Reti

Alessandro S.

4 febbraio 2019

Indice

1 Satelliti

Un satellite di comunicazioni può essere immaginato come un grande ripetitore di microonde collocato nel cielo. Questo dispositivo contiene diversi transponder, ossia ricetrasmettitori satellitari. Ognuno ascolta una parte dello spettro, amplifica il segnale in ingresso e lo ritrasmette su un'altra frequenza per evitare interferenze con il segnale in arrivo. I satelliti sono per natura dei mezzi di comunicazione broadcast: costa uguale mandare un messaggio a una stazione o a mille, però questo è uno svantaggio dal punto di vista della privacy: tutti possono vedere tutto, è quindi necessario criptare dati sensibili. Inoltre hanno le seguenti proprietà: il costo di trasmettere un messaggio è indipendente dalla distanza da percorrere, hanno un eccellente tasso di errore e in caso di necessità sono possono essere dispiegati in tempi rapidi. Al momento per le telecomunicazioni sembra che le fibre ottiche combinate con la rete radio cellulare siano il sistema migliore, tranne per alcuni settori specializzati come sistemi di navigazione come il GPS, trasmissioni broadcast unidirezionali (ad esempio trasmissioni televisive), o comunicazioni in luoghi dove non sono disponibili infrastrutture di terra (luoghi scarsamente popolati o in paesi poveri). Esistono tre tipi di satelliti GEO, MEO E LEO.

GEO: Geostationary Earth Orbit

Sono posti ad un'altezza di 35800km in modo tale da apparire fermi nel cielo e disposti ad intervalli di due gradi sulla circonferenza equatoriale per evitare interferenze: ciò limita il numero totale di tali satelliti a 180. Un solo satellite di questo tipo può coprire da un minimo di qualche centinaio di km fino a un terzo del globo. I satelliti GEO hanno un ritardo di circa 270ms data la distanza dalla terra, però non devono essere inseguiti essendo geosincroni. Sono usati per fare rilevazioni meteorologiche e per le tv satellitari. VSAT(Very Small Aperture Terminal) sono delle microstazioni, utili sopratutto in aree rurali, che ricevono dati direttamente dal satellite, invece per inviare dati si affidano a un hub di terra che amplifica il segnale e lo manda al satellite.

MEO: Medium Earth Orbit

A circa 20000km di altezza, tra le due fasce di Van Allen, si trovono i satelliti MEO. Non sono geosincroni e impiegano 6 ore a fare il giro del pianeta, vanno di conseguenza seguiti. Poiché sono in un orbita più bassa dei GEO richiedono dispositivi meno potenti per essere raggiunti. Sono usati per si-

stemi di navigazione come il GPS statunitense, il GLONASS russo, o Galileo dell'Unione Europea.

LEO: Low Earth Orbit

Sono posizionati ad altezze molto più basse degli altri due tipi di satelliti, quindi hanno dei tempi di latenza di pochi ms, però muovendosi molto velocemente sono necessari un alto numero di satelliti per un sistema completo. Sono usati sia per comunicazioni dati che voce. Ci sono due esempi di sistemi completi di satelliti LEO: Iridium e Globalstar. Una interessante proprietà di Iridium è che l'instradamento(routing) tra utenti distanti viene effettuato nello spazio: i satelliti si "passano" il messaggio finché il più vicino di loro al destinatario glielo consegna. Globalstar usa invece una tecnica più tradizionale chiamata "bentpipe", cioè appena riceve il segnale lo ritrasmette a terra e lì verrà instradato verso il destinatario (ciò potrebbe implicare ancora l'uso di satelliti). Usati per telefoni satellitari impiegati ad esempio da scienziati, militari o viaggiatori in aree remote del pianeta ma anche per le comunicazioni su stazioni petrolifere e nelle boe di segnalazione di tsunami.

2 Modulazione delta

Nel sistema telefonico il segnale analogico viene campionato un certo numero di volte al secondo (numero deciso tramite il teorema di Nyquist) e viene quindi digitalizzato: questa tecnica è chiamata Pulse Code Moldulation. Dopo aver digitalizzato la voce si usano delle tecniche di compressione per ridurre il numero di bit da inviare nella rete. Queste di basano sul principio che il segnale cambia in modo relativamente lento rispetto alla frequenza di campionamento, perciò gran parte dell'informazione è ridondante. La modulazione delta è una di esse. Infatti non trasmette l'ampiezza digitalizzata bensì un singolo bit a 1 se il nuovo campione è maggiore del precedente oppure a 0 se è minore. Se il segnale fa dei salti ampi, la codifica richiederà diversi periodi di campionamento per mettersi in pari: cioè l'errore viene ignorato, ciò è accettabile nel campo delle comunicazioni vocali. Questa tecnica ha il vantaggio di consumare poche risorse (particolarmente preziose su dispositivi mobili) e di impiegare poco tempo per essere portata a termine, ha però lo svantaggio che in caso il segnale cambiasse troppo rapidamente si perdono delle informazioni.

3 CDMA

Il costo di istallazione e mantenimento di una linea di trasmissione con grande larghezza di banda rispetto a una con con larghezza di banda stretta è poca, quindi i gestori di una linea cercano di usare una sola linea per quanti più utenti possibili. La tecnica di condividere il canale tra più utenti viene detta multiplexing e può essere realizzato in modi differenti. In TDM(Time Division Multiplexing) gli utenti periodicamente secondo una politica round-robin prendono uno alla volta il controllo dell'intera banda disponibile per un breve intervallo di tempo, si aggiungono piccoli tempi di guardia per permettere piccoli aggiustamenti tra due intervali. In FDM(Frequency Division Multiplexing) lo spettro delle frequenze viene suddiviso in sotto insiemi chiamti canali ognuno assegnato in uso esclusivo ad un utente, analogamente al TDM si aggiungono delle bande di guardia per tenere ben separati i canali. CDM(Code Division Multiplexing) noto anche come CDMA(Code Divsion Multiple Access) lavora in maniera completamente diversa. CDMA è una forma di comunicazione a spettro distribuito in cui un segnale a banda stretta viene sparso su una banda di frequenze più ampia, ciò lo rende più tollerante alle interferenze e permette a più segnali di utenti diversi di condividere la stessa banda di frequenza. Infatti CDMA permette a tutte le stazioni di trasmettere su tutto lo spettro di frequenza in ogni momento. Le trasmissioni simultanee secondo la teoria dei codici si sommano linearmente. La chiave di CDMA è quindi essere in grado di estrarre il segnale desiderato e di scartare tutto il resto come se fosse rumore. A ogni stazione è assegnato un codice di m bit chiamato sequenza di chip. Per trasmettere un bit a 1 la stazione trasmette la sua sequenza di chip, mentre per trasmettere un bit a 0 invia la negazione della sequenza. Quindi per inviare un bit a 1 o a 0 in realtà bisogna inviare m bit: ognuno di essi viene chiamato chip (tipicamente ci sono dai 64 ai 128 chip per bit). Tutte le sequenze di chip sono a due a due ortogonali, cioè il prodotto interno normalizzato di qualunque coppia di sequenze distinte $S_x e S_y$ è 0, invece il prodotto di una sequenza per se stessa è uguale a 1. Per generare le sequenze si usa un metodo noto come codici di Walsh che usa una notazione bipolare: -1 per indicare gli 0 e 1 per indicare gli 1. Quando due stazioni trasmettono contemporaneamente le loro sequenze di chip si sommano linearmente. Il recupero di effettua calcolando il prodotto interno normalizzato tra la sequenza di bit chip ricevuta e la sequenza di chip della stazione i cui bit si vogliono recuperare (per recuperare il bit inviato il ricevente deve conoscere in anticipo la sequenza di chip della stazione trasmittente). Il prodotto dà -1 se la stazione aveva inizialmente inviato uno zero, 1 se aveva inviato un 1 e 0 se la stazione non aveva inviato nulla. (pag 138 per vedere come funziona il calcolo del prodotto)

4 bit/byte stuffing

Per dare un servizio allo strato network lo strato data link usa i servizi forniti dallo strato fisico: esso trasporta delle grezze e ininterrotte sequenze di bit, senza preoccuparsi di eventuali errori. È compito dello strato data link trovare gli errori ed eventualmente correggerli. Per fare ciò ha bisogno che il flusso dei dati sia diviso in pezzetti chiamati frame. Tale divisione (framing) deve rendere facile per il ricevente di un flusso trovare l'inizio di ogni nuovo frame, ma nel contempo usando la minor banda possibile.

Un primo primitivo metodo di framing usa un campo dati all'inizio del frame per indicare il numero di byte del frame. Il destinatario così sa quanti byte contare per arrivare al fondo del frame corrente e all'inizio del frame successivo. Il problema di questo metodo è che un singolo errore nel campo dati manda irrimediabilmente fuori sincrono trasmittente e ricevente: il ricevente non sa dove inizia il prossimo frame e il trasmittente non sa da dove deve ritrasmettere. Per questi motivi non è quasi mai usato da solo.

Un secondo metodo di framing è il byte stuffing. In questo metodo viene posizionato un byte flag(etichetta) all'inizio e alla fine di ogni frame. Due byte flag indicano la fine di un frame l'inizio del successivo. Questo permette in caso di errore di scartare il frame corrente e passare subito al successivo cercando due flag byte di seguito. In caso il byte flag si presenti nei dati da inviare viene posizionato dal trasmittente un byte di escape che sarà poi rimosso dal ricevente. Stessa procedura se un byte di escape si trova nei dati: prima di esso viene messo un'altro byte di escape che sarà poi rimosso.

Un terzo metodo si chiama bit stuffing che serve ad aggirare la limitatezza del byte stuffing di usare per forza byte da 8 bit, cioè rende possibile l'uso di frame di grandezza qualsiasi. In questo metodo ogni frame inizia e finisce con la sequenza speciale di bit: 01111110 (notare i 6 uno). Se nei dati sono presenti cinque 1 di seguito il trasmittente ci inserisce subito dopo uno 0, invece il ricevente se vede cinque 1 seguiti da uno 0 rimuove quest'ultimo. In caso di perdita di sincronizzazione basta scorrere il flusso di dati fino a trovare la sequenza 01111110, in quanto quest'ultima non può trovarsi nei dati (una volta che sono stati preparati e spediti dal trasmittente) ma solo ai bordi del frame. USB usa bit stuffing, il protocollo PPP usa byte stuffing. Nel caso peggiore cioè quando tutto il carico è composto da byte flag nel byte stuffing la misura del frame aumenta del 100% perchè ogni byte flag va preceduto dal un byte di escape, mentre nel bit stuffing del 12,5% perchè viene aggiunto un bit ogni 8.

5 Aloha

I collegamenti di rete si possono dividere in due categorie: quelli point-to-point, che usano collegamenti individuali tra coppie di stazioni, e quelli broadcast che hanno un unico canale di comunicazione condiviso da tutte le stazioni chiamato canale multiaccesso. Il canale può essere una porzione dello spettro delle frequenze wireless o anche un filo o una fibra ottica. In qualsiasi rete broadcast la chiave è la scelta dell'entità che dovrà acquisire il diritto di utilizzo del canale in caso di competizione. I protocolli che se ne occupano appartengono al sottostrato MAC (Medium Access Control) dello strato collegamento dati (data link). I protocolli multiaccesso sono usati sopratutti nelle LAN e MAN, al contrario le WAN(eccezion fatta per i satelliti) usano connessioni punto a punto. Si parla di allocazione statica del canale quando la capacità del canale è divisa usando delle tecniche di multiplexing come FDM. Ne sono esempio le stazioni radio FM. Quando il numero di utenti è piccolo e ognuno di essi è costante nel trasmettere, questo tipo di allocazione è efficiente. Ma se il numero di utenti è grande e variabile o il carico della rete non è costante, ma è ma "bursty" cioè ci sono dei momenti dove il carico è molto basso e altri dove è molto alto, l'allocazione statica non si adatta bene alla situazione. Il problema principale è che quando un utente non invia nulla, la porzione del canale a lui assegnata è sprecata perché nessun altro la può usare. In molti sistemi di computer quindi per i problemi sopra descritti si usa l'allocazione dinamica del canale. Essa si basa sul seguente modello: è disponibile un solo canale per tutte le comunicazioni, tutte le stazioni possono sia ricevere che inviare, le stazioni sono indipendenti e se stanno trasmettendo non fanno nient'altro finché non hanno finito, le stazioni sono in grado di rilevare quando due frame si sovrappongono, tale evento è chiamato collisione. Nei canali cablati dei componenti hardware possono rilevare le collisioni e terminare la trasmissione. Questo tipo di rilevamento è molto più difficile nei canali senza fili (wireless) e perciò le collisioni in tali canali sono dedotte dalla mancanza di acknowledgement del frame.

Aloha è un protocollo per trasmissioni su un canale multiaccesso ad allocazione dinamica, è stato sviluppato da Norman Abramson negli anni 70 all'università delle Hawaii dove avevano la necessità di collegare varie stazioni sparse sulle isole dell'arcipelago, ma non esisteva un servizio telefonico funzionante da poter usare per lo scopo e stendere dei cavi appositi nell'oceano non era un'opzione. Il sistema che svilupparono si basava su radio a corto raggio, ma il protocollo Aloha si può applicare a qualsiasi sistema dove utenti non coordinati competono per l'uso di un singolo canale condiviso. Ci sono due versioni di Aloha: l'originale detta pura e la slotted, inventata nel 1972 da Roberts come miglioramento dell'originale. L'idea di fondo di Aloha(pura) è

semplice: consentire agli utenti di trasmettere ogni volta che hanno dati da inviare. Ovviamente ci potranno essere delle collisioni che danneggeranno i frame, ma grazie alla proprietà di feedback della trasmissione broadcast, un trasmettitore potrà sempre scoprire, ascoltando il canale, se il suo frame ha subito una collisione. Le collisioni avvengono quando due frame sono trasmessi simultaneamente, quei due frame sono rovinati e non possono essere recuperati andranno quindi ritrasmessi. La ritrasmissione avviene dopo aver atteso un tempo **casuale**, altrimenti le stesse stazioni ritrasmetterebbero nello stesso istante provocando così una nuova collisione. È come se venisse tirato un dado con molte facce e il tempo da aspettare fosse il valore della faccia del dado. Il numero di facce del dado deve essere aumentato al crescere del numero di utenti della rete. I frame hanno tutti la stessa dimensione perché la capacità di trasporto in Aloha è massima quando i frame hanno dimensione uniforme. In Aloha si può sperare di utilizzare al massimo il 18,4% del canale, tale numero si ottiene supponendo che una popolazione infinita di utenti generi nuovi frame in accordo con la distribuzione di Poisson. Ha quindi il vantaggio che all'aumentare degli utenti la sua efficienza non diminuisce: è quindi facilmente scalabile. In Aloha slotted il tempo è diviso in intervalli, ognuno corrispondente a un frame. Questi intervalli sono stabiliti da una speciale stazione che emette un segnale all'inizio di ogni nuovo intervallo. Una stazione non può inviare dati quando vuole ma deve attendere l'inizio di un intervallo per farlo, ciò trasforma il sistema continuo di Aloha puro in un sistema discreto. Grazie a questo due frame inviati sul canale o collidono completamente, o non collidono, mentre in Aloha pura i frame poteva collidere anche parzialmente. Quindi poiché il periodo vulnerabile è dimezzato viene raddoppiata la capacità di trasporto: 36,8%. Aloha(puro e slotted) ha lo svantaggio di non controllare prima di trasmettere se il canale è occupato da un'altra trasmissione.

6 CSMA

CSMA(Carrier Sense Multiple Acess) è una famiglia di protocolli per trasmissioni su un canale ad accesso multiplo con rilevamento della portante. Questo significa che le stazioni rimangano in ascolto di una portante (ossia una trasmissione) e si comportano di conseguenza: in generale finché rilevano che è in corso una trasmissione, per non generare una collisione non iniziano a trasmettere.

In CSMA 1-persistente, se una stazione ha dei dati da trasmettere e il canale è libero la stazione li trasmette, se invece il canale è occupato aspetta finchè il canale non si libera. Il nome è 1-persistente perché la probabilità

di trasmettere a canale libero è 1, mentre persistente deriva dal fatto che continua ininterrottamente a controllare se il canale si è liberato.

In CSMA non persistente se la stazione rileva che il canale è occupato attente un tempo casuale prima di ripetere la rilevazione, se lo trova libero trasmette.

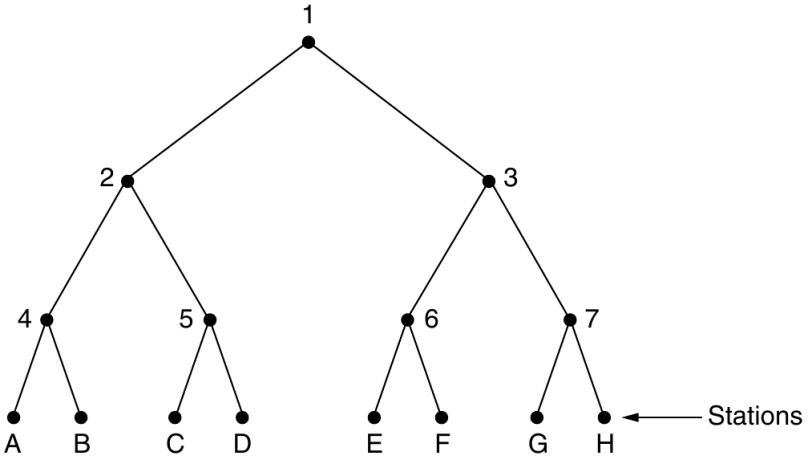
In CSMA p-persistente il tempo è diviso in intervalli e se la stazione rileva che il canale è occupato attente l'intervallo successivo prima di effettuare di nuovo il controllo. Se il canale lo trova libero trasmette subito con una probabilità p e rimanda la trasmissione all'intervallo successivo con una probabilità di 1-p. Il processo si ripete fino a quando il frame non è stato trasmesso. In CSMA/CD (CSMA Collision Detection) le stazioni monitorano il canale e se il segnale che vi leggono è diverso da quello che stanno trasmettendo sanno che mentre stanno trasmettendo sta avvenendo una collisione. In questo caso interrompono immediatamente la trasmissione, quest'accorgimento fa risparmiare tempo e banda. Dopo aver rilevato una collisione attendono un tempo casuale e ritrasmettono il frame. Questo tipo di monitoraggio per motivi ingegneristici viene fatto solo su reti cablate.

CSMA ha il vantaggio di essere più performante di Aloha perché genera molte meno collisioni, ma non le elimina del tutto: nel caso di canale libero con più stazioni pronte a trasmettere inevitabilmente si produrrà una collisione, perché le stazioni non sono coordinate e quindi possono rilevare solo trasmissioni in atto e non possono sapere se le altre stazioni stanno per trasmettere. Una limitazione di CSMA è che nelle LAN wireless permette di scoprire solo se c'è attività intorno alla stazione che sta analizzando la portante, quando invece una stazione avrebbe interesse anche a sapere se c'è attorno al ricevitore. CSMA/CD è la base di Ethernet, IEEE 802.11 usa un versione più raffinata del CSMA p-persistente.

7 Adaptive Tree Walk

Nelle situazioni a carico leggero i protocolli a contesa (come Aloha) sono preferibili per i basso ritardo che comportano. Al crescere del carico, però la contesa diventa sempre più inefficiente perché la quantità di dati necessari all'arbitraggio del canale aumenta. Con i protocolli senza collisioni si verifica esattamente il contrario: a basso carico hanno un ritardo elevato, ma al crescere del carico l'efficienza del canale migliora. I protocolli a contesa limitata combinano il metodo della contesa, per ottenere un ritardo limitato a basso carico, e il metodo senza collisioni, per raggiungere una buona efficienza di canale nelle situazioni a carico più elevato. Per fare ciò dividono le stazioni in gruppi per diminuire il livello di competizione. Adaptive Tree Walk è uno

di questi protocolli. Esso prevede che il tempo sia diviso in intervalli e le stazioni siano rappresentate come foglie di un albero binario. Nel primo intervallo successivo a un frame trasmesso con successo, diciamo l'intervallo 0, tutte le stazioni possono competere per l'acquisizione del canale. Se una di esse ci riesce, bene. Se invece c'è una collisione, durante l'intervallo 1 solo le stazioni sotto al nodo sinistro della radice (2) possono provare a comunicare, se una di loro riesce ad acquisire il controllo del canale, l'intervallo successivo è riservato alle stazioni che si trovano sotto il nodo destro (3). Se invece c'è subito un'altra collisione, nell'intervallo 2 potranno tentare di comunicare solo le stazioni sotto il nodo sinistro del nodo 2 (4) e si ripete ricorsivamente l'algoritmo fino a che una stazione riesce a trasmettere con successo.



Un accorgimento per ottimizzare questo protocollo è iniziare la ricerca nel livello dell'albero binario che ha un numero medio di stazioni contendenti pari a 1. Adaptive Tree Walk prende spunto da un algoritmo che l'esercito statunitense utilizzava durante la Secondo Guerra Mondiale per verificare se il soldati fossero affetti da lue.

8 Codifica Manchester

La codifica Manchester è una codifica binaria per rappresentare i bit a 0 e 1 tramite il voltaggio. Con questa codifica ogni periodo di bit è diviso in due intervalli uguali. L'1 binario è inviato scegliendo un livello di tensione alto durante il primo intervallo e un livello basso durante il secondo. Lo schema contrario è usato per rappresentare uno 0 binario. Ciò permette ai ricevitori di determinare senza ambiguità il punto iniziale, il punto centrale e il punto finale di ogni bit senza fare riferimento a impulsi esterni, si dice cioè che è una codifica autosincronizzante. È meglio della codifica binaria semplice, che associa l'assenza di voltaggio al bit a 0 e la presenza con il bit a 1, perché tale

codifica non riesce a distinguere tra la linea libera e il segnale nullo associato al bit a 0, e perché velocità di campionamento anche solo leggermente diverse tra trasmettitore e ricevente possono mandarli fuori sincronia facendo perdere l'informazione del punto dove iniziano i bit, specialmente dopo una lunga serie di 0 o di 1. Lo svantaggio della codifica Manchester è che occupa il doppio della banda della codifica binaria elementare, perché gli impulsi sono larghi la metà. È usata dalla LAN Ethernet.

La codifica Manchester differenziale è una variante della Manchester elementare. In questa codifica il bit a 1 è indicato dall'assenza di una transizione all'inizio dell'intervallo, il contrario è usato per rappresentare uno 0 binario. In entrambi i casi c'è una transizione nel punto centrale. Questa codifica rispetto a alla Manchester normale richiede dei dispositivi più complessi, ma offre maggiore immunità ai rumori. È usata da alcune LAN ma non da Ethernet, perché si è preferita la semplicità di quella elementare.

9 Flooding

Lo strato di rete (network) fornisce servizi allo strato trasporto principalmente instradando i pacchetti dalla sorgente verso la destinazione che per essere raggiunta richiede il più delle volte di passare da molti router intermedi. La scelta dei percorsi più appropriati viene detta instradamento (routing). Ci sono due tipi di algoritmi di instradamento: statici, che includono il routing basato sul percorso più corto e il flooding, e dinamici che includono il routing basato sul vettore delle distanze e quello basato sullo stato dei collegamenti.

Un algoritmo statico è quello di flooding, in cui ogni pacchetto in arrivo è inviato a tutte le linee tranne a quella da cui proviene. Logicamente questo meccanismo genera un elevato numero di pacchetti duplicati; anzi teoricamente il numero è infinito, a meno che non si prendano delle misure per attenuare il processo. Una possibile tecnica è usare un contatore di salti inserito nell'intestazione di ogni pacchetto e decrementare il suo valore ad ogni salto, in modo da scartare il pacchetto quando il contatore raggiunge lo zero. Idealmente il valore iniziale da assegnare al contatore dei salti dovrebbe corrispondere alla lunghezza del percorso dall'origine alla destinazione. Se il trasmettitore non conosce la lunghezza del percorso, può assegnare al contatore il valore che rappresenta il caso peggiore, vale a dire il diametro dell'intera sottorete. Una tecnica di moderazione alternativa tiene traccia dei pacchetti trasmessi nel flood, in modo da evitare una seconda ritrasmissione. Per ottenere questo risultato, il router d'origine deve inserire in ogni pacchetto che riceve dai suoi host un numero di sequenza. Ogni router ha quindi bisogno di una lista per ciascuno dei router sorgenti, che indichi i nu-

meri di sequenza già trasmessi da quella di origine. Se appare nella lista, il pacchetto in arrivo non viene trasmesso. Per impedire una crescita illimitata ogni lista è integrata con un contatore k, in modo da indicare che sono stati visti tutti i numeri di sequenza fino a k. Una variante è il flooding selettivo nel quale i router trasmettono i pacchetti solo attraverso le linee che vanno approssimativamente nella direzione giusta, questo ha senso a meno che la topologia non sia particolarmente strana.

Nella maggior parte delle applicazioni l'algoritmo di flooding non è molto funzionale, ma comunque ha alcuni utilizzi pratici. Per esempio nelle applicazioni militari, dove un gran numero di router può saltare in aria in un istante, la grande robustezza dell'algoritmo di flooding è molto desiderabile, perchè se esiste una linea funzionante riesce a trovarla. Nelle applicazioni database distribuite a volte è necessario aggiornare simultaneamente tutti i database; in questo caso può essere utile adottare l'algoritmo di flooding, che è quindi adatto alle trasmissioni broadcast. Infine, l'algoritmo di flooding essendo che sceglie tutti i percorsi, tra di essi ci sarà anche quello più corto e quindi può essere usato come metrica per gli altri algoritmi. Inoltre esso richiede poco in termini di configurazione: ogni router deve solo conoscere i propri vicini.

10 Choke packet

Nello strato di rete si possono creare delle congestioni e il choke packet è una tecnica per risolverle. Un modo per gestire una congestione è notificare all'host sorgente di rallentare. Il router invia un choke packet all'host. Dentro al choke packet c'è la destinazione trovata nel pacchetto originale. Infine il pacchetto originale viene etichettato, in modo da impedire la generazione di altri choke packet lungo il percorso, e poi è inoltrato nel solito modo. Quando riceve il choke packet, l'host sorgente deve ridurre il traffico inviato alla destinazione specificata, ad esempio del 50%. Poiché altri pacchetti che puntano alla stessa destinazione probabilmente sono già in viaggio e genereranno altri choke packet, per un intervallo di tempo prefissato l'host li ignorerà. Trascorso questo periodo di tempo se arriva un altro choke packet allora significa che la linea è ancora congestionata perciò l'host riduce il flusso un altro po'. Se non arriva più nessun choke packet l'host può aumentare nuovamente il flusso; questo viene fatto adottando piccoli incrementi per scongiurare un'altra congestione. L'effetto retroattivo implicito di questo protocollo può aiutare a impedire la congestione, rallentando il flusso dei dati solo quando si presenta un problema.

Una variante è il choke packet hop-by-hop. Essa è utile sopratutto ad alte velocità o sulle lunghe distanze, perché in quei casi l'invio di choke packet all'host sorgente non funziona bene poiché la reazione è lenta: molti pacchetti possono essere trasmessi dopo la segnalazione della congestione. In questo approccio alternativo il choke packet non ha effetto solo sull'host destinatario ma su tutti i router attraversati: quindi tutti i router appena ricevono il choke packet riducono il flusso destinato al router che ha generato il pacchetto. Lo schema hop-by-hop ha l'effetto di alleviare rapidamente la congestione nel punto dove si forma, al prezzo di un incremento dell'utilizzo dei buffer di trasmissione nelle parte del percorso precedente. In questo modo la congestione può essere stroncata sul nascere senza perdere alcun pacchetto.

11 Leaky e Token bucket

Ogni flusso di pacchetti ha delle esigenze diverse in base all'applicazione per cui sono usati. Queste esigenze sono caratterizzate da quattro parametri primari: affidabilità, ritardo, jitter e banda. Insieme questi parametri determinano la QoS (Quality of Service), ossia la qualità del servizio richiesta dal flusso. Per ottenere una buona QoS si utilizzano varie tecniche tra cui il sovradimensionamento della rete, l'utilizzo di buffer e l'impiego di tecniche di modellazione del traffico (traffic shaping). Quest'ultima tecnica fluidifica il traffico lato server, invece che lato client, cioè regola la velocità media della trasmissione dei dati, questo è molto importante per i dati in tempo reale come le connessioni audio e video.

Un algoritmo di traffic shaping è il leaky bucket dove ogni host è collegato alla rete da un'interfaccia che contiene una coda interna finita, che può essere rappresentata come un secchio che ha un buco sul fondo e che quindi perde: qualunque sia la velocità d'ingresso dell'acqua (i dati che l'host vuole inviare), l'efflusso (i dati effettivamente inviati) avrà una velocità costante p quando ci sarà acqua nel secchio, e 0 quando il secchio sarà vuoto. Inoltre se il secchio è pieno, l'acqua aggiuntiva versata nel contenitore si riverserà all'esterno e andrà perduta (un pacchetto viene scartato se vuole entrare nella coda ed è piena). In questo algoritmo l'host è autorizzato a inserire nella rete un solo pacchetto per ciclo di clock. Il leacky bucket quindi trasforma un flusso irregolare di pacchetti provenienti dai processi dell'utente dell'host in un flusso regolare di pacchetti immesso nella rete, appianando i picchi e riducendo enormemente le possibilità di congestioni.

L'algoritmo leaky bucket impone un rigido modello di output a velocità media, che non segue la variabilità del traffico. Per molte applicazioni è meglio permettere all'output di accelerare un po' quando arrivano grandi

raffiche di dati. Il token bucket invece lo permette ed è infatti più flessibile. In questo algoritmo ogni pacchetto deve distruggere un token per poter essere inviato. Se non ci sono token disponibili un pacchetto non può essere spedito e deve attendere. Nel secchio ci sono i token che sono generati da un clock ogni tot secondi. I token possono essere accumulati fino alla capienza massima del secchio n. Questa proprietà permette di trasmettere raffiche composte da un massimo di n pacchetti, dando perciò una risposta più veloce a picchi improvvisi in ingresso, ma generando un flusso di output meno regolare. Inoltre il token bucket quando il secchio si riempe completamente non scarta mai i pacchetti, ma i token. Si può ottenere un traffico più uniforme inserendo un leacky bucket dopo il token bucket. La velocità del leacky bucket dovrebbe essere più alta di quella del token bucket ma più bassa della velocità massima della rete.

12 CIDR

IP è utilizzato intensamente da decenni e ha funzionato estremamente bene, come dimostra la crescita esponenziale di Internet. Sfortunatamente, IP sta diventando rapidamente vittima della sua stessa popolarità: sta esaurendo gli indirizzi. Inizialmente esistevano più di 2 miliardi di indirizzi che erano ritenuti più che sufficienti tant'è che negli anni 80 gli esperti pensavano fosse impossibile solo arrivare ad avere 100.000 reti, ciò invece avvenne nel 1996. Il problema di sottostimare il numero di reti fu notevolmente peggiorato dal sistema di suddivisione basato sulle classi che ha sprecato milioni di indirizzi. Il problema è stato che sono stati distribuiti moltissimi indirizzi di classe B: gli indirizzi di classe A con 16 milioni di indirizzi sono troppo grandi per un azienda, ma una rete di classe C da 256 indirizzi è considerata troppa piccola se si prendono in considerazione eventuali espansioni future, peccato che però un indirizzo di classe B con 65.000 indirizzi è di gran lunga troppo grande per la maggior parte delle aziende e che quindi la maggior parte degli indirizzi così assegnati andranno sprecati: delle ricerche hanno dimostrato che più della metà di tutte le reti di classe B hanno meno di 50 host. Sarebbe stato meglio creare delle reti classe C più grandi (1022 host per rete) in modo che le aziende non richiedessero le reti di classe B. È però ingiusto criticare i progettisti di Internet per questo: all'inizio Internet era una rete di ricerca che collegava solo le principali università degli Stati Uniti: nessuno pensava che internet sarebbe diventato un sistema di comunicazione di massa. La soluzione implementata per dare a Internet un po' di respiro si chiama CIDR (Classless InterDomain Routing). L'idea di fondo di CIDR è assegnare gli indirizzi IP rimanenti in blocchi di dimensioni variabili, senza tener conto delle

classi. Questo permette di assegnare gli indirizzi secondo le reali esigenze. L'abbandono delle classi però rende l'inoltro più complicato. C'è una singola tabella di routing per tutte le reti costituita da una matrice di valori nella forma (indirizzo IP, maschera di sottorete, linea di output). Quando arriva un pacchetto, il router prima di tutto estrae il suo indirizzo IP di destinazione; poi esamina la tabella di routing voce per voce, mascherando l'indirizzo di destinazione e confrontandolo con le voci della tabella. In particolare viene svolta un'operazione di AND tra l'indirizzo IP del pacchetto e la maschera di sotto rete dell'indirizzo memorizzato nel router. Se il risultato dell'AND è uguale all'indirizzo base (cioè indirizzo IP della rete di destinazione senza specificare gli host) allora il pacchetto è inviato lungo la linea associata. È possibile che più voci (con diverse lunghezze di maschera di sotto rete) coincidano con il valore cercato, in questo caso è utilizzata la maschera più lunga. Per una gestione efficiente del CIDR, si deve ridurre il numero di voci della tabella e quindi le sue dimensioni. Questo è raggiunto tramite una tecnica chiamata *aggregates entries*: due blocchi che devono essere indirizzati allo stesso router possono essere combinati in un'unica entrata se hanno un prefisso in comune. Le *aggregates entries* funzionano perchè nella maggiorparte dei casi un router ha non più di otto linee in uscita.

13 NAT

Gli indirizzi IP si stanno esaurendo, la soluzione a lungo termine è che tutta Internet passi a IPv6, protocollo che adotta indirizzi a 128 bit. Questa transizione sta procedendo lentamente, e ci vorranno anni prima che si completi. Di conseguenza serviva una soluzione attuabile in tempi brevi che servisse da pezza: questa soluzione si chiama NAT (Network Address Translation). L'idea di base di NAT è assegnare a ogni azienda un singolo indirizzo IP per il traffico di Internet. Dentro l'azienda, ogni computer riceve un indirizzo IP unico utilizzato per l'instradare il traffico interno alla rete locale, ma quando un pacchetto lascia l'azienda e si dirige verso l'ISP, viene eseguita una traduzione di indirizzo. Per rendere fattibile questo schema, sono stati dichiarati privati tre intervalli di indirizzi IP, che le aziende possono utilizzare internamente come preferiscono. L'unica regola è che nessun pacchetto contenente questi indirizzi può apparire su Internet. I tre intervalli sono:

10.0.0.0 – 10.255.255.255/8 (16 milioni di indirizzi)

172.16.0.0 – 172.31.255/12 (1 milione di indirizzi)

192.168.0.0 – 192.168.255.255/16 (65 mila indirizzi)

L'unico problema è che quando arriva una risposta da Internet il dispositivo come sceglie l'indirizzo di destinazione interno? In linea di principio si sarebbe potuta creare una nuova opzione che conservasse il vero indirizzo sorgente: la gestione di questo meccanismo avrebbe però richiesto la modifica del codice IP di tutti i computer di Internet, quindi non sarebbe stata rapida. I progettisti di NAT hanno osservato che la maggior parte dei pacchetti IP trasporta carichi utili TCP o UDP, dei protocolli che contengono una porta sorgente e una porta di destinazione. Le porte sono dei numeri interi lunghi 16 bit che indicano dove inizia e finisce la connessione TCP e UDP. Questi campi dati sono quelli che permettono di far funzionare NAT. L'utilizzo del campo *source port* permette di risolvere il problema della mappatura. Ogni volta che un pacchetto diretto verso l'esterno raggiunge il dispositivo NAT, l'indirizzo sorgente è sostituito dall'indirizzo IP dell'azienda. Inoltre il *source port* è sostituito da un indice che punta alla tabella di traduzione da 65 mila voci del dispositivo NAT. Questa voce di tabella contiene l'indirizzo IP originale e la porta sorgente iniziale. Quando il pacchetto raggiunge il dispositivo NAT della rete di destinazione, la *source port* nell'intestazione TCP(o UDP) viene estratta e utilizzata come indice nella tabella di mappatura del dispositivo NAT. Infine dalla voce individuata, il dispositivo estrapola l'indirizzo IP interno e la *source port* originale.

I puristi delle reti dicono che NAT sia un abominio per quattro ragioni. La prima è che NAT viola il modello di IP, il quale dice che ogni indirizzo dovrebbe corrispondere a una sola macchina. La seconda è che rompe il modelo end-to-end secondo il quale qualsiasi host può inviare un pacchetto a un altro host; questo perché la mappatura del dispositivo di NAT è fatta dai pacchetti che escono, quindi i pacchetti in entrata non possono essere accettati se non dopo quelli in uscita. La terza è che NAT cambia la natura di Internet, perché il dispositivo di NAT deve mantenere le informazioni per ogni connessione che gestisce: una proprietà di una rete orientata alle connessioni, cosa che Internet non è. Questo crea un singolo punto di fallimento peché se crasha il dispositivo di NAT tutte le connessioni che gestisce saranno perse. La quarta è che viola la regola più fondamentale di indipendenza tra gli strati di rete: lo strato k non deve fare assunzioni su cosa lo strato k+1 mette nel suo carico. Così facendo se per qualche motivo TCP(o UDP) dovesse cambiare, NAT non funzionerebbe più; stesso risultato se degli utenti decidessero di usare altri protocolli di rete diversi da TCP(o UDP).

14 ARP

Anche se ogni macchina di Internet ha un indirizzo IP, in realtà questi non possono essere utilizzati per inviare pacchetti perché l'hardware che opera sullo strato data link non comprende gli indirizzi Internet. Oggi la maggior parte degli host nelle aziende e nelle università è collegata a una LAN attraverso una scheda di rete che conosce solo indirizzi LAN. La domanda che sorge spontanea è: in che modo gli indirizzi IP vengono associati agli indirizzi dello strato data link (per esempio agli indirizzi Ethernet)? Quasi tutte le macchine di Internet usano un protocollo chiamato ARP (Address Resolution Protocol). ARP funziona nella maniera così descritta.

Si supponga che sia arrivato al router collegato alla LAN un pacchetto IP e che il software IP scopra che la destinazione del pacchetto si trovi sulla stessa rete. Il problema è che i computer sulla rete LAN sono identificati da un indirizzo LAN e non si sa a quale indirizzo LAN corrisponda quale IP. Il problema potrebbe essere risolto mediante un file di configurazione, collocato da qualche parte nel sistema, che associ gli indirizzi IP agli indirizzi LAN. Questa soluzione è certamente possibile, ma in un'azienda con migliaia di macchine l'aggiornamento di questi file sarebbe soggetto a errori e richiederebbe molto tempo. La soluzione migliore è che sia emesso sulla LAN un messaggio broadcast che chiede: chi è il proprietario dell'indirizzo IP x? La trasmissione arriva a tutte le macchine che controllano il proprio indirizzo IP, ma solo la stazione che ha quell'indirizzo IP risponde e manda il proprio indirizzo LAN. Il gestore del sistema non deve fare altro che assegnare a ogni macchina un indirizzo IP e decidere le maschere di sottorete, ARP si occupa del resto. Una volta eseguita l'operazione ARP, il computer memorizza in cache il risultato, caso mai dovesse ricontattare lo stesso computer. Varie migliorie possono eseere fatte: la prima consiste nel portare nel messaggio broadcast, in piggyback, anche la propria corrispondenza IP, indirizzo strato colleganto dati. Inoltre si può far si che anche la macchine non direttamente interessate si salvino le configurazioni scambiate, questo è possibile perché la comunicazione è broadcast. Un ulteriore miglioramento si ottiene facendo in modo che ogni computer trasmetta in broadcast la propria associazione durante l'accensione. Quest'operazione non genera alcune risposta normalmente, ma se arriva, ciò significa che due computer hanno ricevuto lo stesso indirizzo IP. Questo evento viene riferito all'amministratore. Per consentire la modifica delle associazioni, dovute ad esempio a schede Ethernet guaste sostituite con schede nuove (con dei nuovi indirizzi Ethetnet), le voci nella cache ARP dovrebbero scadere dopo pochi minuti.

15 UDP

Internet possiede due protocolli principali nello strato trasporto, un protocollo senza connessione (UDP), e uno orientato alla connessione (TCP). UDP offre alle applicazioni un modo per inviare datagrammi IP incapsulati senza dover stabilire una connessione. UDP trasmette segmenti costituiti da un'intestazione di 8 byte (64 bit) seguita dal carico utile. UDP è simile a IP, ma ha il "bonus" di avere le porte. L'intestazione ha quattro campi dati: *source port*, *destination port*, *UDP length* e *UDP checksum*, quest'ultimo facoltativo. Le due porte servono a identificare i punti finali all'interno dei computer di origine e destinazione. Quando arriva un pacchetto UDP, il suo carico utile è consegnato al processo associato alla porta di destinazione. La porta di origine serve principalmente quando si deve inviare una risposta all'origine. Copiando il campo *source port* dal segmento di ingresso nel campo *destination port* del segmento in uscita, il processo che invia la risposta può specificare il processo sulla macchina destinataria che deve riceverla. Il campo *UDP lenght* include l'intestazione di 8 byte e i dati. *UDP checksum* è facoltativo.

UDP non si occupa del controllo del flusso, del controllo degli errori o della ritrasmissione dopo la ricezione di un segmento errato. Questi processi sono lasciati ai processi utente. UDP si occupa invece di fornire un'interfaccia al protocollo IP, con la caratteristica aggiunta del demultiplexing di più processi utilizzando le porte. Questo è tutto.

Un'area in cui UDP è particolarmente utile riguarda le situazioni client/server: il codice è semplice e richiede meno messaggi rispetto a un protocollo che necessita dell'impostazione iniziale. Un'applicazione che utilizza UDP è DNS. UDP è un protocollo semplice con alcuni utilizzi di nicchia, come le interazioni client/server e il multimediale, ma per la maggior parte delle applicazioni Internet è necessaria una consegna affidabile in sequenza che UDP non può fornire, pertanto il motore di Internet è considerato essere TCP.

Introduzione alla sicurezza nelle reti

I problemi di sicurezza delle reti si possono dividere in quattro aree interconnesse fra loro: segretezza, autenticazione, non-ripudio e controllo dell'integrità. La segretezza si occupa di mantenere le informazioni fuori dalla portata dei non autorizzati. L'autenticazione si occupa di stabilire l'identità del soggetto con si sta comunicando, prima di rivelare informazioni sensibili. Il termine non ripudio riguarda le firme e tratta del dimostrare che un utente abbia davvero mandato un messaggio. L'integrità si occupa di stabilire se il

messaggio ricevuto è identico a quello spedito, cioè non è stato modificato mentre era in transito.

La sicurezza può essere implementata (in modi diversi) in tutti gli strati di rete, ed eccezion fatta per lo strato fisico, tutti questi modi si basano su principi crittografici. Crittografia deriva da due parole di greco antico che significano "scrittura segreta". Con cifrario s'intende una trasformazione carattere per carattere(o bit per bit), senza considerare la struttura linguistica del messaggio. Al contrario, un codice rimpiazza ogni parola con un'altra parola o simbolo. I codici non sono più utilizzati, ma in passato hanno avuto impieghi importanti: ad esempio durante la SGM, nel Pacifico, gli Stati Uniti impiegarono indiani Navajo che comunicavano tra loro con le parole Navajo che definiscono termini militari. I Giapponesi non riuscirono mai a interpretare quel codice data l'estrema complessità della lingua Navajo e il fatto che era a loro sconosciuta.

Storicamente la crittografia è stata usata sopratutto dai militari. Il notevole volume di messaggi non rendeva fattibile impiegare una squadra di specialisti di alto livello, quindi un tempo uno dei limiti principali della crittografia era la capacità degli addetti alla codifica di riuscire a compiere le operazioni necessarie, spesso in campo di battaglia e con equipaggiamento minimo. Un altro vincolo era la necessità di poter cambiare velocemente da un metodo crittografico ad un altro, in quanto un addetto alla codifica poteva essere catturato dal nemico.

Dati questi requisiti si arrivò all'idea che i messaggi da cifrare, detti **testo in chiaro**, sono trasformati tramite una funzione parametrizzata da una **chiave**. L'output del processo di cifratura, noto come **testo cifrato**, viene generalmente trasmesso tramite un messaggero o la radio. Assumiamo che il nemico, o intruso, ascolti il testo cifrato, al contrario del destinatario legittimo, l'intruso non conosce la chiave di decifrazione e quindi non riesce facilmente decifrare il testo cifrato. In alcuni casi l'intruso, oltre ad ascoltare il canale di comunicazione (intruso passivo), riesce anche a registrare i messaggi per poi ritrasmetterli aggiungendo delle informazioni, oppure per modificare i messaggi legittimi prima che raggiungano il destinatario (intruso attivo). L'arte di decriptare i cifrari, chiamata criptoanalisi, e l'arte di inventarli (crittografia) sono note con il nome collettivo di crittologia. Decriptare è l'attività di decifrazione da parte del criptoanalista (intruso), mentre la decifratura è l'operazione legittima di lettura di un messaggio cifrato. Detti C il codice criptato(Crypted), P il testo in chiaro (Plain), E la funzione di cifratura (Encryption), D la funzione di decifrazione (Decryption) e K la chiave (Key) vale:

$$C = E_k(P)$$

$$P = D_k(C)$$
$$D_k(E_k(P)) = P$$

Un principio fondamentale della crittografia è il principio di Kerckhoff. Esso afferma che bisogna sempre assumere che il criptoanalista conosca il metodo usata per la cifratura e la decifrazione. Lo sforzo necessario di inventare, testare e installare un nuovo algoritmo ogni volta che quello vecchio è compromesso ha sempre reso impraticabile la strada di tenere segreto l'algoritmo della cifratura. Le chiavi invece vanno tenute segrete, cambiare la chiave può essere fatto anche frequentemente senza creare problemi.

16 Cifrari a sostituzione e a trasposizione

In un cifrario a sostituzione, ogni lettera o gruppo di lettere viene rimpiazzato da un'altra lettera o gruppo di lettere per mascherare il messaggio, conservano quindi l'ordine dei simboli del testo in chiaro. Uno dei cifrari più antichi che si conoscano è il cifrario di Cesare, che consiste nello spostare circolarmente le lettere dell'alfabeto del testo cifrato di k lettere; k viene considerata la chiave del metodo. Il miglioramento successivo consiste nel far sì che ognuna delle lettere corrisponda ad altre lettere, tale sistema viene chiamato sostituzione monoalfabetica; la chiave è la stringa di lettere che corrisponde all'intero alfabeto. In questo sistema ci sono $26! = 10^{26}$ possibili chiavi, quindi provare tutte le chiavi non è un attacco fattibile, però essendo che viene conservato l'ordine dei simboli, il cifrario può essere forzato facilmente grazie alle proprietà statistiche dei linguaggi naturali. Per farlo si procede contando la frequenza di tutti i simboli presenti e associando al simbolo più comune la lettera più comune dell'alfabeto, al secondo più comune la seconda lettera più comune è così via, passando poi ai digrammi e ai trigrammi. Il criptoanalista ricostruisce per tentativi il testo in chiaro. Un altro approccio consiste nell'indovinare una parola che probabilmente si trova dentro al testo e cercare di capire a che parte di testo criptato corrisponde grazie ad alcune proprietà della parola: ad esempio una vocale che si ripete intervallata da altre quattro lettere.

I cifrari a trasposizione riordinano le lettere ma non le mascherano. Un tipico cifrario a trasposizione è quello colonnare che ha come chiave una parola senza lettere ripetute. Lo scopo della chiave è quello di numerare le colonne: la colonna numero 1 è quella sotto la lettera della chiave più vicina all'inizio dell'alfabeto, e così via. Il testo in chiaro viene scritto orizzontalmente, per righe, fino a riempire la matrice. Il testo per essere cifrato viene letto per colonna, cominciando con quella che ha la lettera chiave più bassa. Il criptoanalista per forzare un cifrario a trasposizione, per prima cosa,

deve accorgersi che si tratta proprio di un cifrario a trasposizione. Questo viene fatto confrontando la frequenza dei caratteri del testo cifrato rispetto a quella del linguaggio naturale: se coincidono si tratta di cifrario a trasposizione. Il passo successivo è fare delle ipotesi sul numero di colonne. Questo si può fare indovinando dal contesto una parola che con buona probabilità appare nel testo. Se due lettere di questa parola normalmente sono distanti k caratteri, ma nel testo cifrato le trovo adiacenti, si può ipotizzare che la chiave sia lunga k caratteri. Il passo conclusivo consiste nel trovare l'ordine delle colonne. Quando il numero delle colonne è piccolo, ognuna delle possibili coppie di colonne può essere esaminata per vedere se le frequenze di occorrenza dei suoi digrammi corrispondono a quelle tipiche della lingua in cui si suppone sia scritto il messaggio. Si suppone che la coppia con il miglior risultato in questo test sia posizionata correttamente. A questo punto si procede con le colonne rimanenti per identificare la colonna successiva alla coppia, che sarà quella che permette di avere la miglior distribuzione di trigrammi. Questo processo viene portato avanti finché non si identifica un possibile ordinamento delle colonne.

17 Blocchi monouso (One-Time Pads)

Costruire un cifrario imbattibile è molto facile e la tecnica per farlo è nota da decenni. Per prima cosa si converte il testo in chiaro in una stringa di bit, per esempio usando la rappresentazione ASCII per i caratteri. Poi si prende una chiave formata da una stringa di bit generati a caso della stessa lunghezza della stringa di bit che rappresenta il testo in chiaro. Infine si calcola lo XOR (OR esclusivo) delle due stringhe, bit per bit. Il testo cifrato non può essere forzato, in quanto per ogni pezzo sufficientemente grande del testo cifrato, ogni lettera apparirà con la stessa frequenza, lo stesso vale per i digrammi, i trigrammi, ecc. Questo metodo, noto come blocco monouso (one-time pad), è immune a tutti gli attacchi presenti e futuri indipendentemente dalla potenza di calcolo a disposizione: perché un messaggio così cifrato non contiene nessuna informazione in quanto contiene tutti i possibili testi in chiaro di una data lunghezza con uguale probabilità. I blocchi monouso sono ottimi in teoria, ma hanno diversi svantaggi nella pratica. Per cominciare, la chiave non può essere imparata a memoria, quindi sia il mittente sia il destinatario devono averne una copia. Le flash paper erano usate per quello scopo: una volta accartocciate prendevano fuoco. Inoltre come il nome suggerisce le chiavi di questo metodo devono essere usate una sola volta, altrimenti un malintenzionato potrebbe fare lo XOR di due messaggi cifrati con la stessa chiave e ottenere lo XOR di due messaggi

in chiaro, base di partenza per provare a decifrare i due messaggi. Infine la quantità dei dati che può essere trasmessa è limitata dalla lunghezza della chiave: è quindi essenziale avere una scorta di chiavi pronte all'uso. Questo metodo è usato nei contesti dove è richiesta la massima segretezza, è quindi usato da organizzazioni come la CIA e FBI, nonché nelle linee rosse che collegano i capi di governo degli stati più importanti come Stati Uniti, Cina e Russia. Ha lo svantaggio che se a un certo punto mittente e destinatario perdono la sincronizzazione il resto dei dati sarà illeggibile.

18 DES e DES triplo

La crittografia moderna usa le stesse idee base della crittografia classica (come la trasposizione e la sostituzione), ma con un enfasi diversa. Tradizionalmente i crittografi hanno sempre usato degli algoritmi semplici. Al giorno d'oggi è vero il contrario: l'idea è quella di rendere l'algoritmo di cifratura così complesso che anche se il criptoanalista avesse a disposizione un'ampia scelta di testo cifrato a sua scelta, non riuscirebbe comunque a farci nulla senza avere la chiave crittografica. Sono detti algoritmi a chiave simmetrica gli algoritmi di cifratura che usano la stessa chiave per cifrare e decifrare i messaggi. I cifrari a blocco sono quelli che prendono un blocco di n bit dal testo in chiaro e li trasformano, usando la chiave, in n bit cifrati. DES(Data Encription Standard) sviluppato nel 1977 da IBM è un cifrario a blocchi a chiave simmetrica. DES fu scelto dal governo degli Stati Uniti come standard crittografico nelle comunicazioni non riservate. Il testo in chiaro viene diviso in blocchi da 64 bit che verranno cifrati in altrettanti blocchi da 64 bit. L'algoritmo è parametrizzato da una chiave da 56 bit e ha 19 stadi distinti. Ogni stadio fa uso di una combinazione di P-box e S-box. P-box (scatola di permutazione) è usata per fare trasposizioni di bit, mentre una S-box (scatola di sostituzione) sostituisce certi bit con altri bit. Sia P-box che S-box sono parametrizzati dalla chiave. DES è considerato obsoleto data la chiave di soli 56 bit, poiché oggi esistono macchine abbastanza potenti da trovare la chiave con la forza bruta. IBM ha quindi inventato DES triplo, nel quale vengono usate due chiavi da 56 bit (112 bit in totale sono considerati sufficienti) e tre stadi. Nel primo stadio, il testo viene cifrato con DES usando la prima chiave nel solito modo. Nel secondo stadio, DES viene usato in modalità decifrazione usando la seconda chiave. Infine nel terzo stadio viene fatta un'altra cifratura con la prima chiave. La modalità cifra, decifra, cifra (EDE) al posto di una tripla cifratura (EEE) è stata scelta per compatibilità all'indietro con i sistemi DES a chiave singola. EDE e EEE dal punto di vista

crittografico sono egualmente forti. Nel 2001 DES e la variante più potente DES triplo sono stati sostituiti da AES (Advanced Encryption Standard).

19 Modalità di cifratura

I cifrari a blocco spezzano il testo del messaggio in chiaro in blocchi di n bit e li trasformano in n bit cifrati usando la chiave.

ECB

La modalità più ovvia di cifratura è quella di cifrare (ad esempio con DES o AES) con la stessa chiave uno dopo l'altro i blocchi in cui si è suddiviso il testo. Questa tecnica è nota come ECB (Elettronic Code Book), però viene di rado usata perché un intruso può cambiare l'ordine dei blocchi. Questo attacco è reso ancora più facile dal momento che l'attaccante non ha bisogno di decifrare il messaggio. L'attacco è efficace perché non è evidente quando arriva il messaggio che esso è stato manomesso. Per evitare questi tipi di attacchi si usano altre modalità che collegano tutti i blocchi cifrati in diversi modi. Così facendo, quando si tenta di rimpiazzare dei pezzi di testo cifrato con altri, l'effetto in fase di decifrazione sarà quello di avere dei dati senza significato a partire dal punto in cui è stata operata la sostituzione.

Cipher block chaining

Nella modalità cipher block chaining (collegamento dei blocchi cifrati) ogni blocco di testo in chiaro viene messo in XOR con il precedente blocco cifrato prima di eseguire la cifratura vera e propria. Per il primo blocco lo XOR viene calcolato con un blocco di dati casuali detto IV (Initialization Vector) che è trasmesso (in chiaro) insieme al testo cifrato. La decifrazione usa di nuovo lo XOR per rovesciare il processo. Questa modalità ha il vantaggio che lo stesso testo in chiaro dà origine a diversi testi cifrati a seconda della sua posizione, questo rende la criptoanalisi più difficile.

Stream cipher

Questa modalità funziona cifrando un vettore di inizializzazione(IV) con una chiave crittografica per ottenere il primo blocco in uscita. Quest'ultimo viene cifrato per produrre un secondo blocco in uscita, quindi si procede analogamente con il terzo, ecc. La sequenza (di lunghezza arbitraria) di blocchi cifrati in uscita, chiamata keystream (flusso delle chiavi), viene utilizzata come blocco monouso e applicata in XOR sul testo in chiaro per ottenere

il testo cifrato. Si noti che l'IV è usato solamente nel primo passo e che il keystream è indipendente dai dati e quindi può essere calcolato in anticipo. La decifrazione viene eseguita generando lo stesso keystream dal lato del ricevente. Questa modalità ha il vantaggio che, poiché il keystream dipende solo da IV e dalla chiave, non è sensibile agli errori di trasmissione del testo cifrato. Quindi, un errore di un bit nel testo cifrato trasmesso genera solamente un errore di un bit nel testo decifrato. È essenziale che nella modalità stream cipher non venga mai riutilizzata la coppia (chiave, IV), perché questo vorrebbe dire generare lo stesso keystream. L'uso ripetuto dello stesso keystream espone il testo cifrato ad attacchi di tipo keystream riutilizzato. Immaginiamo che due blocchi di testo in chiaro, P_0 e Q_0, siano cifrati con lo stesso keystream k_0. Un intruso che riesca a catturare entrambi i blocchi cifrati può trovare lo XOR dei due messaggi in chiaro perché $(P_0\ XOR\ K_0)XOR(Q_0\ XOR\ K_0) = P_0\ XOR\ Q_0$. Se uno dei due blocchi è noto o può essere indovinato, l'intruso ha trovato il valore dell'altro blocco. Inoltre lo XOR dei due testi in chiaro può essere attaccato usando le proprietà statistiche dei linguaggi.

Counter

Si supponga che un file sia trasmesso attraverso la rete e poi memorizzato su disco in formato cifrato; ciò è utile in caso il computer del ricevente sia a rischio di furto. Spesso si accede ai file su disco in un ordine non sequenziale. Con un file cifrato usando il cipher block chaining, l'accesso casuale a un blocco richiede che prima siano decifrati tutti i blocchi precedenti, un'operazione troppo onerosa. Per questo motivo è stata inventata la modalità contatore. In questa modalità non viene cifrato direttamente il testo; si cifra invece un vettore di inizializzazione, e il risultato è messo in XOR con il testo il chiaro. Incrementando di 1 il vettore d'inizializzazione a ogni blocco, diventa facile riuscire a decifrare un blocco in qualunque posizione si trovi, senza dover decifrare i suoi predecessori. Questa modalità può essere soggetta ad attacchi di keystream riutilizzato, quindi vettore d'inizializzazione e chiave vanno scelti indipendentemente e in modo casuale.

20 Modi di attacare DNS

DNS Amplification Attack

È un attacco di tipi DDoS (Distribuited Denial of Service). In questo attacco si usano i server DNS per attaccare una vittima. L'attaccante invia pacchetti IP con il campo dell'indirizzo sorgente falsificato (spoofed) a un server DNS.

L'indirizzo falsificato è quello dello della vittima. Ogni pacchetto inviato fa una richiesta al DNS; la richiesta viene scelta in modo da comportare la più grande risposta possibile. Il fatto che un pacchetto piccolo possa generare una risposta di grande dimensioni, fa si che l'attacco sia amplificato, da questo il nome dell'attacco. Il DNS dopo aver ricevuto la richiesta, ignaro dell'attacco in corso, manda la risposta all'indirizzo IP che crede essere la vera sorgente del pacchetto. La vittima così è travolta dalla quantità di pacchetti ricevuti, esaurisce le proprie risorse e non riesce più a garantire il servizio agli utenti non malintenzionati.

DNS Flood Attack

È un attacco di tipo DDoS. Il destinatario di questo attacco è un server DNS: si cerca di far finire le risorse del server, in modo che non possa rispondere alle richieste legittime. Questo viene ottenuto facendo arrivare al server una valanga (flood) di pacchetti UDP. I server DNS sono le "tabelle di internet": danno cioè la corrispondenza tra indirizzo IP e nome del dominio. Se si butta giù un DNS, è come se si buttasse giù un pezzo di Internet!

DNS Spoofing

È un attacco contro i dati di un server DNS. Esso consiste nell'ingannare un DNS per installarci un indirizzo IP falso. La cache che contiene un indirizzo falsificato si dice poisoned cache. Un attaccante, diciamo Trudy, vuole ingannare il DNS dell'ISP di Alice inviandogli una query per cercare l'indirizzo di Bob. Sfortunatamente, visto che DNS usa UDP, il server non ha modo di sapere chi fornisce la risposta alla query. Se l'ISP di Alice non ha informazioni sull'IP del sito web bob.com, allora passa la richiesta al server top-level per il dominio.com, ma Trudy batte in velocità il server com inviando per risposta il proprio indirizzo IP. La risposta vera verrà scartata perché la query è già stata soddisfatta. Quindi se Alice digiterà bob.com nel browser verrà indirizzata al sito di Trudy e non quello di bob. Per proteggersi da questo attacco è stata messa una pezza: ogni richiesta ha un numero di sequenza per poter controllare che richieste e risposte corrispondano. Trudy può aggirare questa protezione registrando un nuovo dominio. Poi deve chiedere al server DNS di Alice di risolvere qualcosa nel suo sito. Non avendo questo sito nella cache, manda una richiesta verso l'esterno che contiene un numero di sequenza. Questo numero di sequenza viene usato da Trudy, naturalmente incrementato di 1 (ma anche di 2,3 etc per essere sicuri), per mandare una risposta falsificata sul qual è l'indirizzo IP di bob.com. Per proteggersi da questo tipo di attacco è stata messa una seconda pezza: al posto di un

numero di sequenza sequenziale verrà usato un numero casuale. Però anche questa soluzione può essere superata con attacchi massivi e/o paralleli. La soluzione definitiva sarà data da DNSsec, nel quale ogni messaggio ha due campi in più: KEY che contiene la chiave pubblica e SIG che contiene la firma digitale.

21 HMAC

IP nella sua forma originale non comprende nessuna misura di sicurezza, per questo può essere soggetto a IP spoofing: un attacco che si basa sul cambiare indirizzo IP del mittente con uno diverso dal proprio. Per risolvere questo problema è stato sviluppato IPsec. IPsec aggiunge a IP delle intestazioni che gli permettono autenticare i pacchetti e di evitare attacchi di tipo ripetizione grazie a dei numeri di sequenza. Ci sono due tipi di intestazioni: AH (Authentication Header) e ESP (Encapsulating Security Payload), il primo autentica solo il pacchetto, il secondo autentica il messaggio e opzionalmente può cifrare il pacchetto. Entrambi hanno un campo per l'autenticazione chiamato HMAC (Hashed Message Authentication Code). HMAC è un campo a lunghezza variabile che contiene la firma digitale del pacchetto. Per la firma del pacchetto si utilizzano algoritmi a chiave simmetrica, perché quelli a chiave pubblica sono troppo lenti per questo ambito di utilizzo.

Una funzione di message digest prende in input un testo di lunghezza arbitraria e calcola una stringa di bit di lunghezza fissa chiamata hash. Il metodo dell'HMAC consiste nel calcolare l'hash sui contenuti del pacchetto e della chiave **insieme** . Questo è molto più veloce rispetto a calcolare prima l'hash con una funzione di message digest, e poi usare RSA(o altro algoritmo a chiave pubblica) sull'hash prodotto.