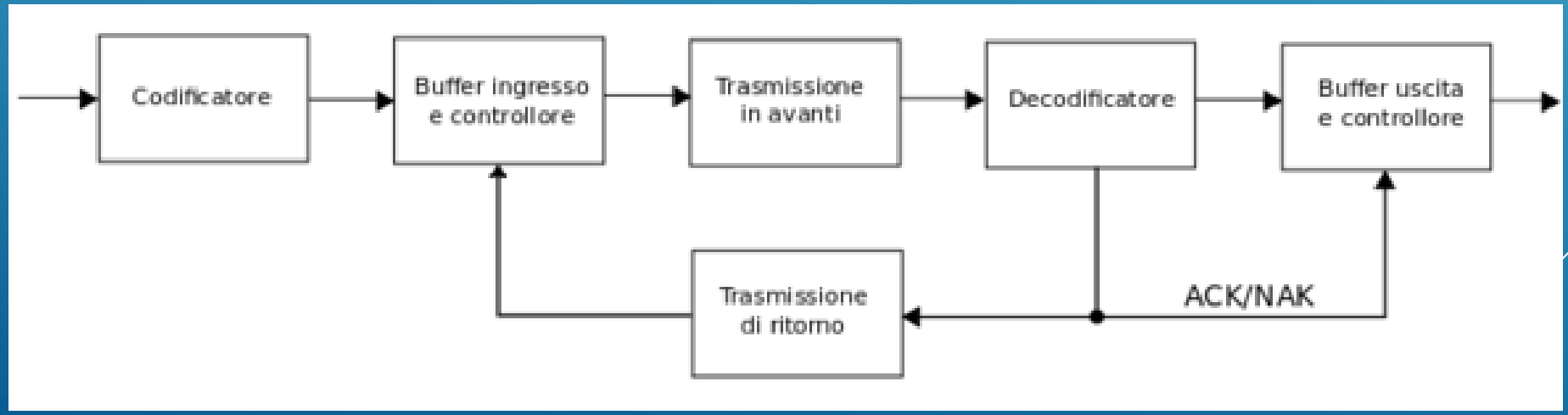
PROTOCOLLO STOP AND WAIT

INTRODUZIONE:

Il protocollo ***stop and wait*** fa parte di un sistema di telecomunicazione definito : ***Automatic Repeat-reQuest(ARQ***) è una strategia di controllo di errore, che svolge il compito di rivelare un errore (ma non di correggerlo). I pacchetti corrotti vengono scartati e viene richiesta la loro ritrasmissione.

Affinché il sistema sia capace di riconoscere i messaggi corrotti è necessario che questi vengano preliminarmente codificati da un codificatore. Dopo la trasmissione il decodificatore decodifica il messaggio e, a seconda che questo sia integro o meno, si comporta in base a uno dei 3 diversi protocolli più comuni:

1. **PROCOLLO STOP AND WAIT**; il mittente invia un messaggio e attende dal destinatario una conferma positiva (**ACK**, acknowledge), negativa (**NACK**, contrazione di negative acknowledge) o un **comando**; se scade il tempo di attesa (time-out) per uno di questi tre, il mittente provvederà a rispedire il pacchetto e il destinatario si incaricherà di scartare eventuali repliche. Nel caso in cui si verificasse un errore nella trasmissione del segnale di conferma (ACK), il mittente provvederà a rinviare il pacchetto; il destinatario riceverà in questo modo una copia del pacchetto già ricevuto, credendo che gli sia pervenuto un nuovo pacchetto. Per evitare questo problema si può procedere numerando i pacchetti trasmessi, ovvero inserendo un bit di conteggio.
2. **Go-Back-N**; il mittente dispone di un **buffer dove immagazzina 'N' pacchetti** da spedire, man mano che riceve la conferma ACK svuota il buffer e lo riempie con nuovi pacchetti; nell'eventualità di pacchetti persi o danneggiati e scartati avviene il rinvio del blocco di pacchetti interessati. I pacchetti ricevuti dal destinatario dopo quello scartato vengono eliminati.
3. **Selective Repeat**; in questo caso anche il destinatario dispone di un **buffer dove memorizzare i pacchetti** ricevuti dopo quello/quelli scartati; quando i pacchetti interessati vengono correttamente ricevuti, il buffer viene svuotato (mittente) o i pacchetti contenuti salvati (destinatario).



ACKNOLODGEMENT(ACK)

Prima di parlare più approfonditamente del protocollo stop and wait si deve conoscere come funziona l'ACK.

In ambito telecomunicazioni e informatico, è il simbolo che identifica un segnale di Acknowledge emesso in risposta alla ricezione di un'informazione completa.

Tipico esempio è il pacchetto di controllo previsto dal protocollo TCP(transmission control protocol)che viene trasmesso dal ricevente al mittente per segnalare la corretta ricezione di un pacchetto dati.

L'ACK può anche essere di tipo cumulativo (quello usato dal TCP), indicando cioè l'avvenuta corretta ricezione di più pacchetti di dati.

Per esempio: un ACK4 indica che il pacchetto 4 che la stazione trasmittente ha inviato è stato ricevuto correttamente; implicitamente però l'ACK4 cumulativo sta ad indicare che anche i pacchetti 3, 2, 1, 0 sono stati ricevuti e non sono andati persi.

Similmente un NACK (Not Acknowledge) indica la mancata ricezione di un pacchetto (nel caso di NACK selettivo), cioè la mancata ricezione di 1 pacchetto (NACK4= il pacchetto 4 non è arrivato, ma il 3, 2, 1 e 0 si).

APPROFONDIMENTO

Il protocollo Stop-and-wait è il più semplice protocollo di comunicazione di richiesta di ripetizione automatica (ARQ) della trasmissione di un pacchetto informativo in caso di rilevazione di errore nel pacchetto stesso del ricevitore. Un mittente manda un solo frame alla volta. Dopo che ogni frame è stato inviato, non viene inviato più nulla sino a quando il mittente non riceve un segnale ACK. Il quale viene inviato al mittente dopo che il destinatario ha ricevuto un frame corretto. Se l'ACK non raggiunge il mittente prima di un certo tempo (timeout), rimanda nuovamente il frame.

Un altro problema si verifica quando, l'ACK inviato dal destinatario è danneggiato o perso: in questo caso, il mittente non riceve l'ACK, va in timeout e invia nuovamente il frame. Questo comportamento genera un problema per il destinatario, che si trova con due copie dello stesso frame, senza sapere se ciò che ha ricevuto è un frame duplicato oppure un frame contenente gli stessi dati di quello precedente. Può anche accadere che vi siano due segnali di ACK per lo stesso pacchetto questo succede a causa del lento tempo di risposta.

Per evitare questi problemi, la soluzione più comune è quella di definire un numero di sequenza di 1 bit(che si alterna sempre) nell'header del frame. Quando il destinatario invia un ACK, include anche il numero di sequenza del prossimo pacchetto che si aspetta. In questo modo, il ricevente è in grado di riconoscere i frame duplicati tramite il semplice controllo del bit di sequenza. Questo significa che, se due frame consecutivi hanno lo stesso numero di sequenza, sono duplicati e quindi il secondo viene scartato. Lo stesso discorso si può applicare agli ACK.

Anche questo metodo non risolve tutti i problemi, poiché si può verificare l'eventualità in cui il mittente invia un pacchetto con bit di sequenza pari a zero non ricevuto dal destinatario, e immediatamente dopo riceve un ACK con bit di sequenza pari a uno: in questo modo vengono persi due pacchetti a causa della asincrona dei due contatori. Una soluzione possibile per limitare questo genere di errori è aumentare il numero di bit di sequenza, anche se questo rende solo più improbabile l'evenienza di avere delle copie.

Concludendo, il metodo ARQ stop-and-wait è parecchio inefficiente rispetto ad altri protocolli ARQ, specialmente a causa del tempo che intercorre tra l'invio dei vari pacchetti, e contando anche il fatto che essendoci gli ACK, il tempo per terminare la comunicazione raddoppia, limitando di fatto la capacità del canale di comunicazione.