creare una VM è migrarne una fisica già configurata e funzionante presente nella rete.

Installando un applicativo dedicato sulla macchina da migrare denominato **VMware Converter**, si avvia la migrazione dal sistema fisico al sistema virtuale, specificando solo l'indirizzo IP del server *ESXi* di virtualizzazione e assegnando un nome alla VM che sarà creata. Una volta completata la procedura, va avviata una fase di installazione dei driver di dispositivo relativamente alla nuova VM, fase completamente automatica, che garantisce la virtualizzazione delle specifiche periferiche a bordo del sistema fisico appena virtualizzato: ora la nuova VM si ritrova funzionante e operativa sulla rete. Il server *ESXi* ha configurato la condivisione dell'interfaccia di rete fisica con le configurazioni delle interfacce di rete proprie del server appena virtualizzato.



NB. Il sistema VMware Server *ESXi* (scaricabile presso il sito in formato ISO) viene installato in modalità trial per 60 giorni, con tutte le funzionalità sbloccate. Per installare la licenza Free e prolungarla indefinitamente è necessario registrarsi sul sito di VMware e inserire poi il codice che verrà fornito per la versione Free Hypervisor.

Gli applicativi client VMware Client e VMware Converter sono invece direttamente disponibili liberamente per sistemi operativi Windows (2012).

Altri famosi sistemi di virtualizzazione sono: Microsoft Hyper-V, Citrix XwnServer, Xen, Linux KVM.

# Internetworking: accesso da remoto

Una volta installata e configurata una rete privata (per esempio, TCP/IP), installati i suoi servizi tramite sistema operativo (per esempio workgroup, Active Directory o Samba) e abilitati per essa i servizi su Internet (per esempio, sia di tipo client sia di tipo server), è molto interessante verificare come sia possibile accedere a una rete privata attraverso la rete pubblica (Internet).

Le persone, infatti, non sempre sono materialmente disponibili per l'accesso diretto alla propria rete privata, per questioni legate alla vita personale, ai tempi del lavoro o alle sue caratteristiche (lavoratori mobili o senza sede fissa), a contingenze particolari quali spostamenti o migrazioni temporanee di vario tipo.

La stessa esigenza è presente per le aziende o le organizzazioni che hanno realizzato più reti sul territorio e che intendono connettere sedi, uffici distaccati, filiali, ecc. per condividere i servizi presenti nelle varie infrastrutture.

In particolare, l'esigenza di disporre di un accesso remoto è molto importante per gli *amministratori di rete* che, per avere sotto controllo la propria infrastruttura, spesso devono intervenire da luoghi differenti rispetto a quelli in cui è richiesto l'intervento. Con programmi di accesso remoto gli amministratori di rete possono controllare una rete anche di grandi dimensioni attraverso una sola console fisica e anche da postazioni pubbliche esterne alla rete da controllare (**amministrazione remota**).

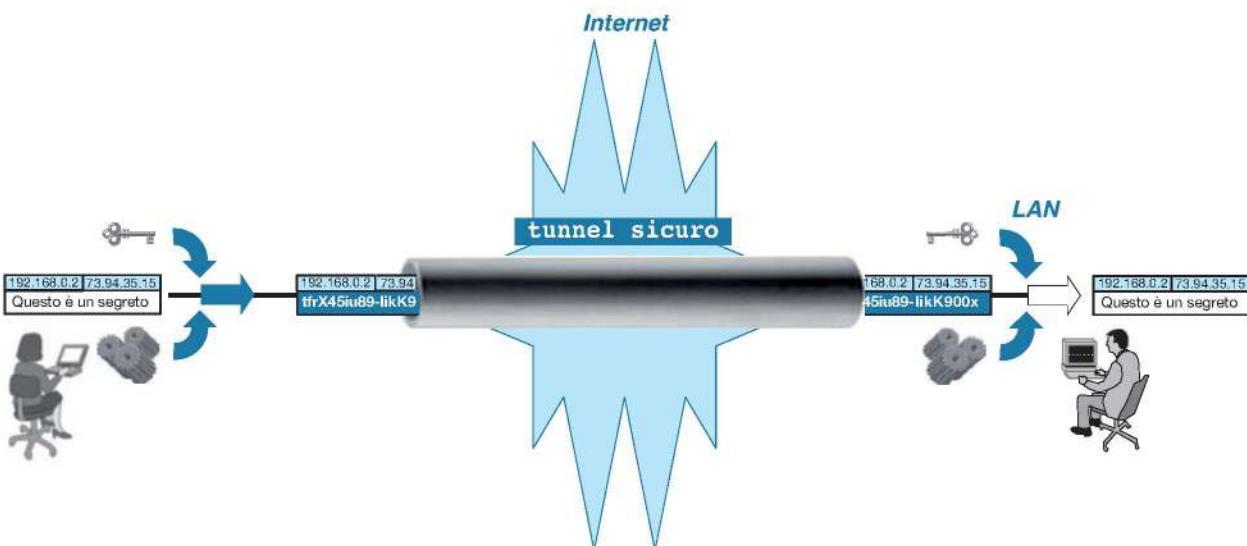
L'accesso remoto a una rete può avvenire a vari livelli e offrire differenti opportunità e funzioni. Quasi sempre un accesso remoto, però, utilizza una tecnica specifica per realizzare le sue funzioni, denominata **tunneling**.

## Tunneling

Il tunneling è una tecnica per cui un protocollo viene completamente incapsulato all'interno di un altro protocollo che ha la funzione di trasportarlo da mittente a destinazione. Lo scopo è fare in modo che il protocollo «incapsulato» possa utilizzare le funzioni tipiche del protocollo «incapsulante» per attraversare un tratto di rete che altrimenti non avrebbe potuto percorrere, o attraverso il quale non avrebbe potuto transitare con la necessaria sicurezza.

Normalmente il protocollo da incapsulare è di livello (OSI) superiore a quello che lo incapsula, ma può anche essere dello stesso livello.

Per esempio, su una rete LAN il protocollo IP è incapsulato all'interno del protocollo Ethernet in pacchetti MAC; oppure il protocollo TCP è incapsulato nel protocollo IP.



Un altro caso evidente di tunneling è il protocollo PPP incapsulato in frame Ethernet, ovvero PPPoE. In questo caso il protocollo PPP, che è punto-punto, usa i frame Ethernet per passare attraverso una rete broadcast e raggiungere il modem: senza l'incapsulamento PPP non sarebbe stato in grado di farlo.

Il tunneling ha il pregio di mantenere il protocollo incapsulato totalmente indipendente dal protocollo incapsulante, pertanto i due estremi del tunnel possono utilizzare il protocollo estratto dal tunnel dopo aver sfruttato le capacità di trasporto del protocollo che lo ha veicolato.

In ogni caso per l'accesso remoto sono fondamentali alcuni requisiti di sicurezza, tra i quali l'**autenticazione** dell'identità, la **riservatezza** dei contenuti e l'**integrità** dei dati (*authentication, confidentiality, integrity*).

Questi requisiti sono fondamentali dato che l'accesso remoto avviene praticamente sempre attraverso la rete pubblica (Internet) e quindi l'identità e i dati che devono essere trasportati per realizzare una connessione remota sono potenzialmente alla portata di chiunque operi su Internet.

Per l'accesso remoto il tunneling garantisce anche i requisiti di sicurezza: i dati transitanti su rete pubblica sono prima cifrati e poi incapsulati in tunnel che trasportano i dati cifrati dal mittente al destinatario. Una volta giunti a destinazione attraverso il tunnel potranno essere decifrati dal destinatario.

**NOTA:** Questi requisiti fondamentali e i protocolli di rete relativi alla sicurezza sono trattati nella Sezione B di questo volume.

## 1 Terminale remoto

Una prima modalità per fruire di accesso remoto è costituita dal cosiddetto **terminale remoto** (storicamente anche noto come *rlogin* o *telnet*).

In questo caso un applicativo (client) di livello 7 è in grado di fornire un'intera sessione di un host remoto da gestire tramite interfaccia a riga di comando. In questo modo sull'host remoto si possono riconfigurare servizi, verificare il file system, avviare programmi e, in generale, fare tutto ciò che è consentito dalla shell a carattere dell'host remoto (*amministrazione remota*).

Il protocollo dominante per effettuare accessi con terminale remoto è **SSH** (*Secure Shell*, RFC 4253), successore di **telnet**, ma che offre il servizio in sicurezza garantendo autenticazione, riservatezza e integrità sia con schemi a chiave simmetrica sia asimmetrica.

Si tratta di un protocollo client/server trasportato in un tunnel TCP, pertanto l'host remoto deve avere avviata la parte server del protocollo tramite un'applicazione dedicata affinché un host client possa effettuare la connessione.

Siccome il terminale remoto offre un'interfaccia a caratteri è lo strumento di amministrazione remota più idoneo per controllare sistemi Linux.

## ESEMPIO

### PuTTY

Il programma PuTTY è il più utilizzato client SSH sia in ambiente Linux sia in ambiente Windows. La parte server di SSH su Windows può essere ottenuta usando un software libero denominato OpenSSH for Windows.

Il server SSH su una macchina Linux può essere avviato con:

```
[root@linux root]# /etc/init.d/sshd start
sshd started
[root@linux root]# /etc/init.d/sshd status
sshd (pid 31408 31393 19133 1830) in esecuzione...
[root@linux root]#
```

Il client SSH PuTTY è liberamente scaricabile da Internet, per esempio per un sistema Windows.

La configurazione per una connessione a un server è semplice. Nelle *Basic options* di configurazione del programma basta indicare: *Host Name or IP address* (l'indirizzo IP su cui opera un server SSH), il numero di *Porta TCP* (22) e il tipo di connessione (SSH), dato che PuTTY è in grado di gestire più protocolli di accesso remoto.

Avviata la connessione, se il server SSH è in linea, PuTTY apre una finestra console su cui richiede prima di tutto un login (nell'esempio si effettua con l'utente root del sistema Linux a cui ci si è connessi), quindi viene concessa la console del sistema remoto (la password non viene mostrata):

```
finestra Windows della console remota del sistema R.sistemi.lan
login as: root
root@linux's password:
Last login: Wed Dec 19 21:03:22 2012 from a20.sistemi.lan
[root@linux root]# cd /var/
[root@linux var]# ls
amavis cache empty lib local lock log mail named run spool tmp yp
[root@linux var]#
```

Nell'esempio la connessione è richiesta al sistema **R.sistemi.lan** riportato nell'Appendice (cfr. [Appendice. Configurazione DHCP e DNS](#)).

## 2 Desktop remoto

**Desktop remoto** (*Remote desktop*) offre la medesima funzionalità di terminale remoto ma estende il controllo dalla tastiera al mouse e quindi presenta al client l'interfaccia grafica del server che si intende amministrare da remoto.

Nell'ambito dell'amministrazione remota con interfaccia grafica operano sostanzialmente due tipi di protocolli client/server con analoghe funzionalità: **RDP** (*Remote Desktop Protocol*), protocollo proprietario Microsoft e

utilizzato per l'amministrazione remota nativamente sui sistemi operativi Windows; oppure **RFB** (*Remote Frame Buffer*, RFC 6143), protocollo che poi ha dato origine a numerosi software per l'amministrazione remota anche con versioni a licenza gratuita (software VNC).

Entrambi i protocolli garantiscono autenticazione, riservatezza e integrità della connessione tramite protocolli SSH o TLS quasi sempre inseriti in un tunnel TCP.

Siccome desktop remoto offre una completa interfaccia grafica è lo strumento di amministrazione remota più idoneo per controllare sistemi Windows.

### Desktop remoto

Il programma `mstsc.exe` è il client di desktop remoto nativo dei sistemi Windows. Il servizio server, invece, è attivato da processo `svchost.exe (-k termsvc)` ed è disponibile sia sui sistemi Windows Server sia sui sistemi Workstation.

Sui sistemi Microsoft Server sono possibili solo due sessioni contemporanee allo stesso server di Desktop Remoto, più la sessione **console**.

La sessione **console** è la speciale sessione che viene avviata quando il login viene eseguito fisicamente sulla macchina server. Per utilizzare un numero superiore di sessioni allo stesso server di Desktop Remoto vanno abilitati i cosiddetti servizi *Terminal*.

Sui sistemi Microsoft Workstation il server di Desktop Remoto è avviabile tramite *Pannello di controllo*.

Per Windows XP, 2003: *Pannello di controllo-Sistema-Conessione remota*.

Per Windows Vista, 7, 8: *Pannello di controllo-Sistema-Impostazioni di connessione remota*.

In entrambi i casi (Server o Workstation) vanno specificati gli utenti autorizzati ad accedere tramite il client di Desktop Remoto.

Sono poi disponibili varie applicazioni client/server di terze parti che offrono il servizio di desktop remoto (TeamViewer, RealVNC, ecc.) che, a differenza del Desktop Remoto di Microsoft offrono anche la modalità interattiva: l'utente connesso alla **console** del server «vede» l'attività svolta dal client connesso da remoto (e viceversa) e può interagire con esso.

## 3 VPN

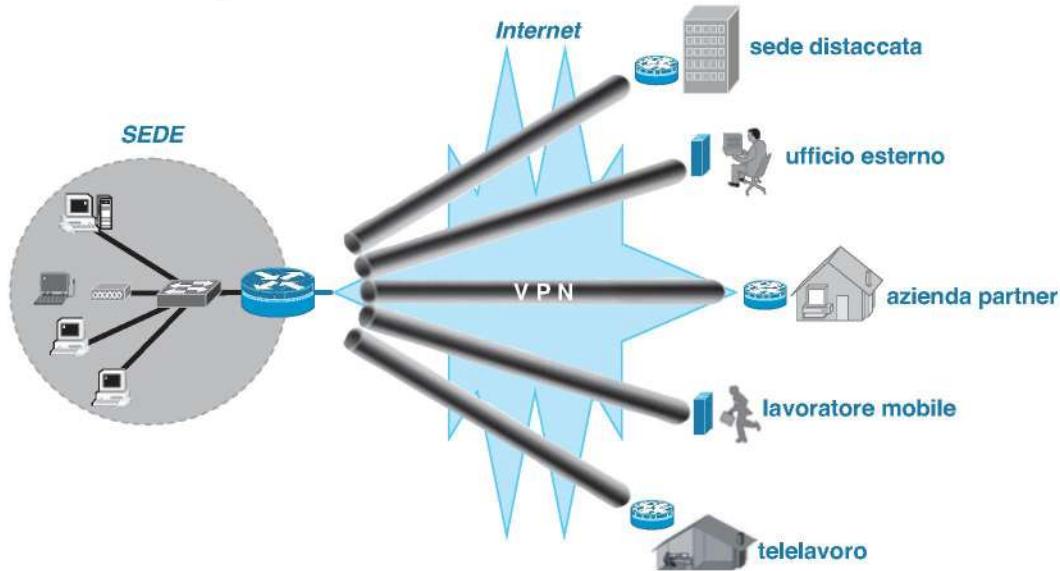
Una **VPN** (*Virtual Private Network*) è un modo di utilizzare la rete pubblica (per esempio, Internet) per veicolare un traffico privato, cioè relativo a una rete privata (per esempio, una rete locale TCP/IP residenziale o aziendale).

Così facendo i servizi tipici di una rete privata (per esempio il programma di contabilità di un'azienda) può essere utilizzato da un utente anche se non si trova materialmente all'interno della rete privata ma, per esempio, in un ufficio estero. L'accesso remoto dell'utente alla propria rete privata viene ottenuto utilizzando l'infrastruttura pubblica di Internet e l'utente remoto opera come se si fosse connesso (login) dall'interno della rete privata.

Il concetto può essere esteso, pertanto una VPN può consentire di mettere in comunicazione, come se agissero sulla stessa LAN, due o più sedi distaccate di una stessa organizzazione, e quindi anche due (o più) LAN private.

Sebbene il campo operativo delle VPN possa essere molto esteso (per esempio, *Trusted VPN*, acquistate e organizzate da un fornitore di servizi

Internet che può anche prevedere connessioni dedicate), in questo testo ci si limita a discutere delle **secure VPN**, ovvero di quelle VPN che sono create e utilizzate in completa autonomia da un'organizzazione privata, e che usano come transito dei dati l'infrastruttura TCP/IP Internet esistente tramite tunneling.



Il tunneling è necessario affinché i dati trasferiti attraverso la rete pubblica rimangano isolati all'interno di flussi ben definiti e siano trasportati in maniera sicura: i dati privati circolano dentro il tunnel sulla rete pubblica senza avere contatti con essa.

Le tecnologie di virtual private networking sono un esempio di integrazione di applicazioni di sicurezza. Per realizzare una VPN, infatti, sono necessarie soluzioni di *autenticazione* (affinché alla VPN possano accedere solo utenti autorizzati), *riservatezza* (affinché i messaggi in transito non siano trafugati) e *integrità* (affinché i dati trasmessi coincidano sempre con quelli ricevuti) tramite tecniche di *tunneling* anche per attraversare reti differenti.

Alle VPN, inoltre, possono essere associate anche tecnologie di Quality of Service (QoS), a seconda dei servizi che devono essere garantiti su queste connessioni.

Il protocollo più consolidato per organizzare VPN è sicuramente **IPsec** (*IP security*, RFC 1825 e seguenti), che in realtà è una serie di protocolli che agiscono a livello 3 Rete come sostituti di IP.

IPsec è in grado di garantire l'autenticazione con il sottoprotocollo **IKE** (*Internet Key Exchange*, RFC 2401). IKE è la parte di IPsec che interviene inizialmente, consentendo lo scambio sicuro delle chiavi private dei due capi della VPN e in alcuni casi inizializzando il tunnel VPN. Inoltre definisce anche la chiave di sessione per cifrare la comunicazione seguente (riservatezza). È un protocollo di livello Applicazione ed è trasportato sulla porta UDP 500.

In seguito le sessioni di IPsec sono organizzate con il sottoprotocollo **ESP** (*Encapsulating Security Payload*) che sostituisce tutto il payload del pacchetto IP originale (il campo *dati*) con un payload cifrato, autenticato e integro.

#### **VPN workstation**

Client e server VPN sono disponibili su Windows (2000+, anche workstation come XP, 7 e 8) tramite «**Crea nuova connessione**», opzioni:

- **connessione a rete aziendale** per la parte client;
- **connessione avanzata** per la parte Server.

All'atto della creazione della VPN sono necessarie alcune operazioni di impostazione.

#### *Lato server*

**1)** Specificare le utenze della rete LAN locale che possono effettuare una connessione VPN da rete pubblica.

**2)** Specificare gli indirizzi IP, coerenti con la LAN privata, che dovranno stabilirsi sul server locale e sull'host remoto per ogni connessione VPN entrante.

#### *Lato client*

**1)** Indicare l'indirizzo IP del server VPN che dovrà accettare la connessione VPN entrante.

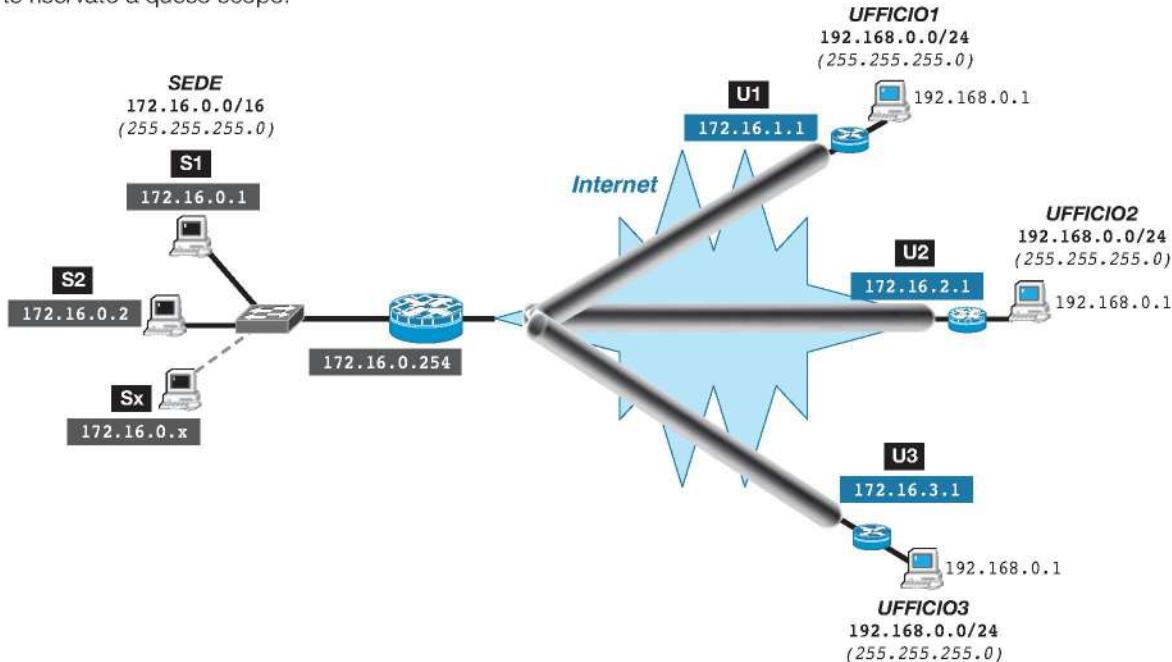
**2)** Autenticarsi come uno degli utenti previsti nella configurazione del server VPN.

**VPN**

La Sede di un'organizzazione strutturata su LAN privata, dispone di numerosi *Uffici* sparsi sul territorio ai quali deve concedere l'utilizzazione dei propri programmi e dei propri database interni.

Affinché i vari *Uffici* possano accedere attraverso la rete pubblica alla LAN privata della Sede, si possono organizzare tanti tunnel VPN, uno per ogni ufficio.

I vari *Uffici* pertanto potranno accedere alla rete privata della Sede acquisendone un indirizzo IP opportunamente riservato a questo scopo.



Nel disegno si nota come i tre host **U1**, **U2** e **U3** di tre *Uffici* distaccati abbiano ognuno due indirizzi IP privati: uno appartenente alla propria LAN privata; l'altro appartenente alla LAN della Sede e acquisito tramite connessione VPN.

Le reti Microsoft usano IPsec per realizzare VPN, e inseriscono a livello 2 OSI (ovvero sotto IPsec) un protocollo di supporto: **L2TP** (*Layer 2 Tunneling Protocol*) che, dopo l'autenticazione iniziale di IPsec, ha il compito di creare il tunnel su cui verrà trasportato IPsec. L'accoppiata dei due protocolli L2TP/IPsec è standardizzata in un'apposita RFC (3193).

## 4 Cloud computing

Con **cloud computing** si intende un modello di gestione della rete tale per cui gli utenti di una rete privata delegano a terzi la gestione di uno o più servizi della propria rete interna.

La delega viene decisa per sollevare gli amministratori dalla gestione in prima persona dei servizi, gestione interna che viene ritenuta *o troppo costosa o troppo tecnicamente impegnativa o non all'altezza del servizio richiesto*.

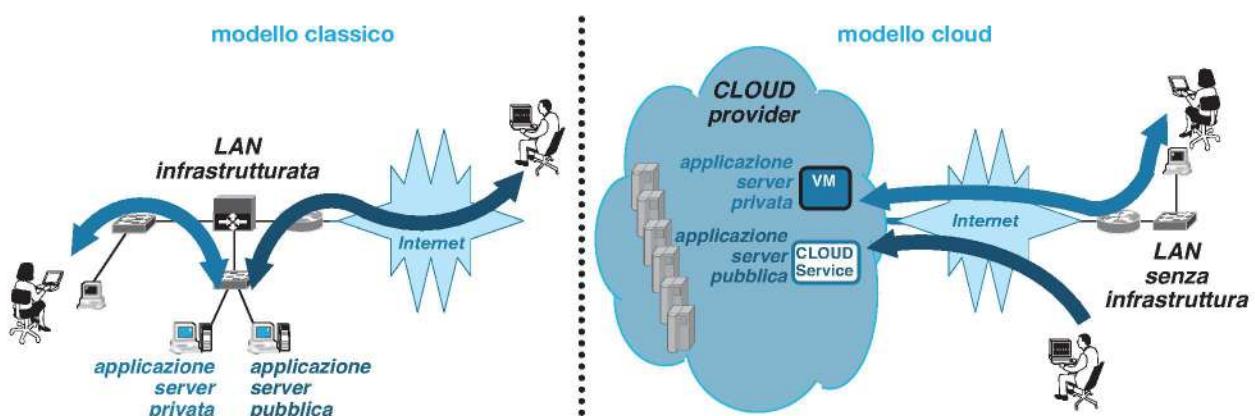
In questo senso si può affermare che il primo caso rilevante di cloud computing attuato in massa dall'utenza Internet è stato la **posta elettronica** con la migrazione verso il *Web mail* ben prima che il termine cloud si imponesse come modello specifico.

Modi di gestione che hanno anticipato il cloud computing sono anche

l'**hosting** e l'**housing** offerto dagli ISP o da altri operatori. Essi realizzano sui propri server, dislocati geograficamente distanti dal cliente, i servizi Web più importanti, come siti, posta elettronica, spazio disco (*storage*), sottraendone la gestione dalle macchine private del cliente e ospitandone l'implementazione sui propri datacenter raggiungibili via Internet.

Normalmente i fornitori di cloud computing offrono una notevole schiera di macchine ad alte prestazioni in grado di ospitare i servizi di molte aziende, sia sotto forma di processi dedicati su macchine specifiche, sia sotto forma di processi attivati all'interno di sistemi virtualizzati (VM, *Virtual Machine*).

Nel contempo questi grandi datacenter sono provvisti di una connettività sulla rete pubblica ad altissime prestazioni.



### ROI

In generale il cloud computing investe soprattutto le aziende che operano in rete. Anche in questo caso gli utenti privati di una rete aziendale richiedono sempre maggiori servizi e l'introduzione immediata delle nuove tecnologie. L'infrastruttura IT interna si ritrova a dover rispondere velocemente alle richieste e l'amministrazione della rete è impegnata continuamente alla configurazione e alla gestione di nuovi servizi.

Questo sforzo spesso non è compensato dal **ROI** (*Return On Investment*): i costi di gestione dell'infrastruttura interna della rete superano i benefici ricavati da essi.

Questi **costi di gestione** riguardano le voci di:

- **investimento iniziale** (ovvero l'acquisto dell'hardware quali server, apparati e cablaggi);
- **mantenimento** (la gestione nel tempo della manutenzione efficiente del servizio);
- **risorse umane** (la forza lavoro degli amministratori di sistema);
- **licensing** (i costi delle licenze dei prodotti software);
- **consumi energetici** (per mantenere accesi i server e gli apparati interni);
- **formazione** (per mantenere tecnicamente aggiornata la forza lavoro interna).

Per queste ragioni molte aziende IT ricorrono al cloud computing.

In questo modo le risorse interne, ora delegate a terzi *in cloud*, non rientrano più all'interno della propria infrastruttura, ma vengono fruite tra-

### Web mail

Nello schema originale il servizio di posta elettronica viene gestito da un applicativo di livello 7 (il **client di posta elettronica**) configurato e avviato sulla macchina dell'utente.

Nel corso del tempo i client di posta elettronica sono stati sostituiti progressivamente dal servizio di gestione della posta elettronica fornito direttamente dal fornitore delle caselle di posta tramite un altro applicativo di livello 7, il browser Web (**Web mail**).

Molti utenti di posta elettronica trovavano inutilmente complesso installare e configurare un client di posta per leggere le proprie mail, mentre l'equivalente servizio disponibile via Web dal gestore risultava più semplice e immediato.

Nello stesso tempo il servizio diventava più efficiente: gli utenti possono ora gestire la posta elettronica anche da host diversi da quello personale, conferendo al servizio un importante attributo di *mobilità*.

mite TCP/IP e Internet e distribuite tra diversi datacenter di terze parti. I costi suddetti sono praticamente tutti a carico della struttura che fornisce il cloud computing. Tutta la gestione dei nuovi servizi è completamente trasparente per l'utilizzatore, che «vede» un unico punto virtuale di accesso alle proprie risorse delegate.

Per organizzare un sistema cloud normalmente sono previsti tre soggetti:

- il **cliente** (*end user*): l'utente o l'azienda che richiede il servizio e sottoscrive un contratto per esso;
- l'**amministratore** (*cloud user*): un utente tecnico che sceglie e configura i servizi offerti dal fornitore. Mette in contatto il cliente finale con il fornitore di cui conosce l'infrastruttura e ne cura i rapporti tecnici. Questa figura di intermediazione è presente soprattutto quando il servizio è richiesto da un'azienda;
- il **fornitore** (*cloud provider*): l'azienda che possiede le piattaforme hardware e software (datacenter) con cui il servizio viene erogato: connettività, macchine, archiviazione, applicazioni, ecc., e che si fa pagare in base all'uso (*pay-per-use*).

I costi del cloud computing si concentrano quasi esclusivamente sul contratto di fornitura che il cliente stipula con l'azienda fornitrice e/o con l'amministratore, il cosiddetto **SLA** (*Service Level Agreement*).

Questi contratti sono evidentemente critici e vanno valutati con molta attenzione, dato che, tra gli altri oneri, l'azienda fornitrice deve assicurare garanzie adeguate per quanto riguarda il **rischio** (cioè eventi che causano guasti, malfunzionamenti, perdite dati e interruzioni del servizio) e la **riservatezza** (cioè la garanzia che i dati «delegati» non siano dispersi).

Aldilà delle ragioni puramente economiche, il cloud computing promette una funzionalità sempre più importante per i servizi di rete: la **mobilità**.

Con il cloud computing i servizi di una rete privata diventano realmente accessibili da ogni postazione generica su Internet, che sia un host privato dell'utente, un notebook personale, uno smartphone o dei tablet mobili.

Naturalmente la bontà di un servizio di cloud computing dipende in maniera stringente dalla **qualità della connessione** alla rete pubblica: la riduzione del *digital divide* e il miglioramento della qualità della connessione (banda larga) sono fattori essenziali per la sua diffusione.

Le subscriber line (*l'ultimo miglio*) sono fondamentali per un servizio cloud efficace. Oggi in Italia (2013) i costi per realizzare un cavidotto da un punto di accesso all'utente finale (per esempio per un collegamento in fibra ottica, che ha un indice di guasto di due ordini di grandezza inferiore rispetto al rame) valgono dai 5 ai 10 euro/metro se non bisogna scavare i condotti, mentre si attesta sull'ordine di 50-100 euro/metro se bisogna eseguire gli scavi.

## 4.1 Servizi cloud

Un fornitore di cloud computing può offrire alcune classi di servizi che realizzano altrettante modalità d'uso da parte dei clienti.

Il modello più semplice è definito **SaaS** (*Software as a Service*).

In questo caso il cliente utilizza il software messo a disposizione del fornitore per ottenere servizi quali file sharing, archiviazione, gestione posta elettronica, messaggistica, applicativi d'ufficio, ecc. Il *SaaS* si interfaccia agli utenti prevalentemente via HTTP e con Web Browser: il cliente, oltre all'uso delle applicazioni cloud, può solo configurarne i servizi e non ha altro controllo sul loro funzionamento.

Esempi di cloud *SaaS*: *GoogleDocs* (file sharing), *Gmail* (posta elettronica), *Dropbox* (file sharing), *Microsoft Office 365* (applicativi di produttività personale), ecc.

A un livello superiore si pone il **PaaS** (*Platform as a Service*).

In questo caso il cliente usa ambienti di sviluppo messi a disposizione del fornitore e, tramite SDK specializzati, può scrivere applicazioni personalizzate che verranno poi messe a disposizione come servizi cloud.

Il cliente ha un buon controllo sul servizio, potendone programmare le funzionalità direttamente.

Esempi di cloud *PaaS*: *Google App Engine*, *Microsoft Windows Azure Compute*, *Amazon Elastic Beanstalk*, *Joyent public cloud* (usata da Facebook e LinkedIn), ecc.

Infine il servizio più promettente viene denominato **IaaS** (*Infrastructure as a Service*).

In questo caso il cliente affitta spazio di calcolo presso il fornitore in termini di quantità di CPU, spazio disco e connettività in base alle esigenze dei servizi che intende sviluppare. Il cliente può quindi scrivere o migrare le proprie applicazioni private nell'infrastruttura del fornitore che, tipicamente, le farà «girare» su *macchine virtuali* (VM) opportunamente configurate.

Esempi di cloud *IaaS*: *Google Compute Engine*, *Amazon EC2*, *Microsoft Windows Azure VM*, *HP Cloud*, *IBM Blue cloud*, *VMware IaaS*, *MobileMe* (Apple), *GoGrid*.