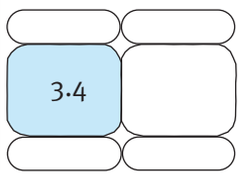
ANALISI E IDENTIFICAZIONE DEL TRAFFICO INTERNET

Paolo Giacomazzi

Per gestire in modo efficiente le risorse della sua rete, l'*Internet Service Provider* (ISP) ha la necessità di conoscere le caratteristiche dei flussi di traffico trasportati attraverso la sua infrastruttura e di saper individuare l'applicazione che ha generato i flussi stessi. Con queste informazioni, l'ISP può stabilire come gestire ogni flusso. Inoltre, conoscendo l'applicazione che sta generando un determinato flusso, si possono stabilire con precisione i requisiti di qualità del servizio ad esso associati che, se rispettati, determinano un maggior grado di soddisfazione degli utenti. L'analisi del traffico Internet e la sua identificazione possono fornire all'ISP queste informazioni di importanza strategica.



1. LE MOTIVAZIONI DELL'ANALISI E DELL'IDENTIFICAZIONE DEL TRAFFICO INTERNET

Per un *Internet Service Provider* (ISP), l'analisi e l'identificazione del traffico generato dai suoi clienti è propedeutica ad un insieme di operazioni critiche relative alla gestione delle risorse della rete e del suo rapporto, anche economico, con i clienti stessi. Se, da un lato, per un ISP è relativamente semplice misurare il volume complessivo di traffico che transita sui collegamenti (*link*) della propria rete, l'informazione ottenuta da una misurazione così aggregata e priva di dettagli si restringe ad una valutazione complessiva del carico al quale sono sottoposti i link e i nodi (*router*). Eventualmente, tramite una collezione di serie storiche di questo carico, l'ISP è in grado di eseguire un'analisi della tendenza del carico e di decidere se è opportuno potenziare qualche *link/router* che si sta avvicinando ad una soglia di carico che esso ritiene critica. Da questo punto di vista, l'analisi del carico complessivo delle risorse della rete abilita una basilare attività di *capacity planning* (pianificazione della capacità) e *resource provisioning* (approvvigionamento delle risorse).

Per gestire in modo più preciso ed efficiente le risorse della sua infrastruttura, L'ISP ha bisogno di informazioni più dettagliate. In primo luogo, L'ISP deve conoscere le caratteristiche dei flussi di traffico che fa transitare nella sua rete, in particolare: la durata dei flussi, il volume di traffico che ognuno di essi trasporta, la loro velocità di trasmissione e il grado di variabilità di questa velocità. Questo tipo di analisi è denominata *flow analysis* (analisi a livello di flusso) e permette all'ISP una gestione delle risorse più efficiente di quella abilitata da una mera misurazione del livello di carico complessivo dei link. Il secondo problema che deve affrontare l'ISP è quello di conoscere la tipologia di applicazione il cui traffico è trasportato in rete dai flussi. L'analisi che permette di ottenere questa informazione è l'*identificazione* (o anche classificazione) del traffico che mette in grado l'ISP di gestire al meglio i flussi nella sua rete, tenendo conto dei requisiti di qualità e delle caratteristiche specifiche delle applicazioni utilizzate dagli utenti finali.

La conoscenza dell'aliquota di traffico generata da ogni tipologia di applicazione fornisce all'ISP uno strumento per meglio pianificare le proprie strategie di offerta di servizi e della relativa tassazione. Per esempio, è proprio grazie all'identificazione del traffico che si è potuto capire che attualmente il traffico generato dalle applicazioni di tipo *Peer-to-Peer file sharing* (per esempio eMule) **ammonta a circa l'80% del complessivo traffico in Internet** [1]. Visto che molto di questo traffico è generato da utenti dotati di connessioni ADSL con tassazione di tipo *flat* (a canone, indipendentemente dal volume di traffico scambiato dall'utente) gli ISP sono consci, grazie all'identificazione del traffico, di trovarsi in una situazione nella quale l'occupazione delle risorse di rete è in crescita veloce e, tale aumento (che comporta costi di investimento per il potenziamento dell'infrastruttura) non corrisponde ad un commisurato incremento del fatturato, con conseguente erosione dei margini operativi. Anche l'identificazione di applicazioni di telefonia *peer-to-peer* (per esempio Skype [2]) è critica per gli ISP. In questo caso, il traffico telefonico transita sulla linea ADSL del cliente, e non attraverso la rete telefonica, privando il provider della possibilità di tassare la chiamata. Questi due esempi dimostrano come l'ISP debba necessariamente conoscere la tipologia di applicazioni utilizzate dai propri clienti per individuare le criticità e le motivazioni che portano ad una diminuzione dei margini operativi e per poter elaborare contromisure. Da questo punto di vista, l'identificazione del traffico è per l'ISP un'attività di importanza strategica.

L'identificazione del traffico è anche un utile strumento per una gestione della Qualità del Servizio (QoS) differenziata per le diverse applicazioni. Infatti, se si riconosce l'applicazione che sta generando un dato flusso di traffico, si possono identificare i requisiti di QoS specifici per quell'applicazione (per esempio, in termini di *throughput* e ritardo) e quindi allocare le risorse necessarie per garantire che questi requisiti siano rispettati, con conseguente soddisfazione dei clienti. Questa attività è usualmente denominata *QoS management*.

Un'altra attività di gestione del traffico realizzata dagli ISP è il cosiddetto *traffic engineering*, un'operazione che consiste nel determinare, per un dato flusso di traffico o per una data categoria di flussi, il percorso migliore (*route*) all'interno della rete, per meglio rispettare i requisiti di qualità del servizio delle applicazioni. Un'attività di *traffic engineering* mirata alla gestione della QoS richiede che i requisiti di qualità del servizio siano noti e, quindi, che si conduca un'attività di identificazione del traffico.

Dal punto di vista della gestione della sicurezza, l'identificazione del traffico può rendere più efficiente l'attività di *anomaly detection*, che consiste per esempio nell'individuare un carico di traffico anomalo per una data applicazione in rete. Una situazione di questo tipo può essere generata da attacchi di tipo *Denial of Service o Distributed Denial of Service*, nel quale un insieme di computer "infettati" genera un carico di traffico focalizzato verso risorse mirate (per esempio un sito web che si intende inabilitare). Se si rileva una distribuzione anomala del traffico, si possono intraprendere misure reattive per controbattere questo tipo di attacchi.

È opportuno citare infine la possibilità che gli ISP hanno di fornire risorse differenziate a diverse tipologie di applicazioni. È chiaro, infatti, che se il traffico *peer-to-peer* consuma la maggior parte delle risorse di rete senza produrre un fatturato commisurato, un ISP potrebbe fornire meno banda alle applicazioni di tipo peer-to-peer, a favore di applicazioni più remunerative. Questo è un argomento molto controverso che vede entrare nel dibattito la questione della *network neutrality*. La *network neutrality* è un principio in base al quale un ISP non dovrebbe discriminare il tipo di applicazione utilizzato dai suoi clienti, quando fornisce ad essi un collegamento ad Internet. In pratica, l'ISP dovrebbe agire secondo il principio "*un bit è un bit*", cioè, tutti i bit vanno trattati allo stesso modo, senza sfavorire il traffico di alcune tipologie di applicazioni per favorirne altre più convenienti dal punto di vista dei profitti. D'altra parte, la *network neutrality* è un principio non ancora formalizzato da normative, oggetto di dibattito, con sostenitori e oppositori. Si cita il caso (negli USA) di Comcast [3] che ha rallentato il traffico delle applicazioni di *file sharing*. Nel 2008, la Federal Communications Commission (FCC) sanzionò Comcast per questo comportamento, ma recentemente (Aprile

2010) la corte d'appello federale del distretto di Columbia ha ribaltato la decisione. Dunque, da un punto di vista tecnico, l'identificazione del traffico permette all'ISP di penalizzare alcune categorie di applicazioni, ma l'effettivo utilizzo di una politica di questo tipo non è ancora stato regolato dal Legislatore.

2. ANALISI DEI FLUSSI DI TRAFFICO

L'analisi dei flussi e l'identificazione del traffico Internet avvengono tramite l'osservazione dei pacchetti IP che costituiscono il flusso stesso. Un flusso di traffico è definito come una sequenza di pacchetti IP che condividono gli stessi indirizzi IP di sorgente e di destinazione (*Source Address e Destination Address* nella Figura 1), gli stessi numeri di porta di sorgente e di destinazione (Source Port e Destination Port) e lo stesso protocollo (il campo *Protocol* dell'*header* IP) di trasporto. Questi cinque campi, che nella figura 1 sono evidenziati con uno sfondo arancione, sono usualmente denominati la "*quintupla*" e identificano efficacemente il flusso al quale un pacchetto IP appartiene. Come conseguenza di questa definizione, un flusso di traffico è monodirezionale e, se si intende esaminare il comportamento complessivo di un'applicazione (traffico di andata e traffico di ritorno), si dovranno esaminare congiuntamente i due flussi monodirezionali che trasportano il traffico dell'applicazione nei due sensi (nel caso di un'applicazione *client-server*, i due flussi sono quello da *client a server* e quello da *server* a *client* che, insieme, formano il flusso logico bidirezionale di collegamento remoto tra *client* e *server*).

2.1. Caratteristiche dei flussi di traffico in Internet

Dal punto di vista degli ISP, le caratteristiche più critiche di un flusso di traffico sono la durata, il volume complessivo di traffico trasportato, la velocità di trasmissione e il grado di variabilità di questa velocità. Queste caratteristiche presentano una grande variabilità [3] e sulla base di queste si usa suddividere i flussi di traffico in un insieme di categorie. La categorizzazione più utilizzata è quella illustrata nella figura 2. Almeno il 45% dei flussi ha un tempo di vita di meno di 2 s - sono i flussi denominati *dragonfly* (libellula) - e circa il 98% dei flussi dura meno di 15 min. Il restante 2% dei flussi ha una durata di parecchie ore o giorni, sono denominati *tortoise* (tartaruga), e trasportano il 50%-60% dei byte trasmessi su un link. Per quanto riguarda il volume di byte trasportati, un flusso è un *mouse* (topo) o un *elephant* (elefante). Un *mouse* trasporta pochissimo traffico, ma i *mouse* sono decisamente numerosi. D'altra parte, i pochi flussi *elephant* sono

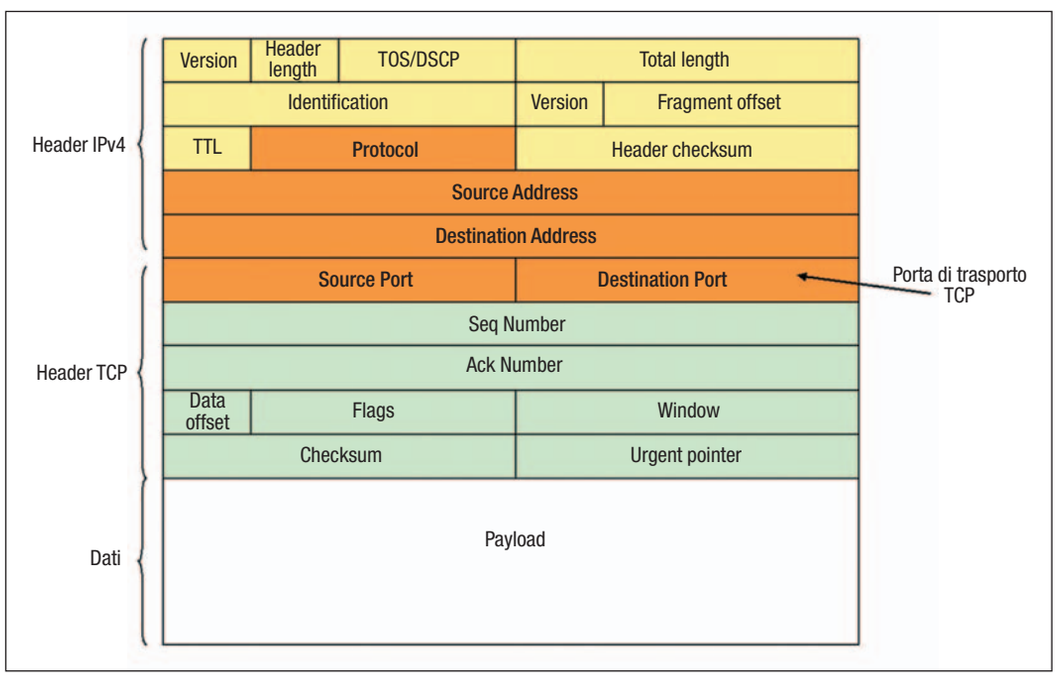


FIGURA 1
La porta di trasporto in un pacchetto IPv4/TCP

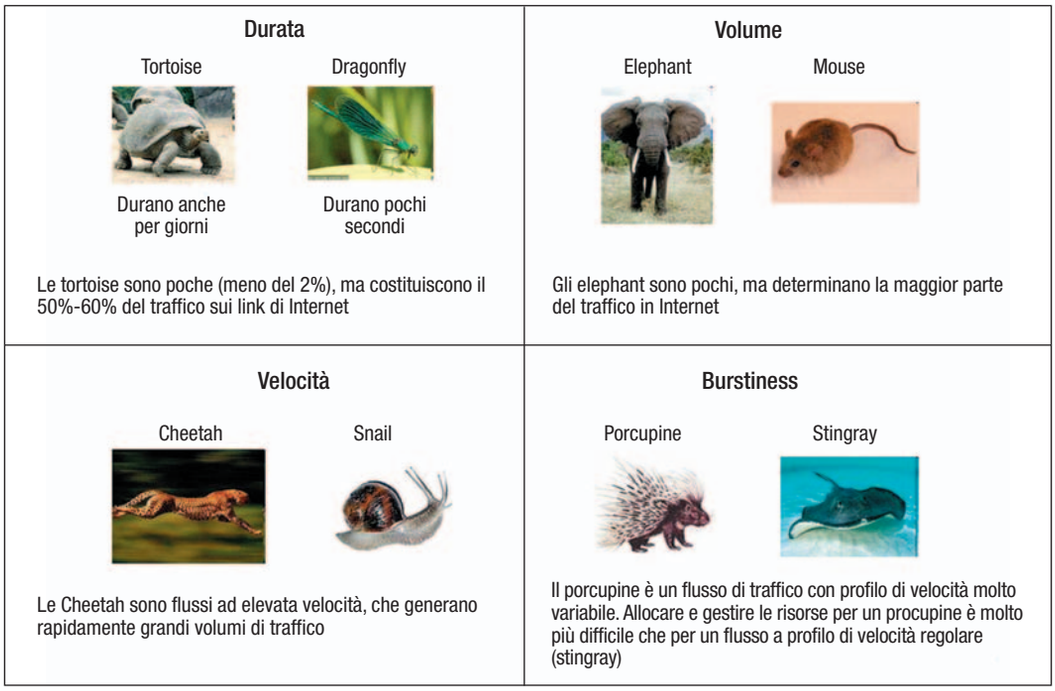


FIGURA 2
Tipologie di flussi in Internet, classificati secondo il volume, la durata, la velocità e la burstiness

responsabili della maggior parte del traffico sui *link* della rete. La terza caratteristica significativa di un flusso di traffico è la sua *burstiness*. Un flusso con elevata *burstiness* presenta una velocità molto variabile nel tempo e passa rapidamente da uno stato in cui lavora a bassa velocità ad uno stato in cui trasmette a velocità molto elevata, e viceversa. Un flusso ad elevata *burstiness* è un *porcupine* (porcospino). Al contrario, flussi che presentano un profilo di velocità molto regolare e con poche variazioni sono denominati *stingray* (razza) [5]. Infine, un flusso può essere caratterizzato da una velocità media molto elevata, e in questo caso è denominato *cheetah* (ghepardo), o molto piccola, è il caso dello *snail* (lumaca).

L'ISP, individuando le categorie di appartenenza dei flussi di traffico presenti sui link della propria rete, potrà evidenziare i flussi più critici, ponendo per esempio particolare attenzione agli *elephant*, che determinano l'aliquota maggiore del traffico in rete, per poterli gestire nel migliore dei modi. Per esempio, è logico instradare gli *elephant* attraverso i link provvisti di più risorse disponibili. In pratica, i flussi di traffico che devono destare più attenzione sono quelli ad alto volume (*elephant*), quelli veloci (*cheetah*) perché richiedono l'allocazione di molta banda sui link, quelli lunghi (*tortoise*) perché le risorse allocate rimangono impegnate a lungo e quelli molto variabili (*porcupine*), perché risulta molto delicata la determinazione delle risorse richieste per servirli con un livello di qualità adeguato (al contrario, determinare la banda richiesta per servire uno *stingray* è semplice, in quanto il profilo di velocità è poco variabile). Flussi che presentano tutte le quattro caratteristiche al livello di criticità massimo sono molto rari e sono più frequenti situazioni intermedie come per esempio flussi ad alto volume (*elephant*) e lunghi (*tortoise*), ma non particolarmente veloci o *bursty*. Anche questi flussi sono critici e da trattare in modo adeguato.

Vi sono diversi metodi per individuare le quattro categorie di un flusso di traffico [1]. Partendo dalla constatazione che in Internet una piccola percentuale dei flussi è responsabile della maggior parte del traffico, una tecnica utilizzata per determinare se un flusso è un *elephant* o un *mouse* consiste nel calcolare il volume medio e la deviazione standard del volume di traffico generato dai flussi sul link in esame. Un flusso è classificato come *elephant* se il suo volume è più grande del volume medio dei flussi, più tre volte la deviazione standard, ed è classificato come mouse altrimenti. Secondo lo stesso principio, si classifica un flusso come *tortoise* se la sua durata è più grande della durata media dei flussi, più tre volte la deviazione stan-

dard della durata stessa, altrimenti il flusso è un *dragonfly*. Nello stesso modo si procede per classificare un flusso come *cheetah/snail e porcupine/stingray*.

3. L'IDENTIFICAZIONE DEL TRAFFICO INTERNET

Come già evidenziato, l'identificazione del traffico Internet si differenzia sostanzialmente dall'analisi dei flussi, in quanto la prima intende individuare l'applicazione che ha generato un flusso di traffico e la seconda ha lo scopo di misurare alcune caratteristiche complessive del traffico trasportato dai flussi.

3.1. L'identificazione del traffico Internet tramite il numero di porta

Tradizionalmente, l'identificazione dell'applicazione che genera un flusso di traffico sotto osservazione è stata eseguita tramite il riconoscimento delle porte a livello di trasporto. L'analisi delle porte di trasporto è stata (fino a che la si è potuta utilizzare) un metodo semplice ed efficace per identificare le applicazioni tramite l'osservazione dei pacchetti appartenenti al flusso. Come mostrato nella figura 1, un pacchetto IPv4 che trasporta i dati di un'applicazione che utilizza il *Transmission Control Protocol* (TCP) come livello di trasporto, riporta esplicitamente il numero di porta di destinazione (*Destination Port*) nell'*header* del TCP. La porta di destinazione ha lo scopo di identificare esplicitamente l'applicazione che utilizza i dati contenuti nel pacchetto. Esistono porte allocate ufficialmente dall'*Internet Assigned Numbers Authority* (IANA) [6]; per esempio, la porta 25 è assegnata al protocollo *Simple Mail Transfer Protocol* (e-mail), la porta 23 al *The Secure Shell* (SSH) *Protocol*, la porta 80 ad HTTP, originalmente utilizzata per il *web browsing*, e così via. Se un'applicazione utilizza una delle porte assegnate per raggiungere l'applicazione prevista (per esempio, il World Wide Web sulla porta 80), il riconoscimento dell'applicazione alla quale appartiene un pacchetto è direttamente desumibile dall'ispezione diretta della porta di destinazione. Anche nel caso in cui un'applicazione non sia assegnataria di una porta ufficiale, ma usi sempre una porta o un insieme di porte ben definito (come facevano per esempio alcuni sistemi *peer-to-peer* di prima generazione), il riconoscimento è immediato tramite lo stesso metodo. Si nota infine che le stesse considerazioni si applicano nel caso in cui il protocollo di trasporto utilizzato sia lo *User Datagram Protocol* (UDP) invece che il TCP.
Questo metodo, purtroppo, oggi è applicabile in un numero ridotto di casi, in quanto molte applicazioni, e proprio quelle che generano la maggior parte del traffico, selezionano le porte dinamicamente e, soprattutto, utilizzano porte generiche, per esempio la porta 80, originalmente dedicata all'applicazione World Wide Web, con l'obiettivo di mascherarsi e di rendere un'applicazione (per esempio, di *peer-to-peer file sharing*) indistinguibile da un normale web browsing. Intorno agli anni 2003-2004, quando queste tecniche entrarono massivamente in campo, sembrò di registrare un calo del traffico *Peer-to-Peer*. Questa fu una deduzione errata, ma presto riconosciuta come tale: in realtà le applicazioni *Peer-to-Peer* avevano cominciato a nascondersi (da qui il titolo significativo "Is P2P *dying or just hiding*?" dell'articolo [7] pubblicato nel 2004, quando si iniziò a riscontrare questo fenomeno).
In conclusione, la mera analisi del numero di porta di destinazione ormai non fornisce all'ISP un'identificazione affidabile dell'applicazione che un flusso di traffico sta trasportando, quindi, è necessario utilizzare metodi diversi e più complessi.

3.2. La packet inspection

Un'altra metodologia tradizionale per l'identificazione del traffico Internet è la *packet inspection*, che consiste nell'osservare il contenuto dei pacchetti per identificare gli scambi protocollari che avvengono attraverso i flussi e, quindi, identificare l'applicazione. Potenzialmente questo è un metodo molto efficace che, esaminando una molteplicità di caratteristiche dei pacchetti, può superare il problema del mascheramento della porta di trasporto. D'altra parte, sussistono alcune problematiche che, in realtà, rendono la *packet inspection* in generale insufficiente. In primo luogo, un'analisi protocollare deve essere di tipo *stateful*, cioè, è necessario memorizzare e tenere aggiornato uno stato per ogni flusso esaminato, al fine di registrare lo stato corrente del protocollo ipotizzato e, quindi, verificare la

bontà dell'ipotesi. Questo produce chiaramente problemi di scalabilità su link ad alta capacità e per un elevato numero di flussi. Inoltre, si deve tenere presente che molte applicazioni cifrano il *payload* dei pacchetti e, di conseguenza, gli scambi protocollari tendono a diventare sempre meno osservabili. Infine, l'osservazione del *payload* dei pacchetti si scontra con problematiche di privacy dei dati personali degli utenti in molti Paesi.

La *packet inspection* può essere resa più scalabile tramite un approccio *stateless*, rinunciando ad intercettare esplicitamente gli scambi protocollari e ricercando particolari stringhe nel *payload* dei pacchetti. Per esempio, l'applicazione *peer-to-peer* eDonkey contiene la stringa '\xe3\x38' nel *payload* del pacchetto IP, alcune *query* Web contengono la stringa '\GET', e così via. Una tale ispezione del *payload* dei pacchetti permette una buona precisione dell'identificazione, ma presenta gli stessi svantaggi dell'ispezione *stateful*, per quanto riguarda la cifratura dei pacchetti e le problematiche relative alla privacy.

Vi sono diversi strumenti, anche aperti, disponibili per l'esecuzione della *packet inspection*. Per esempio, Snort [8] è un software aperto che può eseguire analisi del traffico in tempo reale, effettuando ricerche di particolari stringhe o sequenze di stringhe nei pacchetti. Snort, in tal modo, è anche in grado di rilevare la presenza di vari tipi di attacco nel momento in cui l'attacco stesso si sviluppa. In conclusione, la *packet inspection* è uno strumento efficace, ma la cui validità non è completa soprattutto a causa dell'impossibilità di individuare stringhe caratteristiche nel *payload* di un pacchetto criptato.

3.3. Classificazione basata su caratteristiche statistiche del traffico

Sia la mera analisi della porta di trasporto che (in misura minore) la *packet inspection*, presentano limiti nell'identificazione del traffico Internet i quali possono essere superati da approcci più moderni che esaminano le caratteristiche statistiche dei flussi di traffico, rilasciando la necessità di esaminare il *payload* applicativo trasportato dai pacchetti. Caratteristiche statistiche dei flussi di traffico possono essere le distribuzioni dei tempi di interarrivo dei pacchetti e la loro correlazione, nonché la distribuzione delle lunghezze dei pacchetti, e la loro correlazione. Il presupposto di questo tipo di analisi è che diverse applicazioni presentino differenze osservabili nel processo di generazione dei pacchetti. Questa ipotesi è stata verificata con successo [9]; si è rilevato, per esempio, che diverse applicazioni TCP/IP sono contraddistinte da caratteristiche significativamente diverse nel processo di interarrivo dei pacchetti, delle lunghezze dei pacchetti [10] e della distribuzione delle lunghezze dei pacchetti [11, 12, 13].

Un esempio pratico di questo fatto è illustrato nella figura 3 **A** [14], che mostra il risultato della misurazione della lunghezza dei primi due pacchetti di flussi di traffico generati dalle applicazioni *Simple Mail Transfer Protocol* (SMTP) ed EDonkey. I grafici della figura riportano sul-

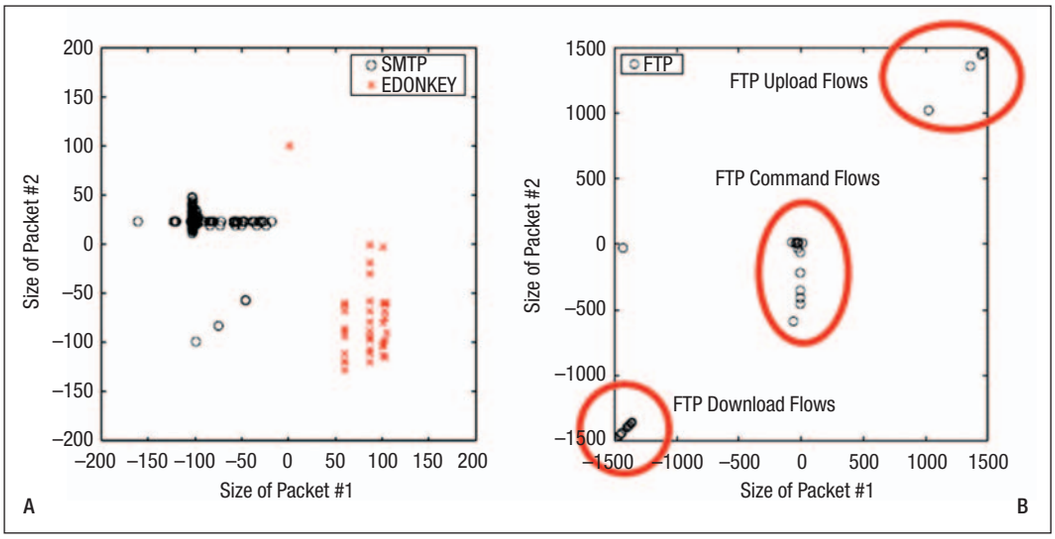


FIGURA 3
Lunghezze dei primi due pacchetti di **A** *- flussi SMTP ed EDonkey e* **B** *- flussi FTP (Figura tratta da [14])*

l'asse *X* la lunghezza del primo pacchetto di ogni flusso esaminato e, sull'asse *Y*, la lunghezza del secondo pacchetto. In questo modo, nella figura ad ogni flusso osservato corrisponde un punto la cui ascissa e ordinata sono rispettivamente la lunghezza del primo e del secondo pacchetto del flusso. Si nota che i punti relativi alle due applicazioni tendono a concentrarsi in regioni decisamente separate, pertanto, l'insieme dei due attributi selezionati costituisce una buona base per la definizione di regioni di decisione efficaci per distinguere SMTP da EDonkey. La figura 3 **B** [14] mostra, per la stessa coppia di caratteristiche, i punti misurati per un insieme di flussi *File Transfer Protocol* (FTP). In questo caso, si nota che i flussi *upload* (da *client a server*) e *download* (da *server* a *client*) occupano regioni separate del piano e altrettanto si può stabilire per i flussi di controllo (che sono necessari per il funzionamento di FTP).

Sulla base di questa constatazione, sono stati introdotti nuovi metodi di classificazione che osservano il processo di generazione dei pacchetti in un flusso e ne identificano alcune caratteristiche statistiche distintive. Questo tipo di approccio risolve contemporaneamente il problema della cifratura dei pacchetti e le questioni di privacy legate all'ispezione dei *payload* dei pacchetti.

Se si desidera identificare l'applicazione che ha generato un flusso di traffico misurandone e valutandone alcune caratteristiche (dette anche attributi) statistiche, è importante tenere conto del fatto che, in generale, un'applicazione stabilisce almeno due flussi unidirezionali. Per esempio, anche nel semplice caso del Web Browsing, allo scaricamento di una pagina che genera un traffico da Web Server a utente, corrisponde un flusso di riscontri TCP da utente verso Web Server. Per ottenere una buona precisione nell'identificazione dell'applicazione è vantaggioso considerare contemporaneamente i due flussi applicativi nelle due direzioni. Tali flussi sono chiamati *forward* (per esempio un file scaricato) e *backward* (per esempio, i riscontri TCP relativi allo scaricamento nella direzione *forward*). Alcuni attributi usualmente esaminati per identificare i flussi di traffico sono riportati nella tabella 1.

La tabella 1 riporta quattro gruppi di attributi. Si osserva che i gruppi 1 e 2 fanno riferimento a caratteristiche complessive di un flusso, che possono essere determinate solo dopo che il flusso ha terminato la sua fase attiva e viene abbattuto. Infatti, l'attributo 1 (valore minimo, massimo, medio e deviazione standard della lunghezza dei pacchetti nella direzione "*forward*") può essere quantificato solo dopo che si è registrata la lunghezza di tutti i pacchetti del flusso esaminato e si è elaborata la statistica complessiva di queste lunghezze. Le stesse considerazioni valgono per gli attributi del gruppo 2. Al contrario, gli attributi dei gruppi 3 e 4 richiedono l'osservazione di pochissimi pacchetti (dei primi tre o dei primi cinque) di un flusso. Quindi, gli attributi dei gruppi 3 e 4 permettono l'identificazione dei flussi "*on the fly*", velocissima e in tempo reale, che può essere determinata già nei primi istanti di vita del flusso. Gli attributi dei gruppi dei gruppi 1 e 2 corrispondono ad un'identificazione in generale molto più lenta e che non può essere considerata in tempo reale. Come si vedrà nel seguito, gli attributi dei gruppi 3 e 4 sono (forse sorprendentemente) molto efficaci - soprattutto quelli del gruppo 4 - e quindi permettono un'identificazione sia veloce che precisa.

Un sistema di classificazione dei flussi seleziona un insieme di *N* attributi, definendo così uno spazio *N*-dimensionale dove la *i*-esima dimensione quantifica il valore che assume l'*i*-esimo attributo. Osservando un flusso di traffico si possono misurare gli attributi selezionati e, quindi, attribuire al flusso esaminato una coordinata (cioè, un punto) nello spazio *N*-dimensionale degli attributi. Se gli attributi sono selezionati in modo appropriato, si osserva che i flussi di traffico generati da una specifica applicazione tendono a concentrarsi, nello spazio *N*-dimensionale degli attributi, in certe regioni e non in altre occupate da altre applicazioni. L'esempio riportato nella figura 4 illustra il procedimento nel caso di due attributi. Nella figura 4 **A** è mostrata la regione dello spazio bidimensionale, relativo ai due attributi selezionati, nella quale vanno a concentrarsi i punti relativi ai flussi di traffico generati da una specifica applicazione *A*. Questa regione può essere ottenuta mediante osservazione di flussi di traffico per i quali è noto che l'applicazione che li ha generati è proprio *A*.

Una volta che la regione relativa all'applicazione *A* è definita, si è diviso il piano di identificazione in due regioni: la regione associata

	Gruppo di attributi 1
1.	Valore minimo, massimo, medio e deviazione standard della lunghezza dei pacchetti nella direzione "*forward*";
2.	Valore minimo, massimo, medio e deviazione standard della lunghezza dei pacchetti nella direzione "*backward*";
3.	Valore minimo, massimo, medio e deviazione standard dei tempi di interarrivo dei pacchetti nella direzione "*forward*";
4.	Valore minimo, massimo, medio e deviazione standard dei tempi di interarrivo dei pacchetti nella direzione "*backward*";
5.	Valore del campo "*protocol*" dell'*header* dei pacchetti IP;
6.	Numero totale di flag TCP URG e PUSH nella direzione "*forward*";
7.	Numero totale di flag TCP URG e PUSH nella direzione "*backward*";
	Gruppo di attributi 2
8.	Durata del flusso;
9.	Numero totale di byte e pacchetti nella direzione "*forward*";
10.	Numero totale di byte e pacchetti nella direzione "*backward*";
	Gruppo di attributi 3
11.	Lunghezza dei primi tre pacchetti nella direzione "*forward*";
12.	Lunghezza dei primi tre pacchetti nella direzione "*backward*";
13.	Tempi di interarrivo dei primi tre pacchetti nella direzione "*forward*";
14.	Tempi di interarrivo dei primi tre pacchetti nella direzione "*backward*";
	Gruppo di attributi 4
15.	Lunghezza dei primi cinque pacchetti nella direzione "*forward*";
16.	Lunghezza dei primi cinque pacchetti nella direzione "*backward*";
17.	Tempi di interarrivo dei primi cinque pacchetti nella direzione "*forward*";
18.	Tempi di interarrivo dei primi cinque pacchetti nella direzione "*backward*".

TABELLA 1
Gruppi di attributi frequentemente utilizzati nella classificazione del traffico Internet

all'applicazione *A* e la regione complementare, associata a tutte le applicazioni diverse da *A*. A questo punto, osservando un flusso per il quale l'applicazione che lo ha generato è ignota, si misureranno le due caratteristiche selezionate e si identificherà quindi un punto nel piano. Se tale punto cade all'interno della regione associata all'applicazione *A*, si deciderà che il flusso è stato generato da *A*, altrimenti, si decide che il flusso è stato generato da un'applicazione diversa. La figura 4 **B** esemplifica il procedimento, mostrando con punti neri i flussi di traffico che **realmente** sono stati generati da *A* e con punti bianchi i flussi di traffico che **in realtà** sono stati generati da un'applicazione diversa da *A*. I punti neri che sono compresi nella regione associata ad *A* corrispondono a classificazioni corrette, così come i punti bianchi che cadono al di fuori della regione associata ad *A*. D'altra parte, sono possibili errori. Infatti, un flusso generato da *A* potrebbe essere classificato come "non-*A*". Nella figura 4 **B** quest'evenienza è rappresentata dal punto nero fuori dalla regione associata ad *A*, ed è denominato "Falso Negativo". Un altro tipo di errore è il "Falso Positivo", cioè, un flusso che viene classificato come originato dall'applicazione *A*, ma in realtà non lo è. Nella figura 4 **B** ciò è rappresentato dal punto bianco compreso nella re-

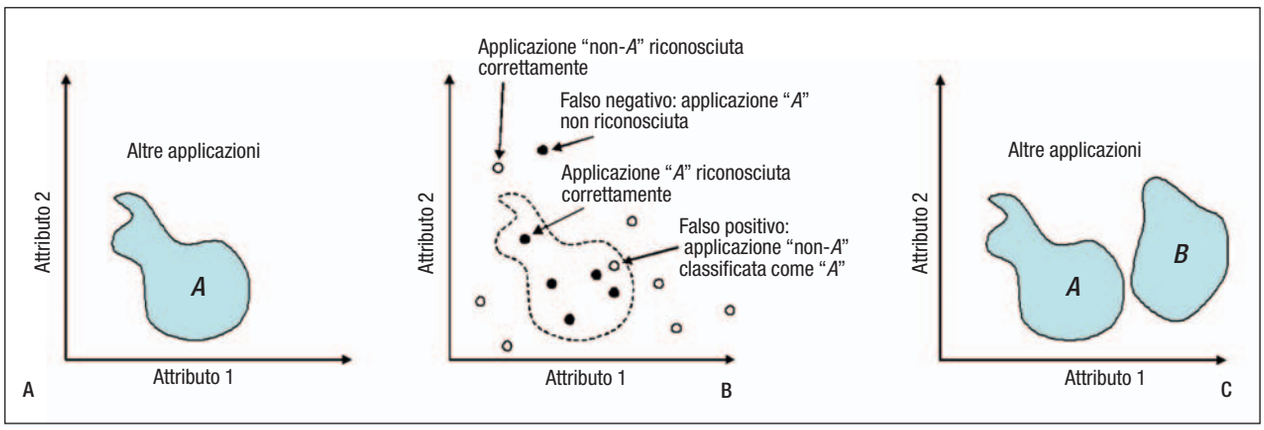


FIGURA 4

Classificazione delle applicazioni sulla base di caratteristiche preselezionate dei flussi

gione associata ad *A*. In generale si desidera identificare contemporaneamente più di un'applicazione e ciò si ottiene, come mostrato nella figura 4 **C**, definendo nello spazio *N*-dimensionale le regioni associate ad ognuna delle applicazioni di interesse (le applicazioni *A* e *B* nell'esempio).

Due importanti metriche per la quantificazione della precisione di un sistema di identificazione delle applicazioni sono la percentuale di falsi negativi e di falsi positivi, definiti rispettivamente *False Negative Rate* (FNR) e *False Positive Rate* (FPR), che dovrebbero essere piccoli. A queste metriche si aggiungono: il *True Negative Rate* (TNR) e il *True Positive Rate* (TPR).

3.4. Il machine learning supervisionato

Una delle principali problematiche nell'applicazione delle tecniche di identificazione basate sull'analisi degli attributi dei flussi di traffico è la costruzione delle regole tramite le quali si identifica l'applicazione che ha generato un flusso di traffico. Alcuni metodi di *machine learning* costruiscono effettivamente lo spazio *N*-dimensionale degli attributi e le relative regioni di decisione. Altre tecniche elaborano insiemi di regole di decisione che non coinvolgono direttamente la costruzione delle regioni nello spazio degli attributi. Anche in quest'ultimo caso, la selezione di attributi che distinguono chiaramente le differenze tra le applicazioni in esame è un prerequisito essenziale.

Alcune tecniche di *machine learning* utilizzano un insieme di esempi pre-classificati per inferire automaticamente una serie di regole di classificazione tramite le quali si procede all'identificazione dei flussi di traffico non compresi nell'esempio fornito inizialmente. Questo tipo di tecnica è detta "*supervisionata*", in quanto l'algoritmo di identificazione del traffico è preliminarmente addestrato con informazioni preconfezionate. Tramite le regole di classificazione costruite sulla base degli esempi forniti, l'algoritmo di classificazione fornisce, a fronte di un input costituito da un flusso di traffico, una classificazione dello stesso in una delle possibili categorie (applicazioni) definite nell'esempio iniziale utilizzato per l'addestramento. L'inizializzazione di un algoritmo di *machine learning* supervisionato prevede:

- la fase di *addestramento*, nella quale il motore di classificazione esamina l'esempio fornito ed elabora il modello di classificazione;
- la successiva fase di *testing*, nella quale si forniscono al motore di classificazione alcuni flussi generati da applicazioni note, per verificare la precisione della classificazione operata. Una volta esaurita la fase di *testing*, il motore di classificazione può essere messo in produzione.

Nel caso della figura 4 **C**, un esempio d'addestramento potrebbe essere strutturato come una serie di flussi f_i, ad ognuno dei quali è associata la terna (x_{1i}, x_{2i}, y_i), dove x_{1i} e x_{2i} sono il valore dell'attributo 1 e 2, rispettivamente, dell'*i*-esimo flusso dell'esempio, e y_i assume il valore *A* o *B*, a seconda che il flusso f_i sia stato generato dall'applicazione *A* o da *B*. Sulla base di un esempio sufficientemente corposo, il motore di *machine learning* è in grado di costruire le regioni relative ad *A* e *B* mostrate

nella figura 4 **C**. La generalizzazione nel caso di un insieme di *N* caratteristiche è intuitiva.
Un problema significativo degli algoritmi di *machine learning* supervisionati è che gli esempi per l'addestramento devono essere costituiti da flussi (numerosi) correttamente pre-identificati. Anche la fase di test deve avvenire alimentando il motore di classificazione con flussi non appartenenti all'esempio iniziale e anch'essi correttamente pre-identificati. La creazione di collezioni di flussi correttamente identificati per l'addestramento e per il testing è una fase lunga e costosa.
Esempi di tecniche di machine learning supervisionate sono il *Naïve Baye, le Bayesian Network*, il *C4.5 Decision Tree*, e le *Support Vector Machines*.

3.4.1. Il Naïve Bayes

Il metodo *Naïve Bayes*, come dice il nome, si basa sul noto teorema sulle probabilità condizionate di Bayes, che è utilizzato in quest'ambito per individuare l'applicazione che più verosimilmente ha generato un flusso di traffico osservandone alcuni attributi selezionati, per esempio, tra quelli della tabella 1. Per ipotesi, si assuma di avere due classi (applicazioni), C_1 e C_2, e un insieme di due attributi, X_1 e X_2, che possono essere le lunghezze in byte dei primi due pacchetti di un flusso. Durante la fase di addestramento, osservando un insieme di flussi noti generati dalle applicazioni C_1 e C_2, si determinano le distribuzioni di probabilità delle lunghezze dei primi due pacchetti dei flussi generati dall'applicazione C_1, e il calcolo è ripetuto per l'applicazione C_2. In seguito, dato un flusso ignoto, se ne misurano gli attributi X_1 e X_2 (cioè, le lunghezze dei primi due pacchetti) e applicando il teorema di Bayes si individua l'applicazione che più verosimilmente ha generato il flusso.

Attributo	Possibili valori
Tempo atmosferico	Sole, nuvoloso, pioggia
Temperatura	Variabile continua
Umidità	Variabile continua
Vento	Si, no

TABELLA 2
Attributi e i possibili valori ad essi associati

3.4.2. L'algoritmo C.45 "decision tree"

L'algoritmo C4.5 crea una struttura decisionale ad albero, nel quale i nodi rappresentano gli attributi e gli archi rappresentano i valori che connettono gli attributi. La costruzione dell'albero decisionale è la fase più complessa di C.45 e avviene durante l'addestramento applicando procedure complesse per ottimizzarne la costruzione, tentando di minimizzare i casi ambigui, che portano ad errori di classificazione. In questa sede si propone un semplice esempio per illustrare i concetti generali applicati dall'algoritmo. Si suggerisce la lettura di [15] per un approfondimento.
Si consideri una casistica di giocatori di golf che, in base alle condizioni atmosferiche, decidono se giocare o non giocare. Gli attributi del tempo atmosferico considerati sono illustrati nella tabella 2 e ad essi sono associati i possibili valori, che possono essere variabili discrete o continue. La sequenza di addestramento registra le decisioni effettivamente prese da alcuni giocatori ed è illustrata nella tabella 3.
L'elaborazione dei dati da parte di C.45 porta all'albero decisionale mostrato nella figura 5 **A**. Si nota che, se il tempo è nuvoloso si prenderà sempre la decisione di giocare, mentre se c'è il sole o se piove la decisione dipende da altri fattori come umidità e vento. La temperatura è stata eliminata come variabile decisionale, in quanto l'algoritmo di costruzione dell'albero ha stabilito, sulla base della sequenza di addestramento, che il guadagno informativo che si otterrebbe includendo tale variabile non è sufficiente a giustificarne l'utilizzo e, quindi, a rendere più complesso e ramificato l'albero. Questa eliminazione di un attributo è generata da speciali procedure di "potatura" dell'albero decisionale operate da C.45.
La figura 5 **B** mostra che la costruzione dell'albero decisionale di C.45 porta a definire, nello spazio degli attributi, delle zone di decisione. Lo spazio degli attributi è tridimensionale e non quadridimensionale, in quanto la variabile temperatura è stata eliminata da C.45.

3.4.3. Bayesian networks

Una *Bayesian Network* è una struttura decisionale che consiste in un grafo dove i nodi rappresentano attributi o classi e gli archi de-

Tempo atmosferico	Temperatura	Umidità	Vento	Si gioca?
Sole	29.4	85%	No	No
Sole	26.6	90%	Si	No
Nuvoloso	28.3	78%	No	Si
Pioggia	21.1	96%	No	Si
Pioggia	20	80%	No	Si
Pioggia	18.3	70%	Si	No
Nuvoloso	17.7	65%	Si	Si
Sole	22.2	95%	No	No
Sole	20.5	70%	No	Si
Pioggia	23.8	80%	No	Si
Sole	23.8	70%	Si	Si
Nuvoloso	22.2	90%	Si	Si
Nuvoloso	27.2	75%	No	Si
Pioggia	21.6	80%	Si	No

TABELLA 3
Dati di addestramento

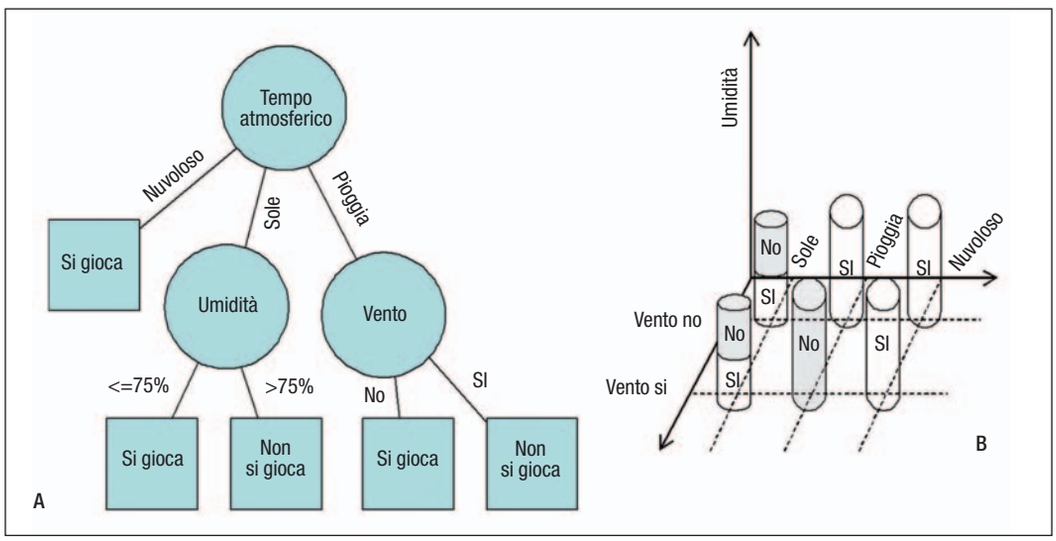


FIGURA 5
A - *Albero decisionale nell'esempio di applicazione dell'algoritmo C.45;* **B** - *rappresentazione delle zone di decisione costruite dall'albero decisionale nello spazio degli attributi*

scrivono le mutue relazioni tra i nodi. Il peso di un arco è una probabilità condizionata e quantifica l'entità della relazione tra i due nodi che esso collega. Dato il grafo decisionale (costruito nella fase di addestramento) e le probabilità condizionate che pongono in relazione i nodi del grafo, la classificazione di un flusso è un'operazione relativamente semplice e, anche in questo caso, la vera criticità è la costruzione del grafo decisionale in fase di addestramento. Per un approfondimento sulle Bayesian Networks si veda [16].

3.4.4. Support Vector Machines

Le *Support Vector Machines* (SVM) sono una tecnica generale di *pattern recognition* che può essere utilizzata per la classificazione del traffico Internet mediante *machine learning*

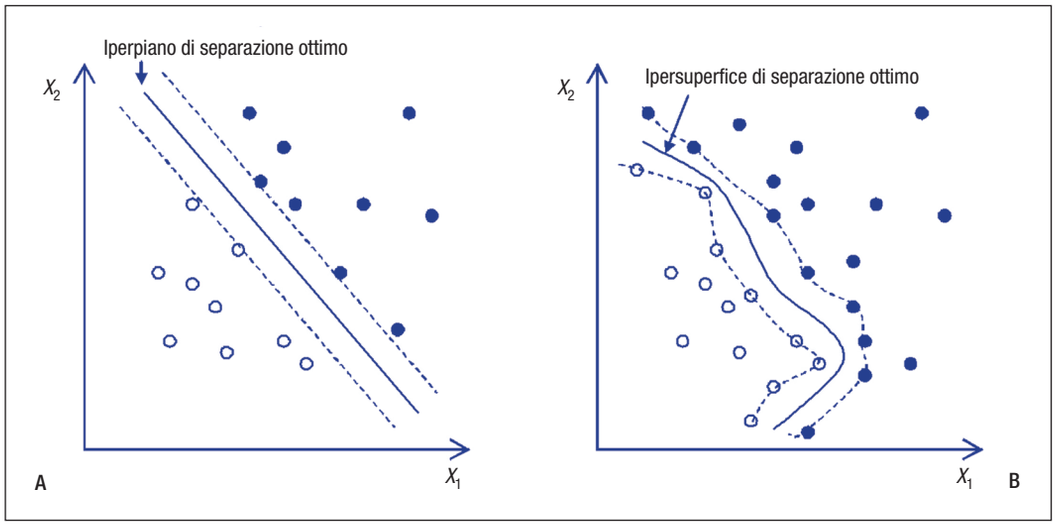


FIGURA 6
Separazione delle zone di decisione operata dalle Support Vector Machines

supervisionato. Le SVM affrontano direttamente il problema della costruzione delle zone di decisione nello spazio degli attributi. La figura 6 mostra, nel caso bidimensionale di due attributi, i valori (punti) delle istanze presenti nella sequenza di addestramento. Sono presenti due classi (applicazioni) distinte da un cerchio vuoto e un cerchio nero. Nella figura 6 **A** si mostra il caso in cui le istanze delle due classi si raggruppano in due distinte zone, separabili da una linea. Questa linea in generale è denominata l'iperpiano di separazione (la denominazione "iperpiano" deriva dal fatto che nel generale caso di *N* attributi la linea di separazione è in realtà un piano in *N*-1 dimensioni). L'attribuzione di una nuova istanza ad una classe, quando il motore di classificazione è in esercizio, risulta molto semplice se la linea di separazione è un iperpiano (e quindi lineare). L'operazione è più complessa nel caso generale in cui la linea di separazione non è più un iperpiano, ma una superficie più generica, come mostrato nella figura 6 **B**. In questo caso, le SVM operano una trasformazione dello spazio degli attributi mediante la quale la linea di separazione diventa un iperpiano e quindi operano nello spazio trasformato.

3.5. Tecniche di clustering

Le tecniche di classificazione supervisionate sono basate sulla conoscenza a priori di quali siano le classi (applicazioni) da discriminare. Le tecniche di *clustering*, invece, non sono provviste di questa informazione e cercano autonomamente correlazioni e somiglianze/differenze negli attributi dei flussi osservati, rintracciando schemi ricorrenti. In tal modo, un algoritmo di *clustering* raggruppa autonomamente le istanze osservate in regioni nello spazio degli attributi, andando ad individuare quei gruppi di istanze che formano nello spazio degli attributi "costellazioni" abbastanza raggruppate (si veda la Figura 3 **A**). I gruppi così creati possono essere *hard* (le regioni sono completamente separate) o *soft* (le regioni sono parzialmente sovrapposte).
Un buon algoritmo di *clustering* tende a produrre *cluster* che sono *dominati* da un'applicazione (cioè, una percentuale elevata dei flussi concentrati nel cluster appartengono ad una singola applicazione, che domina il *cluster*) e nei quali la presenza di flussi appartenenti ad altre applicazioni è marginale. Ciò garantisce che, quando il motore di classificazione utilizzerà i *cluster* per classificare i flussi, si avrà una buona precisione, cioè, pochi falsi positivi e pochi falsi negativi.
Un esempio di algoritmo di *clustering* è presentato in [14, 17] (si veda la Figura 3 **A**), dove il grado di somiglianza tra due flussi è quantificato dalla distanza euclidea dei rispettivi punti nel piano che rappresenta le due caratteristiche misurate. Una volta definiti i cluster, un flusso ignoto in osservazione è assegnato al cluster il cui centro, nel piano di Figura 3 **A**, è più vicino al punto associato al nuovo flusso.
Una volta creati i *cluster*, è necessario assegnare ad ogni cluster l'applicazione che lo

domina, in modo tale da abilitare la classificazione di nuove istanze quando il classificatore è messo in esercizio. È a questo punto che anche un algoritmo di *clustering* deve essere addestrato con una sequenza per la quale le applicazioni che hanno generato le istanze sono note. L'algoritmo di *clustering* associerà ad ogni *cluster* l'applicazione che più lo rappresenta. In tal senso, anche un classificatore che adotta una tecnica di *clustering* prevede una fase supervisionata, denominata anche *fase di labeling*. In [14], si osserva che utilizzando i primi 5 pacchetti di un flusso si ottiene una buona separazione tra i *cluster* e un'efficiente classificazione.

3.5.1. Prestazioni degli algoritmi di *machine learning*

Non esiste in letteratura uno studio completo che analizzi in condizioni omogenee le prestazioni di tutti gli algoritmi di *machine learning* esistenti. Esistono comunque alcuni lavori che esaminano e raffrontano alcuni casi significativi, per esempio [18], nel quale si comparano le prestazioni di un algoritmo di classificazione supervisionato basato sulle Bayesian Networks [16], tecniche di *clustering* basate sull'algoritmo delle K-means [17], algoritmi Bayesiani (basati su tecniche derivate dal *Naïve Bayes*, ma dotate di soluzioni aggiuntive avanzate per la riduzione degli errori di classificazione) che includono i tempi di interarrivo dei pacchetti negli attributi esaminati [19] e, infine, algoritmi supervisionati, che applicano C.45 [18].
Gli algoritmi sono addestrati e testati fornendo in input tre diverse tracce pubblicamente disponibili: le tracce auckland-vi-20010611 e auckland-vi-20010612, che sono state raccolte sul medesimo link di rete e la traccia nzix-ii-20000706, raccolta su un link diverso. Gli algoritmi sono addestrati con la traccia auckland-vi-20010611 (il *campione A-addestramento*), sono poi verificati tramite la traccia auckland-vi-20010612 (il *campione A-test*) e con la traccia nzix-ii-20000706 (il *campione B-test*). Lo scopo di verificare gli algoritmi in due casi, con una traccia raccolta sullo stesso link utilizzato per l'addestramento e su di un link diverso, è quello di analizzare la "portabilità" dell'algoritmo di classificazione, cioè, la possibilità di eseguire un unico addestramento e poi di utilizzare l'algoritmo in punti della rete diversi da quello sul quale l'addestramento è stato svolto. I vantaggi di un motore di classificazione "portatile" sono evidenti: si possono ridurre notevolmente i costi di addestramento in quanto un solo ciclo di addestramento è sufficiente per operare su tutti i link della rete.
Le metriche utilizzate per quantificare le prestazioni dei diversi algoritmi sono il *True Positive Rate* (TPR) e il *False Positive Rate* (FPR). Le categorie di applicazioni esaminate sono HTTP, SMTP, POP3, FTP, DNS, Telnet.
La tabella 4 riporta il tasso di veri positivi (TPR) dei quattro algoritmi esaminati, addestrati con il *campione A-addestramento* e testati con il *campione A-test*, utilizzando i gruppi di attributi 1 e 4 (Tabella 1). Nella tabella 5 si riporta il tasso di falsi positivi, nella stessa situazione sperimentale. Si nota che non esiste un algoritmo migliore degli altri in tutti i casi. D'altra parte, l'algoritmo basato su C.45 è in molti casi il migliore, con poche eccezioni. Si riscontra inoltre che la precisione degli algoritmi esaminati è abbastanza buona (alto tasso di veri positivi e basso tasso di falsi positivi).
Una quantificazione della portabilità dell'algo-

Protocollo	Bayesian Networks [16]	Clustering [17]	Tecniche Bayesiane avanzate [19]	C.45 [18]
HTTP	89.2%	96.2%	91.8%	99.7%
SMTP	97.2%	90.1%	94.5%	98.6%
POP3	97.2%	93.4%	94.6%	-
FTP	97.9%	92.4%	-	94.8%

TABELLA 4

Tasso di veri positivi (TPR) nel caso di algoritmi addestrati con il ***campione A-addestramento*** *e testati con il* ***campione A-test****; sono stati utilizzati gli attributi dei gruppi 1 e 4 (Tabella 1)*

Protocollo	Bayesian Networks [16]	Clustering [17]	Tecniche Bayesiane avanzate [19]	C.45 [18]
HTTP	10.4%	1.3%	6.4%	0.1%
SMTP	2.3%	0.1%	3.1%	1.4%
POP3	2.3%	0.7%	3.1%	-
FTP	1.8%	0.4%	-	0.5%

TABELLA 5

*Tasso di falsi (FPR) positivi nel caso di algoritmi addestrati con il **campione A-addestramento** e testati con il **campione A-test**; sono stati utilizzati gli attributi dei gruppi 1 e 4 (Tabella 1)*

	Campione A-test		*Campione B-test*	
	TPR	FPR	TPR	FPR
DNS	100%	0.4%	99%	0.1%
FTP	95%	0.5%	77%	1.4%
Telnet	92%	0.1%	84%	0.4%
SMTP	98%	1.4%	95%	3%
HTTP	99%	0.1%	99%	0.2%

TABELLA 6

*Per l'algoritmo basato su C.45, tasso di veri e falsi positivi con addestramento con il **campione A-addestramento** e test con il **campione A-test** e il **campione B-test**; sono stati utilizzati gli attributi dei gruppi 1 e 4 (Tabella 1)*

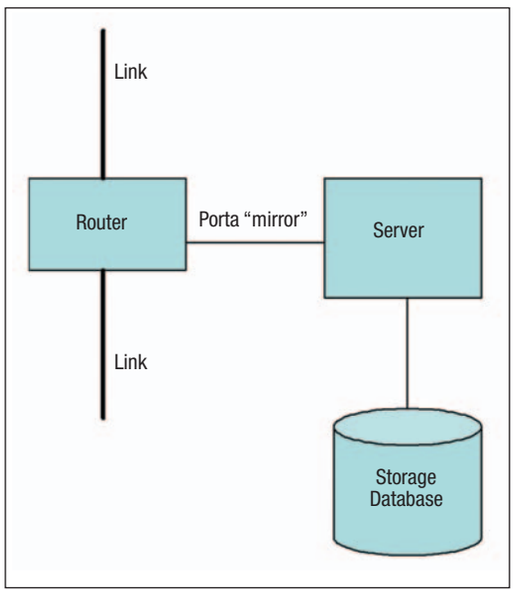


FIGURA 7

Allestimento per la misurazione dei pacchetti e dei flussi di traffico

ritmo basato su C.45 è riportata nella tabella 6, dove si considerano le applicazioni DNS, FTP, Telnet, SMTP e HTTP e il test è effettuato sia con il *campione A-test* sia con il *campione B-test*. Le buone prestazioni dell'algoritmo che agisce sullo stesso link di rete sul quale è stato addestrato non si riscontrano in ugual misura quando il classificatore è messo in esercizio su di un link diverso. In questo caso, le prestazioni non sono pessime (a parte il tasso di veri positivi per l'applicazione FTP). In ogni caso, sono necessari ulteriori miglioramenti per poter ottenere un algoritmo veramente portabile.

3.6. Misurazione e classificazione del traffico in pratica

Sia che si desideri misurare le caratteristiche aggregate dei flussi di traffico, sia che si voglia operare un'identificazione delle applicazioni, è necessario prelevare il traffico dalla rete per poi analizzarlo. Nella figura 7 è mostrato un tipico allestimento per la raccolta e l'analisi dei dati di traffico. Per misurare il traffico su di un link, si deve dotare il *router* che lo gestisce di una *mirror port* la quale replica il traffico presente sul link selezionato. La *mirror port* è collegata direttamente ad un server che memorizza il traffico su un database. Un tale allestimento presenta diverse criticità tecniche e ha costi non trascurabili. Infatti, se il link da esaminare ha grande capacità e i flussi pre-

senti sono numerosi, il numero di pacchetti da memorizzare è enorme (può essere dell'ordine delle centinaia di migliaia di pacchetti al secondo per un link a più di un Gbit/s) e questo richiede ingenti capacità di memorizzazione (per esempio circa 1 Tbyte/h di registrazione su un link da 1 Gbit/s). Inoltre, la velocità di accesso del database deve essere estremamente elevata, in quanto si deve poter lavorare alla velocità di linea del link. Similmente, il server deve essere adeguatamente attrezzato per lavorare a velocità così elevate. Esistono hardware specializzati disponibili sul mercato nella forma di schede da installare nei server, in grado di collegarsi alla *mirror port* e di lavorare alle velocità richieste, sollevando così la CPU del server dal compito di lavorare a velocità estremamente elevate. Naturalmente, il costo di questo tipo di schede è alto.
Per la misurazione dei flussi lo strumento più adottato è NetFlow della Cisco, applicazione inclusa nel sistema operativo IOS dei router Cisco. Tramite NetFlow è possibile registrare e analizzare le caratteristiche dei flussi di traffico che attraversano un router sotto osservazione, seguendo per esempio lo schema applicativo mostrato nella figura 7.
Per la classificazione delle applicazioni uno strumento largamente utilizzato è NBAR (*Network Based Application Recognition*), sempre della Cisco, in grado di integrarsi con NetFlow. La classificazione delle applicazioni da parte di NBAR è eseguita essenzialmente sull'osservazione dei numeri di porta e sulla packet inspection. Esistono anche strumenti forniti da altre case, come il NetFlow Analyzer [20], che sono in grado di acquisire le misurazioni di traffico effettuate da NetFlow e di elaborare analisi dei dati, offrendo viste personalizzabili (tramite interfaccia web) sulla tipologia e caratteristiche del traffico in rete. Per esempio, si possono ottenere rapporti tabellari e grafici sulle applicazioni utilizzati, gli utenti e più in generale sul traffico presente nei diversi punti della rete (velocità, volume, numero di pacchetti, utilizzazione dei link), il tutto anche in tempo reale.
Per quanto riguarda l'applicazione pratica delle tecniche di *machine learning* per la classificazione del traffico si è ancora in uno stadio iniziale (la ricerca in questo settore è ancora molto intensa). Alcuni operatori di telecomunicazioni stanno effettuando sperimentazioni, in alcuni casi, in collaborazione con gruppi di ricerca universitari.

4. CONCLUSIONI

L'analisi del traffico trasportato dai flussi e l'identificazione delle applicazioni che generano i flussi di traffico sono due attività di importanza strategica che permettono all'ISP di gestire al meglio le risorse della sua rete e la qualità del servizio di trasporto del traffico generato dalle applicazioni in rete. L'analisi del traffico fornisce utili informazioni su quali sono i flussi critici (che trasportano un volume di traffico maggiore e che presentano velocità di trasmissione molto elevate) che vanno gestiti con particolare attenzione. L'identificazione del traffico permette all'ISP di conoscere le applicazioni che generano i flussi di traffico e dà quindi indicazioni su quale sia il miglior modo di trattare i flussi di traffico cercando di fornire i livelli di qualità del servizio richiesti dalle applicazioni e, quindi, dagli utenti.
Entrambe le attività sono complesse e richiedono un costoso dispiegamento di mezzi hardware/software. Inoltre, la difficoltà dell'identificazione del traffico comporta l'utilizzo di algoritmi complessi ed è impossibile ottenere un'identificazione sicura al 100% dell'applicazione che ha generato un dato flusso di traffico. Si è visto come le più avanzate tecniche di riconoscimento delle applicazioni implementano metodologie di *machine learning*, nelle quali l'algoritmo di identificazione è in primo luogo addestrato con sequenze di traffico preidentificate, tramite le quali l'algoritmo costruisce autonomamente una serie di regole di decisione. Le migliori tecniche di machine learning portano a percentuali di errore in generale contenute (qualche punto percentuale, o frazioni di percento per alcune applicazioni facilmente riconoscibili). D'altra parte, se si mette in esercizio un identificatore di traffico su un link diverso dal quale si è prelevata la sequenza di traffico utilizzata per il suo addestramento, si nota che la precisione dell'identificazione tende a degradare significativamente. La concezione di un algoritmo di *machine learning* "portatile" che possa funzionare correttamente anche su link sui quali non è stato addestrato è attualmente oggetto di ricerca.

Bibliografia

[1] Callado A., Kamienski C., Szabó G., Geró B.P., Kelner J., Member S.F., Sadok D.: A Survey on Internet Traffic Identification. *IEEE Communications Surveys & Tutorials*, Vol. 11, n. 3, Third Quarter 2009, p. 37-52.

[2] DecinaM., Giacomazzi P.: Il futuro del protocollo IP. *Mondo Digitale*, n.2, giugno 2007, p. 17-29.

[3] http://mediablog.corriere.it/2010/04/-da_corte_usa_colpo_alla_neutra.html

[4] Brownlee N., Claffy K.C.: Understanding Internet traffic streams: dragonflies and tortoises. *IEEE Communications Magazine*, Vol. 40, n. 10, October 2002, p. 110-117.

[5] Wallerich J., Dreger H., Feldmann A., Krishnamurthy B., Willinger W.: A methodology for studying persistency aspects of internet flows. *SIGCOMM Comput. Commun. Rev.*, Vol. 35, n. 2, April 2005, p. 23-36.

[6] http://www.iana.org/assignments/port-numbers

[7] Karagiannis T., Broido A., Brownlee N., Claffy K.C., Faloutsos M.: *Is P2P dying or just hiding*? IEEE Global Telecommunications Conference, November 2004.

[8] http://www.snort.org/

[9] Paxson V.: Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Trans. Networking*, Vol. 2, n. 4, 1994, p. 316–336.

[10] Dewes C., Wichmann A., Feldmann A.: *An analysis of Internet chat systems*. In: ACM/SIGCOMM Internet Measurement Conference, 2003, Miami, Florida, USA, October 2003.

[11] Claffy K.: *Internet traffic characterisation*. PhD Thesis, University of California, San Diego, 1994.

[12] Lang T., Armitage G., Branch P., Choo H.-Y.: *A synthetic traffic model for Half-life*. In: Proc. Australian Telecommunications Networks and Applications Conference, ATNAC2003, Melbourne, Australia, December 2003.

[13] Lang T., Branch P., Armitage G.: *A synthetic traffic model for Quake 3*. In: Proc. ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE2004), Singapore, June 2004.

[14] Bernaille L., Teixeira R., Akodkenou I., Soule A., Salamatian K.: Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, Vol. 36, n. 2, 2006, p. 23–26.

[15] Kohavi R., Quinlan J.R., (Will Klosgen, Jan M. Zytkow, editors): *Decision-tree discovery*. Handbook of Data Mining and Knowledge Discovery. Oxford University Press, 2002, p. 267-276.

[16] Auld T., Moore A.W., Gull S.F.: Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, Vol. 18, n. 1, Jan. 2007, p. 223–239.

[17] Bernaille L., Teixeira R., Salamatian K.: *Early application identification*. In: The 2-nd ADETTI/ISCTE CoNEXT Conference, Dec. 2006.

[18] Verticale G., Giacomazzi P.: *Performance Evaluation of a Machine Learning Algorithm for Early Application Identification*. Proc. of the International Multiconference on Computer Science and Information Technology, 2008, p. 845-849.

[19] Crotti M., Dusi M., Gringoli F., Salgarelli L.: Traffic classification through simple statistical fingerprinting. *SIGCOMM Comput. Commun. Rev.*, Vol. 37, n. 1, 2007, p. 5-16.

[20] http://www.manageengine.com/products/-netflow/netflow-traffic-analysis.html

PAOLO GIACOMAZZI si è laureato in Ingegneria Elettronica presso il Politecnico di Milano nel 1990 ed ha conseguito il Master in tecnologia dell'informazione al CEFRIEL. Dal 1992 al 1998 è stato ricercatore con il Politecnico di Milano dove ora è professore associato di telecomunicazioni. L'attività didattica e la ricerca riguardano la qualità del servizio nella rete Internet multimediale, le reti radiomobili B3G e la sicurezza nelle reti di telecomunicazioni. È editor del *IEEE Network Magazine ed è editor della Book Reviewing Feature del IEEE Network Magazine*.
E-mail: giacomaz@elet.polimi.it