

numero di peer  $N$ . Quindi, le applicazioni con architettura P2P possono essere scalabili e la scalabilità è una diretta conseguenza del fatto che i peer re-distribuiscono i bit oltre che a scaricarli.

### BitTorrent

BitTorrent è un diffuso protocollo P2P per la distribuzione di file [BitTorrent 2007]. Nel gergo di BitTorrent, l'insieme di tutti i peer che partecipano alla distribuzione di un particolare file è chiamato **torrent** (*torrente*). I peer in un torrent scaricano **parti** (*chunk*) del file di uguale dimensione uno dall'altro, con una dimensione tipica di 256 kbyte. Quando un peer entra a far parte di un torrent per la prima volta, non ha parti del file. Col passare del tempo, accumula sempre più pezzi che, mentre scarica, invia agli altri peer. Una volta che un peer ha acquisito l'intero file, può (egoisticamente) lasciare il torrent o (altruisticamente) rimanere nel torrent e continuare a inviare parti agli altri peer. Inoltre, qualsiasi peer può lasciare il torrent in qualsiasi momento con solo un sotto-insieme delle parti del file e rientrare a far parte del torrent in seguito.

Illustriamo ora più da vicino come funziona BitTorrent. Dato che si tratta di un protocollo piuttosto complicato, descriveremo solo i suoi meccanismi più importanti, lasciando da parte alcuni dettagli. Ciascun torrent ha un nodo di infrastruttura chiamato **tracker**. Quando un peer entra a far parte di un torrent, si registra presso il tracker e periodicamente lo informa che è ancora nel torrent. In questo modo, il tracker tiene traccia dei peer che stanno partecipando al torrent. Un certo torrent può avere centinaia o migliaia di peer che partecipano in un dato istante.

Come mostrato nella Figura 2.26, quando un nuovo peer, Alice, entra a far parte di un torrent, il tracker seleziona in modo casuale un sottoinsieme di peer (diciamo 50) dall'insieme dei peer che stanno partecipando a quel torrent, e invia l'indirizzo IP di questi 50 peer ad Alice. Avendo la lista dei peer, Alice cerca di stabilire delle connessioni TCP contemporanee con tutti i peer della lista. Chiamiamo i peer con i quali Alice riesce a stabilire una connessione TCP "peer vicini" (*neighboring peer*). Nella Figura 2.26 si vede che Alice ha solo tre peer vicini (ma normalmente ne avrebbe molti di più). Col passare del tempo, alcuni di questi peer possono lasciare il torrent, mentre altri (al di fuori dei 50 iniziali) possono cercare di stabilire una connessione TCP con Alice: quindi i peer contigui a un dato peer cambiano nel tempo.

In un certo istante, ciascun peer avrà un sottoinsieme delle parti di un file e peer diversi avranno differenti sottoinsiemi. Periodicamente Alice chiederà a ciascuno dei suoi vicini, sulle connessioni TCP, la lista delle parti del file in loro possesso. Tramite questa conoscenza, Alice invierà richieste, di nuovo sulle connessioni TCP, per le parti del file che ancora le mancano.

Di conseguenza, in un dato istante Alice avrà un sottoinsieme delle parti del file e saprà quali parti hanno i suoi vicini. Con queste informazioni, Alice deve prendere due importanti decisioni: in primo luogo quali parti deve richiedere per prime ai suoi vicini e, secondariamente, a quali vicini dovrebbe inviare le parti richieste. Nel decidere quali parti richiedere, Alice adotta la tecnica del **rarest first** (il più raro per primo). L'idea è determinare, tra le parti che ancora le mancano, quelle che sono più rare; cioè le parti con il minor numero di copie ripetute tra i suoi vicini, e richiederle per prime. In questo modo, le parti più rare vengono re-distribuite più velocemente, cercando, approssimativamente, di rendere uguale il numero di copie di ciascuna parte nel torrent.

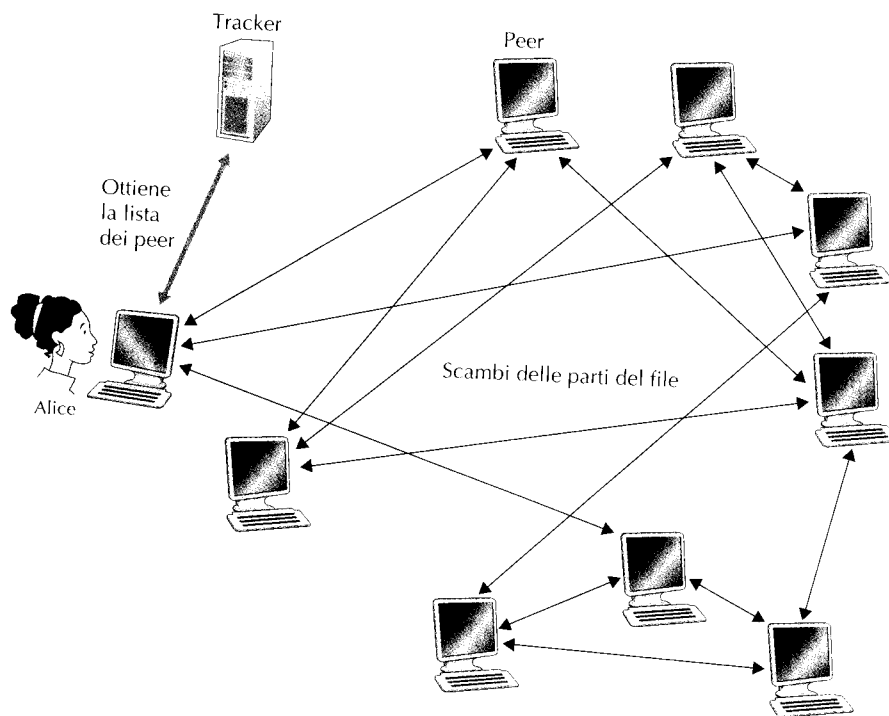


Figura 2.26 Distribuzione di file con BitTorrent.

Per determinare a quali richieste Alice debba rispondere, BitTorrent usa un intelligente algoritmo di trading (scambio). L'idea di base è che Alice attribuisca priorità ai vicini che le stanno inviando dati in questo momento alla frequenza più alta. Specificatamente, per ciascuno dei suoi vicini, Alice misura continuamente la frequenza alla quale riceve i bit e determina i quattro peer che le stanno passando i bit alla frequenza più elevata. Alice poi contraccambia inviando le parti del file a quegli stessi quattro peer. Ogni 10 secondi ricalcola la frequenza e può darsi che modifichi l'insieme dei quattro peer. Aspetto importante è che ogni 30 secondi sceglie casualmente un vicino in più e gli invia le parti. Chiamiamo Roberto il peer scelto casualmente. Dato che Alice sta inviando dati a Roberto, potrebbe diventare uno dei quattro "uploader" principali di Roberto, nel qual caso Roberto inizierebbe a inviare dati a Alice. Se la frequenza alla quale Roberto manda i dati ad Alice è abbastanza alta, Roberto, a sua volta, potrebbe diventare uno dei quattro "uploader" principali di Alice. In altre parole, ogni 30 secondi Alice sceglie casualmente un nuovo compagno di scambi e inizia a scambiare parti con quello. Se i due peer sono soddisfatti degli scambi, l'uno metterà l'altro nella propria lista dei quattro "uploader" principali e continueranno a effettuare scambi tra loro fino a che uno non trova un partner migliore. L'effetto è che i peer in grado di inviare file a frequenze compatibili tendono a trovarsi. La selezione casuale dei vicini consente anche a nuovi peer di ottenere le parti del file, in modo che abbiano qualcosa da scambiare. Tutti gli altri peer, a parte quei cinque (i quattro peer "top" e un peer di prova), sono "soffocati", cioè non ricevono alcune porzioni da Alice.

Nella condivisione di file P2P, un problema diffuso è il **free-riding** (sfruttamento), nel quale un peer scarica file dal sistema di condivisione, senza inviarne. L'algoritmo di trading di BitTorrent elimina virtualmente il problema del free-riding, in quanto, se Alice vuole scaricare bit da Roberto a una frequenza decente per un periodo di tempo esteso, deve nello stesso tempo inviare bit a Roberto a una frequenza decente. BitTorrent ha molti altri meccanismi interessanti che non vengono qui trattati (mini chunk, pipelining, prima selezione casuale, modalità endgame e anti-snubbing) [Cohen 2003].

### 2.6.2 Ricerca di informazioni in una comunità P2P

Una componente critica di molte applicazioni P2P è un indice delle informazioni, cioè una corrispondenza tra le informazioni e la loro posizione negli host. In questo tipo di applicazioni, i peer *aggiornano* dinamicamente l'indice e *cercano* in esso. Poiché la nozione di "corrispondenza tra le informazioni e relativa posizione sugli host" può sembrare un po' astratta, consideriamo un paio di esempi concreti.

- Generalmente, in un sistema di condivisione file P2P abbiamo un numero elevato di peer connessi, e ciascuno dispone di oggetti da condividere, tra cui MP3, video, immagini e software. Un sistema di condivisione file P2P ha un indice che dinamicamente tiene traccia dei file che i peer rendono disponibili nella condivisione. Per ciascuna copia di ciascun file messo in condivisione dalla comunità dei peer, l'indice mantiene una voce che fa corrispondere le informazioni sulla copia (per esempio, se è una canzone MP3, il titolo della canzone, l'artista e così via) all'indirizzo IP del peer che ha la copia. L'indice è aggiornato dinamicamente con l'andare e venire dei peer e quando i peer ottengono nuove copie del file. Per esempio, quando un peer entra a far parte del sistema, informa l'indice dei file che possiede. Quando un utente particolare, diciamo Alice, vuole ottenere un dato file, cerca nell'indice per localizzare le copie del file desiderato. Dopo aver localizzato i peer che hanno le copie del file, può scaricare il file da quei peer. Una volta che ha ottenuto l'intero file, l'indice viene aggiornato per tenere conto della nuova copia del file di Alice.
- In un'applicazione di messaggistica istantanea, c'è un indice che fa corrispondere i nomi utenti alle posizioni (indirizzi IP). Per comprendere l'importanza dell'indice in queste applicazioni, considerate due utenti "Alice-bellissima" e "Roberto-simpatico", ciascuno ha l'altro nella propria lista degli amici (buddy list). Quando "Roberto-simpatico" fa partire il suo client per la messaggistica istantanea su un host con indirizzo IP X, il suo client informerà l'indice che "Roberto-simpatico" è on-line all'indirizzo IP X. Più tardi, quando "Alice-bellissima" fa partire il suo client per la messaggistica istantanea, dato che "Roberto-simpatico" è nella sua buddy-list, il suo client cerca nell'indice "Roberto-simpatico" e scopre che è on-line all'indirizzo IP X. "Alice-bellissima" può quindi stabilire una connessione TCP diretta all'host con indirizzo IP X e iniziare a mandare messaggi a "Roberto-simpatico". Oltre alla messaggistica istantanea, anche altre delle attuali applicazioni usano un indice per tenere traccia della presenza, compresi i sistemi di telefonia su Internet (si veda il Paragrafo 2.6.3).

Diciamo brevemente che il protocollo BitTorrent, che è solamente un protocollo per la distribuzione di file, non fornisce alcuna funzionalità per indicizzare e cercare i file. In segui-