IL FUTURO DEL PROTOCOLLO INTERNET

Maurizio Dècina
Paolo Giacomazzi

Nonostante lo straordinario successo dell'IP (*Internet Protocol*) versione 4, è necessario far evolvere questo protocollo per rendere scalabili le comunicazioni con utenti mobili. Strategie rivoluzionarie che concepiscono in vitro i futuri protocolli presentano serie criticità. La rete ha infatti dimostrato che i suoi cambiamenti avvengono naturalmente, secondo le esigenze del mercato reale. Strategie evoluzionarie più aderenti ai trend del mercato possono avere un maggior successo e dovrebbero tenere conto della grande diffusione delle reti overlay come la telefonia Skype e lo *streaming* video su IP.

1. STRATEGIE EVOLUZIONARIE E RIVOLUZIONARIE: GLI ERRORI DEL PASSATO

Sono ormai passati oltre trent'anni dalla nascita del protocollo IPv4 e può essere sorprendente constatare che il denominatore comune più basilare di questo protocollo, il trasporto del traffico in modalità *best effort*, è ancora oggi uno dei suoi punti di forza più importanti ed è ancora il servizio di trasporto fondamentale di internet. Questo nonostante i ripetuti tentativi di miglioramento che si sono susseguiti negli anni ed hanno portato alla standardizzazione di IPv6, cioè, quella che dovrebbe essere la successiva versione del protocollo IP. Dopo dodici anni dal rilascio del primo documento di standard, da parte della *Internet Engineering Task Force* (IETF), il protocollo IPv6 ha una diffusione molto limitata, ciò fa intuire che non sia affatto semplice modificare il protocollo IP per farlo evolvere. Un'indicazione intuitiva della mole di lavoro svolta nella standardizzazione di IPv6 è riportata nella figura 1, che mostra il numero di standard per anno ed il numero complessivo di standard IPv6 emanati dalla IETF. Il numero totale di standard è ragguardevole, più di 160 a tutt'oggi; un tale numero di standard già approvati è indice di una tecnologia ormai in via di maturazione. È inoltre interessante notare che l'attività di

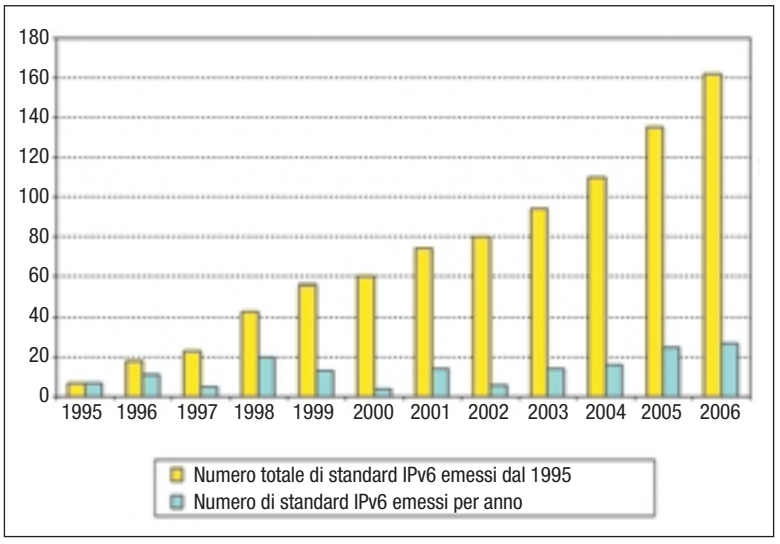

FIGURA 1
Numero di standard di IPv6 emanati dalla IETF dal 1995 al 2006

Glossario

Best-effort: la modalità di trasporto dell'informazione classica di Internet e di IPv4. Nella modalità *best-effort*, la trasmissione di un pacchetto dati avviene senza garanzie. In particolare, il pacchetto può essere perduto, i pacchetti successivi di una comunicazione possono giungere alla destinazione in modo disordinato, cioè, con una sequenza diversa da quella di immissione in rete alla sorgente. Nella modalità *best-effort* i ritardi di trasferimento dei pacchetti non sono controllati e, infine, un pacchetto può giungere duplicato alla destinazione. Può sembrare che una modalità di comunicazione con queste caratteristiche sia di bassa qualità. D'altra parte, la rete internet deve il suo successo alla comunicazione *best-effort* che a tutt'oggi è ancora la modalità di trasferimento più utilizzata.

IPv4: protocollo IP, versione 4, utilizzato in internet per il trasporto uniforme di tutte le tipologie di traffico come dati, voce e video. Il protocollo IPv4 è ormai molto vecchio, dato che la sua prima standardizzazione risale al 1980. D'altra parte, IPv4 è tutt'ora il protocollo di rete utilizzato in internet e la sua radicale semplicità ne ha finora spinto l'evoluzione rapida.

IPv6: il protocollo IP di generazione successiva a IPv4. Nato dall'esigenza iniziale di espandere lo spazio degli indirizzi di IPv4, IPv6 implementa una serie di miglioramenti ulteriori. Le difficoltà incontrate da IPv6 nella penetrazione nel mercato reale dipendono in parte dalla difficile coesistenza di IPv6 con IPv4, che richiede funzioni di traslazione piuttosto complesse.

Multicast: la comunicazione *multicast* o punto-multipunto vede coinvolte una sorgente d'informazione e molteplici destinazioni, mentre la più nota e diffusa comunicazione punto-punto coinvolge una singola sorgente e una singola destinazione. La comunicazione *multicast* è particolarmente adatta alla distribuzione di contenuti multimediali audio/video (per esempio film, eventi sportivi) dove un singolo fornitore di contenuti intende raggiungere un'ampia utenza, con il medesimo contenuto. In questi casi, la comunicazione *multicast* può ridurre significativamente i costi di distribuzione del contenuto rispetto all'utilizzo di una comunicazione punto-punto dalla sergente del contenuto ad ogni destinazione.

Overlay: un sistema *overlay* è costituito da componenti applicative eseguite dai computer degli utenti del sistema stesso. I computer degli utenti diventano i nodi del sistema *overlay* e costituiscono una rete virtuale costruita su internet. Il concetto architetturale di sistema *overlay* è utilizzato dai sistemi *peer-to-peer* per la condivisione di file (per esempio Kazaa, Gnutella), per la telefonia su internet (per esempio Skype) e per la distribuzione di contenuti multimediali su internet in modalità streaming (per esempio CoolStreaming).

Streaming: modalità di trasporto di contenuti multimediali in tempo reale, con fruizione del contenuto contestuale alla distribuzione. Nella fase iniziale del trasferimento streaming, una prima parte del contenuto viene accumulata in un *buffer* nel client dell'utente (il *playout buffer*). Quando il livello di riempimento del *buffer* raggiunge una predeterminata soglia minima, inizia la riproduzione del contenuto tramite il client dell'utente. Il *buffer* viene letto (svuotato) dal client alla velocità di riproduzione specifica del contenuto e contemporaneamente viene riempito con le successive parti di contenuto che sono nel frattempo trasferite via internet. Il *playout buffer* disaccoppia la rete dal sistema di riproduzione del client, infatti, anche se la rete rallenta temporaneamente e la velocità di svuotamento del *buffer* è più elevata di quella di riempimento, il buffer continua a fornire contenuto al sistema di riproduzione del client. Ovviamente, se il periodo di rallentamento della rete ha una durata eccessiva, il *buffer* prima o poi si svuota e si ha un "buco" nella riproduzione del contenuto. Uno dei compromessi più critici del trasporto in modalità streaming è la soglia di riempimento del *playout buffer*. Se nel *buffer* è grande la frequenza dei "buchi" nella riproduzione del contenuto è piccola, ma ciò è pagato con un ritardo sistematico più elevato. Il compromesso dipende dalla tipologia di contenuto: per la visione di un film si può essere disposti a sopportare un ritardo iniziale consistente, ma per eventi sportivi in tempo reale, per esempio una partita di calcio, il ritardo ammissibile dall'utenza è molto più basso.

standardizzazione è tutt'ora in accelerazione, anzi, lo scorso anno, il 2006, è in assoluto l'anno in cui sono stati emanati più standard di IPv6. L'indicazione che ricaviamo è che IPv6 è una tecnologia di interesse crescente. Il fatto che la diffusione di IPv6 sia molto inferiore, rispetto a quanto ci si potrebbe aspettare dallo stato avanzato della sua standardizzazione, porta a domandarci quali siano i motivi che possono aver generato, e che attualmente mantengono, questo differenziale tra aspettative e realtà. Questi motivi possono concretizzarsi in mere difficoltà di *rollout*, in difficoltà tecnologiche più ampie e profonde, in soluzioni di fatto che si sono ormai cristallizzate e che sono difficilmente gestibili tramite IPv6 o, addirittura, in una mancata corrispondenza tra le aspettative di chi fruisce dei servizi offerti dalla rete internet e le nuove potenzialità rese disponibili da IPv6. L'analisi che verrà sviluppata in questo articolo ci porterà a concludere che tutte queste difficoltà esistono e che non si può realisticamente pensare di affrontare tutte le problematiche tecnologiche, architetturali e di servizio con un unico protocollo, per quanto complesso esso possa essere. Per iniziare l'analisi, è utile descrivere in breve un precedente tentativo di realizzare qualcosa di simile alla "Internet del futuro": l'ATM (*Asyncronous Transfer Mode*), una tecnologia emersa nella metà degli anni '80 e che per qualche anno, fino all'inizio degli anni '90, è stata considerata come la vera tecnologia di comunicazione del futuro. Ci si sbagliava.

ATM: *Asynchronous Transfer Mode* o "*Another Terribile Mistake?*" Questo è ciò che moltissimi professionisti addetti ai lavori, ricercatori universitari e aziendali e consulenti si sono chiesti nei primi anni '90, quando dopo quasi un decennio di sforzi di ricerca e sviluppo su ATM si produsse una tecnologia di comunicazione di qualità eccezionale ma di diffusione molto limitata. Dopo vent'anni, ATM esiste ancora come tecnologia di nicchia, essenzialmente utilizzata dai *carrier* per virtualizzare il trasporto sulla rete fisica SDH. Una fine misera per uno sforzo mondiale che ha visto per molti anni coese le industrie e le università di tutto il mondo per progettare la tecnologia di comunicazione "definitiva". Senza entrare nei dettagli, il sostanziale falli-

mento di ATM è stato l'approccio puramente tecnologico che si è adottato nella sua concezione e nel successivo progetto e sviluppo. Grazie a molto impegno e lavoro, sono stati individuati i requisiti della "migliore" tecnologia di comunicazione. ATM è stato realmente creato e prodotto e gli scopi tecnologici sono stati raggiunti, ma nel frattempo il mondo è andato avanti e IPv4, nella sua semplicità, ha conquistato il mercato, di conseguenza ATM è stato relegato in mercati di nicchia. È interessante ed istruttivo constatare che il mercato della comunicazione è stato conquistato da una tecnologia radicalmente semplice, IP.
L'analisi delle cause del sostanziale insuccesso di ATM è propedeutica alla comprensione di come le correnti esigenze dell'utenza della rete internet possano essere efficacemente soddisfatte da un nuovo insieme di tecnologie e di quali siano gli errori più pericolosi da evitare. ATM è stato concepito con poca considerazione rispetto a ciò che stava accadendo nel mondo IP. In particolare, le applicazioni distribuite, già ai tempi di ATM, comunicavano tramite TCP/IP e la rapida ascesa del mercato di tali applicazioni ha portato ad accrescere l'importanza strategica di TCP/IP. Il circolo virtuoso che si è creato tra l'ampia ed eterogenea potenzialità di servizio delle applicazioni distribuite e la semplicità di comunicazione ad esse offerta dal TCP/IP ha permesso la rapida penetrazione di tale protocollo nel mercato dell'ICT. Nel frattempo, ATM fu teorizzato e sviluppato nei laboratori. Pronto per l'entrata nel mercato, ATM si è trovato di fronte ad applicazioni progettate e sviluppate per essere trasportate tramite TCP/IP e un protocollo di rete (IP) già scelto dal mercato reale come il vero "collante" delle reti. Le prospettive di ATM, a quel punto, erano già chiuse prima di aprirsi. Certo, ATM offriva una qualità del trasporto in rete incommensurabilmente superiore ad IP, ma il legacy che ormai si era creato aveva (ed oggi ha ancor di più) un'inerzia tale che la "mera" elevatissima qualità tecnologica offerta da ATM non è stata sufficiente a superare.
È molto importante fare tesoro delle lezioni del recente passato: la definizione dogmatica, tramite teorizzazione aprioristica e standardizzazione di un nuovo protocollo si scontra con l'imponente forza di inerzia del legacy e delle nuove fasi tecnologiche e di servizio che si susseguono a ritmo crescente. Una delle più importanti lezioni che possiamo imparare dal passato è che la migrazione verso l'"*Internet del Futuro*" dovrebbe in primo luogo individuare le vere esigenze di cambiamento e miglioramento, sia dal punto di vista puramente tecnologico sia per quanto riguarda i servizi e le necessità del mercato. Queste esigenze vanno soddisfatte con miglioramenti tecnologici o, se necessario, con vere e proprie "rivoluzioni", ma in armonia con gli sviluppi che il mercato, indipendentemente dagli enti di standardizzazione e dai costruttori di apparati di rete, propone e immette nella realtà di internet. Uno dei più importanti di questi fenomeni è il *networking overlay*.

2. I SISTEMI *OVERLAY* E IL *PEER-TO-PEER*

I sistemi *overlay* sono noti universalmente come "sistemi *peer-to-peer*", anche se non sono esattamente sinonimi: i sistemi peer-to-peer sono un caso particolare di sistema *overlay*. Attualmente, la grande maggioranza dei sistemi overlay utilizzati in internet è di tipo *peer-to-peer*, per cui si utilizzeranno *overlay* e *peer-to-peer* come sinonimi. I sistemi *peer-to-peer* sono nati come comunità per la condivisione di file (Gnutella, Bit Torrent, Kazaa ecc.): l'utente di un sistema *peer-to-peer*, attraverso la condivisione di file, può scaricare da altri utenti, in quel momento in linea, i file che gli interessano (musica, video ecc.). L'evoluzione successiva dei sistemi peer-to-peer ha rivoluzionato il mercato della telefonia su internet. Per esempio, Skype è una vera e propria rete telefonica *overlay* che permette a due persone che stanno utilizzando computer in rete di telefonarsi tramite internet, senza utilizzare la rete telefonica fissa o radiomobile. La terza ondata dei sistemi *peer-to-peer*, tuttora in fase di rapida ascesa, riguarda lo *streaming* audio e video. Attraverso questi sistemi (Coolstreaming, PPlive, QQlive, Octoshape ecc.) è possibile fruire, tramite la propria connessione ad internet, di contenuti audio/video, addirittura in tempo reale (per esempio eventi sportivi) o quasi in tempo reale.
I sistemi *overlay* costituiscono un fenomeno di

importanza strategica primaria. Basta considerare, infatti, che per un tipico carrier europeo, il traffico trasportato relativo ad applicazioni *peer-to-peer*, supera ormai il 50% del totale. Il *peer-to-peer*, quindi, costituisce la maggiore aliquota, tuttora in crescita, del traffico trasportato nella rete internet. È altrettanto interessante notare che il traffico relativo alla telefonia trasportata su IP e ai dati business, che costituisce un'aliquota sensibilmente inferiore al 50% del totale traffico di un carrier, produce fino al 60-70% del fatturato complessivo. I servizi *peer-to-peer* sono frequentemente fruiti dall'utenza residenziale tramite connessioni ADSL pagate spesso a canone fisso. In tal modo, si è creato un modello di business nel quale la telefonia da computer a computer è tassata a canone e non a tempo, in contrasto con il classico modello di business telefonico. I carrier che forniscono sia servizi telefonici sia connessione ADSL a internet vedono erodersi progressivamente il fatturato relativo alla telefonia, dovuto alla parziale migrazione di una parte delle telefonate su Skype.

I sistemi *peer-to-peer*, che costituiscono un'innovazione tecnologica totalmente indipendente dal corrente sforzo di standardizzazione di IPv6, hanno già prodotto uno spostamento pari circa al 10% del business dei carrier. Lo *streaming* multimediale, già fruibile in modalità *peer-to-peer*, promette ulteriori sensibili mutamenti nel business dei distributori di contenuti. Le prime avvisaglie di questo fenomeno, che si prospetta dirompente almeno tanto quanto quello della telefonia *peer-to-peer*, sono già state riscontrate, per esempio, nelle cause legali intentate da alcuni distributori di contenuti nei confronti di chi abilita o facilita la fruizione in modalità *peer-to-peer* di contenuti protetti dai diritti d'autore o di distribuzione.

Nell'identificare le migliori linee di sviluppo della internet del futuro, il *peer-to-peer* e, più in generale, il *networking overlay*, è un fenomeno da tenere in grande considerazione. Qualsiasi nuova concezione tecnologica e qualsiasi relativo sviluppo avulsi da questa realtà sono probabilmente destinati a rimanere esercizi teorici, di scarso o nullo interesse per le applicazioni pratiche e quindi superati dalla realtà ancor prima di essere inseriti nel mercato. Seguendo questa filosofia, si cercherà ora di individuare le problematiche tecnologiche e di servizio più importanti e significative dei sistemi *peer-to-peer*, in modo tale da identificare i requisiti ai quali dovrebbe aderire uno sviluppo di internet che assecondi il fenomeno del *peer-to-peer*.

2.1. La telefonia peer-to-peer: Skype

Il concetto di base di sistema *overlay* è illustrato nella figura 2, con particolare riferimento al sistema Skype per la realizzazione della telefonia su internet in modalità *peer-to-peer*. La rete internet "fisica", costituita dai link di comunicazione, dai router, dagli switch, cioè, da tutti gli apparati che concorrono alla creazione della comunicazione *end-to-end*, è la rete *underlay*. Alla rete underlay partecipano anche i computer degli utenti della rete internet. Il sistema *overlay* è costituito da componenti applicative che risiedono principalmente sui computer degli utenti della rete internet (chiameremo queste componenti applicative gli *overlay client*). Nel sistema *overlay*, gli *overlay client* assumono il ruolo di nodo della rete *overlay*. In pratica, si può pensare all'*overlay client* come ad un router del sistema overlay. Gli *overlay client* costituiscono una rete logica, a livello applicativo, sulla rete *underlay*. La figura 2, quindi, mostra due reti sovrapposte, ma solo la rete *underlay* è costituita da veri apparati, la rete *overlay* è virtuale. Questa rete virtuale, nel caso di Skype, ha a sua volta due livelli gerarchici e ha strette analogie con la tradizionale rete telefonica. I nodi del livello gerarchico superiore, detti *supernodi*, hanno forti analogie con le classiche centrali telefoniche di transito e, in effetti, la rete logica di Skype ha qualche analogia con la classica rete telefonica a due livelli degli operatori. Ogni elemento della rete *overlay* ha un corrispondente elemento nella rete *underlay*, nella maggior parte dei casi questo elemento è il computer che implementa in software le funzionalità del nodo al livello *overlay*. Nella figura 2, è mostrata con una linea tratteggiata la corrispondenza tra alcuni nodi e supernodi di Skype con i computer che, al livello *underlay*, li istanziano.

In Skype, e in tutti i sistemi *overlay*, la rete *overlay* è creata sfruttando le risorse di computazione, memoria e disco dei client, nonché la banda trasmissiva della connessione di rete

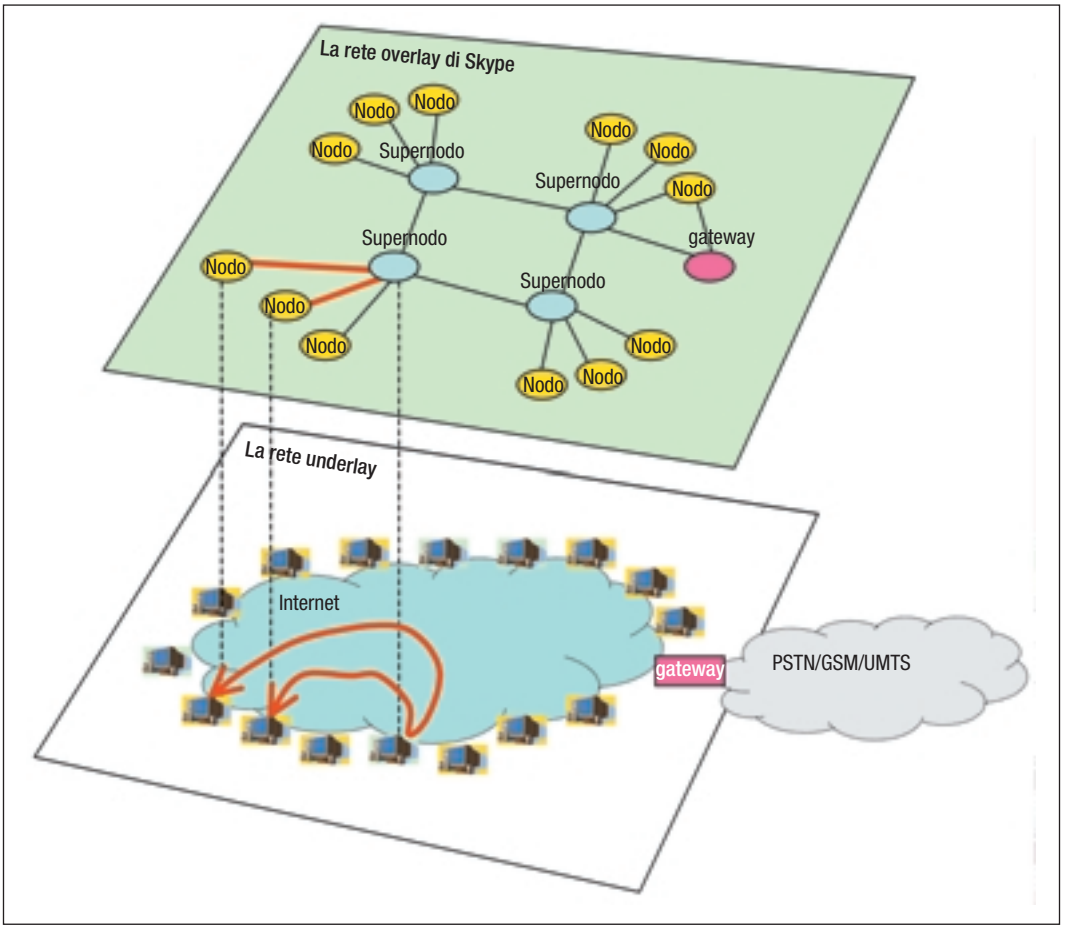

FIGURA 2
I sistemi overlay: la rete overlay di Skype

degli stessi client nei due sensi di trasmissione verso la rete internet. Si osservi che la banda di *upload*, da computer a rete, è ugualmente importante rispetto a quella di *download* e che i supernodi offrono alla rete *peer-to-peer* grandi risorse di calcolo e di banda. Il software applicativo organizza queste risorse per costituire il sistema complessivo per l'erogazione del servizio. Client con una CPU potente, ampia memoria, alta affidabilità e una buona connessione a internet sono riconosciuti dal sistema e, prima o poi, diventano supernodi (in alcuni sistemi, anche a insaputa del proprietario del computer) assumendo un ruolo preminente nella rete *overlay*, per esempio, instradando chiamate telefoniche di terzi.

2.2. La peer-to-peer streaming TV e il multicasting overlay

Gli utenti dei sistemi *peer-to-peer* streaming TV fruiscono di contenuti televisivi, anche in tempo reale, tramite il loro computer connesso a internet e, come fattore distintivo, questi sistemi hanno un numero elevatissimo di utenti che possono raggiungere, su scala planetaria, sfruttando pienamente le potenzialità del *multicasting*, cosa che, con il protocollo IP utilizzato in modo naturale (*underlay*), non è praticamente possibile. Come già evidenziato, i computer degli utenti del sistema *overlay* costituiscono i nodi della rete di distribuzione *overlay*. Questa rete logica, nel più semplice dei casi, è un vero e proprio albero di distribuzione. Il contenuto televisivo è immesso in rete dal nodo "radice" dell'albero, che lo distribuisce ad un numero limitato di utenti. Ogni utente ripropaga il contenuto verso altri utenti realizzando a tutti gli effetti una progressione geometrica del numero di utenti che fruiscono dello stesso contenuto. In pochi passaggi, il numero di utenti che fruisce del contenuto può diventare enorme.

Le risorse per realizzare questa distribuzione sono la capacità di calcolo e la memoria dei computer connessi al sistema e la banda del collegamento (per esempio, banda asimme-

trica nei sistemi ADSL) ad internet dei computer. Con computer e con collegamenti ad internet di fascia media, è possibile creare un sistema di distribuzione su scala planetaria. Le risorse del sistema, infatti, aumentano all'aumentare della dimensione dell'utenza, creando una notevole scalabilità naturale del sistema stesso che vede aumentare le risorse complessive di computazione e di trasmissione di pari passo all'aumentare della domanda di fruizione. Dato che ogni utente che si inserisce nel sistema mette a disposizione ulteriori risorse di computazione e trasmissione, la dimensione complessiva dell'utenza del sistema di *peer-to-peer* streaming TV può crescere senza alcun evidente limite teorico.
L'albero di distribuzione multicast è al livello *overlay* ed è quasi ironico ricordare che al livello *underlay* ogni ramo dell'albero di distribuzione è realizzato tramite una comunicazione punto-punto (unicast) e che il livello *underlay* è completamente agnostico rispetto al fatto che al livello overlay la distribuzione è multicast: i sistemi di *peer-to-peer* streaming TV sfruttano la capacità della rete underlay di creare comunicazioni punto-punto per realizzare ciò che la rete underlay non è sostanzialmente capace di fare su scala globale: la comunicazione *multicast*.

2.3. Le problematiche tecniche dei sistemi overlay

A valle delle precedenti discussioni, è ragionevole concludere che la rete internet del futuro non può svilupparsi in modo indipendente dai sistemi *overlay*, anzi, si dovrebbe cercare di facilitare ancor di più l'espansione di servizi così di successo e di ridurne i costi in termini, per esempio, di risorse necessarie per farli funzionare. È soddisfacente constatare che IPv6 implementa in modo naturale uno dei principali fattori di facilitazione tecnologica dei sistemi *peer-to-peer*: indirizzi IP lunghi, di 128 bit, anziché di soli 32 bit degli indirizzi IPv4. Può sembrare strano che la lunghezza degli indirizzi possa costituire un problema per i sistemi *peer-to-peer*, ma mostreremo ora che il problema è serio, tanto da assumere una valenza strategica.
Con riferimento alla figura 2, in un qualunque sistema *peer-to-peer* si arriva al momento nel quale tra due *client* deve avvenire uno scambio di informazioni, anche intenso, per trasportare un file, per effettuare una chiamata telefonica o per trasferire uno stream audio/video. La cosa può sembrare semplice: i due *client* si conoscono reciprocamente e sanno come raggiungersi tramite i rispettivi indirizzi IP, quindi, sembra che lo scambio possa avvenire senza problemi. Purtroppo, spesso questo non è vero, a causa della NAT (*Network Address Translation*), che in molti casi costringe i *client* che devono scambiarsi informazioni a transitare attraverso un supernodo, come mostrato nella figura 2, invece di comunicare direttamente. Questo comporta un notevole consumo di risorse ma, prima di affrontare questo aspetto del problema, analizziamo i motivi per i quali la *Network Address Translation* è necessaria e perché comporta a sua volta la necessità del transito attraverso i supernodi.

2.3.1. Il conflitto tra Network Address Translation e i sistemi overlay

La NAT è necessaria perché gli indirizzi IPv4 sono pochi. Quando fu definito il protocollo IPv4, 32 bit per codificare gli indirizzi di rete sembravano sovrabbondanti (2^{32} = 4.294.967.296), ma ora i quattro miliardi di indirizzi IPv4 disponibili sono ormai esauriti. Il protocollo IP richiede che ogni interfaccia di rete sia contraddistinta da un indirizzo, cioè, ogni dispositivo, computer, stampante, server, ogni interfaccia di ogni router ecc. richiede un indirizzo, diverso da tutti gli altri. Anche con un'approssimativa valutazione di buon senso si conclude che gli indirizzi disponibili sono meno di quelli già in campo. Ciò è reso possibile dalla tecnica della *Network Address Translation* (NAT).
La NAT espande artificialmente, senza veri e propri limiti superiori, lo spazio degli indirizzi IPv4. Esistono alcuni intervalli di indirizzi IPv4 detti "indirizzi privati" o, con un curioso e molto utilizzato neologismo, gli indirizzi "*nattati*", che sono riusabili a piacere all'interno di reti chiuse. Per esempio, un'azienda può numerare le interfacce della sua rete con indirizzi "*nattati*". Le comunicazioni all'interno dell'azienda sono garantite in modo standard. Le comunicazioni verso l'esterno sono invece da gestire opportunamente, perché altre aziende o persone fisiche possono utilizzare gli stessi indirizzi "*nattati*"ed è assolu-

tamente necessario evitare ambiguità. L'evidente collisione degli indirizzi "*nattati*"è evitata vietando di utilizzare indirizzi nattati al di fuori delle reti private. Quando un pacchetto generato da un computer con indirizzo "*nattato*"deve transitare sulla rete internet pubblica, un dispositivo apposito, la *NAT box*, posto all'interfaccia tra la rete pubblica e la rete privata, provvederà a traslare l'indirizzo "*nattato*"in modo che il pacchetto si presenti sulla rete pubblica con un indirizzo valido globalmente. In molte tipologie di NAT box, tutti i pacchetti che transitano attraverso la NAT box verso la rete pubblica presentano come indirizzo IP di sorgente l'indirizzo della NAT box, indipendentemente dall'indirizzo del computer che li ha generati. Questo risultato, apparentemente paradossale, è possibile in quanto la NAT box, mediante complicati meccanismi che non trattiamo qui, riesce comunque a garantire la coerenza end-to-end delle connessioni che la attraversano. Il risultato è notevole: con un solo indirizzo valido globalmente, quello della NAT box, è possibile gestire le comunicazioni di un'intera rete privata di computer. Questo garantisce un effetto di leva talmente elevato sul numero di indirizzi disponibili che, tramite le NAT box, è possibile espandere in modo virtualmente illimitato lo spazio degli indirizzi IP.

Purtroppo, questa espansione virtuale degli indirizzi IPv4 è pagata con diversi tipi di svantaggi e, in questa sezione, ci occupiamo di quelli direttamente connessi alle attività dei sistemi *overlay*, che necessitano di comunicazione tra i client. Se i due client che devono comunicare sono "*nattati*", una comunicazione diretta tra i client è, in molti casi impossibile, dipendentemente dal tipo e dalla configurazione delle NAT box. Infatti, l'indirizzo "*nattato*"dei due client non è noto al di fuori della rispettiva rete privata e, in mancanza di questo, non si riesce a stabilire una connessione. In questo caso, entrambi client devono prima connettersi a un supernodo. Questo è sempre possibile perché, per definizione, i supernodi sono dotati di indirizzi IP pubblici. A questo punto, la connessione tra i client è spezzata in due segmenti (da client a supernodo e da supernodo all'altro client). Questo è il caso mostrato nell'esempio di figura 2: due nodi, che assumiamo "*nattati*", devono comunicare e per farlo utilizzano come relay il supernodo.

Il costo di questa modalità di comunicazione è elevato. Quando la comunicazione tra client "*nattati*"avviene in due tratte nella rete *overlay*, nella rete *underlay* questo corrisponde a due connessioni *end-to-end* distinte che ovviamente aumentano la quantità di risorse necessaria per effettuare il trasferimento di informazione tra i due client. La percentuale di *client* "*nattati*"è già molto elevata ed è in continua crescita, per cui un'aliquota significativa (anche se difficile da quantificare precisamente) del traffico *peer-to-peer* è costituito dai rilanci delle connessioni tra client "*nattati*"attraverso i supernodi. Questo fenomeno assume un'importanza ancora maggiore se si ricorda che il traffico *peer-to-peer* costituisce ormai più del 50% del totale traffico di internet. Si può concludere che in prospettiva una parte del traffico di internet, che potrebbe forse raggiungere l'ordine del 20% (una previsione precisa è difficile), sarà ridondante e servirà solamente per aggirare il problema della scarsità degli indirizzi. Già oggi questa aliquota di traffico "*address-related*" è notevole, tuttavia mancano studi precisi che ne valutino esattamente il volume. In prospettiva, pagare con una frazione importante della capacità di internet il fatto di avere indirizzi corti sembra essere un prezzo abnorme. In tal senso, è consolante constatare che IPv6 ha indirizzi di ben 128 bit e che, una volta terminata la messa in campo di IPv6, il problema sarà rimosso alla radice. L'impatto quantitativo del problema NAT-*overlay* è talmente intenso che da solo sembra essere sufficiente a giustificare una migrazione da IPv4 a IPv6, questo nonostante sia ancora abbastanza diffuso un giudizio negativo sulla reale necessità di ampliare lo spazio degli indirizzi: "*perché farlo? tanto c'e' NAT*".

L'inefficiente cooperazione tra i sistemi *overlay* e NAT ha anche altri effetti negativi. Se si considera, per esempio, la telefonia *peer-to-peer*, dalla precedente discussione si capisce che le telefonate tra client "*nattati*"devono transitare attraverso un supernodo, consumandone le risorse di CPU, memoria e la capacità della connessione a internet. Per quanto riguarda Skype, studi empirici hanno mostrato che mediamente un supernodo è in gra-

do di fare da transito per un massimo di poche diecine di telefonate contemporanee. Questo pone un limite alla scalabilità di Skype e, più in generale, dei sistemi *peer-to-peer*: il sistema può scalare fino a che ci sono supernodi disponibili. Visto che i supernodi sono computer di profilo più elevato della media, con indirizzi pubblici (non *nattati*) e con connessione ad internet con capacità elevata, è ovvio che essi rappresentano la vera risorsa scarsa del sistema ed il più stretto collo di bottiglia per la scalabilità dei sistemi *overlay*. Si può concludere che una delle esigenze primarie dello sviluppo futuro di internet è di avere a disposizione uno spazio di indirizzi più ampio: ciò diminuirebbe la necessità del NAT con molti effetti positivi, non solo sui sistemi *overlay*.

3. SEPARAZIONE DI IDENTITÀ E INDIRIZZI E L'HOST IDENTITY PROTOCOL (HIP)

Un grave problema attuale di internet è la molteplicità dei significati e dei conseguenti utilizzi dell'indirizzo IP degli *host*. In primo luogo, l'indirizzo IP è un'informazione di localizzazione, che permette ai pacchetti IP di raggiungere la loro destinazione. D'altra parte, l'indirizzo IP è anche l'effettiva identità dell'*host*, cioè, per un *host* che comunica con un *peer* remoto, l'identità del *peer* è ancora una volta l'indirizzo IP del *peer* stesso. Le conseguenze di questo sovraccarico semantico dell'indirizzo IP sono di portata addirittura imponente. Infatti, il punto di terminazione logica delle connessioni a livello applicativo, per le applicazioni distribuite, è il *socket*, cioè, l'indirizzo IP dell'*host* e la porta logica che corrisponde al servizio indirizzato sull'*host* stesso(Figura 3 **A**). Una comunicazione tra due applicazioni su internet è, dal punto di vista delle applicazioni, un collegamento logico tra due *socket*, uno per ognuno dei componenti applicativi coinvolti nella comunicazione. Dato il doppio significato, identità e locazione, dell'indirizzo IP, il collegamento logico tra due applicazioni viene ad essere molto rigido: all'atto dell'instaurazione della connessione logica, un'applicazione stabilisce che intende comunicare con uno specifico *host* in una specifica locazione, entrambe descritte dall'indirizzo IP di quell'*host*. È naturale chiedersi cosa succede se uno dei due *host* si muove: cambia solo la locazione oppure cambia sia la locazione che l'identità? Purtroppo, cambiano entrambe. Questo pone problemi nella gestione della mobilità che ancor oggi si possono considerare come essenzialmente irrisolti. Quando un *host* viene collocato in un'altra sede e quindi cambia indirizzo IP, cambia anche identità. Addirittura, quando un dispositivo è in movimento, esso cambia ripetutamente identità col passare del tempo. Pur ricordando che la semplicità del trasferimento punto-punto tra due dispositivi in posizione fissa è uno dei fattori che ha permesso il travolgente sviluppo di internet, si deve constatare che ora la comunicazione, fortemente vincolata ad un concetto di posizione statica e poco mutabile dei dispositivi in rete, sta ponendo seri vincoli tecnologici allo sviluppo delle applicazioni e, quindi, della internet stessa. Molte moderne applicazioni fruirebbero di grandi vantaggi da un modello più astratto della comunicazione, nel quale il dispositivo che invia un'informazione non deve necessariamente conoscere l'indirizzo e/o la locazione dei dispositivi che deve raggiungere, ma solo la loro identità. Il mittente dell'informazione può in tal modo essere agno-

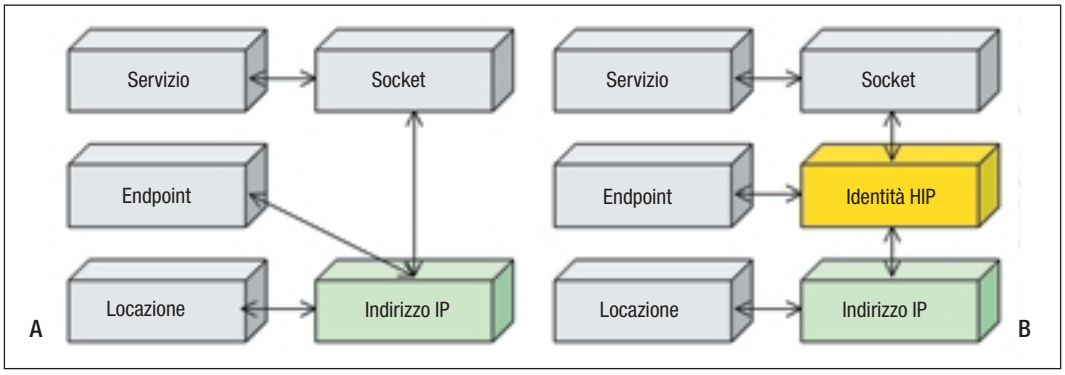

FIGURA 3
A *Il problema della mobilità in IP,* **B** *Identità HIP*

stico rispetto alla effettiva locazione del destinatario e preoccuparsi solo della sua identità. Una tale potente astrazione permetterebbe di gestire in modo naturale la mobilità dei dispositivi, con una conseguente notevole flessibilità e scalabilità delle procedure per la gestione della mobilità. Anche il *multicast* sarebbe applicabile in modo semplice, basti pensare al fatto che la stessa identità può essere assegnata a tutti gli *host* che devono ricevere una data informazione: automaticamente, questa sarebbe instradata verso tutti i destinatari realizzando il *multicast* in modo naturale. In conclusione, la mobilità ed il multicast sollevano problemi di scalabilità praticamente insormontabili con l'attuale suite protocollare di internet. Si pensi per esempio all'enorme differenza tra la rudimentale mobilità dei dispositivi IP *wireless* e la mobilità avanzata di cui si dispone comunicando invece tramite GSM/GPRS o UMTS, che, impiegando una radicale separazione tra nomi e indirizzi, collegano in tempo reale tre miliardi di terminali in piena mobilità. Una penetrazione completa di IP nel segmento d'accesso *wireless* sarebbe auspicabile da molti punti di vista, ma il Mobile IP (la corrente soluzione della internet per la gestione della mobilità) non lo permette su scala ampia e per il *multicasting* la situazione è analoga [1, 2, 3].

Le possibili soluzioni a questo problema di importanza strategica stanno sia al livello *overlay* [4, 5, 6] che al livello *underlay* [7]. Per quanto riguarda il livello *overlay*, le soluzioni finora prospettate e realizzate (si veda per esempio la costruzione *overlay* degli alberi di distribuzione *multicast* dei sistemi di *peer-to-peer* streaming video Coolstreaming o PPlive) sono orientate in modo dedicato all'applicazione specifica che le utilizza. Inoltre i livelli di astrazione dell'identità rispetto all'indirizzamento finora realizzate al livello *overlay* e valide per la distribuzione *multicast* tendono a funzionare male per la gestione della mobilità e viceversa. Sono state teorizzate soluzioni *overlay* generali, in grado di fornire il disaccoppiamento identità-indirizzo in modo indipendente dalla specifica applicazione e dalla funzionalità (*multicast*/mobilità) che si intende realizzare. Certamente, le soluzioni finora proposte sono teoriche, ma indicano chiaramente una possibile via da perseguire. Fra tante proposte, il concetto della i^3 (*Internet Indirection Infrastructure*) merita di essere citata [4], se non altro per l'originalità e l'eleganza, d'altra parte, per ora questa soluzione rimane teorica.

L'unico approccio che finora ha avuto la conferma di una standardizzazione in ambito IETF è l'*Host Identity Protocol* [7] (HIP), ed è una soluzione al livello *underlay* (Figura 3 **B**). I principi e gli obiettivi fondanti di HIP possono essere considerati come una scaletta di alcune delle più critiche ed urgenti problematiche da affrontare per lo sviluppo della rete Internet.

In "*Host Identity Protocol* (HIP) Architecture," di R. Moskowitz e P. Nikander, [7] si rileva che:

- la disponibilità di uno spazio di nomi (che d'ora in avanti useremo come sinonimo di identità) indipendente dagli indirizzi di rete è strettamente necessaria per il futuro sviluppo delle applicazioni *end-to-end*;
- lo spazio dei nomi deve essere *indipendente* dagli indirizzi IP;
- I nomi devono rimpiazzare tutte le occorrenze di indirizzi IP nei payload applicativi dei pacchetti;
  - si nota, infatti, che le necessità pratiche delle applicazioni distribuite hanno portato gli sviluppatori a codificare e trasportare l'indirizzo IP del mittente e/o del destinatario del pacchetto all'interno del payload applicativo del pacchetto stesso. Questo fatto, ovviamente, rende ancora più intricato il problema della traslazione degli indirizzi, in quanto non si deve intervenire solo a livello dell'header del pacchetto, ma anche nel suo payload applicativo. Questo è un intervento molto pesante, che richiederà nel lungo termine un adeguamento delle *Application Programming Interface* delle applicazioni distribuite;
- la lunghezza dei nomi deve essere compatibile con un efficiente trasporto degli stessi nei pacchetti, cioè, non devono essere troppo lunghi per evitare di dover sopportare un overhead eccessivo;
- la probabilità di collisione tra i nomi deve essere bassa, in altre parole, non deve quasi mai accadere che due entità siano denominate nello stesso modo. Purtroppo, questo obiettivo è in netto contrasto con il precedente, in quanto la probabilità di collisio-

ne aumenta al diminuire della lunghezza dei nomi. Il famoso paradosso del compleanno (la probabilità che in un gruppo di sole 23 persone scelte a caso, almeno due persone compiano gli anni nello stesso giorno supera il 50%) fa intuire che il problema possa essere serio, cioè, prese due entità con nomi estratti a caso, la probabilità di avere due entità con lo stesso nome può essere sorprendentemente elevata. Per quanto riguarda il nostro problema specifico, se i nomi sono codificati con 64 bit, la probabilità di avere una collisione in una popolazione di 640 milioni di individui è circa pari all'1% (cioè, molto elevata), è quindi necessario orientarsi su nomi di almeno 160 bit e preferibilmente più lunghi, gli *Host Identity Tag* (HIT). Per contrastare attacchi crittografici prevedibilmente sempre più potenti (vedi gli standard hash correnti) bisogna probabilmente ricorrere a nomi lunghi 512 bit.
Un nome nello spazio dei nomi *Host Identity* (HI) identifica un sistema in rete in modo statisticamente univoco, cioè, univoco a meno di una piccola probabilità di collisione. Un sistema (per esempio un computer o un più generico terminale d'utente) può avere molteplici nomi e nel paradigma di HIP il nome di un sistema (o meglio, uno dei nomi di un sistema) è la chiave pubblica di una coppia di chiavi crittografiche asimmetriche. Il grande vantaggio di questa soluzione è che il nome, contenuto nei pacchetti, può essere utilizzato agevolmente come autenticatore dei pacchetti stessi e costituisce una difesa preventiva e strutturale da molti attacchi che correntemente sono attuati nei confronti delle comunicazioni sia in chiaro sia crittografate.
Il nome di un sistema è, per sua natura, pubblico (la chiave pubblica) e il sistema detiene in segreto la parte privata della chiave. Il possesso della chiave privata che corrisponde esattamente alla chiave pubblica è di per sé un certificato di identità. La stessa chiave privata può essere intenzionalmente distribuita tra molteplici sistemi che in tal caso diventano un gruppo omogeneo. Attualmente si pensa che gli HIT saranno conservati nei server del *Domain Name System* oppure in una apposita infrastruttura dedicata, con caratteristiche simili al DNS.

3.1. Socket, identità e indirizzo IP

Come mostrato nella figura 3 **A**, attualmente l'indirizzo IP di un sistema, individua la locazione dell'interfaccia fisica di rete del sistema stesso, identifica anche il sistema e, inoltre, va a costituire una parte dell'identificativo del punto di terminazione della comunicazione a livello applicativo (il socket). Per disaccoppiare questi molteplici e variegati significati e funzioni, HIP interpone il concetto di identità fra la localizzazione (indirizzo IP), individualità del sistema in rete (endpoint) e punto di terminazione a livello applicativo della comunicazione (socket). Il risultato, mostrato nella figura 3 **B**, è evidentemente più razionale e ordinato. L'identificativo dell'*host* (HIT) concretizza l'identità del sistema in rete, prima affidata all'indirizzo IP. L'indirizzo IP diventa un puro mezzo di localizzazione fisica del sistema e, se il sistema si sposta, varia l'indirizzo IP, ma, giustamente, non la sua identità (cioè, il suo HIT). Dalla figura 3 **A** è facilmente intuibile come nella internet attuale uno spostamento geografico di un *host* possa comportare, paradossalmente, un vero e proprio cambiamento di identità dell'*host* stesso e una variazione del punto di terminazione applicativo della comunicazione, con tutti i conseguenti problemi.
Il disaccoppiamento di identità e locazione di HIP (Figura 3 **B**) permette una gestione della mobilità incommensurabilmente più semplice e scalabile di quella, modesta, offerta attualmente da internet. Infatti, l'*host* in movimento può cambiare locazione (indirizzo) senza influire sulla sua identità e quindi sulla sua rintracciabilità. D'altra parte, l'*host* in movimento deve notificare al suo peer remoto l'avvenuto cambiamento di indirizzo IP (il peer deve essere conscio del cambio di locazione) e il peer remoto, a sua volta, deve accertarsi che l'host in movimento è ancora raggiungibile al nuovo indirizzo.

3.1.1. La comunicazione rendez-vous

Iniziare una comunicazione con un *host* in movimento è più complicato. Il sistema che intende iniziare la comunicazione deve conoscere in anticipo come raggiungere l'*host* che si sta spostando. Nel paradigma di HIP si è identificata una nuova modalità di individuazione del peer con il quale si intende stabilire una comunicazione: la comunicazione *rendez-vous* che, come suggerisce il nome, ha lo scopo di facili-

tare gli "incontri" tra i nodi che intendono comunicare. Il meccanismo *rendez-vous* richiede la presenza di un *Rendez-Vous Server* (RVS), che fornisce un primo punto di contatto noto agli *host* che intendono comunicare. Ovviamente, il *server rendez-vous* può conoscere l'esistenza e l'identità di un *host* solo se l'host stesso si è preventivamente registrato.
Il meccanismo di *rendez-vous* in HIP è attualmente in studio e non è ancora stato standardizzato, per cui è descritto solo in documenti temporanei della IETF, *gli Internet Drafts*. Il meccanismo di *rendez-vous* di HIP è descritto in "*Host Identity Protocol* (HIP) Rendezvous Extension" [8][1].
Un *host* che vuole rendersi rintracciabile si registra presso un server *rendez-vous*, comunicando la propria identità, HIT, e l'indirizzo IP corrente. Il server *rendez-vous* agisce come relay dei pacchetti di segnalazione HIP diretti verso gli *host* registrati. Per rendersi effettivamente rintracciabile, l'*host* deve anche specificare l'indirizzo IP del suo server *rendez-vous* nel suo record DNS [9] (*Domain Name System*), utilizzando un nuovo campo appositamente definito, il record DNS denominato HIPRVS. In aggiunta, l'*host* registra nel proprio record DNS anche l'identità, il suo HIT.
Per esempio, l'*host* con nome DNS *pippo.dominio.it* si registra presso il proprio server *rendez-vous*, *RVS*, e inserisce nel proprio record DNS l'indirizzo IP di RVS, e la sua identità HIT.
A questo punto, un qualunque *host* che intende raggiungere pippo.dominio.it esegue una richiesta DNS, riceve sia l'identità HIT sia l'indirizzo del server *rendez-vous* di pippo.dominio.it e può in tal modo stabilire una comunicazione, sia mutuata dal server *rendez-vous* (inizialmente) ma anche diretta una volta che il server *rendez-vous* ha comunicato all'*host* l'indirizzo IP corrente di pippo.dominio.it.

[1] Il documento citato è un Internet Draft e non uno standard consolidato. Dato che gli Internet Draft sono documenti temporanei, in genere è ritenuto scorretto citarli negli articoli, in quanto prima o poi il documento non sarà più reperibile (per esempio, potrà essere sostituito da uno documento di standard consolidato (RFC)). Si è deciso comunque di citare gli Internet Draft, perché sono gli unici riferimenti ufficialmente disponibili relativi a tematiche molto avanzate e in via di sviluppo, come HIP.

3.1.2. La mobilità in HIP

Allo stato attuale degli studi e della standardizzazione, la mobilità in HIP è gestita in modo elementare. Essenzialmente, quando un *host* in movimento cambia indirizzo IP lo comunica direttamente al suo *peer* remoto che non appena a conoscenza del nuovo indirizzo IP inizierà ad utilizzarlo [10]. Questa gestione della mobilità appare tuttavia di gran lunga meno sofisticata di quella di tipo *carrier-grade*, cioè, con un livello di qualità del servizio per la piena mobilità come quello offerto dagli operatori di telecomunicazioni ai loro clienti su scala mondiale. Gli operatori radiomobili GSM/UMTS offrono attualmente un servizio di piena mobilità evoluto, affidabile, e scalabile su scala planetaria, che coinvolge molte centinaia di operatori e alcuni miliardi di terminali d'utente. La mobilità GSM/UMTS utilizza due tipi di nomi (permanente IMSI, *International Mobile Subscriber Identity*, e temporaneo TIMSI, *Temporary IMSI*) tre tipi di indirizzi (permanente, MSISDN, *Mobile Station ISDN Number*, nel roaming tra operatori, MSRN, *Mobile Station Roaming Number*, e nell'*hand-over* tra celle radio, HON, *Hand Over Number*) e si basa su un sistema di gestione della mobilità che comprende varie tipologie di entità, tra le quali gli HLR (*Home Location Register*) e i VLR (*Visitor Location Register*) che, inseriti nell'architettura di un sistema per la localizzazione e sostenuti da un'affidabile protocollistica di segnalazione, possono tracciare e rendere raggiungibili in tempo reale tutti gli utenti della rete GSM/UMTS.
La semplice mobilità end-to-end di HIP, gestita da uno scambio di messaggi tra i due terminali e senza il supporto di un sistema dedicato alla localizzazione è da considerarsi come un primo tentativo di realizzare un servizio di piena mobilità. Per arrivare a gestire una mobilità di tipo carrier-grade nella internet, sembra essere inevitabile la realizzazione di un sistema per la localizzazione simile, almeno per le caratteristiche funzionali e di qualità, a quello già maturo e consolidato del GSM/UMTS.

3.2. L'evoluzione di HIP rispetto al protocollo IP classico

Il protocollo IP classico fu fondato ipotizzando quattro importanti principi fondanti (Figura 4 **A**):
- *non-mutabilità*: l'indirizzo IP non deve variare nel percorso del pacchetto dalla sorgen-

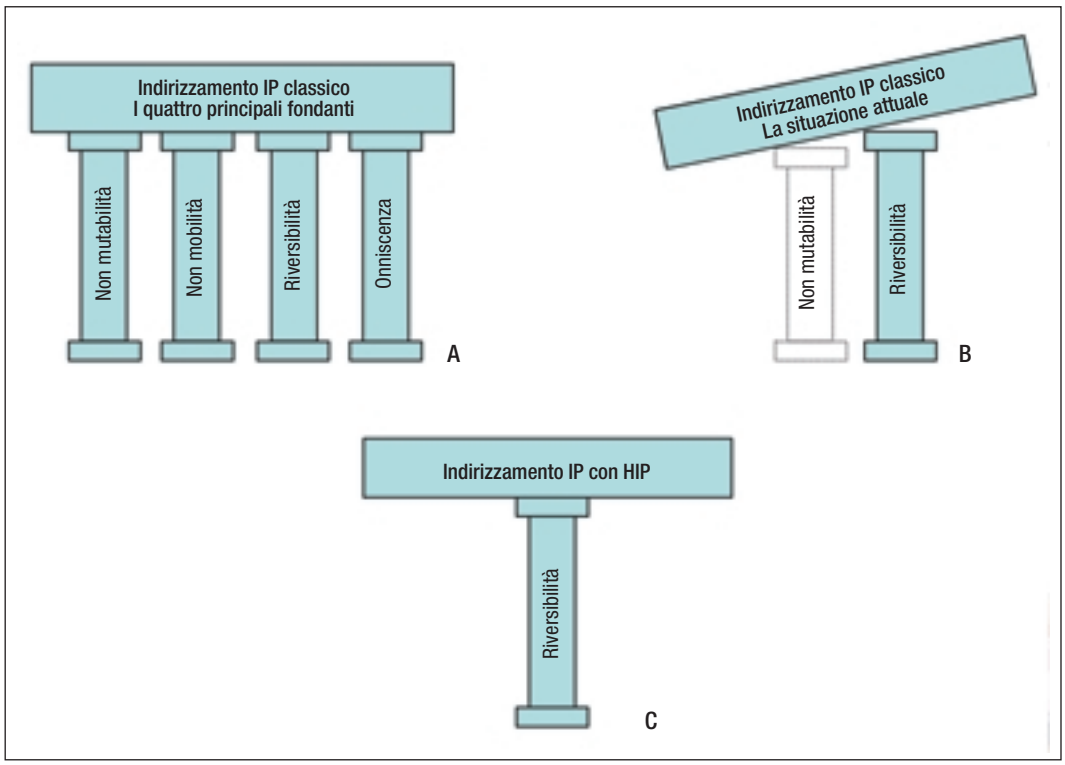

FIGURA 4
A *I principi fondanti dell'indirizzamento IP classico,* **B** *la situazione attuale,* **C** *il paradigma HIP*

te alla destinazione (come già visto, la *Network Address Translation* limita pesantemente la validità generale di questo classico "invariante");

- *non-mobililtà*: dal punto di vista del peer remoto, l'indirizzo IP di un *host* in movimento non deve variare durante la comunicazione; questo è ciò che il *Mobile IP* classico tenta di attuare, con grandissimi e praticamente insormontabili problemi di scalabilità;
- *reversibilità*: un *host* che ha ricevuto un pacchetto può sempre raggiungere la sorgente del pacchetto invertendo gli indirizzi di sorgente e di destinazione;
- *onniscenza*: ogni *host* sa quale indirizzo deve usare un partner remoto per inviargli un pacchetto.

Come mostrato nella figura 4 **B**, nella rete internet attuale il primo e il quarto principio sono già stati pienamente violati (principalmente a causa della *Network Address Translation*) e il secondo, la non-mobilità, è già messo in pericolo da alcuni aspetti del Mobile IP e del multihoming. Nelle precedenti sezioni di questo articolo è stata illustrata la dimensione notevole dei problemi generati da questo venir meno dei paradigmi di base secondo i quali il protocollo IP fu inizialmente concepito. È interessante notare che IPv6 è in realtà un tentativo di ripristinare il primo grande invariante della internet, la non mutabilità, fornendo uno spazio di indirizzi così ampio che non si possa più presentare la necessità di mutare l'indirizzo IP di un pacchetto in transito nella rete.

L'interessante aspetto rivoluzionario di HIP è che l'unica ipotesi fondamentale che sarà ancora necessaria è la terza, la reversibilità (Figura 4 **C**): se il verso della comunicazione è invertibile semplicemente scambiando gli indirizzi di sorgente e di destinazione, allora non è più strettamente necessario contare sulla non-mutabilità, sulla non-mobilità e sull'onniscenza relativa agli indirizzi IP, in quanto sarebbero forniti in modo indiretto dal sistema HIP.

4. CONCLUSIONI

In questo articolo si è cercato di dimostrare che l'approccio strategico corretto per il percorso di crescita della internet è evoluzionario e che un semplice "nuovo protocollo IP", per esempio IPv6, è solo uno dei fattori di crescita della internet, forse necessario, ma da solo assolutamente insufficiente. La rapidissima avanzata delle nuove applicazioni distribuite *peer-to-peer* e, più in generale, *overlay*, pone la questione

di come possano essere superati alcuni gravi problemi tecnologici della internet che la limitano fortemente, oltre a rendere praticamente impossibili la mobilità e la comunicazione multipunto su larga scala. Abbiamo evidenziato che i problemi principali si concentrano nella ormai vecchia concezione del ruolo degli indirizzi di rete IP, che assumono simultaneamente i significati di locazione fisica dell'*host*, di identità dell'*host* e di punto di terminazione a livello applicativo della comunicazione. Un sovraccarico semantico così imponente pone vincoli sempre più stringenti su un utilizzo flessibile degli indirizzi IP e una delle principali problematiche a livello strategico della internet del futuro è il disaccoppiamento dei concetti di identità degli host e della loro locazione geografica. In tal senso, il processo di evoluzione della internet è ancora in uno stato abbastanza prematuro: si stanno studiando soluzioni come l'*Host Identification Protocol*, per stabilire un'identità degli host indipendente dall'indirizzo IP, ma si è ancora ai primi passi. Ciò è in netto contrasto con lo stato ormai molto avanzato della standardizzazione di IPv6 che, però, non è ancora riuscito a penetrare in modo significativo nel mercato. Il fatto che IPv6 non si propone di risolvere il problema del disaccoppiamento di identità e indirizzo, dimostra che effettivamente il nuovo protocollo IP, da solo, non potrà superare tutti gli ostacoli che ci separano dalla internet del futuro.

Bibliografia

[1] Chu Y., Rao S.G., Zhang H.: *A case for end system multicast*. In: Proceedings of ACM SIGMETRICS'00, Santa Clara, CA, June 2000, p. 1–12.

[2] Holbrook H., Cheriton D.: *IP multicast channels: EXPRESS support for large-scale single-source applications*. In: Proceedings of ACM SIGCOMM'99, Cambridge, Massachusetts, Aug. 1999, p. 65–78.

[3] Stoica I., Ng T., Zhanh H.: REUNITE: *A recursive unicast approach to multicast*. In: Proceedings of INFOCOM'00, Tel-Aviv, Israel, Mar. 2000, p. 1644–1653.

[4] Stoica I., Adkins D., Zhuang S., et al: *Internet Indirection Infrastructure*. In: Proceedings of ACM SIGCOMM, Pittsburgh, PA, 2002, http://i3 .cs.berkeley.edu/publications/papers/i3-sigcomm.pdf

[5] Snoeren A.C., Balakrishnan H.: *An approach to host mobility*. In: Proceedings of ACM/IEEE MOBICOM'99, Cambridge, MA, Aug. 1999.

[6] Jannotti J., Gifford D.K., Johnson K.L., Kaashoek M.F., O'Toole J.W.: Overcast: *Reliable multicasting with an overlay network*. In: Proceedings of the 4-th USENIX Symposium on Operating Systems Design and Implementation, San Diego, California, October 2000, p. 197–212.

[7] Moskowitz R., Nikander P.: *Host Identity Protocol (HIP) Architecture*. IETF RFC 4423, May 2006.

[8] Laganier J., Eggert L.: *Host Identity Protocol (HIP) Rendezvous Extension*. IETF Internet-Draft, Network Working Group, draft-ietf-hip-rvs-05draft-ietf-hip-rvs (Experimental), June 7, 2006.

[9] Nikander P., Laganier J.: *Host Identity Protocol (HIP) Domain Name System (DNS) Extensions*. IETF Internet-Draft, Network Working Group, draft-ietf-hip-dns (Experimental), April 13, 2007.

[10] Henderson T.: *End-Host Mobility and Multihoming with the Host Identity Protocol*. IETF Internet-Draft, Network Working Group, draft-ietf-hip-mm-05, March 2, 2007.

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 nelle reti IP, 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

MAURIZIO DÈCINA è professore ordinario di telecomunicazioni al Politecnico di Milano presso il Dipartimento di Elettronica ed Informazione. Ha fondato e diretto il centro CEFRIEL del Politecnico di Milano dal 1987 al 2003.
Dal 1994 al 1995 è stato presidente della Communications Society dell'IEEE.
Ha ricevuto tre premi dell'IEEE: nel 1986 il Fellow Award per la telefonia a pacchetto, nel 1997 l'Award in International Communications per gli standard internazionali, e nel 2000 il premio alla carriera Third Millennium Medal Award per gli esperti di telecomunicazioni più influenti nella seconda metà del 900.
Il prof. Dècina ha lavorato nell'industria, Telecom Italia, Italtel e AT&T-Bell Laboratories, ed stato membro del consiglio di amministrazione di numerose aziende, tra cui Telecom Italia, Italtel, Tiscali e I.Net. Ha anche fondato alcune aziende start-up, tra cui ICT Consulting e Securmatics.
E-mail: maurizio.decina@polimi.it