

Tutorial su World FIP

1. Introduzione

Il World FIP è un **field bus** adatto per applicazioni **tempo-critiche** con caratteristiche di spiccata periodicità dei processi. Il nome deriva dal FIP (**Fieldbus Instrumentation Protocol**) che è stato inizialmente standardizzato come standard nazionale francese. Successivamente è stato inserito nello standard europeo CENELEC EN50170 di Bus di Campo. Il nome è stato cambiato in World FIP successivamente, per esigenze di mercato internazionale. Il protocollo che utilizza il FIP, privilegia il trasferimento di variabili periodiche, mentre non è molto efficiente nel trasferimento delle variabili aperiodiche o dei messaggi.

1.1 Filosofia di base del FIP

La filosofia su cui si basa il FIP è di considerare le variabili utilizzate dalle entità costituenti il processo applicativo come componenti di un database tempo-critico.

Normalmente nei sistemi di controllo di processo esiste un unico database centrale che non è però tempo-critico in quanto non contiene variabili di campo, cioè le variabili di processo, le quali hanno una dinamica talmente veloce da superare i tempi di accesso al database. Nel FIP, invece, si considera la presenza di 2 database: quello tempo-critico che contiene solo le variabili di campo, e il database globale che contiene tutte le altre informazioni utili per la gestione dell'impianto. L'aggiornamento del database tempo critico è di pertinenza del FIP ed è ottenuto con i servizi messi a disposizione dal protocollo. Pertanto, i servizi di lettura/scrittura di un dato del database sono in pratica realizzati mediante i servizi di trasmissione e ricezione di messaggi.

1.2 Traffico generato nel FIP

Nel FIP sono considerati i seguenti tipi di traffico:

- **traffico periodico** per l'aggiornamento delle variabili di processo. Nel FIP si considera questo tipo di traffico come quello presente nel 99% dei casi;
- **traffico aperiodico** per aggiornamento di variabili;
- **traffico aperiodico** per lo scambio dei messaggi.

La differenza tra messaggi e variabili sta nel fatto che una variabile è una informazione molto piccola individuata attraverso un identificatore, mentre un messaggio è una frame di parecchi byte di lunghezza (tipicamente un file) che deve essere trattata in maniera diversa rispetto alle variabili.

1.3 Dispositivi fisici presenti

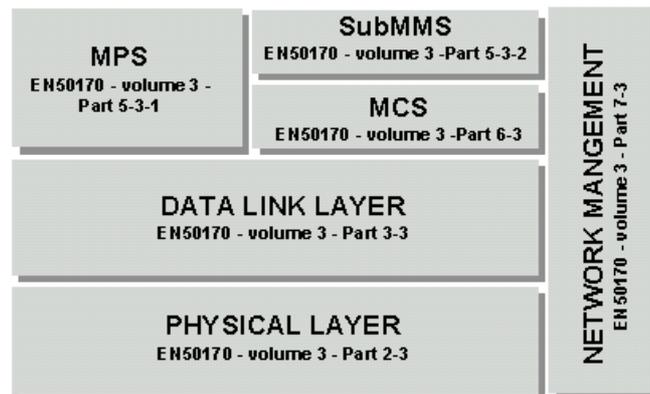
Il sistema minimo supportato dal FIP è costituito da un singolo tronco di bus con un certo numero di stazioni, una delle quali ha il ruolo di gestore del mezzo fisico e prende il nome di Bus Arbitrator (B.A.). Lungo il bus possono essere presenti diversi dispositivi fisici:

- le **Junction Box** (scatole di giunzione) che sono in pratica dei ripetitori. Infatti sono dispositivi che usano uno stesso protocollo FIP e sono usati per realizzare la connessione a gruppi di più dispositivi;
- i **Field Concentrator** (concentratori di campo) che sono in pratica dei gateway. Infatti sono dispositivi che o non usano lo stesso protocollo FIP o sono usati per interconnettere una rete FIP con i bus di altri FIP;

- i **Repeater** (ripetitori) che servono per allungare il bus collegando diversi segmenti di bus. Essi non contengono direttamente i dispositivi, cioè non inglobano nodi al loro interno, come invece avviene per le Junction Box.

1.4 Architettura a livelli del FIP

L'architettura del FIP è quella classica per applicazioni real-time, a tre livelli: Physical Layer (PHL), Data Link Layer (DLL=MAC+LLC), Application Layer (AL).



- Il PHL utilizza una codifica di tipo **Manchester**. Ne esistono 2 versioni legate alle specifiche che sono state definite dall'ISA (International Society for automation and control), in collaborazione con l'IEC: la versione chiamata H1 a velocità più bassa (50kbps) che utilizza doppino intrecciato e la versione chiamata H2 a velocità più alta (1Mbps o 2,5 Mbps) per mezzi fisici più veloci.

Una caratteristica interessante del FIP è la possibilità di potere utilizzare le barriere a sicurezza intrinseca (**Intrinsic Safety Barrier**) per quelle industrie, tipo petrolchimiche, in cui ci sono atmosfere esplosive;

- Il Data link layer è diviso in due sottolivelli: MAC ed LLC. Il MAC utilizza un protocollo di accesso multiplexato a divisione di tempo **TDMA** (Time Division Multiplexing Access), in cui il **Bus Arbitrator** effettua la schedulazione delle varie trasmissioni, inviando di volta in volta, alle varie stazioni, le autorizzazioni di accesso al mezzo, in modo periodico.

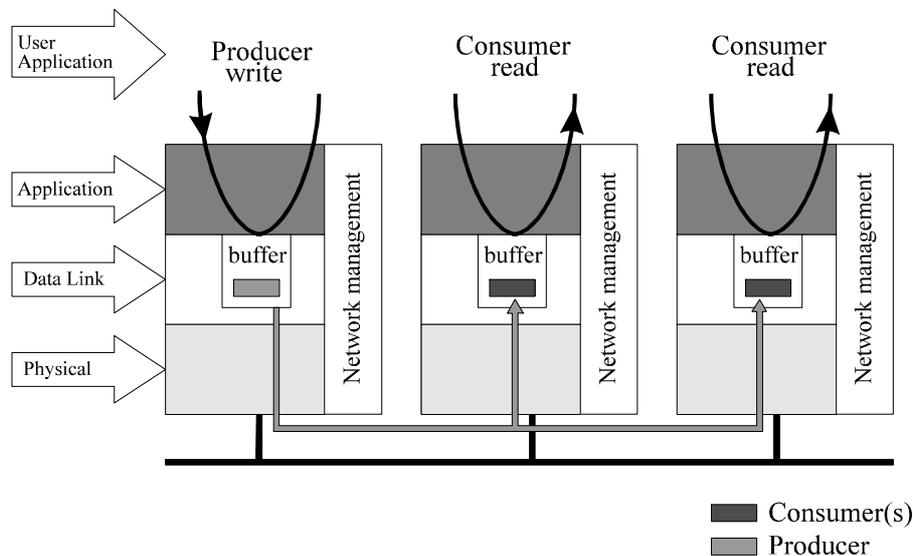
Nell'LLC sono presenti i servizi di:

- **lettura/scrittura/trasferimento** dei dati (valori delle variabili) contenuti nei buffer di trasmissione/ricezione
- **richiesta/trasferimento** con ack e senza ack dei messaggi
- Nell'AL sono presenti dei servizi specializzati adatti alle applicazioni industriali per il trasferimento di dati periodici e aperiodici. E' anche presente una sezione di **field bus management** che risolve tutti i problemi legati alla gestione del FIP e si occupa del problema della configurazione delle funzioni all'interno dei vari livelli, del controllo degli errori e quindi delle prestazioni che si ottengono.

1.5 Meccanismo di trasferimento periodico dei dati.

Nella figura seguente è mostrato il meccanismo di base del sistema, cioè quello di trasferimento di un dato: l'utente produttore passa all'AL l'informazione che va trasferita ai consumatori; l'AL la scrive all'interno di un buffer accessibile dal DLL; quest'ultimo, tramite il mezzo fisico, trasferisce l'informazione nei buffer dei vari nodi consumatori, i quali, tramite l'AL locale possono leggerla.

Questa operazione viene realizzata ciclicamente: quindi ciclicamente il processo produttore scrive nel buffer, ciclicamente i processi consumatori leggono il loro buffer e ciclicamente il DLL trasferisce informazioni nei buffer.



A livello applicativo sono visibili un insieme di oggetti astratti, che sono chiamati "Application Objects". L'insieme di questi oggetti costituisce un Virtual Field Database, che rappresenta un database distribuito.

I servizi disponibili nell'AL sono:

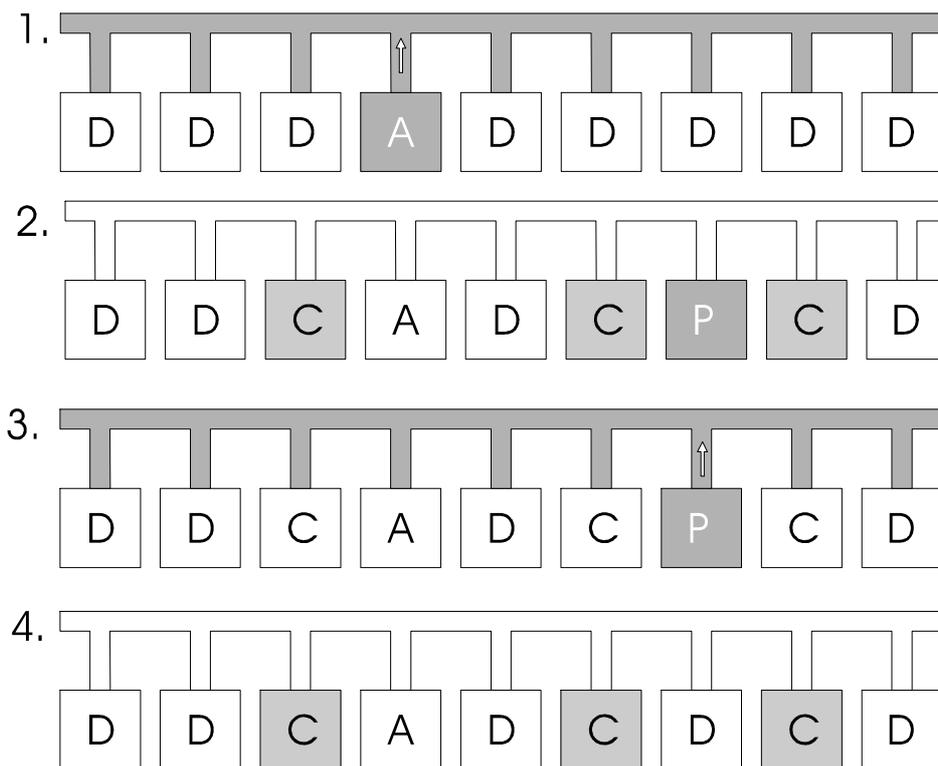
- servizio di lettura e di scrittura locale, usato per leggere o scrivere nel buffer locale;
- servizio di aggiornamento, usato per aggiornare periodicamente quei buffer che sono schedulati nel traffico ciclico;
- servizio di lettura e di scrittura remota, che sono dei servizi composti perché creano una doppia interazione, ad esempio una scrittura su buffer locale + una richiesta di trasmissione come traffico non ciclico;
- servizio indicatore di emissione o ricezione;
- servizio di lettura e scrittura universale, usato per fare dei broadcast, cioè per leggere o scrivere in tutti i buffer della rete;
- servizio di lettura di una lista.

I servizi disponibili nel DLL sono i seguenti:

- scrittura, lettura o trasferimento di un buffer, per il traffico ciclico;
- richiesta esplicita specifica o libera di trasferimento di un buffer, per far sì che il bus arbitrator scheduli una qualche finestra per traffico aciclico;
- richiesta e trasferimento di un messaggio con o senza ack.

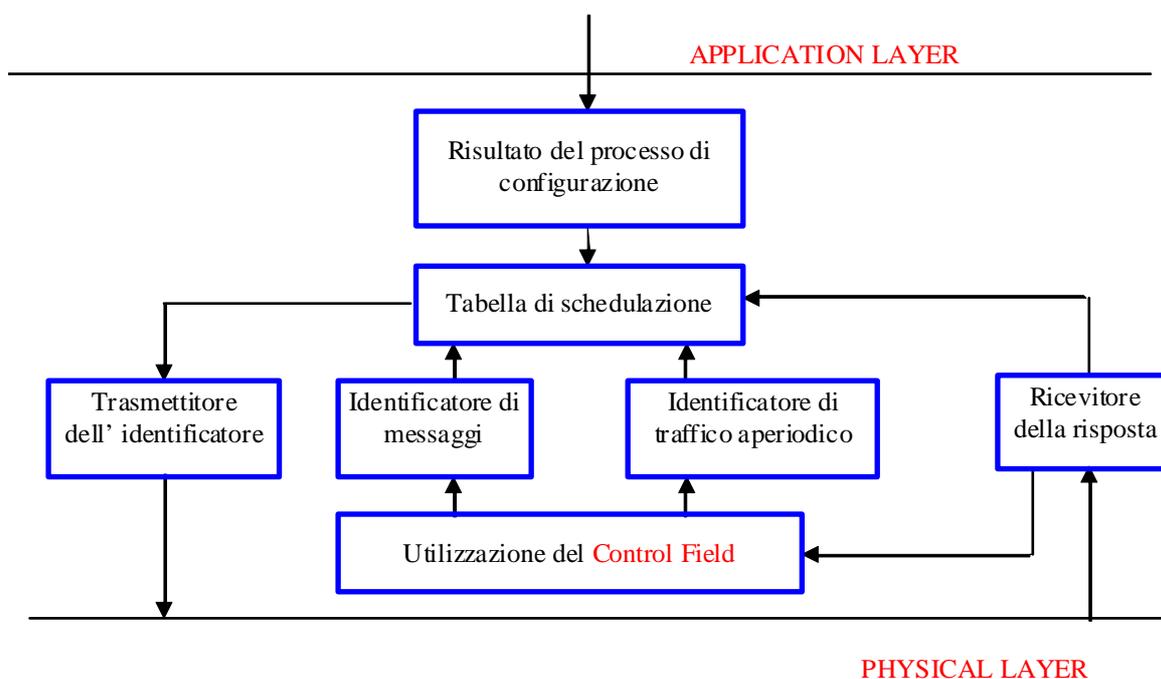
Nella figura seguente è mostrato il meccanismo utilizzato nella trasmissione del traffico ciclico:

1. Il bus arbitrator A spedisce sul bus un identificatore per una trasmissione broadcasting Tale identificatore, che individua i nodi che sono coinvolti nella prossima trasmissione, specifica implicitamente gli indirizzi del processo produttore P e dei processi consumatori C coinvolti nella prossima transazione.
2. L'identificatore è ricevuto dal produttore.
3. Il produttore spedisce sul bus comune i suoi dati.
4. Soltanto i consumatori coinvolti nella transazione, accettano i dati prodotti.



Quindi il Bus Arbitrator, con un solo messaggio identifica univocamente la successiva fase di trasmissione. Tutti i dispositivi D ricevono questo identificatore, ma solo uno si identifica come produttore P e solo alcuni si identificano come consumatori C.

Nella figura sotto è riportata la struttura interna del bus arbitrator: la sua parte più importante è la tabella di schedulazione (che contiene l'elenco delle variabili periodiche ed il loro ordine di trasmissione) dalla quale viene prelevato l'identificatore della variabile periodica o aperiodica (o del messaggio) da trasmettere.



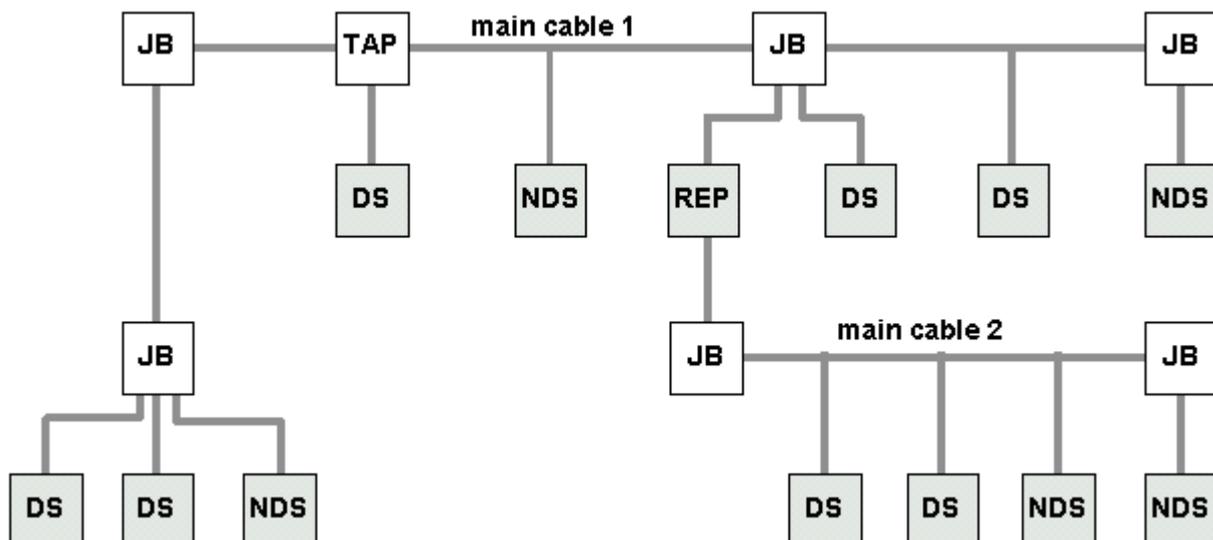
Dopo la trasmissione di un identificatore, il BA ascolta il mezzo e legge la risposta spedita dal produttore autorizzato, che può contenere, nel control field, la richiesta di ulteriori autorizzazioni per trasmettere variabili asincrone o messaggi. Il BA analizza la propria tabella di schedulazione e cerca degli spazi vuoti per autorizzare successivamente le richieste di traffico non periodico.

2. LIVELLO FISICO

Il livello fisico di WorldFIP assicura il trasferimento di bit di informazione da un'apparecchiatura a tutte altre apparecchiature connesse al bus. Il mezzo della trasmissione può essere doppino intrecciato o fibra ottica.

2.1 Configurazione generale

La figura sottostante mostra la struttura di una generica rete WorldFIP in cui sono presenti i principali componenti del sistema.



Le apparecchiature sono denominate:

- **JB** Junction Box. Sono in pratica dei ripetitori, infatti sono dispositivi che usano lo stesso protocollo FIP e sono usati per realizzare la connessione a gruppi di più dispositivi;
- **REP** Ripetitore
- **TAP** Fornisce un punto di collegamento dei dispositivi sul cavo principale.
- **DS** Subscriber localmente disconnettibile
- **NDS** Subscriber localmente non disconnettibile

2.2 Velocità di trasmissione

Per il livello fisico in filo in rame sono state definite tre velocità della trasmissione:

- S1: **31.25 o 50 kb/s** (velocità bassa)
- S2: **1Mb/s** (velocità alta)
- S3: **2.5Mb/s** (velocità alta)

S2 è la velocità dello standard.

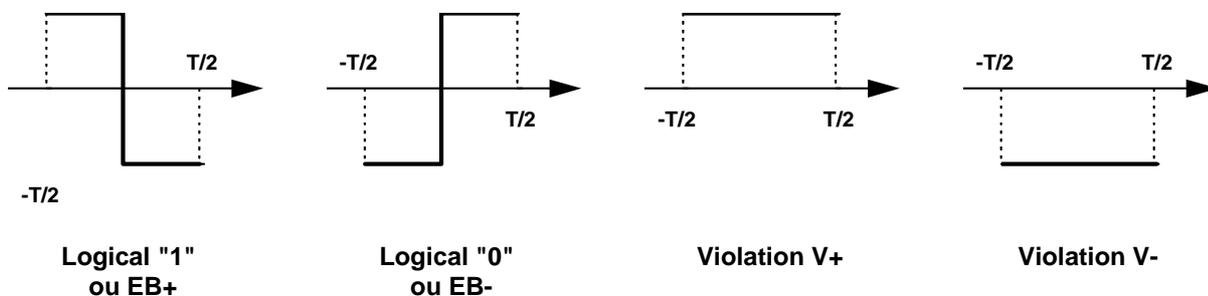
S1 e S3 sono usati solo per applicazioni speciali.

Una velocità addizionale di 5Mb/s è stata definita per livello fisico a fibra ottica.

2.3 Codifica

Il livello fisico codifica i bit trasmessi dal data link layer usando la codifica Manchester. Questo codice permette di trasmettere simultaneamente il segnale di sincronizzazione temporale e i dati. Ciascun intervallo del tempo usato per codificare un bit è diviso in due parti di durata uguale.

I simboli sono rappresentati come segue:

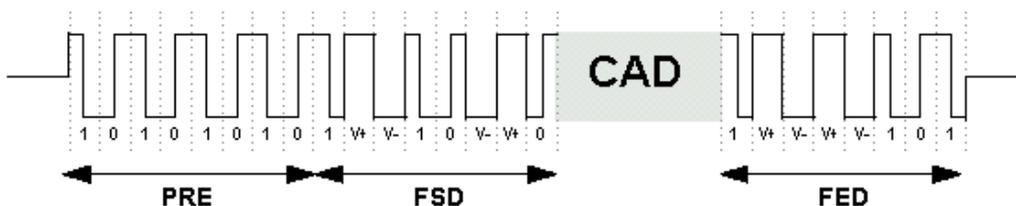


Le violazioni della codifica sono usate nei delimitatori delle frame.

2.4 Codifica delle frame WorldFIP

Tutte le frame sono composte di tre parti:

- Frame start sequence (FSS = PRE+FSD)
- Data and check fields
- Frame end delimiter (FED)



La frame start sequence (FSS) contiene i seguenti campi :

- Preambolo (PRE) Questa serie di 8 bit "1" e "0" è usata dai ricevitori per sincronizzarsi con il clock del trasmettente.
- Frame Start Delimiter (FSD) Questa serie di bit contiene alcune violazioni della codifica ed indica al data link layer l'inizio di informazioni utili (CAD).

Il campo CAD (Controllo e Dati) contiene le informazione fornite dal data link layer.

La sequenza Frame End Delimiter (FED) è usata dal data link layer per localizzare la fine del campo CAD.

Il livello fisico aggiunge quindi 24 simboli ad ogni frame passata dal DLL, che viene trasmessa.

3. DATA LINK LAYER

Il data link layer fornisce due tipi di servizi di trasmissione:

- scambio di variabili identificate
- trasferimenti di messaggi

Questi scambi possono avvenire:

- ciclicamente. Quando il sistema è configurato, i nomi degli oggetti e la loro periodicità sono configurate. Scambi di queste variabili o messaggi avvengono automaticamente senza che l'utente li richieda.
- su richiesta esplicita dell'utente. Queste richieste esplicite faranno circolare sul bus i valori di uno o più variabili, o uno o più messaggi.

3.1 Indirizzamento

Il modello d'indirizzamento WorldFIP ha due spazi distinti d'indirizzamento:

- indirizzamento delle variabili
 - Ciascuna variabile nel sistema distribuito è associata con un identificatore che caratterizza univocamente la variabile. L'indirizzamento è globale.
 - Gli identificatori sono codificati usando numeri interi a 16-bit; teoricamente possono essere identificate 65536 variabili. Per un dato ci può essere uno e solo un produttore, ma diversi consumatori. Gli scambi di variabili sono compiuti in broadcasting.
- Indirizzamento di messaggi
 - Gli scambi di messaggi possono essere di tipo punto-punto o in multipunto su un singolo segmento. Ciascun messaggio emesso contiene l'indirizzo dell'entità emittente e l'indirizzo dell'entità destinazione. Questi indirizzi sono codificati con 24 bit che indicano sia il segmento della rete quanto l'indirizzo della stazione indirizzata.

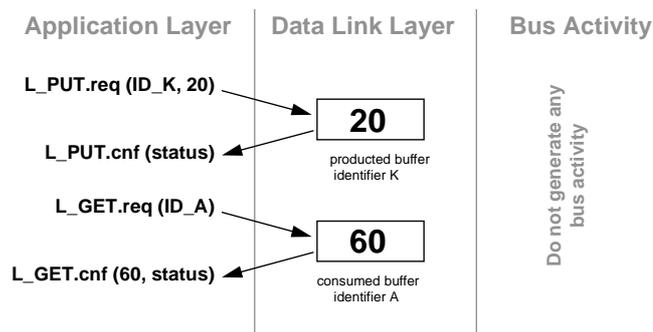
3.2 Interfacce Application Layer-Data Link Layer - Physical Layer

Il data link layer fornisce servizi all'Application layer ed usa i servizi del livello fisico.

Il data link layer può essere rappresentato come una collezione di buffer prodotti e consumati. Questi buffer contengono gli ultimi valori aggiornati dall'utente o dalla rete. Quando un nuovo valore viene introdotto in un buffer, il valore precedente viene sovrascritto.

Le risorse che descrivono questi buffer sono allocate quando la stazione è inizialmente configurata. L'accesso a un buffer è attuato attraverso l'uso dell'identificatore prodotto o consumato.

Il disegno accanto mostra l'interfaccia tra il livello application-data ed Data link layer.



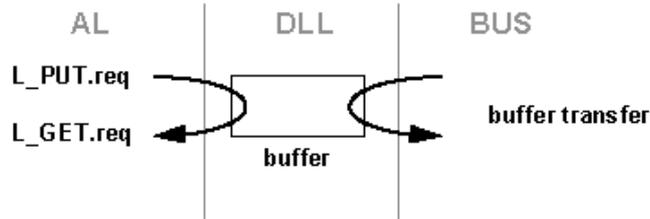
In questo esempio il Data link layer contiene due buffer:

- Un buffer prodotto corrispondente all'identificatore K.
- Un buffer consumato corrispondente all'identificatore A.

Il Livello Application usa un servizio di scrittura (`L_PUT.req`) per inserire un nuovo valore (20) nel buffer prodotto, ed usa un servizio di lettura (`L_GET.req`) per rimuovere il valore di una variabile consumata.

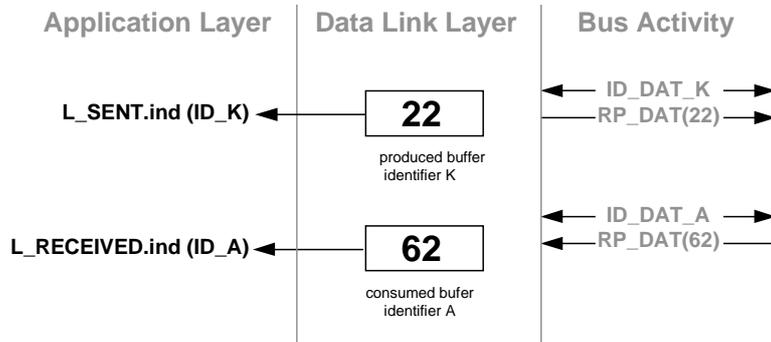
Questi servizi di lettura e scrittura sono locali alla stazione e non generano nessuna attività sul bus.

I buffer prodotti o consumati sono anche accessibili attraverso il bus, mediante il quale è possibile rimuovere o depositare valori remoti. Questo meccanismo è conosciuto come buffer transfer (trasferimento di buffer).



Poiché i buffer hanno due accessi (dal lato Application layer e dal lato Data link layer), il Livello Data Link deve risolvere problemi causati da conflitti di accesso. Il Livello Data Link contiene due Buffer per contenere sia la variabile prodotta che quella consumata in una singola entità di comunicazione.

Per ottenere un trasferimento di un buffer (cioè di una variabile), il Bus Arbitrator trasmette una **question frame** ID_DAT, che contiene l'identificatore della variabile.



Se la stazione è stata dichiarata produttrice dell'identificatore, il Livello Data Link locale risponde col valore della variabile usando una **response frame** RP_DAT. Inoltre, il Livello Data Link spedisce un'indicazione di trasmissione effettuata, al Livello Application (L_SENT.indication).

Similmente, se una stazione è dichiarata consumatore dell'identificatore, il Livello Data Link accetta il valore della response frame RP_DAT sul bus e spedisce un'indicazione di ricezione dell'identificatore al Livello Application (L_RECEIVED.indication).

La massima dimensione del buffer è 128 byte. I buffer contengono solo variabili; non contengono messaggi.

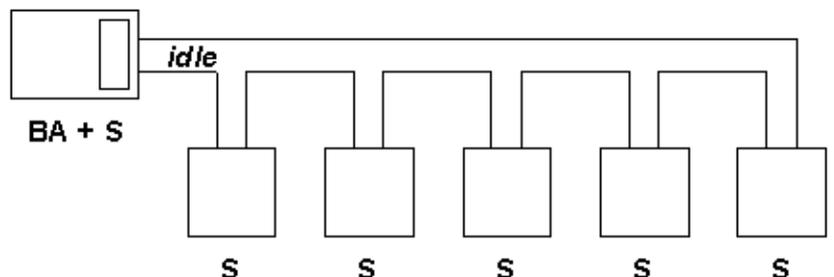
3.3 Meccanismo di allocazione del Mezzo fisico per il trasferimento periodico.

Una rete WorldFIP è composta di stazioni con due tipi di funzionalità:

- bus arbitration: gestione di accesso al mezzo trasmissivo.
- funzioni di produzione/ consumo.

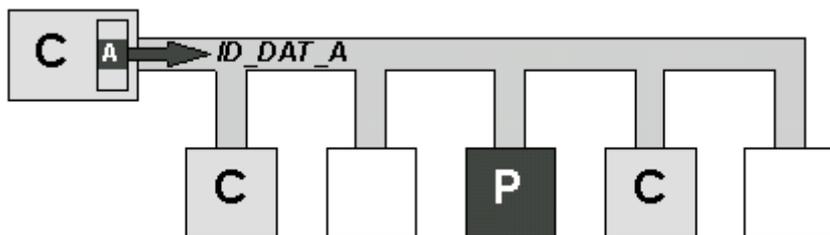
Qualunque stazione WorldFIP può compiere simultaneamente queste due funzioni, ma in un dato istante solo una stazione può compiere la funzione di Bus Arbitrator attivo.

L'arbitro del bus (BA) ha le risorse necessarie per analizzare le variabili agli istanti definiti quando il sistema è stato configurato. L'arbitro del bus ha una tabella di scansione con un



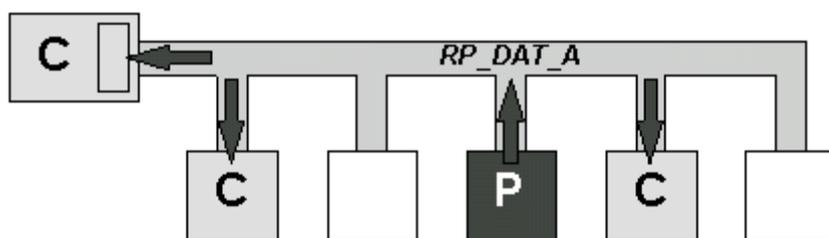
elenco di identificatori da far circolare sul bus.

Il lavoro dell'arbitro del bus è relativamente semplice. Usa la question frame ID_DAT per trasmettere sul bus il nome di un'identificatore. Questa frame è ricevuta simultaneamente dal Data Link layer di tutte le stazioni connesse al bus. Una e solo una di queste stazioni si riconosce come produttore dell'identificatore. Una o più stazioni si riconoscono come consumatori della variabile.



Il produttore della variabile allora trasmette il valore dell'identificatore in una frame di risposta (RP_DAT). Questo valore è catturato simultaneamente da tutte stazioni consumatrici. L'arbitro del bus poi va avanti al prossimo identificatore nella tabella di scansione e ripete lo stesso ciclo di domanda-risposta.

La trasmissione sul bus di un valore prodotto o la ricezione di un valore consumato coinvolge il meccanismo di trasferimento dei buffer. Il trasferimento di buffer avviene per iniziativa dell'arbitro del bus; è totalmente indipendente da qualunque attività dell'utente.



Quando stazioni nuove che consumano uno o più variabili vengono connesse al bus, nessun tempo extra è necessario per fornire quelle variabili alle stazioni nuove. Il meccanismo rimane lo stesso. I temporizzatori nelle macchine di stato del data link layer esaminano continuamente l'attività sul bus. Così se una response question (RP_DAT) viene persa o tarda ad arrivare, consumatori ritornano a una condizione di attesa per una question frame ed ignorano tutti altri tipi di frame.

3.3.1 Tavole di arbitraggio del bus

L'arbitro del bus (BA) può essere considerato il direttore d'orchestra della rete. Quando il sistema è configurato, all'arbitro del bus è dato l'elenco di variabili da analizzare ed le periodicità associate a ciascuna di queste variabili. Se questa configurazione è stata convalidata, e se rispetta i vincoli temporali per gli scambi delle variabili elencate, l'arbitro del bus ripete infinitamente il meccanismo descritto sopra.

La scansione delle variabili è deterministica. WorldFIP può garantire che una variabile con una periodicità data sarà analizzata all'istante appropriato.

L'esempio mostrato sotto indica come configurare l'arbitro del bus per garantire una scansione deterministica.

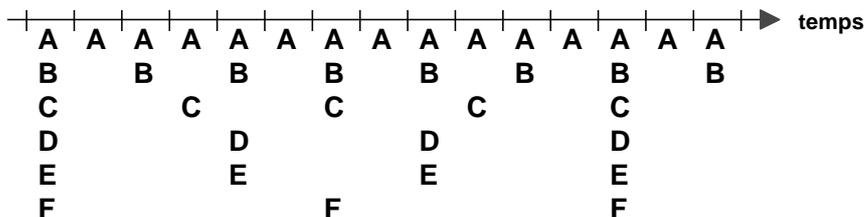
2.2	Variabile	2.3	Periodicità	Tipo	2.4	Tempo
	A		5	INT_8		170
	B		10	INT_16		178
	C		15	OSTR_32		418
	D		20	SFPOINT		194
	E		20	UNS_32		194
	F		30	VSTR_16		290

L'arbitro del bus deve analizzare sei variabili periodiche. Per ciascuna variabile l'arbitro conosce: la sua periodicità espressa in millisecondi, il suo tipo di dato applicativo (e.g: numero intero ad 8-bit, stringa di 32 caratteri, etc.). Usando la velocità di trasmissione (qui è considerata 1Mb/ s) e il tempo di turn-around, l'arbitro del bus può calcolare il tempo necessario per un'operazione elementare composta dal tempo di trasmissione per una **question frame**, seguita dal tempo di trasmissione per la **response frame** associata. (La colonna del tempo è espressa in ms. Corrisponde alla lunghezza di operazioni, dato un tempo di turn-around di 20 mseconds.)

La figura sotto mostra una distribuzione possibile d'identificatori A...F su un asse temporale, come una funzione della periodicità di ciascuna variabile. Ciascuno periodo di tempo costituisce un ciclo elementare (microciclo). Cicli elementari in questo esempio sono tutta della stessa lunghezza: 5 millisecondi.

Nel primo ciclo elementare l'arbitro del bus trasmette tutte le variabili. Nel secondo trasmette solo la variabile A.

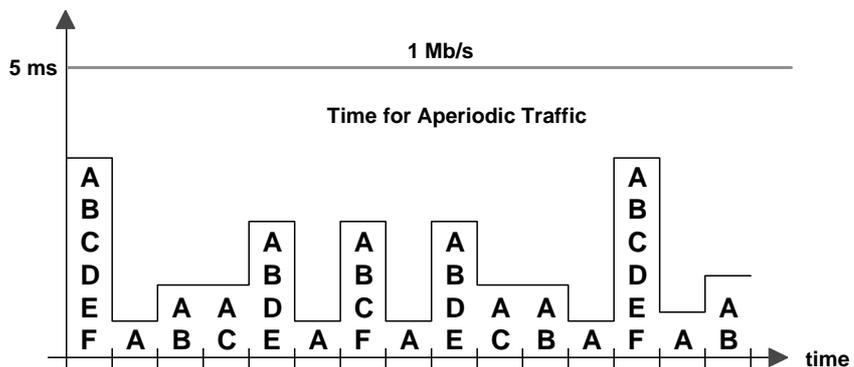
Usando l'informazione di cui sopra, il carico della rete può essere calcolato come mostrato nella figura seguente:



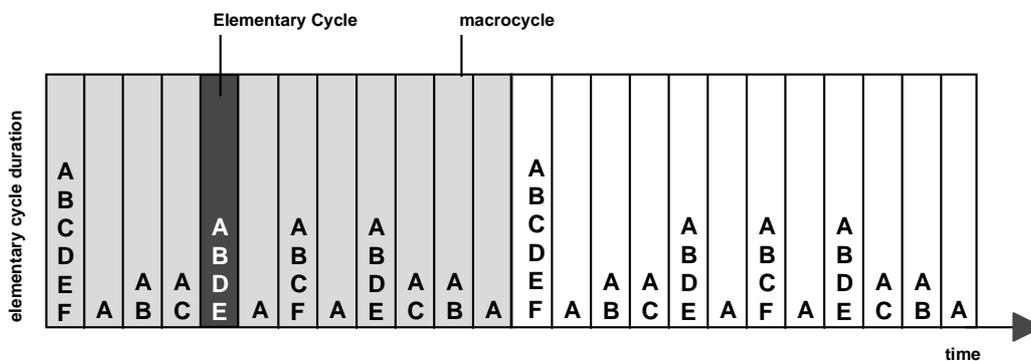
L'asse verticale corrisponde al tempo della scansione per ciascuno ciclo elementare. Questo tempo è la somma dei tempi di tutte operazioni componenti il ciclo elementare. L'asse orizzontale corrisponde al flusso di cicli elementari in tempo.

Dalla figura si nota che nessuno ciclo elementare va oltre il limite dei 5 ms. E' quindi possibile realizzare una tavola di schedulazione del bus che rispetti le periodicità date.

Il tempo che separa la barra orizzontale di 1Mb dalla fine di scansione di ciascuno ciclo elementare è disponibile per il traffico aperiodico.



L'arbitro del bus eseguirà indefinitamente gli stessi cicli elementari, o più precisamente lo stesso macrociclo:



Il macrociclo corrisponde a una sequenza di cicli elementari. Il numero di cicli elementari in un macrociclo è uguale al Minimo Comune Multiplo delle periodicità diviso dal Massimo Denominatore Comune delle periodicità.

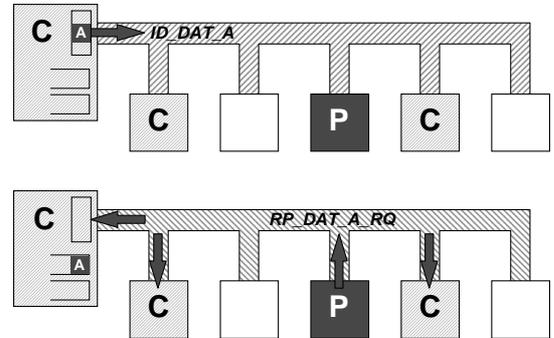
3.4 Richieste di trasferimento aperiodico

Non tutte variabili in un'applicazione distribuita sono incluse necessariamente nelle tavole della scansione cicliche dell'arbitro del bus. Alcune variabili possono essere scambiate solo di quando in quando. WorldFIP fornisce un meccanismo per trasferimento aperiodico di variabili.

Questo meccanismo ha tre stati:

Stato uno:

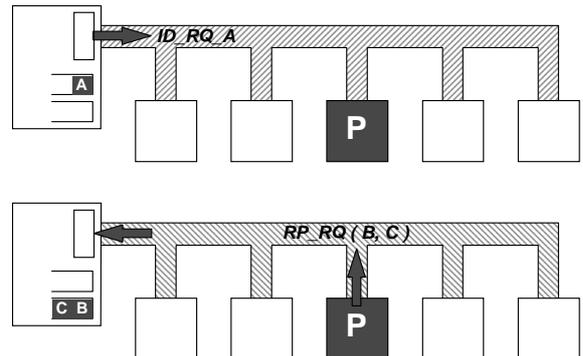
L'arbitro del bus trasmette una question frame che contiene l'identificatore di una generica variabile A in una finestra destinata al traffico periodico (in base alla tabella di schedulazione). Il produttore della variabile A risponde con la variabile corrispondente e se ha l'esigenza di trasmettere una variabile aperiodica setta un bit di richiesta aperiodica nel campo di controllo della sua response frame (RQ). L'arbitro del bus annota l'identificatore A in una coda di richieste per trasferimento di variabili aperiodiche.



Quando viene effettuata la richiesta per trasferimento aperiodico è possibile specificare due livelli di priorità: Urgente o Normale. L'arbitro del bus ha due code, una per ciascuno livello di priorità.

Stato due:

Nella finestra temporale destinata al traffico di variabili aperiodiche, l'arbitro del bus trasmette una request frame (ID_RQ) per richiedere al produttore dell'identificatore A di emettere la sua richiesta completa. Il produttore dell'identificatore A risponde con una frame RP_RQ (elenco di identificatori). Questo elenco di identificatori è tenuto in considerazione dall'arbitro del bus, il quale immagazzina l'elenco in un'altra coda.

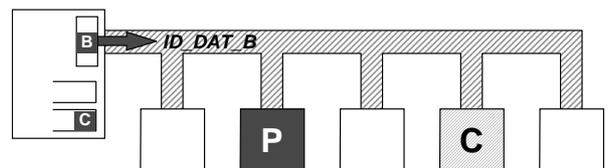


Una response frame può contenere un numero da 1 a 64 identificatori.

L'arbitro del bus può trasmettere la frame di richiesta d'identificazione poco dopo che la richiesta è stata notificata, o molto più tardi. Il ritardo di servizio dipende dal carico del traffico periodico /cioè dagli spazi liberi nella tabella di schedulazione) e dal numero di richieste in corso.

Stato tre:

Questa terza e ultima fase ha luogo in una finestra aperiodica e consiste nel soddisfare le richieste per il trasferimento aperiodico immagazzinate nella coda dell'arbitro del bus. L'arbitro del bus riempie uno o più richieste, dipendentemente dal tempo disponibile nel ciclo elementare per questo tipo di traffico. Facendo così, l'arbitro usa lo stesso meccanismo descritto sopra: l'arbitro trasmette un identificatore di variabile, la quale è prodotta allora da suo unico produttore e consumata da tutte stazioni sottoscrittenti la variabile.



Una stazione può solo richiedere trasferimenti aperiodici usando le risposte a variabili che produce e che sono configurate in traffico ciclico. Un problema che emerge da tale comportamento è che in

WorldFip una stazione non può trasmettere traffico aperiodico se non è schedulata dal Bus Arbitrator per traffico periodico.

3.5 Trasferimento di messaggi non confermati

Il Livello Data Link di WorldFIP fornisce il trasferimento non confermato di messaggi in modalità punto-punto o broadcast. Il principio per un richiesta di trasferimento di messaggi aperiodici è simile a quello di richiesta di trasferimento aperiodico di variabili. La richiesta è associata all'identificatore di una variabile periodica. Il meccanismo prevede tre step:

Primo step:

L'arbitro del bus ordina la trasmissione dell'identificatore A. Il produttore di A risponde col valore di A ed indica nel campo di controllo (MSG bit) della frame, che ha una richiesta per un trasferimento di messaggi.

L'arbitro del bus monitora la frame legata all'identificatore A e inserisce la richiesta in una coda.

Secondo step:

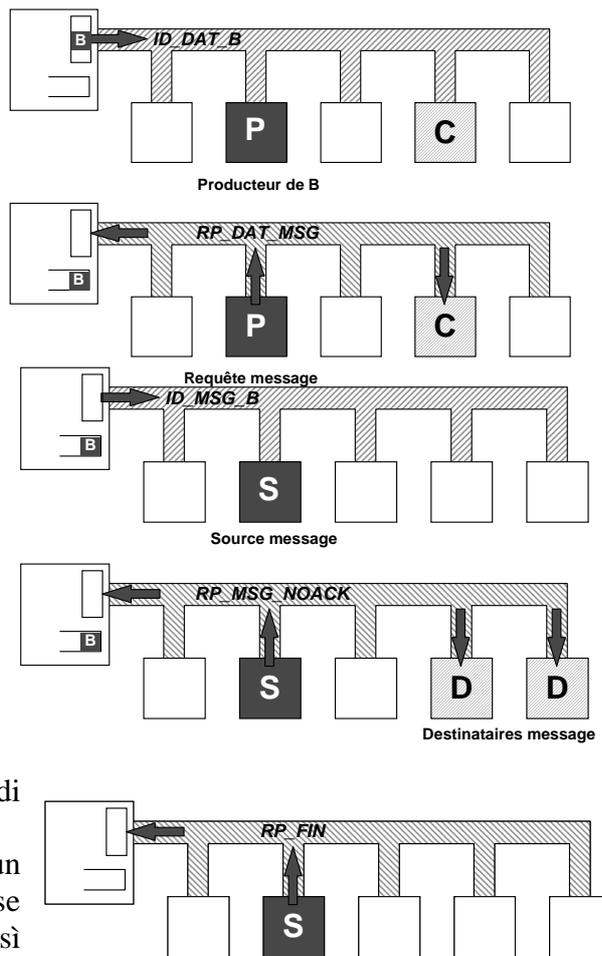
In una finestra per la scansione di richieste di trasferimento di messaggi aperiodico, l'arbitro del bus dà al produttore dell'identificatore A il "diritto parlare." Il produttore di A allora emette il suo messaggio. Questo messaggio include il suo indirizzo (sorgente) e l'indirizzo della stazione/i destinazione in una frame del tipo RP_MSG_NOACK.

L'arbitro del bus, aspetta quindi di ricevere una frame che indichi che il trasferimento di messaggi è finito.

Terzo step:

Dopo aver spedito il messaggio sul bus, il trasmittente emette una frame RP_FIN. Il controllo del bus quindi ritorna all'arbitro del bus.

Quando l'arbitro del bus dà il diritto al produttore di un messaggio di usare la rete, non conosce in anticipo se questa richiesta è confermata o non confermata. Così l'arbitro deve verificare che ci sia abbastanza tempo nel ciclo elementare per soddisfare la richiesta.



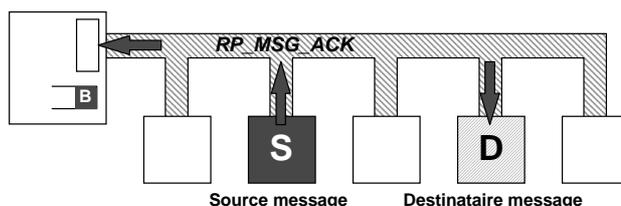
3.6 Trasferimento di messaggi confermati

Il Livello Data Link di WorldFIP fornisce anche servizi di transfer request di messaggi confermati, che rendono gli scambi punto-punto più affidabili.

Il principio operativo è quasi identico a quello del servizio non confermato nei primi due step.

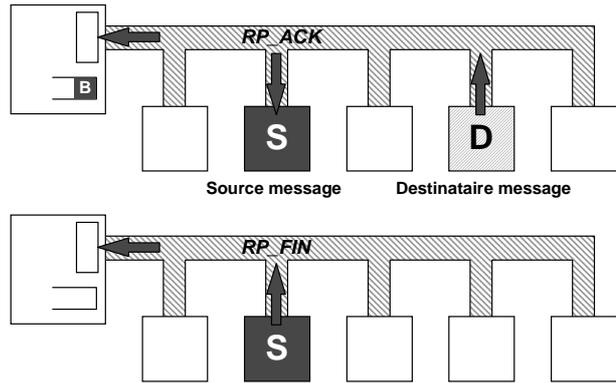
La trasmittente del messaggio risponde con un RP_MSG_ACK frame invece di un RP_MSG_NOACK frame. Lo scambio è punto-punto.

La stazione destinazione risponde con un acknowledgement frame e la trasmittente allora segnala la fine della sua transazione del messaggio all'arbitro del bus (RP _FIN).



Questo servizio usa il meccanismo di numerazione del messaggio modulo 2 che permette alla stazione destinazione di scoprire la perdita o duplicazione di un messaggio.

Il numero di ripetizioni in caso di assenza di un riconoscimento è tra 0 e 2. Questo parametro è un parametro del funzionamento globale della rete WorldFIP.



3.7 Frame

Il formato dei vari tipi di frame usati dal protocollo è diverso nelle diverse fasi operative. A titolo di esempio ne mostriamo alcuni.

- Trasferimento di buffer**

Il meccanismo di trasferimento di buffer usa due tipi di frame: ID_DAT e RP_DAT. Ogni frame trasmessa è delimitata da informazioni aggiunte dal Livello fisico. Come già detto, la frame è posta tra un campo DTR (o inizio di frame) ed un campo FTR (o fine della frame). Tutte le frame WorldFIP iniziano con un byte di controllo che è usato dal subscriber per riconoscere il tipo di frame che riceve. Questo campo di controllo è usato per codificare le richieste di trasferimento di variabili, frame confermate, etc.

Le frame terminano con due byte (FCS: Frame Check Sequence) usati per verificare l'integrità della frame ricevuta.

ID-DAT	DTR	Controle	Identifier	FCS	FTR
	2 bytes	1 byte	2 bytes	2 bytes	1 bytes

Tutte le question frame, siano **ID_RQ** o **ID_MSG** frame, sono del tipo ID_DAT. Sono distinte solo da alcuni bit nel campo di controllo.

La configurazione di una frame emessa come risposta a un ID_DAT frame è mostrata sotto:

RP-DAT	DTR	Controle	DATA	FCS	FTR
	2 bytes	1 octet	n bytes (n <= 128)	2 bytes	1 bytes

Il campo DATA può contenere fino a 128 byte dal Livello Application. Il campo di controllo indica se c'è una qualunque variabile aperiodica o richieste di trasferimento di messaggi.

- trasferimento di messaggi di risposta (RP_MSG_xxx)**

Quando una stazione che ha fatto una richiesta di trasferimento di messaggi, riceve un ID_MSG, essa risponde con un RP_MSG_NOACK o un RP_MSG_ACK frame codificata come segue:

RP-MSG-xx	DTR	Controle	dest. adr.	srce adr.	Message	FCS	FTR
	2 bytes	1 byte	3 bytes	3 bytes	max 256 bytes	2 bytes	2 bytes

un bit nel campo di controllo indica se il trasferimento di messaggi è confermato non confermato. I campi destinazione e sorgente mostrano gli indirizzi delle entità comunicanti.

Quando una stazione destinazione riceve un messaggio con una richiesta confermata, essa emette una frame di conferma.

RP-ACK	DTR	Controle	FCS	FTR
	2 bytes	1 byte	2 bytes	1 bytes

Questa frame è molto corta, poiché l'informazione della conferma è contenuta nel campo di controllo.

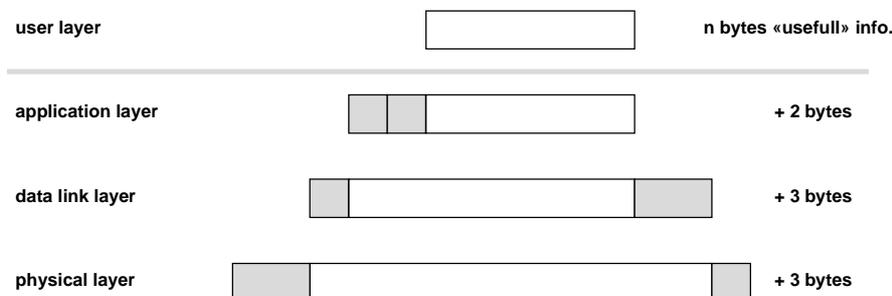
3.8 Livelli di prestazioni nel trasferimento di buffer

L'efficienza di WorldFIP può essere facilmente calcolata per quanto riguarda il trasferimento di buffer ciclici. Tutte le transazioni WorldFIP sono composte dallo scambio di due frame: una frame ID_DAT seguita da una frame RP_DAT. La frame RP_DAT deve apparire entro un tempo chiamato tempo di turn-around. Il Tempo di turn-around è il tempo trascorso tra la fine della ricezione di una frame e l'inizio di trasmissione della frame seguente.



Il tempo di turn-around (TR) è definito nello standard WorldFIP, nella parte di Network management. $10 \text{ TMAC} \leq \text{TR} \leq 70 \text{ TMAC}$.

Ciascun livello dell'architettura aggiunge informazione di servizio a quelle che riceve dal livello soprastante. I tre livelli aggiungono un totale di 64 bit, indipendentemente dalla lunghezza dell'informazione utile.



Il tempo di una transazione include i tempi necessari per:

- trasmissione di una question frame
- tempo di turn-around
- trasmissione della response frame
- tempo di turn-around

L'efficienza è uguale al tempo della trasmissione dell'informazione utile diviso il tempo totale della transazione.

- calcoli basati su $\text{TR} = 10$

Lunghezza d'informazione utile	2.4.1 Efficienza	2.4.2 Throughput utile a 1Mb/s	2.4.3 Throughput utile a 2.5Mb/s
N=1	5.12%	51.28Kb/s	128.21Kb/s
N=2	9.75%	96.56 Kb/s	243.90 Kb/s
N=4	17.77%	177.78 Kb/s	444.44 Kb/s
N=8	30.18%	301.89 Kb/s	754.72 Kb/s
N=16	46.37%	463.77 Kb/s	1159.42 Kb/s
N=32	63.36%	775.76 Kb/s	1939.39 Kb/s
N=64	77.57%	775.76 Kb/s	1939.39 Kb/s
N=128	87.37%	873.72 Kb/s	2184.30 Kb/s

· calcoli basati su TR= 70

Lunghezza d'informazione utile	2.4.4 Efficienza	2.4.5 Throughput utile a 1Mb/s	2.4.6 Throughput utile a 2.5Mb/s
N=1	2.89%	28.99Kb/s	72.46Kb/s
N=2	5.63%	56.34Kb/s	140.85Kb/s
N=4	10.66%	106.67Kb/s	266.67Kb/s
N=8	19.27%	192.77Kb/s	481.93Kb/s
N=16	32.32%	323.23Kb/s	808.08Kb/s
N=32	48.85%	485.55Kb/s	1221.37Kb/s
N=64	65.64%	656.41Kb/s	1641.03Kb/s
N=128	79.25%	792.57Kb/s	1981.42Kb/s

Quando l'informazione utile è molto breve, il tempo di turn-around diviene un fattore importante. Se n= 1bit, l'efficienza è quasi due volte più bassa quando TR= 70 ms.

WorldFIP non sfugge alla regola generale per il calcolo dell'efficienza. Più corta è l'informazione utile, più bassa è l'efficienza. Così è preferibile, allo scopo di aumentare il throughput utile, combinare più informazioni in un struttura dati per la trasmissione.

Quando si compara l'efficienza di WorldFIP con quella di altri tipi di reti, deve essere ricordato che l'informazione prodotta è trasmessa sul bus a tutti consumatori. Quando l'informazione è destinata a un gran numero di consumatori, un'operazione singola è tutto ciò che è necessario per rinfrescare i buffer di questi consumatori. In altri tipi di reti ci sarebbero tante operazioni quanti sono i consumatori.

4. LIVELLO APPLICATION

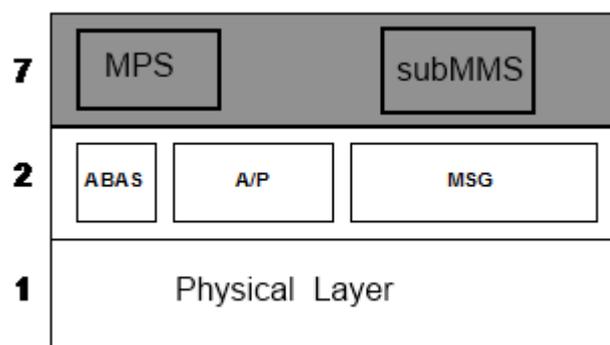
I servizi WorldFIP del Livello Application sono divisi in due gruppi principali:

- MPS (manufacturing periodical/aperiodical services)
- subMMS (sottoinsieme di servizi presenti nel Manufacturing Messaging Specification)

Questa sezione parla solo di MPS.

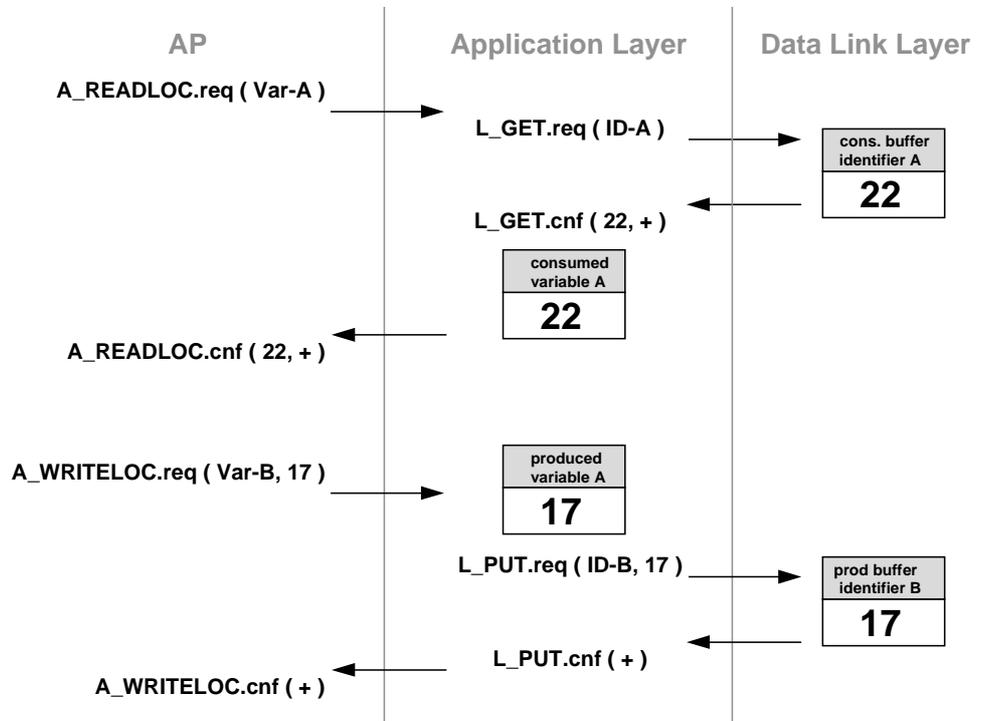
Il modulo MPS del Livello Application fornisce all'utente:

- local read/write services
- remote read/write services
- variable transmission/reception
- informazione sulla freschezza di informazione consumate
- informazione sulla consistenza spaziale e temporale di dati



4.1 Local read/write services

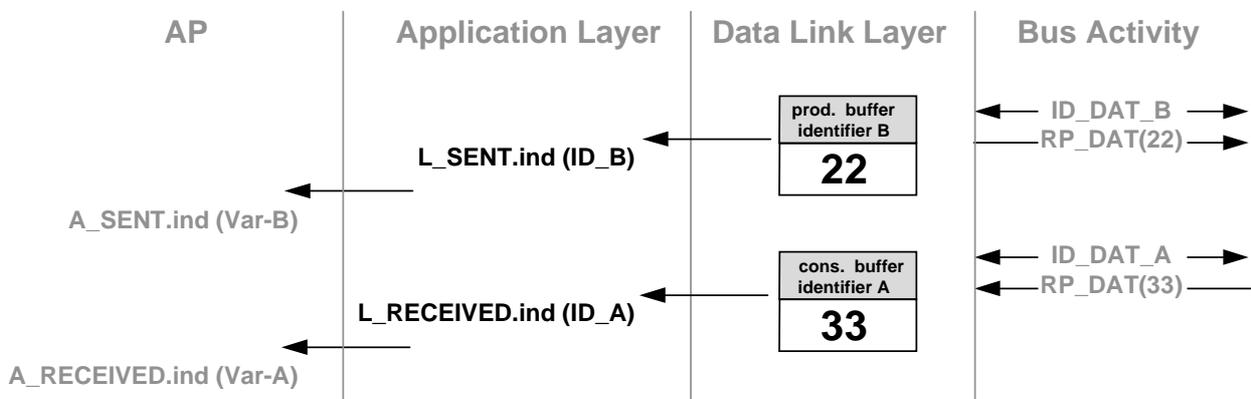
Il Livello Application di WorldFIP fornisce agli utenti servizi **locali** di lettura e scrittura. Questi servizi usano i servizi del Livello Data Link L_PUT.req e L_GET.req per mettere valori in buffer o rimuovere valori da buffer. Questi servizi non generano nessun traffico sul bus.



4.2 Reception/transmission services

Questi servizi, permettono all'utente di essere informato della trasmissione o della ricezione di un identificatore di variabile. Può usare allora questa informazione, per esempio, per sincronizzarsi con una parte di informazione della rete.

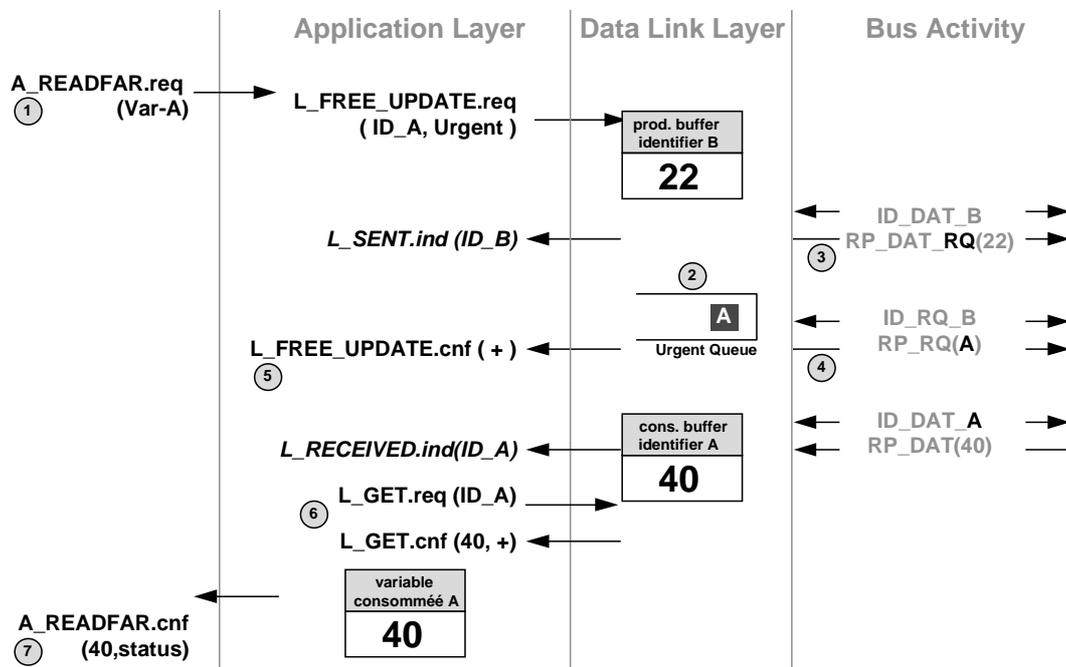
Quando il Livello Application riceve un'indicazione di trasmissione o di ricezione per una variabile prodotta o consumata, inoltra questa indicazione all'utente.



Quando la stazione viene configurata è necessario indicare per ciascuna variabile prodotta o consumata se detta variabile genera uno di queste indicazioni.

4.3 Remote read/write services

Il Livello Application fornisce servizi di lettura e scrittura di variabili remote. Questi servizi permettono di richiedere o trasferire una variabile attraverso il suo identificatore. La figura seguente mostra il meccanismo per la lettura remota di una variabile A (prodotta da altra stazione):



La lettura remota avviene in diversi step:

- 1- L'utente fa una richiesta per lettura remota della variabile A legando la richiesta al buffer periodico contenente la variabile B (valore 22): A_READFAR.req ([var_B]). Il Livello Application esegue la richiesta mediante la primitiva L_FREE_UPDATE.req (ID_B).
- 2- L'identificatore ID_B è aggiunto a una coda di richieste di trasferimento aperiodico
- 3- Usando l'identificatore periodico prodotto dalla stazione (la variabile B), richiesto dall'arbitro del bus, il Livello Data Link setta il bit RQ nel campo di controllo.
- 4- Successivamente, in una finestra della scansione aperiodica, l'arbitro del bus richiede i contenuti della coda della richiesta per il trasferimento aperiodico di variabili.
- 5- La trasmissione di questa coda produce una conferma della richiesta che viene passata all'entità dell' Application layer.
- 6- più tardi, ancora una volta in una finestra aperiodica, l'arbitro del bus autorizzerà la trasmissione della variabile A da parte della stazione remota (valore 40). Un'indicazione del ricevimento di questa variabile sarà spedita al Livello Application. Il Livello Application userà allora un servizio di lettura locale per accedere all'ultimo valore ricevuto.
- 7- La conferma, se positiva, contiene il valore della variabile. Un temporizzatore, nel Livello Application, è usato scoprire tempi dell'attesa eccessivi.

Il meccanismo della scrittura remoto WRITEFAR.req ([var]) funziona quasi alla stessa maniera. Il Livello Application comincia aggiornando il buffer che contiene la variabile, in seguito fa una

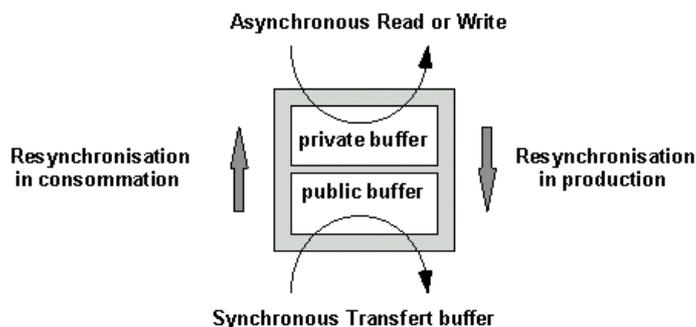
richiesta per il suo trasferimento. Quando il Livello Application riceve un'indicazione di trasmissione della variabile, esso conferma il servizio.

4.4 Risincronizzazione

Il Livello Application rende possibile ai processi asincroni dell'applicazione di partecipare in applicazioni distribuite sincrone fornendo servizi di risincronizzazione in produzione e consumo.

Il meccanismo della risincronizzazione usa una doppia memorizzazione sul Livello Application: un buffer privato accessibile solo attraverso il processo produttore o consumatore dell'applicazione, ed un buffer pubblico accessibile attraverso la rete, la quale rimuove o deposita valori.

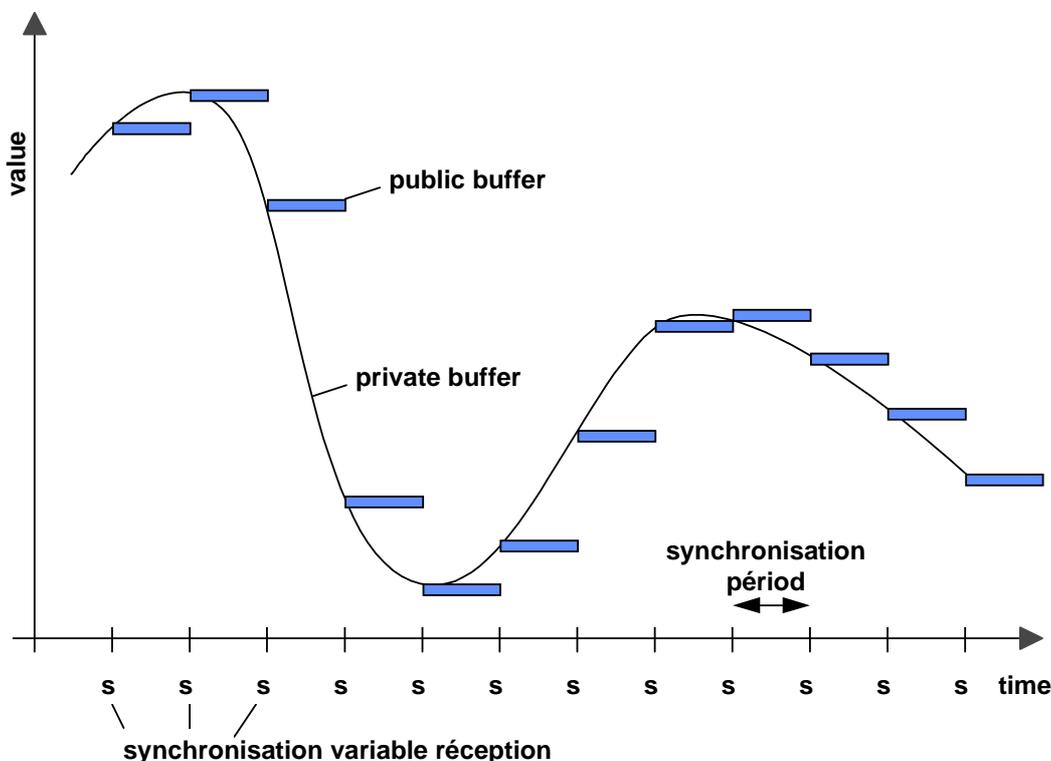
Il meccanismo di risincronizzazione del consumo consiste nel ricopiare i contenuti del buffer pubblico nel buffer privato alla ricezione di una variabile di sincronizzazione.



Il meccanismo di risincronizzazione della produzione consiste nel ricopiare i contenuti del buffer privato nel buffer pubblico su ricezione di una variabile di sincronizzazione.

La figura sotto, mostra i contenuti di buffer privati e pubblici come una funzione del tempo per una variabile prodotta , risincronizzata.

L'utente aggiorna il buffer privato della variabile (curva sulla figura) usando un servizio di scrittura locale.

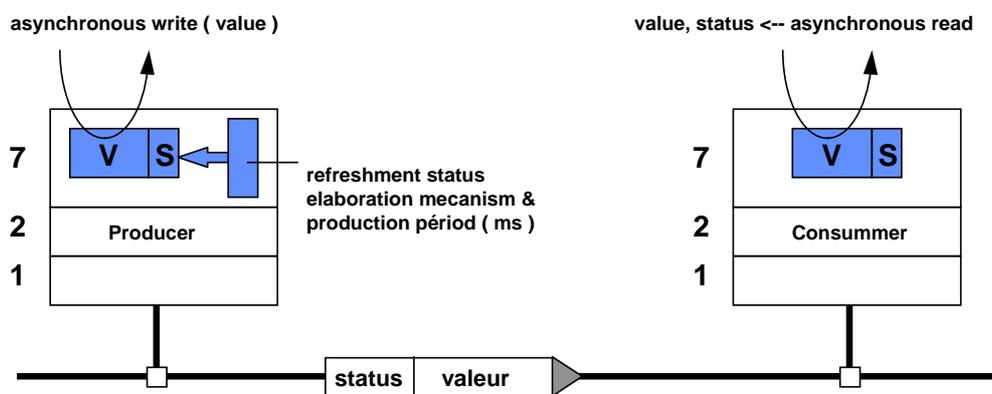


Ogni volta che una variabile di risincronizzazione viene ricevuta, il Livello Application ricopia il buffer privato nel buffer pubblico (rettangoli ombreggiati). Il valore in questo buffer sarà tenuto fino a che viene ricevuta un'altra variabile di sincronizzazione. Il periodo della sincronizzazione

corrisponde alla periodicità della sincronizzazione della variabile nelle tavole della scansione dell'arbitro del bus.

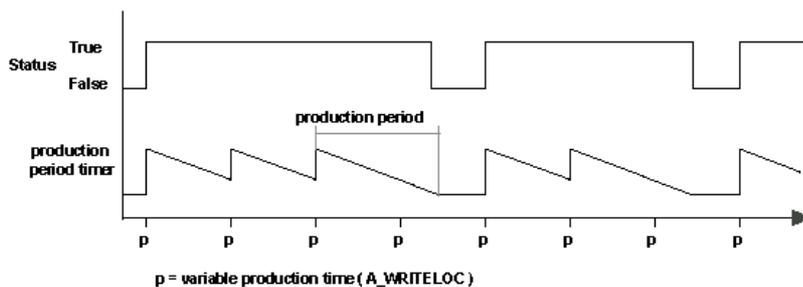
4.5 Rinfresco dell'informazione

Quando un utente legge una variabile nella sua entità di comunicazione locale, esso può ricevere contemporaneamente informazioni qualitative riguardo alla freschezza della variabile. Questa informazione è booleana, e può essere elaborata per qualunque variabile prodotta o consumata. Gli stati di rinfresco sono elaborati dai livelli applicativi dei produttori delle variabili identificate. Per ciascuna variabile prodotta dovrebbe essere indicato se è fornito lo stato del rinfresco. Se c'è un stato di rinfresco, allora deve essere indicato anche il periodo di produzione associato con la Variabile.



Nella figura, il produttore della stazione a sinistra produce valori nuovi al ritmo previsto nella sua entità di comunicazione. Per ogni nuova scrittura, il Livello Application del produttore usa il ritardo di produzione per elaborare un status di rinfresco. Un valore "vero" dello status indica che il processo applicativo del produttore funziona propriamente.

Ciascun consumatore usa un servizio di lettura per accedere allo status e verifica se il produttore della variabile ha rispettato il ritardo di produzione attaccato alla variabile. Il diagramma a destra mostra il meccanismo per elaborare un status del rinfresco asincrono usando il tempo passato e gli istanti di tempo in cui vengono scritti nuovi valori.

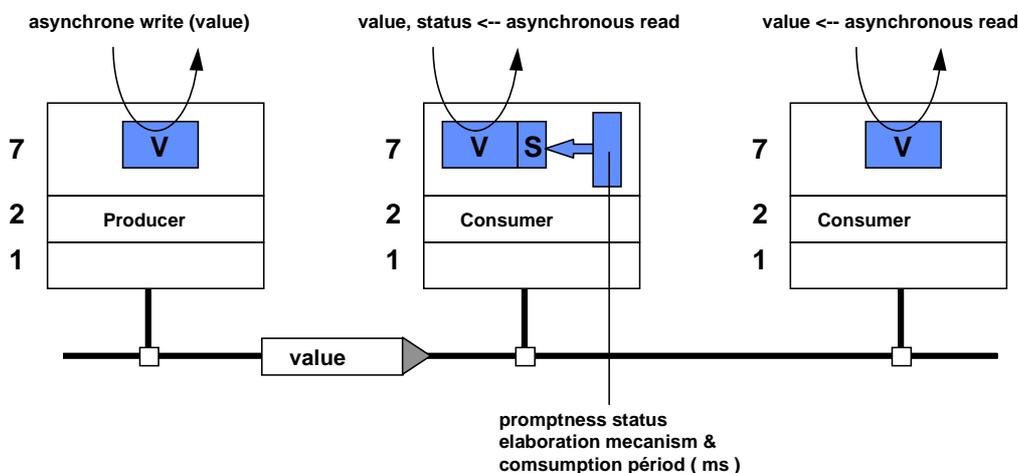


Ogni qualvolta viene scritto un valore nuovo, il Livello Application setta il temporizzatore associato con la variabile al valore del periodo della produzione. Lo Status è "vero" fintanto che questo tempo non è scaduto.

Appena viene ricevuta una variabile di sincronizzazione, il temporizzatore è resettato col valore del periodo della produzione e status diviene "falso." Se il temporizzatore non è scaduto quando l'utente scrive lo status un valore nuovo diviene "vero" e resta "vero" finché il temporizzatore scade o viene ricevuta una nuova variabile di sincronizzazione.

4.6 Prontezza asincrona

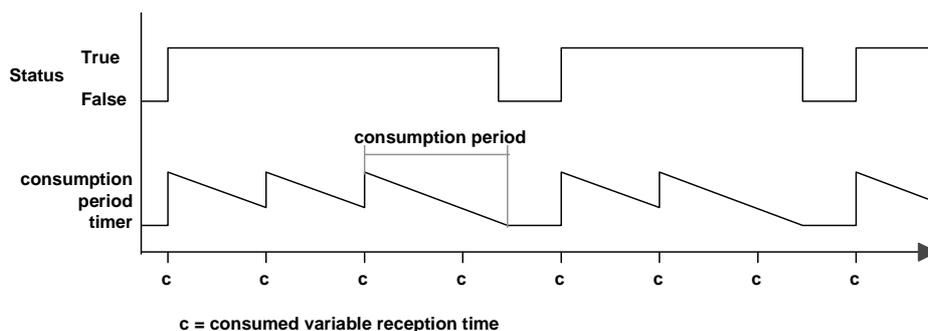
Gli stati di prontezza asincrona sono elaborati dall'Application layer dei consumatori delle variabili identificate. Per ogni variabile consumata in ciascuna entità di comunicazione occorre indicare se è stato calcolato o meno, lo stato di "prontezza". Se lo stato di prontezza è richiesto, occorre indicare anche il periodo di consumo associato alla variabile.



Quando la prontezza asincrona è "true" ciò significa che il meccanismo di trasferimento del buffer ha operato correttamente, cioè il Bus Arbitrator ha rispettato il periodo di schedulazione delle variabili ed i data link layer del produttore e del consumatore funzionano in modo corretto.

Nell'esempio mostrato, una stazione elabora lo stato di prontezza asincrona mentre l'altra non lo fa. Quando la prima stazione legge il valore essa ottiene sia il valore della variabile che lo stato di prontezza. In tal modo, grazie alla coppia di informazioni, la stazione può conoscere il produttore ha rispettato o meno il periodo di produzione e se il meccanismo per il trasferimento di buffer ha lavorato correttamente.

Il diagramma sotto mostra il meccanismo usato per elaborare lo stato di prontezza asincrona, utilizzando il tempo trascorso e l'istante in cui viene ricevuto un nuovo valore.



Quando una entità di comunicazione riceve un nuovo valore, essa setta il timer di prontezza asincrona associato alla variabile consumata, con il valore del periodo di consumo. Lo stato di prontezza diviene "true" e rimane con tale valore finché non scatta il timer.

4.6 Consistenze Spaziali e temporali

Il Livello Application di WorldFIP fornisce un servizio per la lettura di liste di identificatori. Una lista è composta da una collezione di variabili consumate. Questo servizio legge globalmente tutti i valori che compongono la lista e fornisce all'utente un "super-status" che lo informa sulla freschezza dell'informazione che consuma. Può fornire anche un status della consistenza spaziale che indica

che tutte copie di questa lista sono identiche in tutte le entità di comunicazione che consumano la lista.

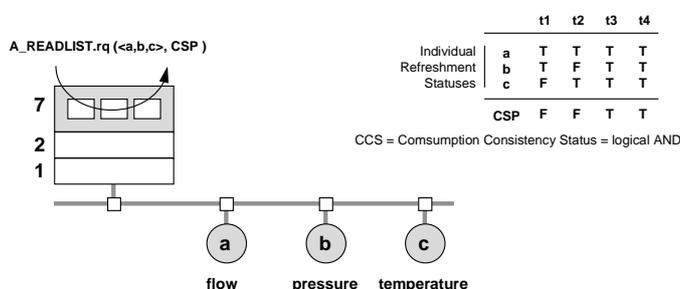
• **Consistenza della produzione temporale**

Le consistenze temporali sono status booleani che possono essere opzionalmente elaborate dai consumatori di una lista di variabili identificate. Consistenze temporali possono essere in riferimento a produzione o consumo. Possono essere asincroni o puntuali.

Nell'esempio sotto tre sensori (flusso, temperatura, pressione) producono ciascuno un valore al quale è associato un status di rinfresco. La stazione consuma l'elenco composto di queste tre variabili.

Il Livello Application del consumatore dell'elenco consuma tre coppie <valore, status> e applica un AND logico agli status di rinfresco individuali. Se tutti questi status sono veri, lo status della consistenza della produzione temporale è vero; altrimenti è falso.

L'utente, mediante una singola operazione di lettura, saprà se tutti i produttori hanno rispettato i periodi della produzione associati alle variabili che producono.



• **Consistenza spaziale**

La consistenza spaziale è un status booleano elaborato opzionalmente dai livelli applicativi che consumano una lista di variabili. Il principio è relativamente semplice: se tutte le copie dell'elenco sono identiche lo status è vero, altrimenti è falso. L'implementazione pratica di questo principio è però molto complesso poiché la singola stazione non può sapere se l'informazione che ha ricevuto è uguale a quella delle altre stazioni. A tale scopo, è richiesta la costruzione di cicli speciali elementari, lo scambio di variabili di consistenza individuali spaziali, e meccanismi per ripartenza in caso di ricevimento difettoso.