Uso di Ethernet nelle applicazioni di Automation: Problematiche, soluzioni e sviluppi

Diffusione di Ethernet

- Ethernet è uno standard di fatto per le applicazioni di Office Automation.
- Vantaggi
  - Basso costo
  - Semplice Installazione
  - Affidabilità
  - Elevato Throughput
  - Ampia disponibilità di componenti Hw e Sw

  -Lo stack TCP/IP su Ethernet è ampiamente disponibile e permette l'uso di Application layer protocols quali FTP, HTTP, ecc.

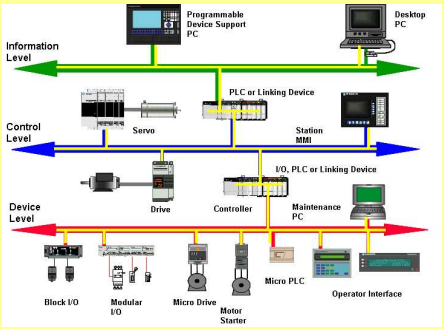
Ethernet nell' Automazione

Caratteristiche dei sistemi di automazione

- Architettura tipicamente gerarchica
- Differenti protocolli sono usati nei vari livelli.

E' possibile usare Ethernet nelle applicazioni di Automazione?

Ethernet permetterebbe agli utenti di usare lo stesso protocollo in tutti I livelli della gerarchia



- Perché non è stato usato nel passato?
  - Tempi di ritardo Indeterministici
  - Problemi con traffico real-time
  - Mancanza di un Application Layer specifico

Ethernet nell'Automation

- Il principale ostacolo nell'uso di Ethernet nelle applicazioni real-time è stato il suo non-determinismo.
- Il protocollo CSMA/CD non può garantire deterministicamente che una frame sarà consegnata entro una deadline prestabilita.

Infatti

- I tempi di ritardo forniti da Ethernet possono essere valutati solo su base statistica (valori medi)
- Non c'è nessuna garanzia che una variabile sia consegnata entro un definito tempo di ritardo.
- Il numero di stazioni, il workload e la dimensione dei dati influenzano pesantemente le prestazioni.

Prestazioni al variare del numero di stazioni

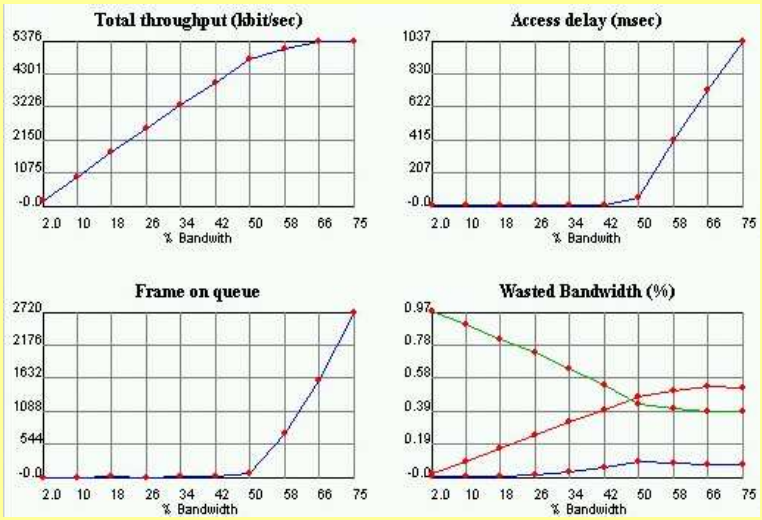
In questo set di prove si vuole osservare in quale maniera il numero di stazioni influenza il comportamento di ethernet, quando il workload complessivo varia in tutte le prove tra due valori fissati (da Rete scarica, a condizioni di Saturazione).

- Workload tot. min : 200 kbit/sec
- Workload tot. max : 7.5 Mbit/sec
- Data rate : 10 Mbit/sec
- Average frame length : 1000bit
- Funzione di distribuzione del traffico : lineare

Il workload sull'asse delle ascisse è misurato come percentuale della larghezza di banda del canale (10Mbps) disponibile.

Prova 1.1 : 5 stazioni

Il workload per ogni stazione varia da 40 a 1500 frame/sec

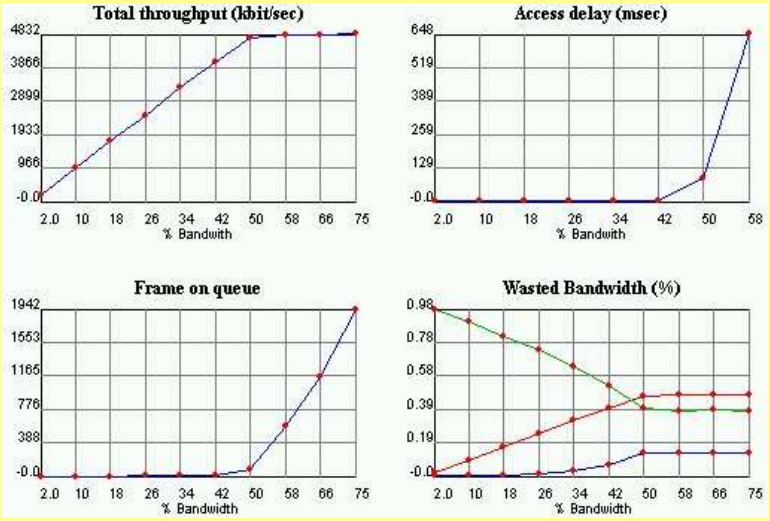


Banda non usata

Sprecata in collisioni

Prova 1.2 : 10 stazioni

Il workload per ogni stazione varia da 20 a 750 frame/sec

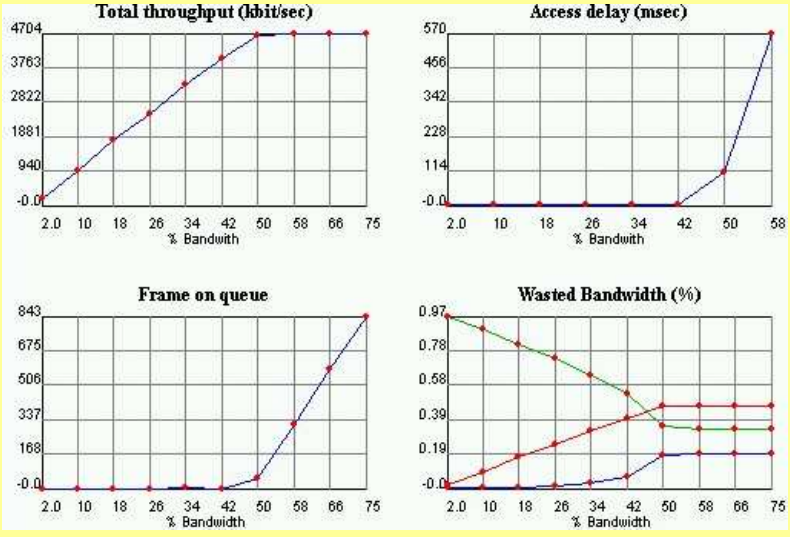


Banda non usata

Sprecata in collisioni

Prova 1.3 : 20 stazioni

Il workload per ogni stazione varia da 10 a 375 frame/sec



Banda non usata

Sprecata in collisioni

Commenti generali

- Come si puo' osservare, l'aumento rapido del numero di frame in coda è correlato al raggiungimento della saturazione del canale: quando il sistema va in saturazione il tempo di servizio di ogni nodo è maggiore del tempo di generazione delle frame.
- Si produce una crescita indefinita della coda.
- Comportamento simile per l'access delay. Si noti che in saturazione il grafico non tiene conto delle frame rimaste in coda che non vengono mai servite. Pertanto le curve, in saturazione, hanno solo un valore qualitativo.
- Ovviamente anche la banda sprecata nelle collisioni non aumenta piu' quando si raggiunge la saturazione.

Analisi del Throughput

- Gli effetti principali dell'aumento del numero di stazioni sul throughput sono due :
  - Si anticipa leggermente il raggiungimento della saturazione
  - Il throughput raggiunto in condizioni di saturazione diminuisce all'aumentare del numero di stazioni

L'aumento del numero di stazioni comporta un incremento nel numero delle collisioni con conseguente aumento di banda sprecata (vedi dispositiva successiva).

5 stazioni

10 stazioni

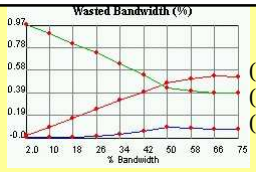
15 stazioni

20 stazioni

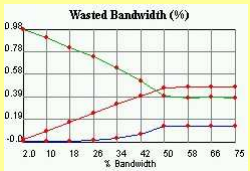
Analisi dello spreco di Banda

- Gli effetti principali sull'utilizzo della banda dell'aumento del numero di stazioni sono :
  - Aumento della banda sprecata nelle collisioni
  - Diminuizione della banda inutilizzata
  - Leggera diminuzione della banda utilizzata per le trasmissioni dati.

LEGENDA
(1) Banda utilizzata
(2) Banda sprecata
(3) Banda persa in collisioni



5 stazioni
(1)
(2)
(3)



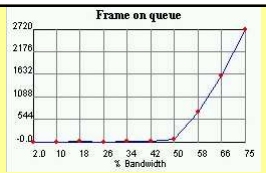
10 stazioni
(1)
(2)
(3)

15 stazioni
(1)
(2)
(3)

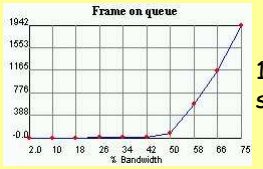
20 stazioni
(1)
(2)
(3)

Analisi delle Frame in coda

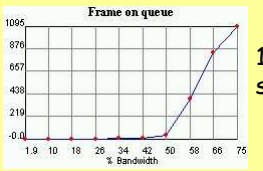
- All'aumentare del numero di stazioni il numero di frame in coda per ogni stazione diminuisce perché a parità di workload totale la frequenza di generazione in ogni stazione è minore.
.
- Ad esempio, nel caso 10 stazioni si ha una generazione max di 750 frame/sec per ogni stazione , mentre nel caso di 20 stazioni è di 375 frame/sec. Quindi in condizioni di congestione la coda del primo caso crescerà piu' velocemente rispetto a quella del secondo caso.
- Se invece consideriamo il numero totale di frame in coda, esso è maggiore quando è maggiore il numero di stazioni. Infatti come già visto all'aumentare delle stazioni diminuisce il throughput complessivo a causa dell'aumento delle collisioni.



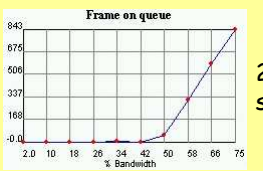
5 stazioni



10 stazioni



15 stazioni



20 stazioni

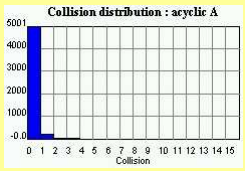
Analisi della Distribuzione di Collisioni (1/2)

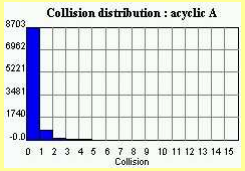
- Si puo' osservare che, per bassi valori di workload la distribuzione delle collisioni non è influenzata dal numero di stazioni. Il basso valore di workload fa si che la differenza del numero di collisioni tra i due casi sia esigua.

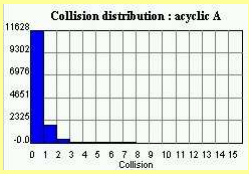
(la prima colonna si riferisce al numero di frame senza collisioni)

Bandwidth (%): 10 % 18 % 26%

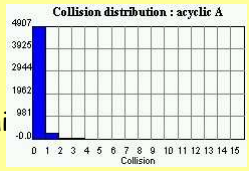
5 stazioni

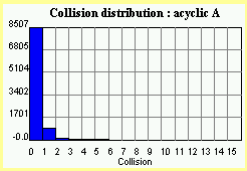


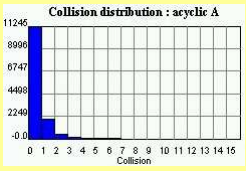




20 stazioni







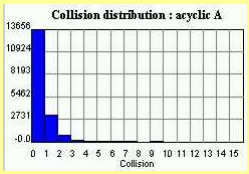
Analisi della Distribuzione di Collisioni (2/2)

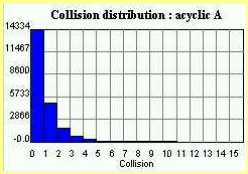
Quando il workload si avvicina al valore di saturazione la distribuzione delle collisioni :

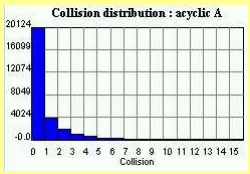
- Mantiene nei due casi un andamento simile, a testimonianza del comportamento fair dell'algoritmo BEB, ma il numero delle collisioni è comunque maggiore quando il numero di stazioni è maggiore.

Bandwidth (%): 34 % 42 % 59%

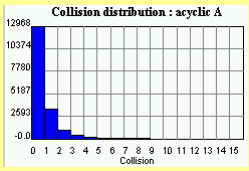
5 stazioni

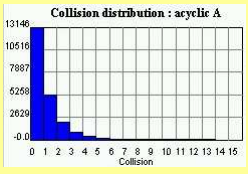


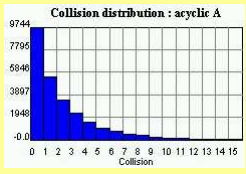




20 stazioni





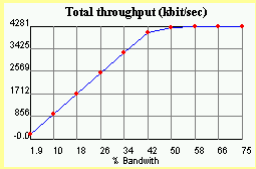


Il vincolo della lunghezza minima delle frame.

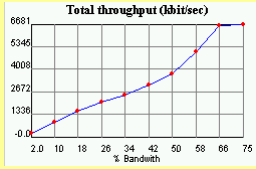
- Ethernet impone una lunghezza minima delle Frame
- Occorre evitare che una stazione possa terminare la spedizione di un pacchetto prima che il primo bit abbia raggiunto l'estremità più lontana del cavo (dove potrebbe collidere con un altro pacchetto) e sia ritornato indietro.
- In una LAN a 10Mbit/sec, lunga 2500 m un pacchetto impiega almeno 51.2 microsecondi per raggiungere l'estremità piu' lontana e ritornare. Cio' equivale ad una lunghezza frame di 512 bit, ovvero 64 byte.
- E' interessante verificare il comportamento della rete quando la lunghezza frame media è al di sotto della soglia di 512 bit
- Sono state eseguite due prove con lunghezza frame di 400 bit e 600 bit.

Effetto sul Throughput

Come si vede, nell'intervallo che precede le saturazioni il throughput è maggiore nel caso di lunghezza 600 bit .



Frame 600 bit



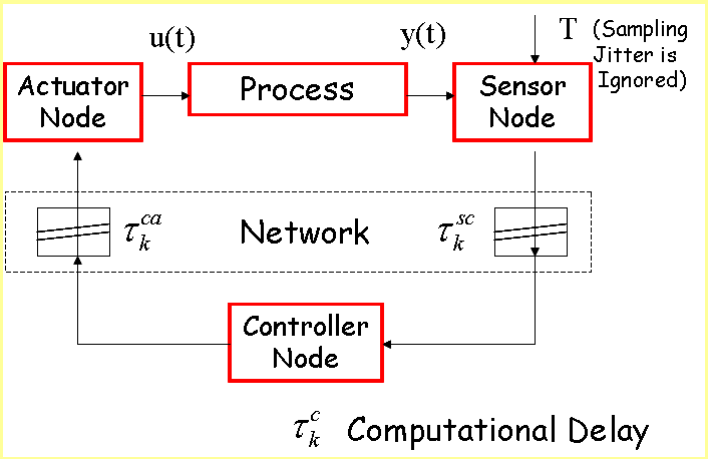
Frame 400 bit

%bandwidth	Frame perse (600 bit)	Frame perse (400 bit)
2 %	0	0
10%	0	28
18%	0	987
26%	0	3493
34%	0	7351
42%	0	12378

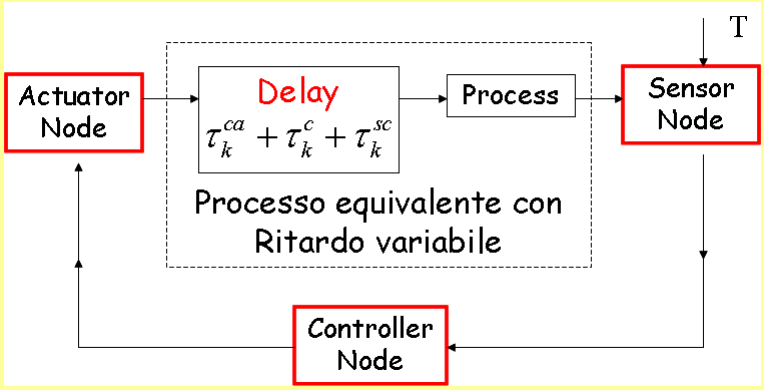
Nel secondo caso le frame possono essere perse, ovvero entrare in collisione senza che la stazione che le trasmette se ne accorga. Il Throughput apparentemente è più elevato

Perché il Tempo di Ritardo è importante

- In un sistema distribuito di automazione le rete introduce dei ritardi fra il Controllore ed il Sensore/Attuatore



Perché il Ritardo è indesiderabile



Il ritardo in un control loop riduce il margine di fase. Ciò può causare instabilità ed in ogni caso degrada le prestazioni del sistema.

Strategia per ridurre il Ritardo: Traffic smoothing

Definizione di canale real-time statistico:

-Se *n* è il numero di tentativi necessari per trasmettere un pacchetto con successo, diremo che esso è stato spedito tramite un canale real-time statistico se vale la seguente condizione:

$$P(n \leq K) > 1 - Z$$

dove Z è una prefissata probabilità di perdita.

Sia Dk* il ritardo, nel caso peggiore, subito da un pacchetto quando viene trasmesso con successo al k-esimo tentativo. Allora si ha la seguente relazione:

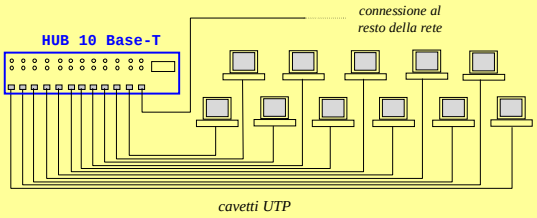
$$P(D \leq D_K^*) > 1 - Z$$

Condizione sufficiente affinché valgano le equazioni *di sopra* è che il *rate* di generazione di nuove *frame* rimanga sotto una soglia prestabilita chiamata:

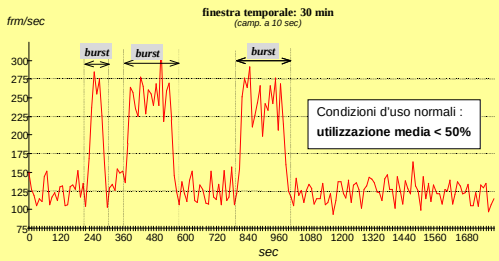
"Network-wide input limit"

Ambiente di riferimento: Il segmento 10BaseT

Realizzato tramite HUB (ripetitore multiporta)



- Il tipico "trace" del traffico di un segmento in normali condizioni operative, alterna periodi di basso carico, a burst di traffico.



Smoothing

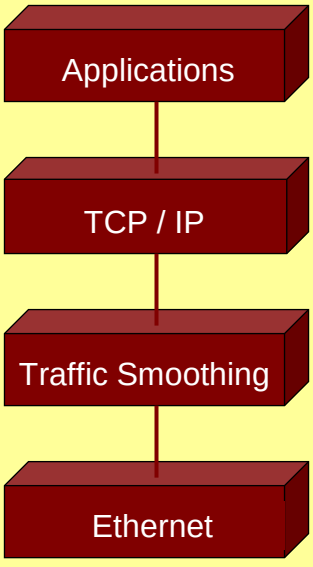
- Condizione sufficiente per garantire, con una prefissata probabilità, che un pacchetto possa accedere al canale entro un tempo prestabilito è che la frequenza di generazione dei pacchetti sia al di sotto di una soglia denominata *network-wide input limit*
- A ciascuna stazione viene quindi assegnata una porzione del network-wide input limit, che viene definita *station input limit*.
- Ciascuna stazione è responsabile di regolare il proprio flusso di pacchetti.

Traffic Smoothing

Il "Traffic Smoothing" distingue il traffico real-time da quello non real-time: il traffico real-time viene subito trasmesso, mentre quello non real-time viene trasmesso secondo l'algoritmo del "credit bucket".

In base alla modalità di assegnamento dello *station input limit*, si distinguono due tipi di traffic smoothing:

- Static Smoothing
- Dynamic Smoothing



Static Smoothing

Il valore dello *station input limit* è assegnato ad ogni stazione a priori, in base alle proprie necessità di larghezza di banda.

Limiti

- Poca flessibilità e scalabilità
- Spreco di larghezza di banda

Dynamic Smoothing

Il valore dello *station input limit* è assegnato ad ogni stazione **dinamicamente**, al variare del carico istantaneo della rete.

Vantaggi

- Maggiore flessibilità e scalabilità
- Il *network-wide input limit* viene ripartito solo tra le stazioni che stanno effettivamente trasmettendo

Limiti

- Dipendenza dalla tecnica di monitoraggio della rete
- Efficacia e velocità del metodo di calcolo dello *station input limit* da assegnare alla stazione in funzione del carico rilevato

Problematiche di monitoraggio della rete

- Dynamic smoothing richiede di acquisire la conoscenza dello stato attuale della rete, attraverso il monitoraggio del trend del traffico

E' necessario attivare degli opportuni processi in ogni macchina per la misura degli indicatori di traffico

- Misura del throughput
- Misura del ritardo
- Misura del numero di collisioni.

Quanta banda per ogni stazione?

- Le informazioni sul workload vanno collezionate per un periodo di tempo P la cui granularità influenza la dinamica della risposta dello smoother.
- La nuova banda che va assegnata ad una stazione è data da:
  - $V_{new} = V_{old}$ **+ quickness** $_*$ **(target-measured_value)**
- **quickness:** determina di quanto il nuovo valore può differire da quello precedente.
- **Target:** valore ottimo da raggiungere.
- **measured_value:** ammontare medio della Banda usata nel precedente periodo di osservazione.

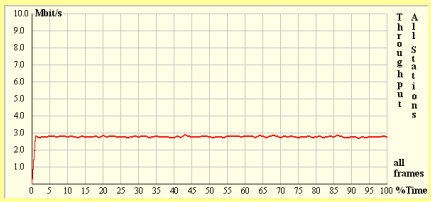
L'algoritmo di dynamic smoothing

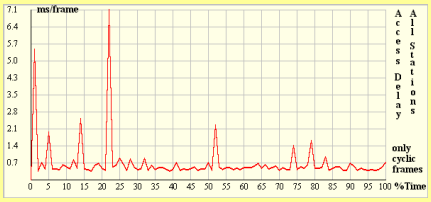
- Una porzione della Banda totale viene assegnata al traffico bursty (non real-time traffic)
- La porzione di banda viene convertita in bytes_per_tick(1 tick=10ms)

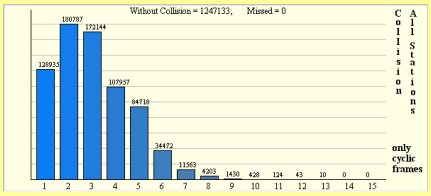
$$bytes\_per\_tick = \frac{bit\_per\_second}{8 \cdot 100}$$

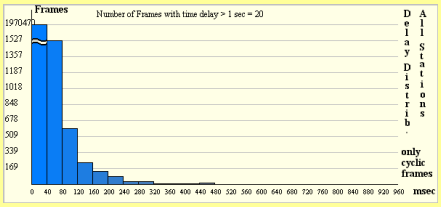
- Ogni nuova frame è convertita nel numero di ticks richiesti per la sua trasmissione.
- Una funzione scheduler contiene il tempo a cui una nuova frame può essere spedita
- Se arrival_time_of_frame< scheduled_time $\Longrightarrow$ la trasmissione è rimandata
- Altrimenti, la stazione possiede un certo numero di crediti e può trasmettere.
- Si fa uso del noto leaky bucket algorithm.

Comportamento in assenza di Burst





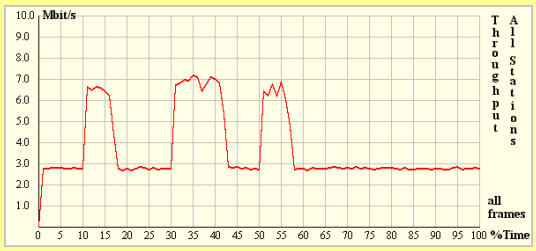




- Il throughput nella prima figura si riferisce a tutte le stazioni.
- Le altre figure si riferiscono solo al traffico ciclico.
- Il tempo di ritardo si mantiene basso (con qualche picco).
- Non vengono perdute frame.

Comportamento in presenza di Burst

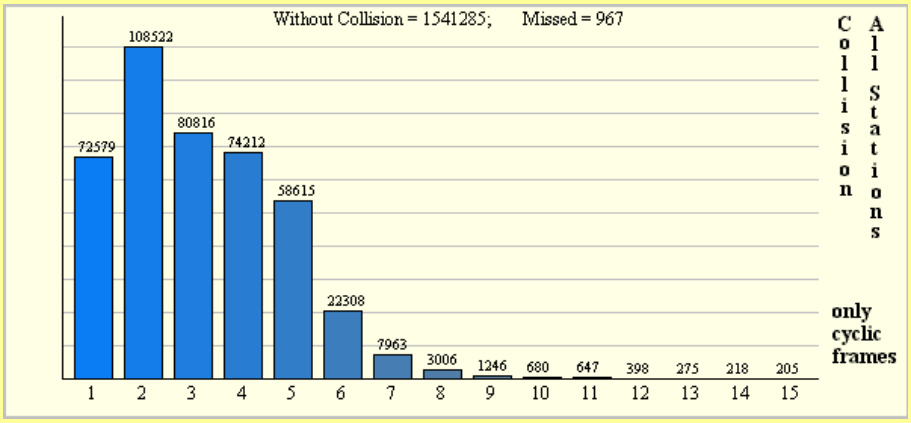
Aggiungiamo alcune stazioni che producono bursts di traffico non real time



- Il throughput cresce rapidamente quando le stazioni trasmettono un Burst di traffico.
- Il ritardo medio del traffico ciclico raggiunge valori molto elevati.

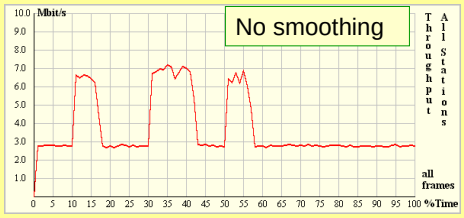
	Workload	Frame Length	Start Burst	End Burst
Station 11	1000 frames/s	12208 bits	10%	12%
Station 12	1000 frames/s	12208 bits	30%	32%
Station 13	1000 frames/s	12208 bits	31%	33%
Station 14	500 frames/s	12208 bits	50%	52%
Station 15	500 frames/s	12208 bits	53%	55%

Comportamento in presenza di Burst



- Molte frames vengono perdute poichè subiscono più di 15 collisioni

Traffico Bursty e Dynamic Smoothing

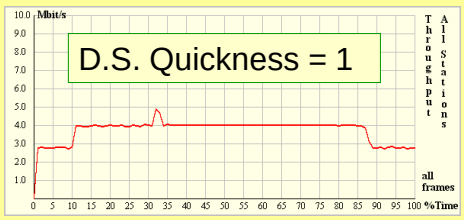


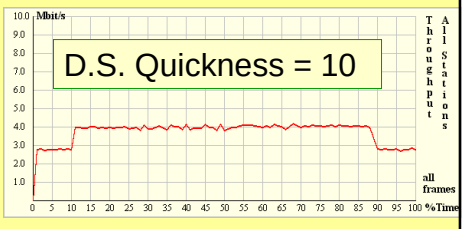
observation period P = 10 msec

Quickness = 1

Throughput limit = 40Kbits/P

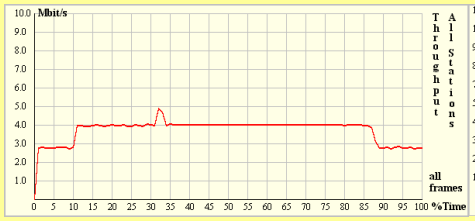
Al traffico Bursty è assegnata una banda massima di 1 Mbps

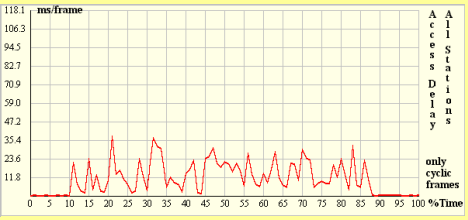


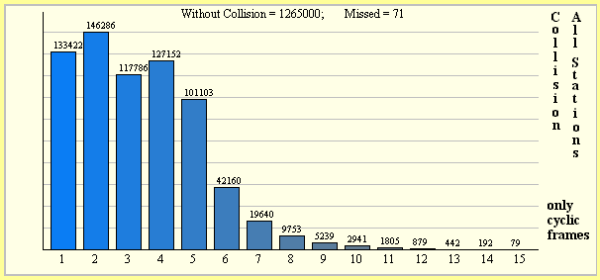


- Il filtro costituito dallo smoothing ha limitato la massima banda usabile a 4 Mbps

Traffico Bursty e Dynamic Smoothing

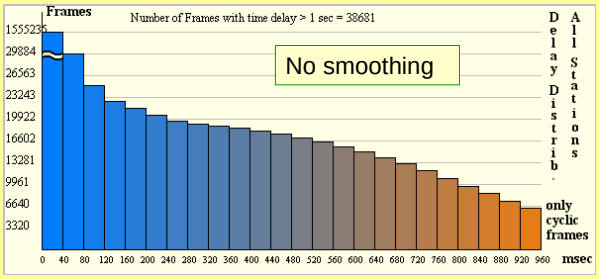




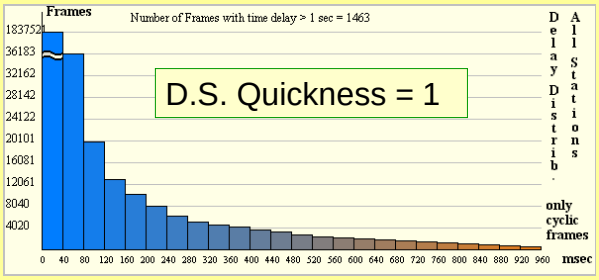


L'access delay medio è notevolmente più basso rispetto al caso senza smoothing

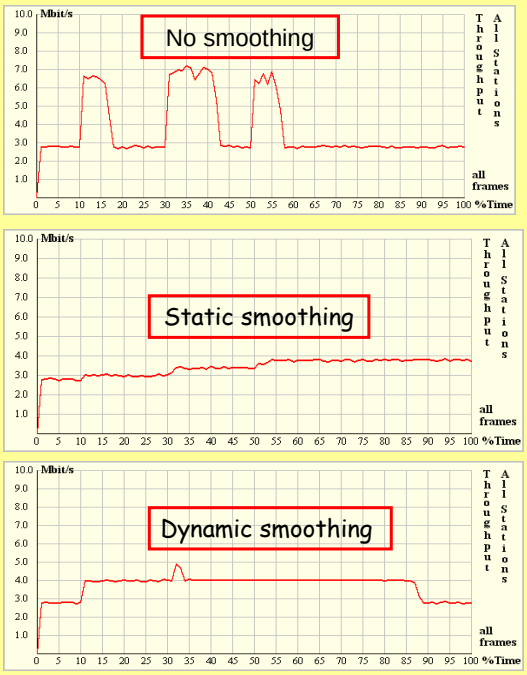
Traffico Bursty e Dynamic Smoothing



Un elevato numero di frames (38681) ha un ritardo che supera 1 Sec.



Il comportamento è fortemente migliorato col dynamic smoothing. Solo 1463 frames superano 1Sec.

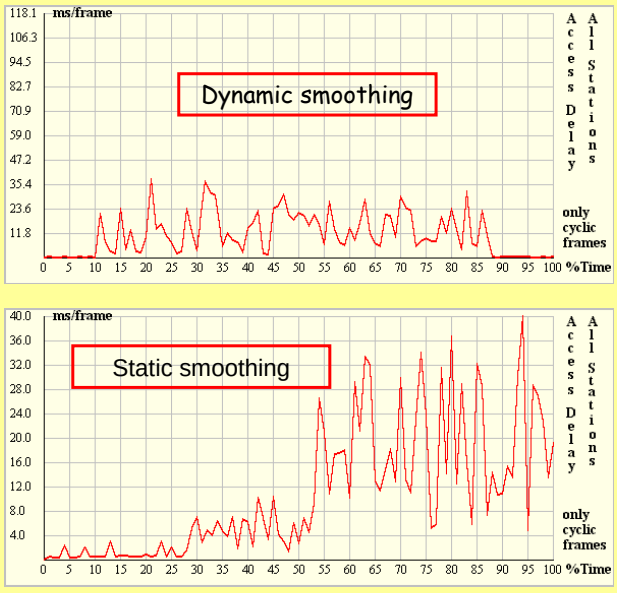


Dynamic versus Static Smoothing

Quando si usa lo static smoothing, il valore del "network-wide input limit" è suddiviso a priori fra le varie stazioni senza tener conto del carico effettivo della rete. Ciò impedisce un uso ottimale della banda.

Invece, il Dynamic smoothing permette alle stazioni real-time stations di superare questo limite senza violare il "network-wide input limit".

Dynamic versus Static Smoothing



Il comportamento in termini di delay differisce a causa della diversa abilità nell'usare la banda disponibile.

Una implementazione del Dynamic Smoothing: Harmonic Increase and Multiplicative Decrease

• *In assenza di collisioni*, lo *station input limit* viene incrementato periodicamente per mezzo di una costante **(incremento armonico)**.

• A seguito di una collisione, la trasmissione di pacchetti non real-time viene bloccata e lo *station input limit* viene decrementato per mezzo di un fattore moltiplicativo costante **(decremento moltiplicativo)**.

Limiti

• Una singola collisione non necessariamente indica un sovraccarico (può essere dovuta al caso)

• L'utilizzo di una singola collisione può determinare **instabilità**

• Regolazione dello smoothing sulla base di **valori costanti** non calibrati sull'effettivo carico della rete

• Necessità di migliorare la conoscenza del carico della rete (uso di più indicatori)

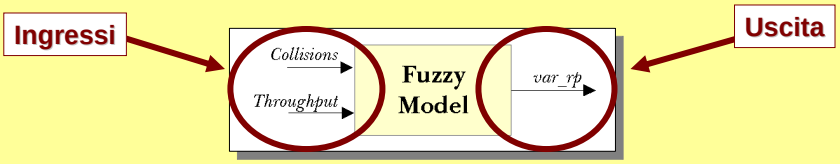
• Necessità di una regolazione differenziata in base al carico della rete

Il controllo Fuzzy

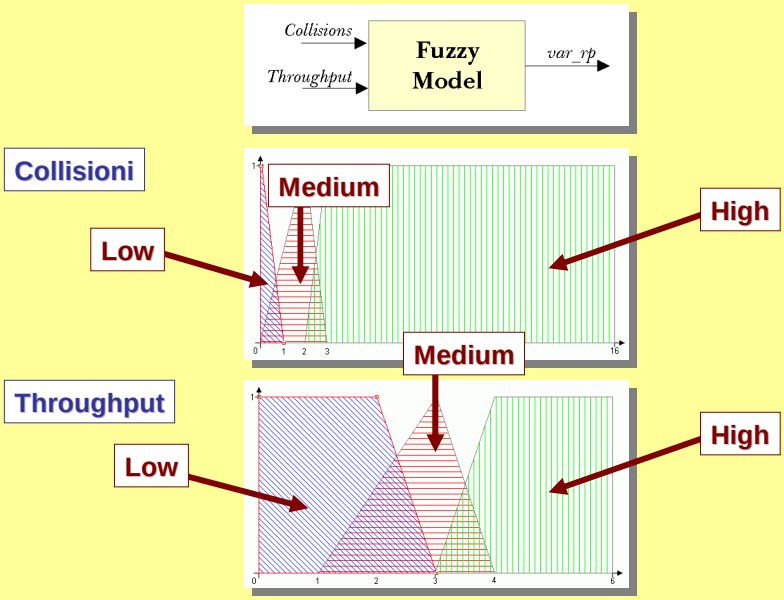
• Il sistema considerato, per la sua **complessità**, è difficile da modellare e controllare con i metodi tradizionali.

• Un controllo efficace richiede l'integrazione con le **conoscenze euristiche** acquisite dagli esperti "sul campo". Il controllo fuzzy permette di integrare facilmente nel controllore la conoscenza acquisita dall'esperienza.

• Il controllo fuzzy è generalmente **più robusto** delle tecniche "classiche" ed è in grado di offrire migliori prestazioni in presenza di variazioni delle condizioni di funzionamento (di parametri, di carico) e di disturbi esterni.



Il modello fuzzy: Membership functions



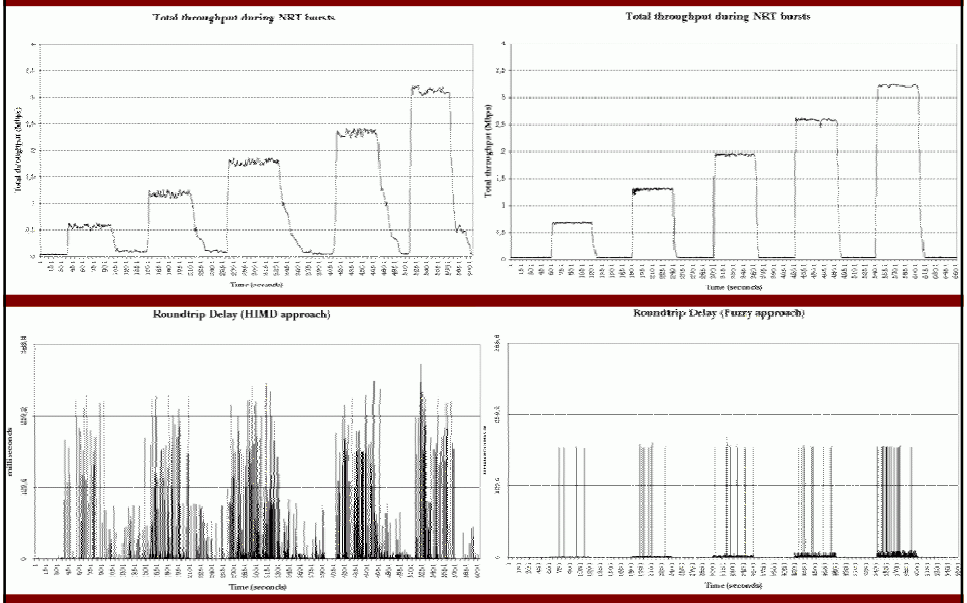
Il modello fuzzy: le regole

• I controllori fuzzy sono costituiti da un insieme di **regole linguistiche** e da un motore di inferenza che processa tali regole.

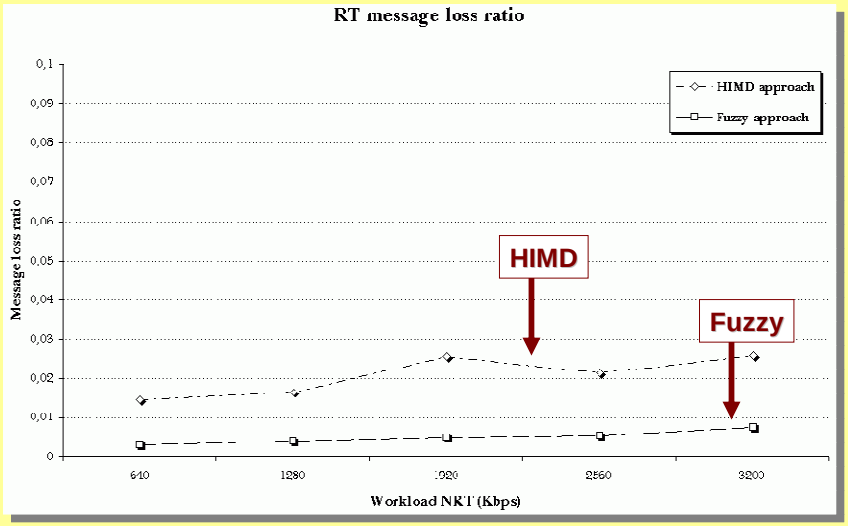
• Le regole fuzzy offrono una trasformazione tra la conoscenza "linguistica" di un esperto e le strategie di controllo automatiche di un controllore.

1) **IF** (Collisions is LOW *AND* Throughput is LOW) **Then** var_rp is -0.5*RP_{min}
2) **IF** (Collisions is LOW *AND* Throughput is MEDIUM) **Then** var_rp is -0.3*RP_{min}
3) **IF** (Collisions is LOW *AND* Throughput is HIGH) **Then** var_rp is 0
4) **IF** (Collisions is MEDIUM *AND* Throughput is LOW) **Then** var_rp is 0.1*RP_{min}
5) **IF** (Collisions is MEDIUM *AND* Throughput is MEDIUM) **Then** var_rp is 0
6) **IF** (Collisions is MEDIUM *AND* Throughput is HIGH) **Then** var_rp is 0.2*RP_{max}
7) **IF** (Collisions is HIGH *AND* Throughput is LOW) **Then** var_rp is RP_{max}
8) **IF** (Collisions is HIGH *AND* Throughput is MEDIUM) **Then** var_rp is 0.7*RP_{max}
9) **IF** (Collisions is HIGH *AND* Throughput is HIGH) **Then** var_rp is 0.7*RP_{max}

Confronto HIMD / Fuzzy

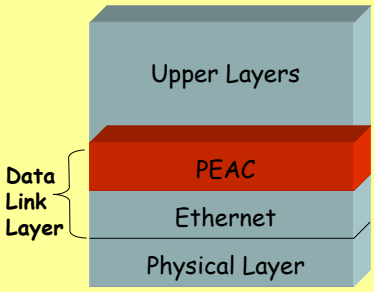


Confronto HIMD / Fuzzy



Una Soluzione per la comunicazione Hard Real-Time su Ethernet :il protocollo PEAC

Il PEAC (Predictable Ethernet Access Control) è collocato nei livelli superiori del Data Link Layer.



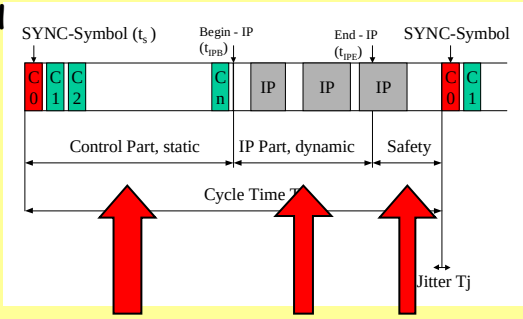
- Consente la gestione integrata di traffico Hard Real-Time(HRT) e Non Real-Time(NRT)
- Realizza un TDMA per la schedulazione del traffico HRT tale da garantire l'accesso al mezzo in modo deterministico
- E' indipendente dalla tecnologia di rete adottata, shared o switched

Politica di gestione della banda in PEAC

La gestione della banda nel PEAC è basata sul concetto di Cycle Time

Il Cycle Time è composto da :

- **Control Part in cui ad ogni stazione è assegnato uno slot di tempo (TDMA)**
- **IP Part dedicata alla schedulazione del traffico NRT**
- **Safety Margin**

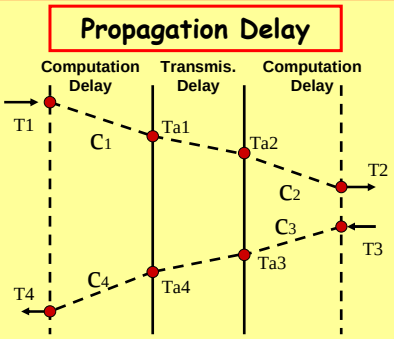


Analisi Temporale di PEAC

Obiettivo : **Calcolo delle lunghezze di Control Part, IP Part e Safety Margin per un dato Cycle Time**

L'analisi si compone di due fasi :

1) Calcolo del tempo di trasmissione di un pacchetto di controllo
2) Schedulazione



Round trip delay = ((T4 – C4) – (T1 + C1)) – ((T3 + C3) – (T2 – C2))

C1 C2 C3 C4 : tempi di computazione opportunamente misurati.

Schedulazione

Il PEAC Cycle Time è costituito da :

$$T_Z = T_{CPL} + T_{IPL} + T_{SM}$$

dove :

- T_{CPL} è calcolato in base al numero di stazioni presenti nella rete
- T_{IPL} e T_{SM} devono essere opportunamente dimensionati per non violare le garanzie Real-Time.

! T_{SM} dipende dalla topologia di rete, dal numero di nodi e dalla scelta di T_{IPL}.

Dimensionamento del T_{SM}

Sotto le seguenti ipotesi :

- **IP frame size fissa**
- **Numero max di IP frames per ciclo**

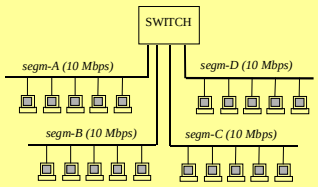
il caso peggiore si verifica quando tutte le stazioni inviano il burst di frames alla fine del T_{IPL}.

Se si verificano delle collisioni fra le frame trasmesse alla fine della IP part queste, a causa delle ritrasmissioni, possono sforare il tempo assegnato e sovrapporsi con la parte ciclica.

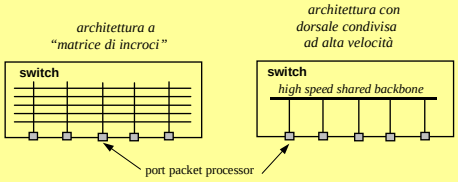
L'uso di uno switch può eliminare il problema.

Switch Ethernet

Una possibile soluzione alla indeterminazione del ritardo di Ethernet sta nell'uso dello switch.



Architettura interna a crossbar e basata su high speed backbone (velocità dell'ordine dei GHz)



Architetture:

- Architettura basata su circuiti integrati dedicati (ASICS).
- Architettura basata su processore multi-tasking di tipo general purpose (una CPU)
- Architettura di tipo misto con strutture ASICS ed una CPU.

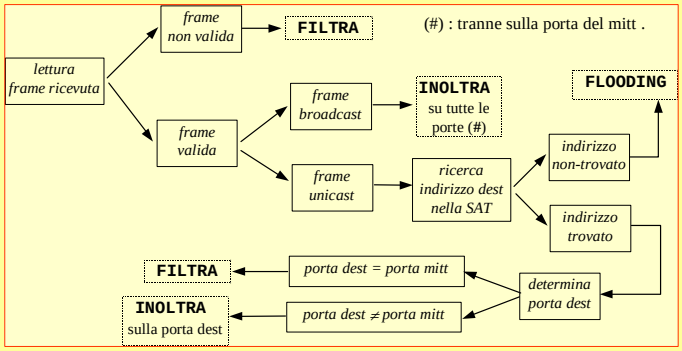
Categorie di Switch:

- Cut-through: switching ad alta velocità.
- Store and Forward: capacità di controllo di errori.

Switch Ethernet

- Appartiene alla categoria dei “Transparent Bridge”.
- Le funzioni di “Filtering & Forwarding” sono supportate tramite una SAT (Source Address Table).
- Inizialmente utilizza il flooding (quando la SAT è vuota).
- Backward learning.

Indici prestazionali:
- latenza (usec)
- throughput max
- Packet loss rate.



The FTT-Ethernet protocol

L’**FTT-Ethernet** protocol è stato sviluppato per soddisfare i requisiti dei sistemi con vincoli Hard real-time. Esso fornisce una composizione di:

- Scalabilità,
- Flessibilità
- Timeliness
- Efficienza.

E’ basato sul paradigma **Flexible Time-Triggered (FTT) che è stato sviluppato all**’Università di Aveiro, ed è stato utilizzato per una implementazione sul Controller Area Network (FTT-CAN)

Il paradigma FTT si basa su due caratteristiche principali:

- Centralized scheduling
- Master/multi-slave transmission control

The FTT-Ethernet protocol

Centralized scheduling:
la politica per il message scheduling e quella per la trasmissione dei messaggi sono localizzati in un solo nodo. (Master)

→ elevata Flessibilità

- Facilita i cambiamenti **on-line** sia del **message set che della scheduling policy**
- Facilita l'implementazione nel master di un meccanismo di **on-line admission control** → TIMELINESS

The FTT-Ethernet protocol

Master/multi-slave:
Lo stesso messaggio del Master viene usato per triggerare diversi messaggi in diversi nodi slave.

- Permette di rinforzare la **timeliness** del traffico nel bus
  (tipico del controllo di trasmissione di tipo Master-Slave)
- **Elevata efficienza di utilizzazione della Banda**
  (se confrontata con quella di un tipico Master-Slave transmission control). Un singolo messaggio del master triggera diversi messaggi in diversi slave.

The FTT-Ethernet protocol

Come opera?

- Il Traffico è allocato in **time slots** di durata fissa

(**Elementary Cycle - EC**)

- Dentro ogni EC possono esistere diverse finestre dedicate a tipi diversi di traffico.
- Il Bus time è organizzato in una infinita successione di Elementary cycles.
- Gli ECs iniziano con un **trigger message (TM)** spedito dal Master per sincronizzare la rete, allocare gli slot e indicare il ruolo (produttore/consumatore) degli slave.

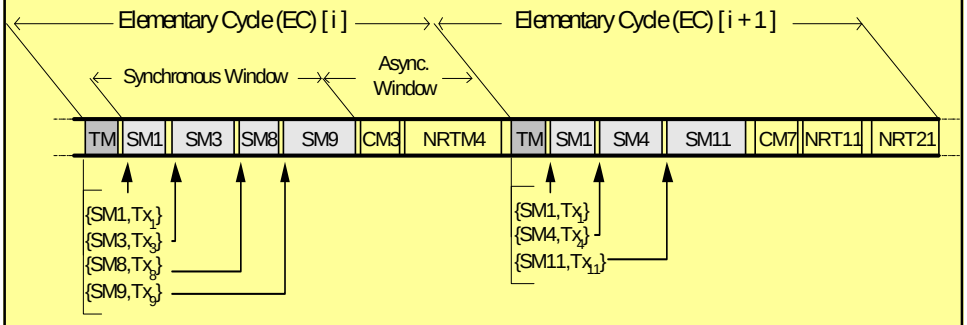
The FTT-Ethernet protocol

Gli ECs sono organizzati in due fasi (windows)

- **Synchronous window**
  - Contiene il traffico **time-triggered**
  - Il traffico è soggetto ad Admission Control e la sua Timeliness è garantita (traffico real-time)
  - Il **trigger message** contiene la **EC-Schedule**, cioè gli identificatori e gli istanti di trasmissione dei messaggi che sono schedulati per essere prodotti nella synchronous window
- **Asynchronous window**
  - Contiene il traffico **Event triggered** traffic, gestito con modalità best effort.
  - Si tratta in genere di **Non real-time** traffic, servito dal Master mediante polling.
  - Se richiesto è possibile pre-analizzare il traffico per supportare **Event triggered Real-time** traffic.

The FTT-Ethernet protocol

Struttura di un Elementary Cycle



Gli istanti di trasmissione dei messaggi sono specificati in modo da evitare collisioni e possibili sovrapposizioni delle finestre → Traffic Timeliness e Temporal Isolation.

FTT-Ethernet protocol

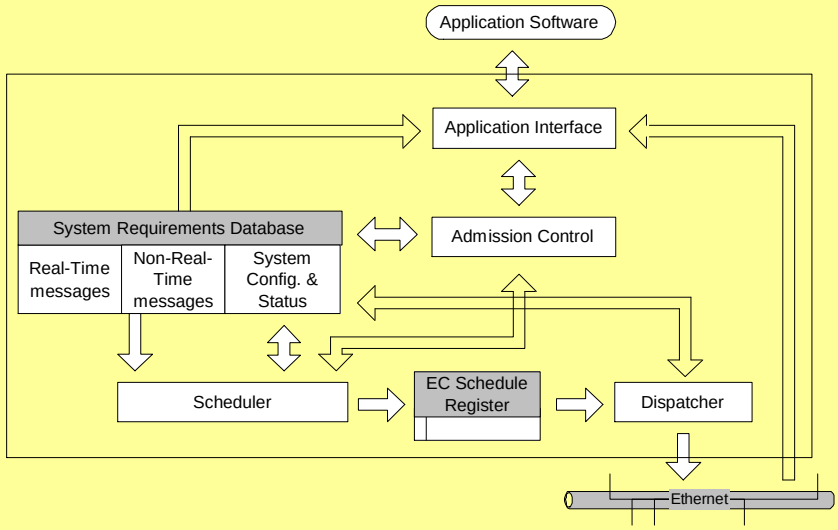
Il nodo Master:

- Mantiene un **database** contenente
  - System configuration (cioè durata dell'EC, ampiezza della synchronous window, numero di message streams, tx rate, ecc.)
  - Communication requirements : Synchronous requirements, Asynchronous requirements.

- **Costruisce** le **EC-schedules** in accordo alla politica di scheduling adottata (ad es. RM, EDF)
- Periodicamente invia in **broadcast** gli **EC trigger message** (contenenti i rispettivi EC- schedule)

FTT-Ethernet protocol

Architettura del Master node

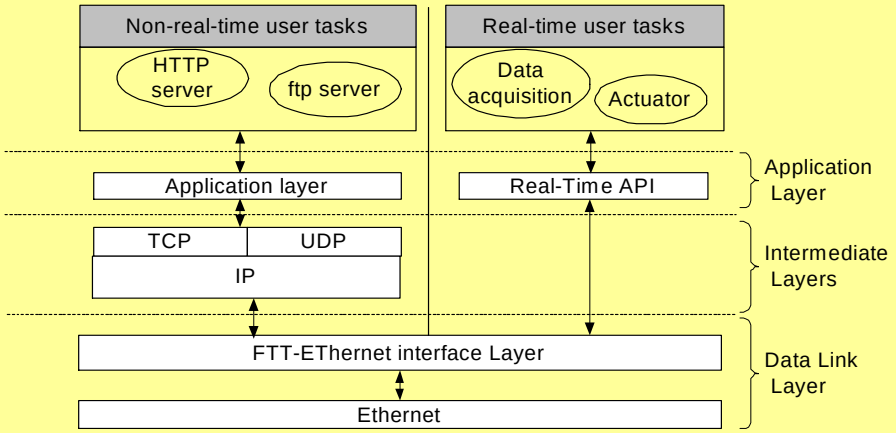


FTT-Ethernet protocol: Slaves

- Eseguono **l'application software** richiesto dall'utente
- Usano i **communication services** per:
  - definire i messaggi prodotti/consumati localmente
  - Aggiornano/leggono i valori delle entità real-time corrispondenti
  - Possiedono due **communication stacks**
  - Uno per traffico non-real-time (standard IP protocol suite)
  - Un altro per traffico real-time(3 layers model)
- L'accesso al bus Ethernet è realizzato attraverso lo **FTT-Ethernet Interface Layer** che :
  - Decodifica gli TM
  - Trasmette i messaggi negli istanti prefissati (rispettando la EC-schedule)
  - Inoltra i messaggi ricevuti al corretto communication stack.

FTT-Ethernet protocol: Slaves

Architettura dello Slave node



The FTT-Ethernet protocol

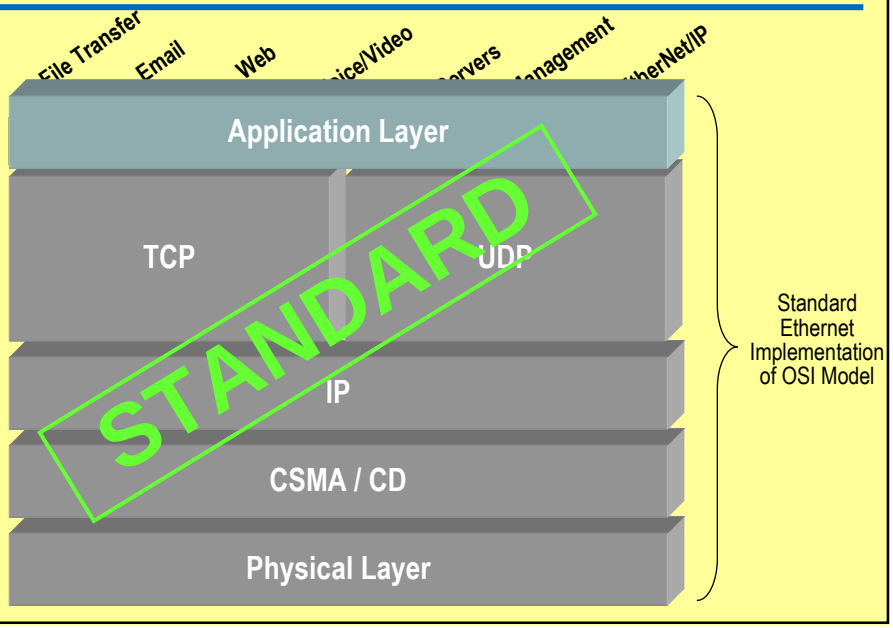
Quali problemi?

Gli aspetti negativi sono:

E' relativamente inefficiente nella gestione di traffico Event Triggered in Ethernet shared, a causa dei tempi imprevedibili di attesa legati all'arbitraggio.

Per mantenere la compatibilità fra gli Elementary Cycles ed il periodo dei messaggi, questo deve essere un multiplo intero dell'EC.

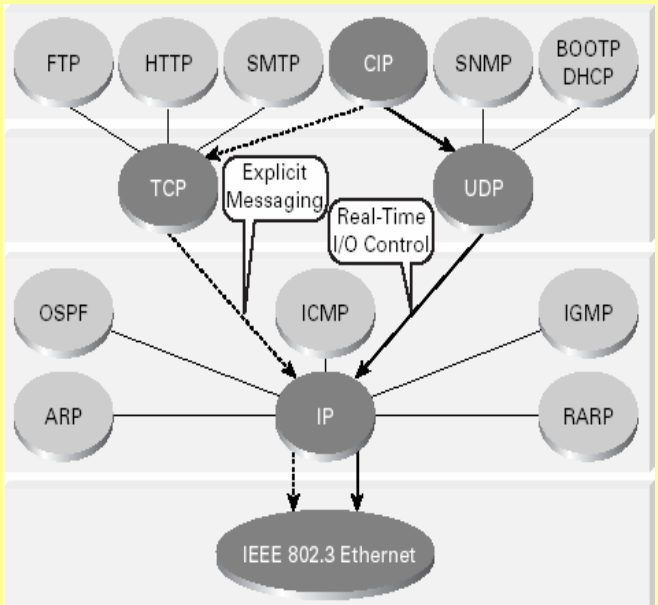
EtherNet/IP – Standard Protocol Model



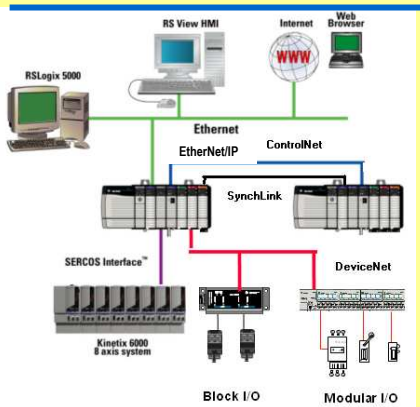
EtherNet/IP – Standard Protocol Model

- CIP (control information protocol): modello a oggetti mappato con tcp/udp/ip, incorpora le seguenti gerarchie di messaggi:
- I/O base: scambiare dati con racks di i/o , gestito tramite udp con modello client/server
- Upload/Download di parametri e programmi : messaggi sporadici "espliciti", usa tcp
- Polled, ciclic, event-driven data
- 1-to-1, 1-to-n , broadcast: tcp è intrinsecamente 1 a 1 . Usa udp quando molti device devono essere aggiornati in breve tempo , perché permette messaggi simultanei a vari nodi

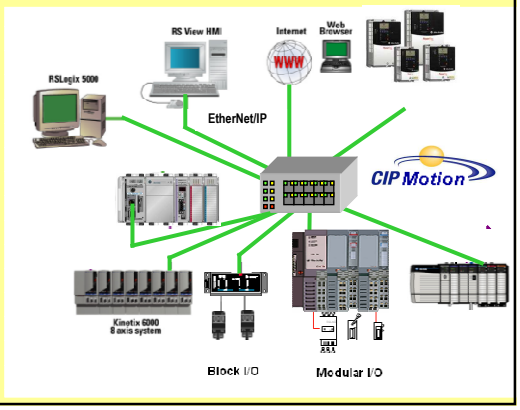
EtherNet/IP – Standard Protocol Model



EtherNet/IP – Flat Network Topology



- Diminuzione dei costi
- Migliori prestazioni
- Integrazione semplificata



L'idea finale è quella di migrare da una organizzazione a livelli ad una topologia Flat, in accordo al modello definito dalla "Society of Manufacturing Engineers" per gli AMS