

Molti di noi avranno sentito parlare di "pacchetti" di dati che girano nella rete. Perché si parla di pacchetti e non di file o messaggi? Perché i messaggi non vengono inviati tutti interi, ma frammentati in piccoli pezzi. E perché si frammenta il messaggio da spedire?

Rispondiamo al quesito andando per ordine e cominciando dall'inizio.

Una linea digitale, dove vengono inviati bit, se non è ideale (cosa che capita sempre, perché le linee sono reali, non ideali) può introdurre degli errori, di vario tipo; per esempio un bit può andare perso oppure – più facilmente – può invertirsi (da 0 a 1 o viceversa). Come si misura la qualità di una linea di comunicazioni digitale?

### BER

La qualità di una linea di comunicazione può essere espressa mediante la percentuale (tasso, o "rate") di bit errati rispetto a quelli inviati. La definizione di BER - Bit Error Rate - viene espressa così:

$$\text{BER} = \text{bit arrivati errati} / \text{bit trasmessi}$$

Per le linee di oggi, sempre migliori, si parla di BER inferiori a  $10^{-6}$ .

Un altro modo, perfettamente equivalente, di esprimere il BER è questo: "il BER è la probabilità che un bit cambi valore durante la trasmissione".

Se conosciamo la quantità di bit trasmessi, possiamo statisticamente prevedere quanti bit arriveranno errati; basta invertire la formula della definizione.

Esempio: in una trasmissione di  $10^7$  (10 milioni) di bit su una linea che presenta un BER di  $0.5 \cdot 10^{-6}$ , quanti bit risulteranno statisticamente errati?

$$\text{bit errati} = \text{bit trasmessi} * \text{BER} = 10^7 * 0.5 * 10^{-6} = 5$$

Bisogna sottolineare l'aspetto statistico della questione. Non è certo che arriveranno 5 bit errati ma facendo molte trasmissioni di questo tipo si troverebbe mediamente questa quantità di bit errati.

### I codici d'errore

Nelle trasmissioni moderne capita spesso di pretendere che il messaggio inviato (file, mail ecc.) arrivi "integro", cioè perfetto, senza alterazioni, neanche un bit diverso.

Per fare questo si sono introdotti i codici d'errore, che sono usati per verificare se quello che è arrivato coincide con quanto inviato; c'è una branca della matematica che studia i migliori codici d'errore.

Il gioco è questo: con un particolare algoritmo (CRC, checksum ecc.) applicato al messaggio da inviare si determina una stringa di bit – il codice d'errore – che verrà inviato insieme al messaggio; il ricevitore esamina il messaggio, applica lo stesso algoritmo al messaggio e confronta il suo calcolo con il codice d'errore inviato: se coincidono ha la ragionevole certezza che il messaggio sia "integro".

Ottimo. Però che facciamo se un messaggio arriva errato? Beh, lo ritrasmettiamo.

### Trasmissioni impossibili

Giusto, lo ritrasmettiamo. Esaminiamo allora questa situazione: si vuole spedire un messaggio di 1Mb su una linea con BER di  $10^{-6}$ . Che probabilità abbiamo che arrivi errato? Si applichi la formula vista prima:

$$\text{bit errati} = \text{bit trasmessi} * \text{BER} = 10^6 * 10^{-6} = 1$$

Statisticamente 1 bit errato c'è. Il codice d'errore ci segnalerà che il messaggio non è "integro". Ma se arriva anche un solo bit errato siamo nelle condizioni di dover ritrasmettere l'intero messaggio. Rinviemo quindi il messaggio, ma le condizioni si ripetono, anche la replica arriverà errata; il gioco potrebbe andare avanti sperando che, per gli effetti statistici, almeno una volta siamo fortunati; ma... nessuna compagnia di telecomunicazioni ragiona in questo modo, affidandosi al caso. Senza considerare che si trasmette un mucchio enorme di bit. Insomma la trasmissione sarebbe tremendamente inefficiente, perché è statisticamente certo che ogni replica del messaggio arriva errata.

### Il framing

Ecco l'idea per risolvere il problema: non trasmettiamo l'intero messaggio in una sola volta, in un unico blocco, ma in pezzi ("frame"). Prendiamo il messaggio di prima, lungo  $10^6$  bit e lo dividiamo per esempio in 10 parti, ognuna da  $10^6/10=10^5$  bit. Poi spediamo questi 10 frame; quanti di questi arriveranno errati, statisticamente? Solo 1, statisticamente, avrà un bit errato perché abbiamo trasmesso un totale di  $10^6$  bit e il BER vale  $10^{-6}$  (è il calcolo fatto prima). Quindi 9 frame sono validi e 1 no, e dovrà essere ritrasmesso ma... qual è la probabilità che questo frame arrivi errato? È lungo  $10^5$  bit, quindi la probabilità che arrivi errato è minore; infatti:

$$\text{bit errati} = \text{bit trasmessi} * \text{BER} = 10^5 * 10^{-6} = 10^{-1}$$

bisogna interpretare statisticamente il risultato, ma dovrebbe risultare evidente che, a differenza di prima, c'è una buona probabilità che arrivi una replica corretta. Infatti  $10^{-1}$  non è zero, ma ci si avvicina.

Siamo a cavallo.

Quasi.

Infatti: cosa dobbiamo fare per stabilire la dimensione ideale del frame? A un primo ragionamento, verrebbe da dire che più piccoli sono i frame, minore è la probabilità di riceverli errati; questo è corretto ma bisogna considerare che il framing è uno spezzettamento che comporta un certo costo in termini di bit.

### L'overhead in un frame

Se trasmetto un messaggio quello che arriva è un messaggio, stop; se lo suddivido in molti pezzi (frame) il RX dovrà capire di quali e quanto pezzi è composto il messaggio originale; chi mastica queste cose sa che i frame devono essere "delimitati"; magari sarà necessario numerare i pezzi per ricostruire fedelmente il messaggio, e dovrò anche sapere quando il messaggio è finito, ecc.

Inoltre tutti i protocolli usati nelle telecomunicazioni aggiungono dei bit in testa al messaggio (a volte anche in coda) per svolgere adeguatamente il proprio lavoro; questi bit messi in testa vengono chiamati "header" del messaggio.

Questi header appesantiscono la trasmissione con dei bit aggiuntivi. Se spedisco un solo frame avrò un solo header, se spedisco 100 frame avrò 100 header con un "costo" in termini di bit molto più elevato. Ebbene questo è il motivo per cui frame troppo piccoli determinano una comunicazione inefficiente, aumentando notevolmente il peso in bit della trasmissione.

Esamina la figura seguente per capire l'appesantimento determinato dagli header; si fa l'esempio di messaggio trasmesso prima in un solo frame e poi con framing differenti:



Misuriamo l'efficienza

Ecco un esercizio assai interessante: creare un foglio elettronico che, data la dimensione di un messaggio (in bit), il "peso" dell'header (in bit) e il BER della linea, metta in un grafico sulle ascisse la dimensione del frame e sulle ordinate l'efficienza della trasmissione, misurata con la percentuale :  $\text{bit\_inviati} / \text{bit\_del\_messaggio}$