Sicurezza dei sistemi e delle reti[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2013/14

Lezione XXIII: Anonimato sul web

Privacy e web

Caricare una pagina web da un server HTTP espone moltissimi *dati personali* (alcuni addirittura *sensibili*)

- IP del client (pseudonym)
- IP del server (il client è interessato a quel server)
- identità (vedi `https://panopticlick.eff.org`)
- Dati del browser (history, cookie,...)
- System profile
- Dati trasmessi con form,...

Chi è il nemico?

1. Un eavesdropper può osservare il traffico: anche quando è criptato (`https`) gli IP e il system profile è accessibile
2. Il server web conserva i dati riguardanti il client
3. Il gestore della rete osserva e/o censura il traffico

Come ci si difende?

1. HTTPS
2. Proxy, cambiando proxy server (e browser!)
3. Onion Routing

Proxy


```
...
REMOTE_ADDR = 194.85.1.1
HTTP_ACCEPT_LANGUAGE = ru
HTTP_USER_AGENT = Mozilla/4.0
HTTP_HOST = www.webserver.ru
HTTP_VIA = 194.85.1.1 (Squid/2.4.STABLE7)
HTTP_X_FORWARDED_FOR = 194.115.5.5
...
```

proxy

Transparent proxy `HTTP_X_FORWARDED_FOR` è l'IP del client originale!

Anonymous Proxies IP nascosto

Distorting Proxies IP falso

High Anonymity Proxies `HTTP_VIA` vuoto

Privoxy

Privoxy è un proxy web con avanzate capacità di filtraggio progettato per la privacy

- gestisce i cookie
- javascript
- personalizzabile

Questa categoria di prodotti è detta protocol cleaner

Onion Routing

Onion Routing (OR) è una tecnica sviluppata dal Naval Research Laboratory di Washington.
Il traffico viene instradato in una serie (mutevole) di *onion router* in maniera tale da rendere difficile il tracciamento delle attività.
Gli onion router costituiscono dei mix di Chaum.

Mix di Chaum

Mix di Chaum

Un mix riceve messaggi di lunghezza fissa, li cripta, aspetta di averne in numero sufficiente da garantire un certo livello di anonimato e inoltra i messaggi (in ordine arbitrario) ad altri mix.

Onion router

- Gli onion router intermedi non hanno informazione sufficiente per tracciare mittente/destinatario e traffico
- Rimane una certa criticità degli exit node e (minore) dei nodi d'entrata

Riassumendo

L'anonimato nella navigazione web è un problema piuttosto complesso. Varie difese:

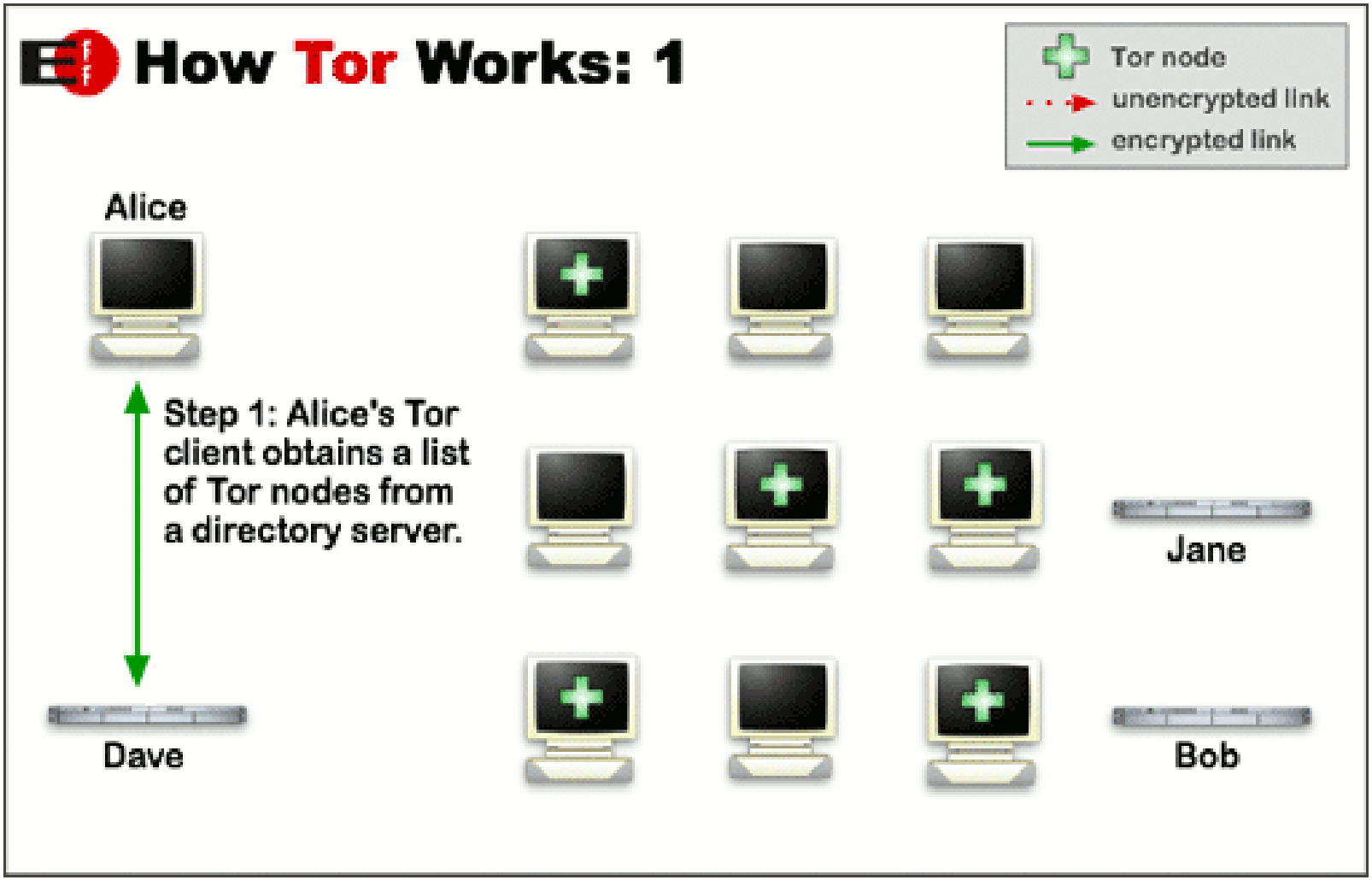
- HTTPS
- Proxy
- Onion Routing

TOR

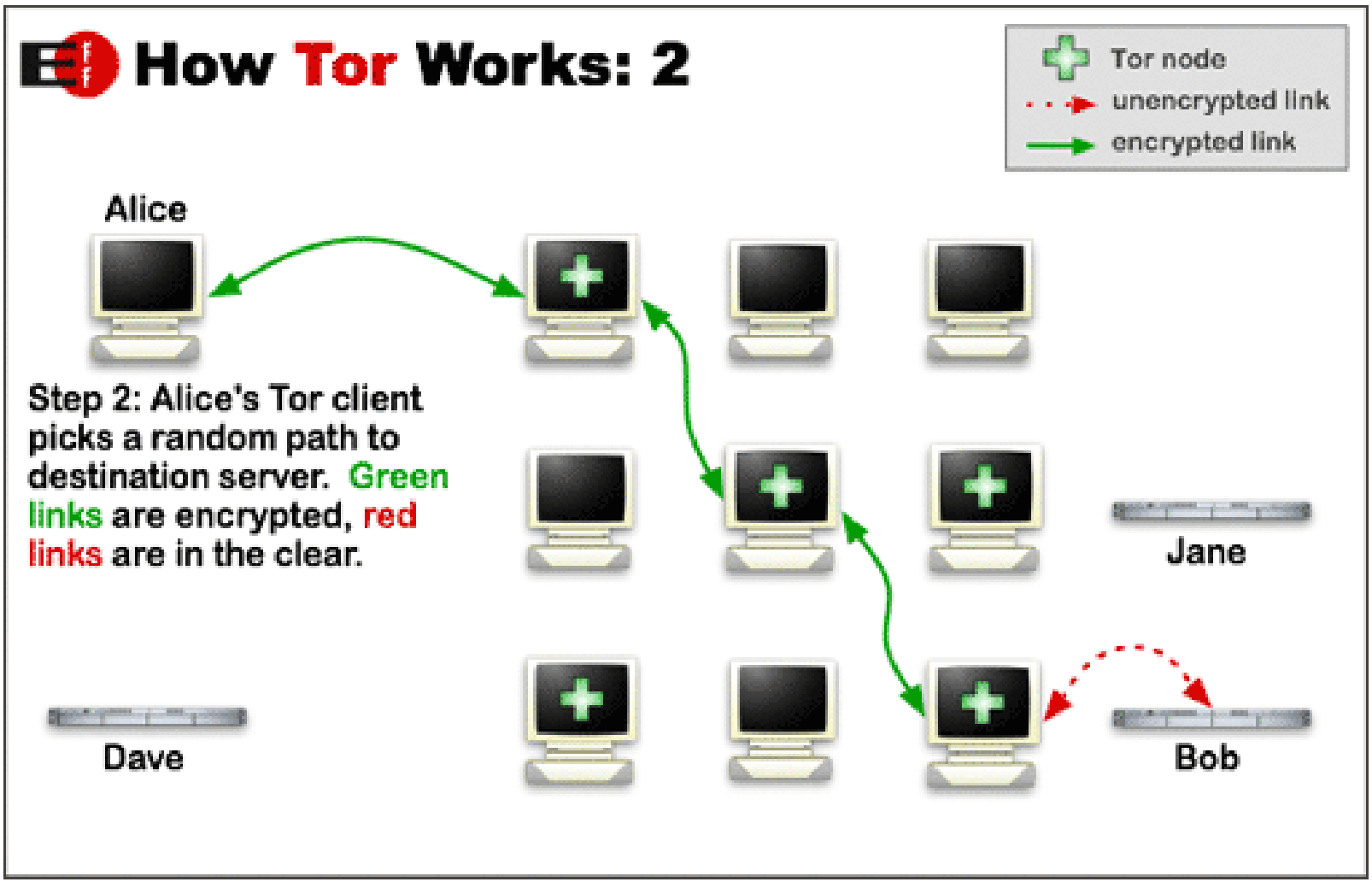
TOR (The Onion Router project) è l'evoluzione piú recente del concetto di OR. Sviluppato da NRL, open source, supportato da EFF.

Il progetto fa parecchi sforzi per rendere il prodotto comprensibile e utilizzabile anche da utenti inesperti (il numero di utenti è un valore per l'anonimato).

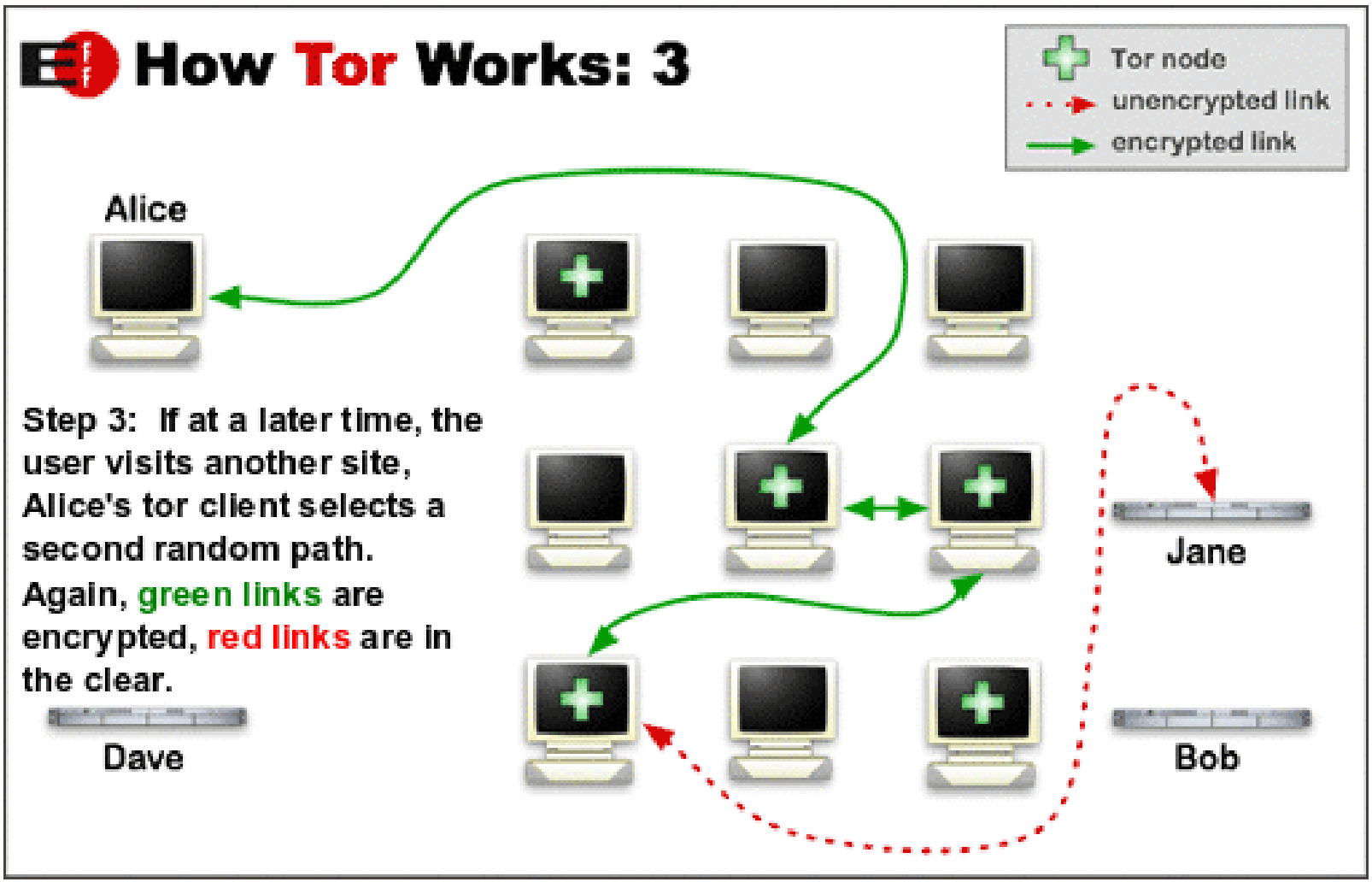
Onion routing con TOR



Onion routing con TOR



Onion routing con TOR



Circuiti

L'instradamento di ogni messaggio viene detto circuito

- Ogni nodo del circuito conosce solo il nodo precedente e successivo (non origine e destinazione)
- Molte richieste diverse vengono multiplexate in un unico circuito
- Robusto rispetto alla compromissione o l'introduzione di onion router malevoli
- Nodi trusted operano da *directory server* iniziali

Struttura di una rete TOR

- Nodi utenti hanno un Onion Proxy (OP)
- Onion Router (OR) connessi tra loro con TLS
- Gli OR hanno una long-term key e short-term "onion" key
- L'unità di trasmissione è la cella, di dimensione fissa di 512 byte

Uso delle chiavi

La *long-term identity key* viene usata per

- firmare la descrizione del router: certificati TLS, chiavi, metadati, *exit policy*
- firmare gli elenchi di router

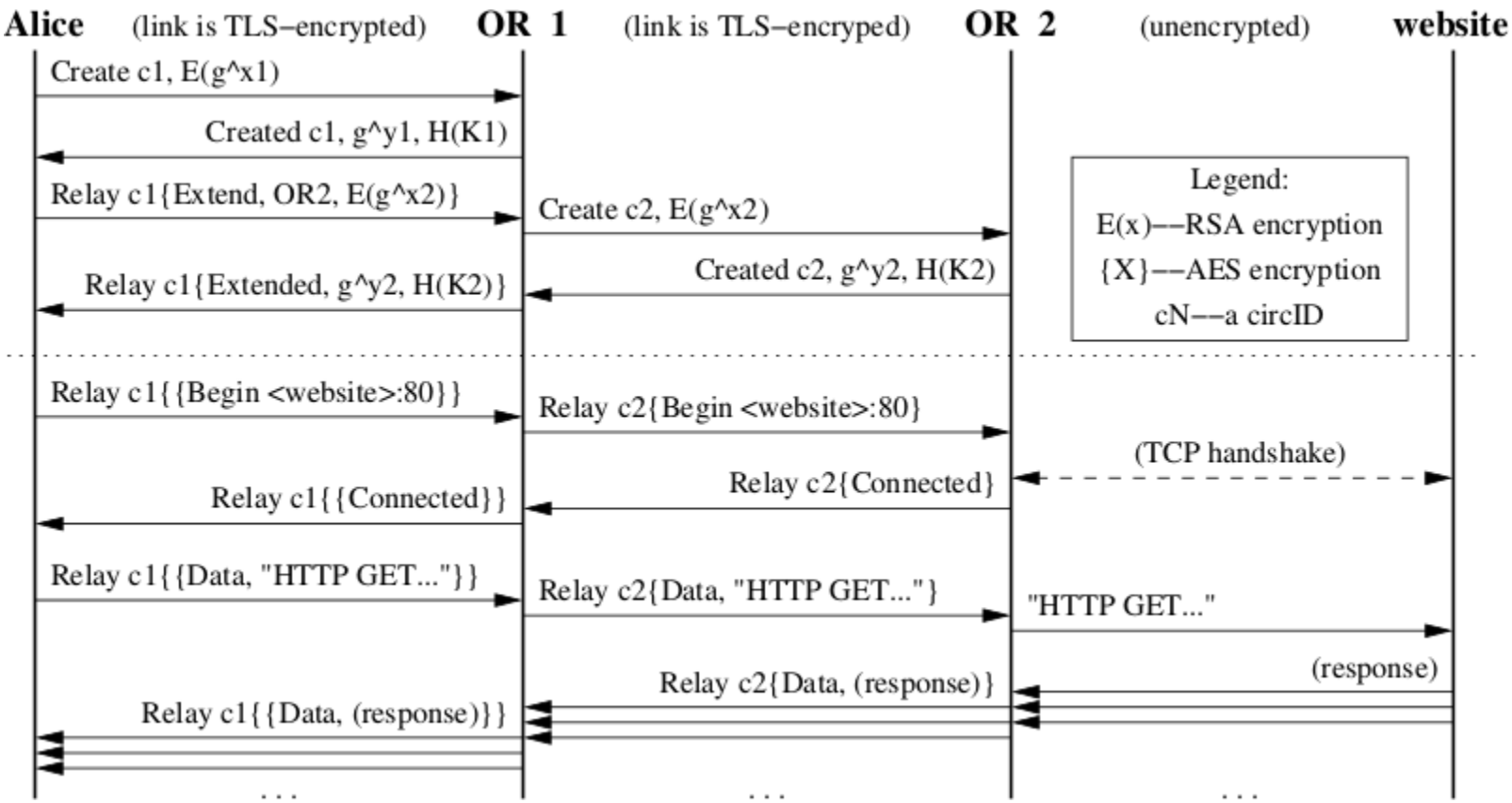
La *short-term key* (onion key) per:

- decrittare le richieste di circuiti
- negoziare chiavi una tantum (ephemeral key) che garantiscono la *forward secrecy*

Circuiti

- l'OP costruisce un circuito in background, e diversi stream utente vengono multiplexati sullo stesso circuito
- Ogni minuto viene creato un nuovo circuito

Creazione di un circuito



La prima fase *estende* il circuito, poi c'è il relay del traffico

Creazione dello stream

L'OP sceglie quale OR può fare da exit node (ognuno ha una *exit policy*)
Solo TCP stream possono essere creati (UDP, e quindi le risouzioni DNS, rimangono problematiche: vedi `http://code.google.com/p/torsocks/` per una soluzione)
Un protocol cleaner è necessario per evitare che informazioni rilevanti finiscano nello stream.

Riassumendo

TOR

- il maggior progetto di Onion Routing
- permette la creazione di circuiti anonimi
- necessita di un protocol cleaner

Reti p2p

peer-to-peer

Un gruppo di nodi che opera sia come client che come server (ogni nodo è in grado di svolgere le stesse operazioni)

Napster-like un server centrale conserva un indice dei servizi

Gnutella-like anche l'indice è distribuito fra i peer (eccetto un elenco di bootstrapping)

Privacy delle operazioni

- L'indice (chi fornisce che cosa) è sostanzialmente pubblico
- La fruizione del servizio generalmente è HTTP (vedi privacy web)
- In alcuni casi (p.es. BitTorrent) i metadati contengono molte informazioni personali
- Potrebbero essere necessarie anche operazioni in cui non si è direttamente interessati (In Svizzera p.es., dove è permesso il download di materiale protetto da copyright, è vietato condividerlo)

Freenet

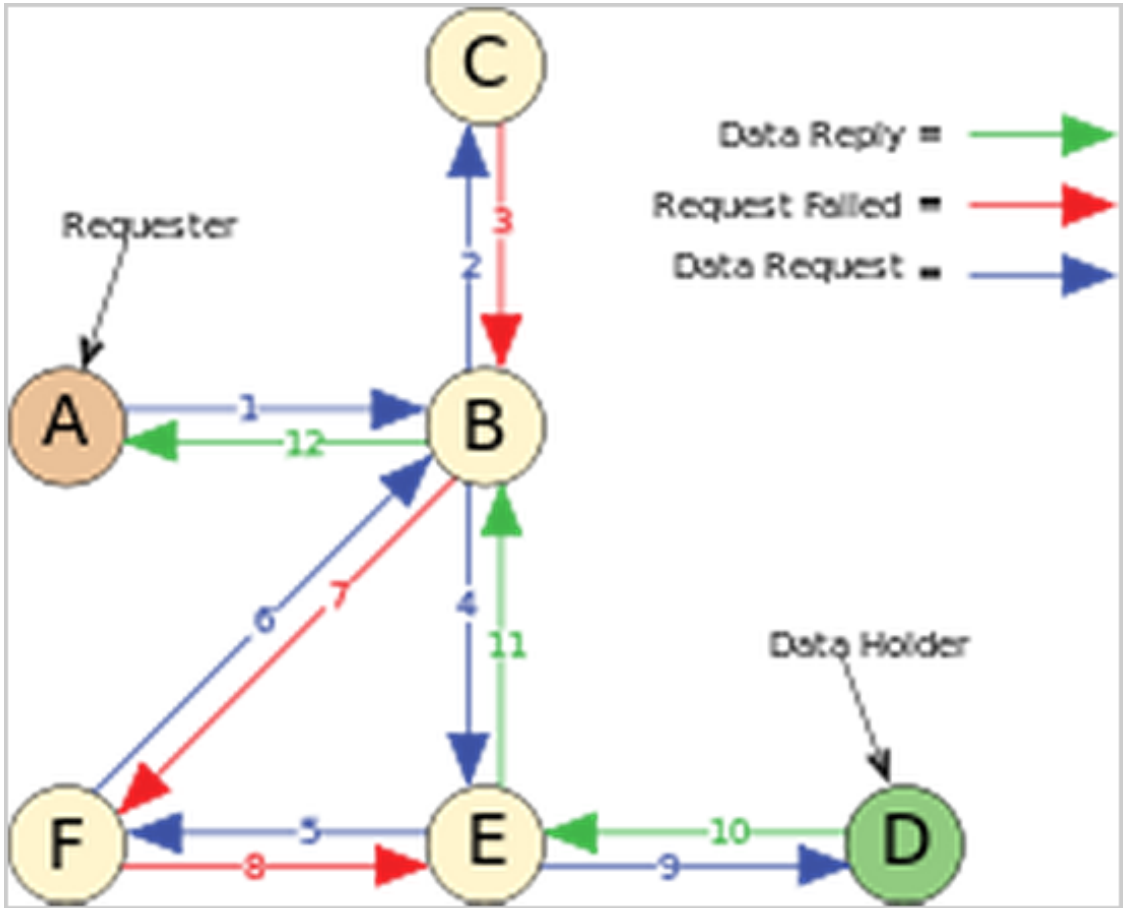
Un tentativo di realizzare un sistema di pubblicazione di contenuti resistente alle censure

- peer-to-peer e completamente decentralizzato
- i dati vengono criptati e replicati su molti nodi
- diventa estremamente difficile sapere chi ha che cosa
- i singoli nodi non hanno modo di sapere cosa mettono a disposizione

Architettura di Freenet

- Ogni contenuto è identificato solo da un hash SHA-256 (non c'è supporto diretto alle ricerche)
- Ogni nodo "conosce" solo un numero ristretto di altri nodi che può raggiungere direttamente
- I contenuti vengono passati ai vicini (e posti in una cache locale), senza sapere se è la destinazione finale
- key-based routing euristico

Freenet routing



Freenet

- Un nodo inserisce un file *nella rete*: a quel punto può anche disconnettersi, perché il file viene spezzato e conservato fra i peer attivi
- I contenuti piú richiesti vengono inseriti piú frequentemente nelle cache (mentre quelli non richiesti tendono a sparire)
- Opennet (chiunque può connettersi) e Darknet (rete fra trusted node con topologia manuale)

Freenet

Lettura obbligatoria:
Clarke, Ian, et al. "Protecting free expression online with Freenet." Internet Computing, IEEE 6.1 (2002): 40-49.