Università degli Studi di Trieste
Dipartimento di Elettronica, Elettrotecnica ed Informatica

Tesi di Laurea in Sistemi di Telecomunicazioni

Studio e Specificazione del Protocollo CoAP per Sistemi Embedded

Relatrice:
Ing. Lia Deotto

Correlatore:
Valentino Zio

Laureanda:
Agata Leo

Anno accademico 2009-2010

Contenuti della presentazione

PARTE I

- **Introduzione**
- **Il protocollo CoAP**

PARTE II

- **Il model-checker Spin**
- **La specificazione e validazione del CoAP**

Conclusioni e sviluppi futuri

PARTE I

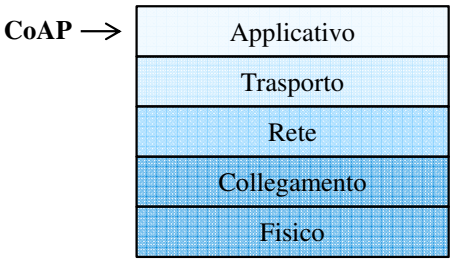
Introduzione
Il protocollo CoAP

CoAP

Constrained Application Protocol

Draft IETF (Internet Engineering Task Force)

- Applicazioni M2M (Machine to Machine)
- Ambienti “constrained”



Stack TCP/IP

Applicazioni M2M:

- Domotica
- Automazione degli edifici
- Controllo industriale
- Telematica sanitaria
- …

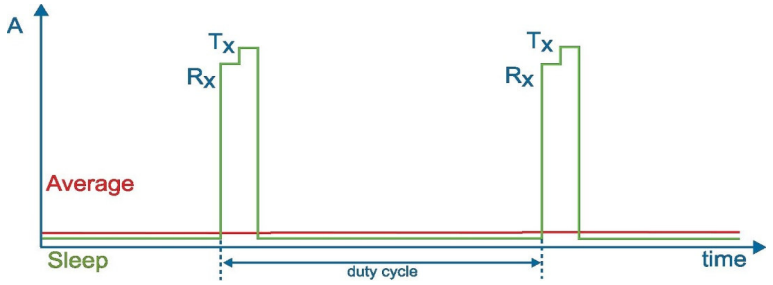
Ambienti “constrained”

Dispositivi embedded

- Semplici microprocessori
  - Microprocessori a 8 o 16 bit
  - Frequenze di clock intorno ai 16 MHz
- Poca memoria
  - Decine di kB di RAM
  - Decine di kB di ROM/Flash
- Potenza limitata
  - Alimentazione a batteria
    - Bassi consumi
    - Ciclo di lavoro ridotto (~0,1%)

Ad esempio:

Texas Instruments MSP 430
CPU 16 bit
Max CPU speed 25 Mhz
Low Power
RAM: 1 - 10 kB
Flash: 512 B - 60 kB



Ambienti “constrained”

Reti wireless personali

- VANTAGGI
  - Facilità di installazione
  - Flessibilità
  - Basso costo
- CARATTERISTICHE
  - Bassa potenza
  - Scarso throughput
  - MTU piccola
  - Alte perdite

Ad esempio:

Standard IEEE 802.15.4

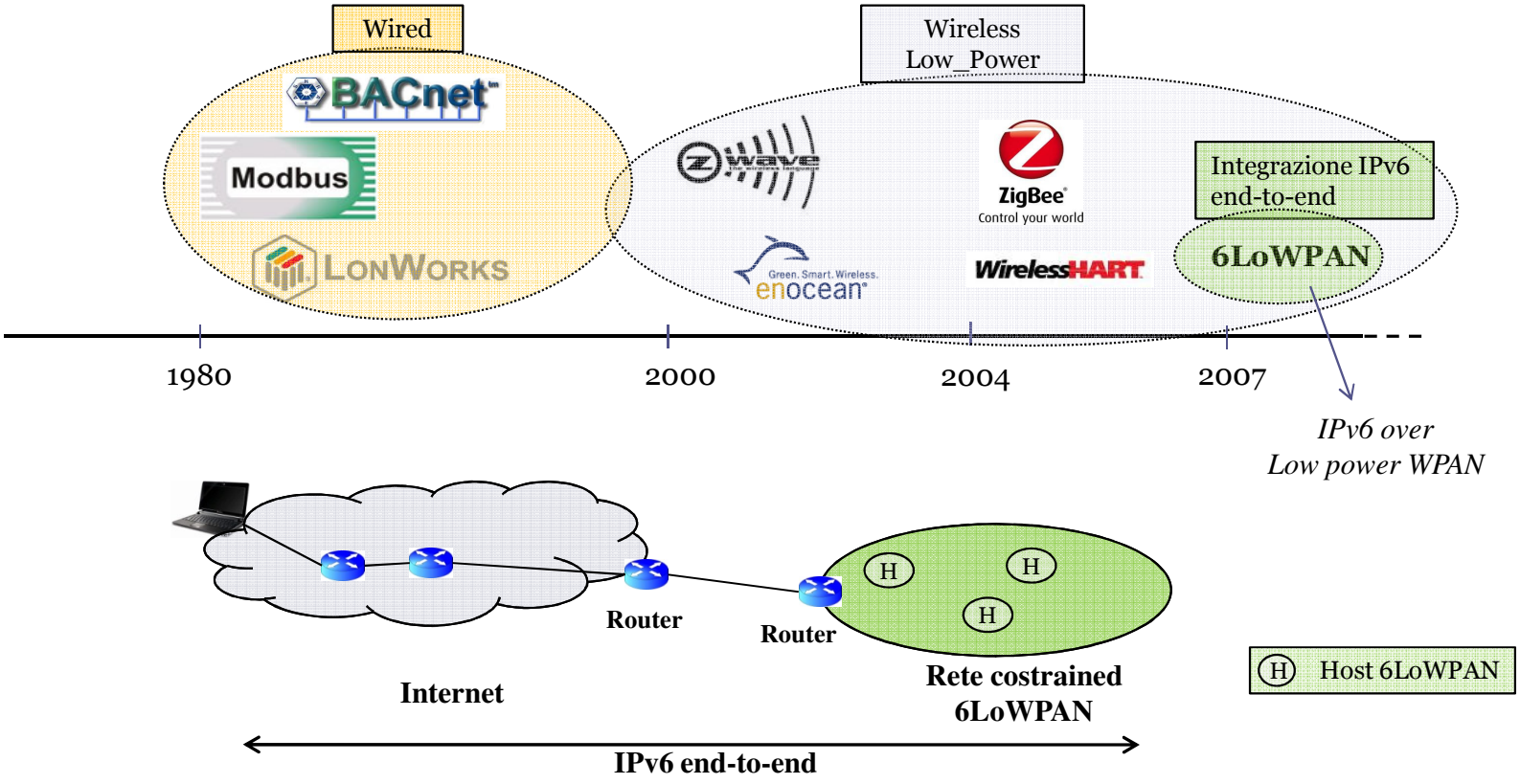
LR-WPAN

(Low Rate Wireless Personal Area Network)

- Output power: ~ 1 mW
- Raggio tipico di 10-20 m
- Massima bit-rate: 250 kbit/s
- MTU: 128 byte

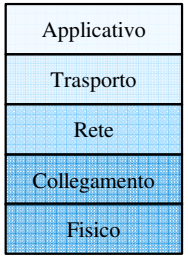
MTU: Maximum Transmission Unit

Comunicazione in applicazioni M2M

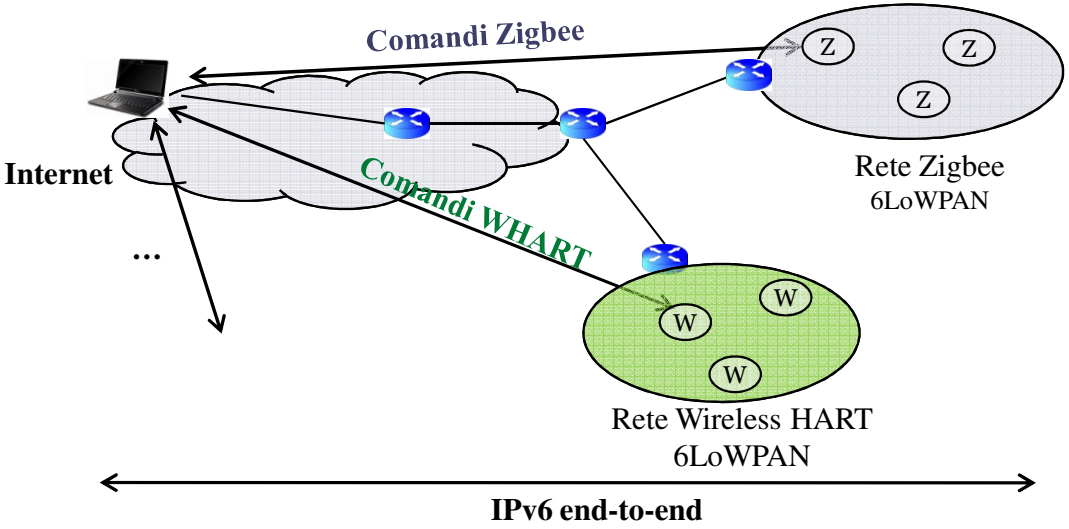


Applicazioni interoperabili?

- 6LoWPAN: IPv6 anche in ambienti constrained
- Ma svariati protocolli applicativi -> frammentazione



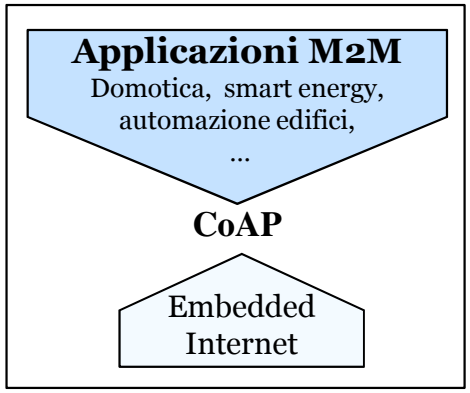
Stack TCP/IP



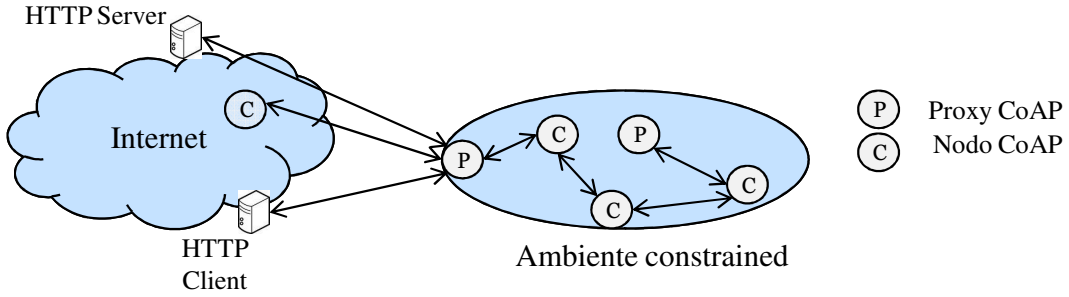
Embedded web

Vantaggi:

- Un solo protocollo applicativo per molte applicazioni M2M
- L'architettura del web è scalabile, flessibile ed efficiente



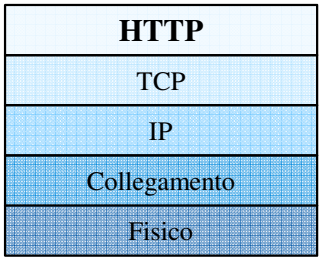
HTTP : web = CoAP : embedded web



Perché HTTP non è adatto?

HTTP

- Progettato per l'interazione con l'utente
- Progettato per il trasferimento di grandi quantità di dati
- Formato di messaggio "pesante"
- Modello di interazione "pull"



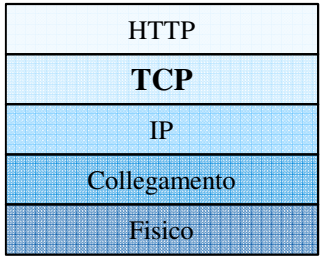
Le applicazioni constrained

- Devono essere progettate per l'interazione M2M
- Le interazioni sono spesso molto brevi
- Generano payload di piccole dimensioni
  - Poche decine di byte
- Hanno un payload disponibile limitato
- Devono risparmiare potenza e memoria
- Hanno bisogno anche di un modello di interazione "push"
- Hanno bisogno del resource discovery

Perché TCP non è adatto? (1/3)

TCP

- Connection-oriented
- Affidabile
- Controllo del flusso
- Controllo della congestione
- Unicast



Le applicazioni constrained

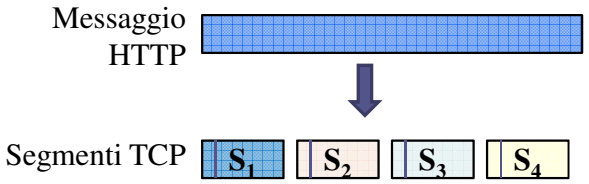
- Non sempre richiedono garanzie di affidabilità
- Possono aver bisogno di limitare la congestione di rete

  Ma con una tecnica più semplice
- Hanno bisogno del multicast

Perché TCP non è adatto? (2/3)

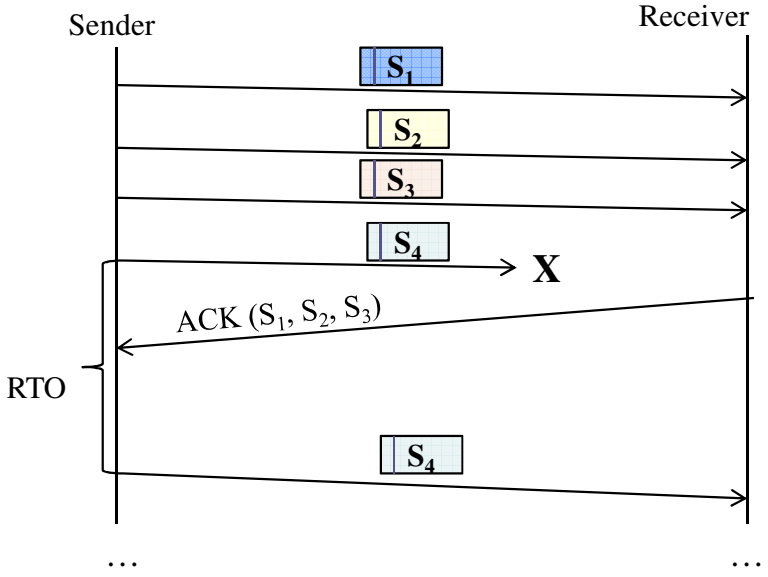
TCP: Tecnica *sliding window*

- Affidabilità
- Controllo del flusso



Implementare una tecnica sliding window a livello di trasporto è un'**ottimizzazione**:

- Per messaggi "**grandi**"
- Se **tutti** i messaggi hanno bisogno di garanzie di affidabilità



Perché TCP non è adatto? (3/3)

TCP: Controllo della congestione

TCP assume:

RTO scade => congestione

TCP sceglie il RTO in base alla stima del RTT (aggiornata dinamicamente)

Approccio opportuno solo quando il sender ha molti dati da trasmettere

Ambienti constrained:

- Alte perdite dovute alle caratteristiche del mezzo radio

~~**RTO scade => congestione**~~

- Interazioni di breve durata

Stima di RTT??

RTO: Retransmission Time Out
RTT: Round Trip Time

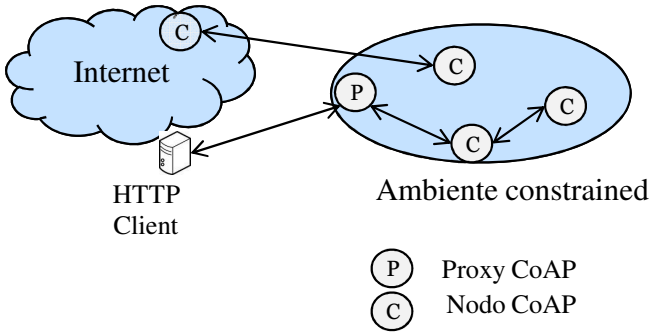
PARTE I

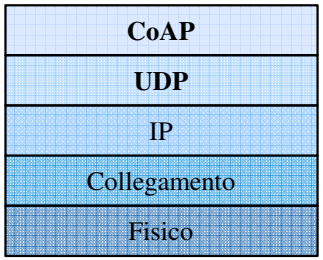
Introduzione

Il protocollo CoAP

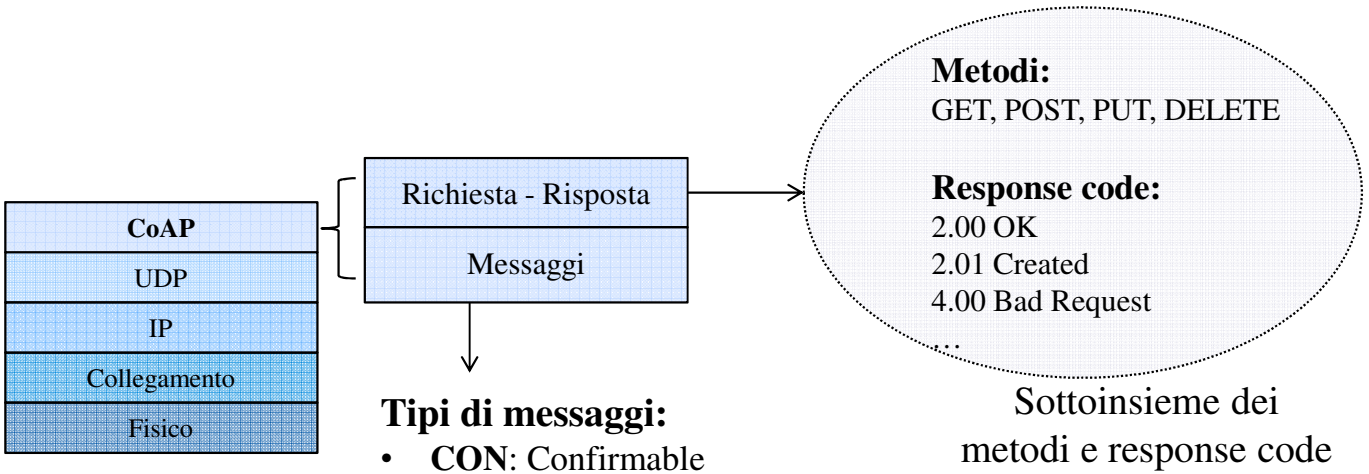
CoAP (1/2)

- Protocollo di trasferimento web embedded
- Simile ad HTTP
  - Mapping CoAP – HTTP
- Usa UDP
  - Ma è compatibile anche con TCP
  - Affidabilità opzionale
    (Tecnica Stop & Wait)
- Funzionalità specifiche M2M
  - Resource discovery
  - Multicast
  - Modello di interazione push





CoAP (2/2)

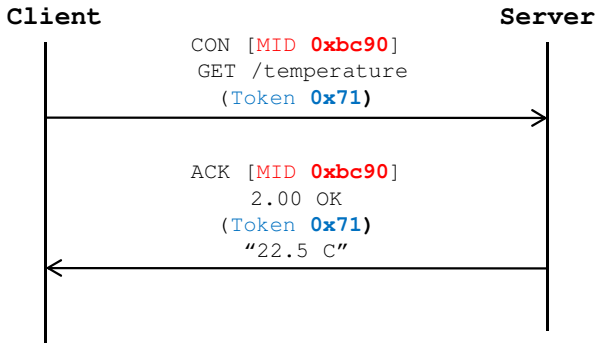


Tipi di messaggi:

- **CON**: Confirmable
- **NON**: Non Confirmable
- **ACK**: Acknowledgement
- **RST**: Reset

Sottoinsieme dei metodi e response code HTTP

Esempio: interazione immediata

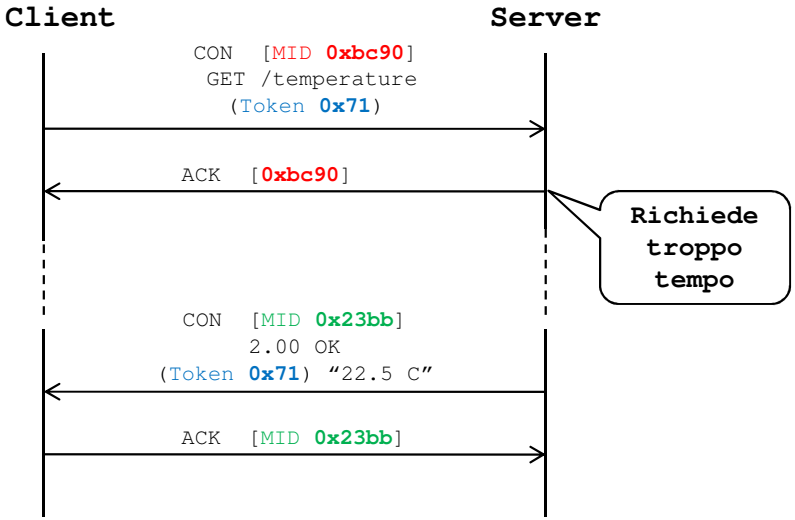


Richiesta CON unicast

Il client identifica una risposta immediata grazie a:

Message ID (MID)
Token

Esempio: interazione deferred



Richiesta CON unicast e risposta deferred

Il client identifica una risposta deferred grazie al

Token

Le relazioni di sottoscrizione

- Il client effettua una **richiesta di sottoscrizione** presso un server per una data risorsa
  - Specificando il *Lifetime* della sottoscrizione
- Il server invia una **notifica** al client ogni volta che lo stato della risorsa osservata cambia (interazioni push)
  - Finché non scade il *Lifetime*
- Il client può rinnovare la sottoscrizione (**refresh**)
  - Poco prima che scada il *Lifetime*
- Sia il client che il server possono **terminare** la relazione di sottoscrizione

PARTE II

Il model-checker SPIN

La specificazione e validazione del CoAP

SPIN e ProMeLa

- SPIN (Simple ProMeLa Interpreter)
  - Tool per verificare la correttezza logica di un sistema distribuito
    In particolare di un protocollo di comunicazione
  - Il sistema distribuito è descritto in linguaggio ProMeLa
- ProMeLa (Process Meta Language)
  - Linguaggio di specificazione per creare un modello di un protocollo di comunicazione
  - Modello rappresentato da una macchina a stati finita
  - Simile al linguaggio di programmazione C

Modello ProMeLa (1/2)

Descrive la **logica** delle procedure di comunicazione di un protocollo

- Deve **astrarsi** dai dettagli implementativi del protocollo
- Deve focalizzare sulle **interazioni**

Obiettivo:

- Individuare **errori logici** nelle procedure di comunicazione di un protocollo
  - Procedere all'implementazione del protocollo sulla base del modello ProMeLa

Limitazione:

Un modello ProMeLa non può descrivere i vincoli temporali del protocollo

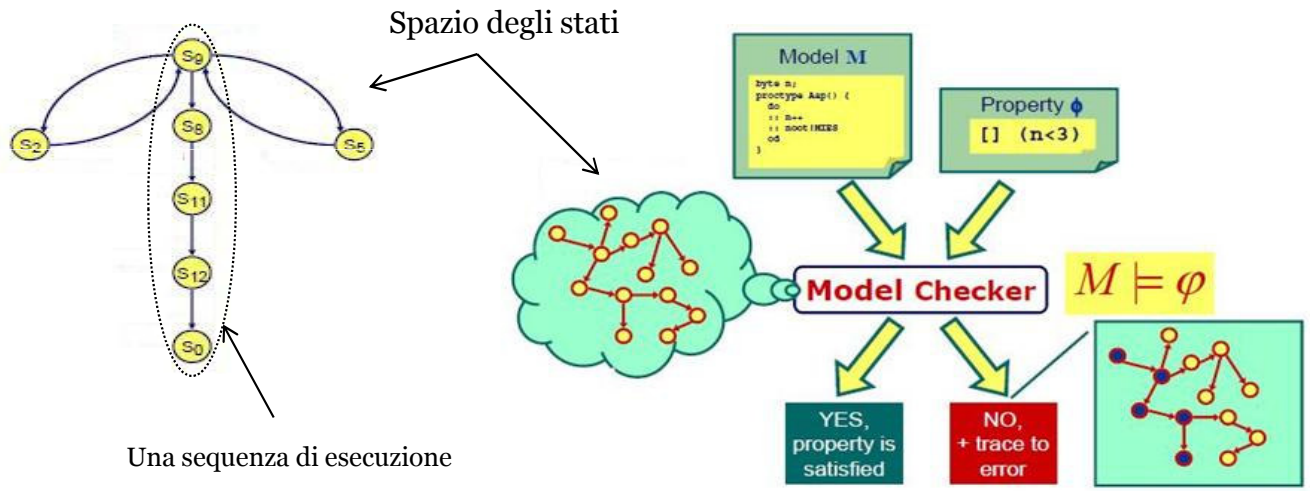
Modello ProMeLa (2/2)

Un modello ProMeLa è costituito da:

- Variabili
- Processi
  - Esecuzione concorrente
  - Possono esistere più istruzioni eseguibili nello stesso processo
    ⇒ **Non determinismo**
- Canali
  - Comunicazione tra processi
    - Asincroni
    - Sincroni

La validazione

Dimostra che in **tutte** le sequenze di esecuzione viene rispettata una **proprietà logica**



Proprietà logiche

Safety	**Liveness**
"Non accadrà mai niente di errato"	*"Prima o poi accadrà qualcosa di corretto"*
Ad esempio: • Invarianza *x è sempre minore di y*	Ad esempio: • Risposta *Quando accade l'azione X prima o poi dovrà accadere l'azione Y*
VALIDAZIONE Spin ricerca una sequenza di esecuzione nella quale accade l'evento non desiderato	VALIDAZIONE Spin ricerca una sequenza di esecuzione nella quale non accade mai l'evento desiderato

PARTE II

Il model-checker SPIN

La specificazione e validazione del CoAP

La specificazione del CoAP

Obiettivi:

- Specificare le procedure di comunicazione del CoAP con riferimento ad una futura implementazione su un dispositivo M2M
  - Interazioni richiesta – risposta
  - Relazioni di sottoscrizione

- Verificare l'assenza di errori logici nella specificazione e individuare incompletezze nella bozza CoAP

Vincolo:

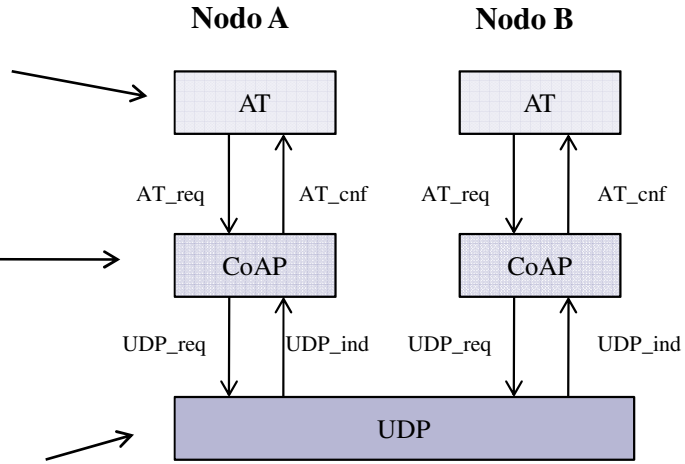
In ProMeLa non si possono specificare le dinamiche temporali del protocollo

Processi e canali

AT (Interfaccia dei comandi per il dispositivo M2M)
Invia una richiesta al CoAP
e attende una risposta

CoAP
Client e server

UDP (Rappresenta il canale di comunicazione)
Può perdere messaggi



“Il modello deve astrarsi dai dettagli implementativi”

Validazione
delle interazioni richiesta-risposta

Proprietà di liveness (di risposta):

- *"Ad ogni richiesta da parte del client fa seguito una risposta immediata, oppure soltanto un messaggio vuoto che riporta riporta l'ACK da parte del server, oppure deve essere inviata ad AT una notifica di timeout"*
  VALID

- *"Ad ogni richiesta del client fa seguito una risposta immediata, oppure una risposta deferred da parte del server, oppure deve essere inviata ad AT una notifica di timeout"*

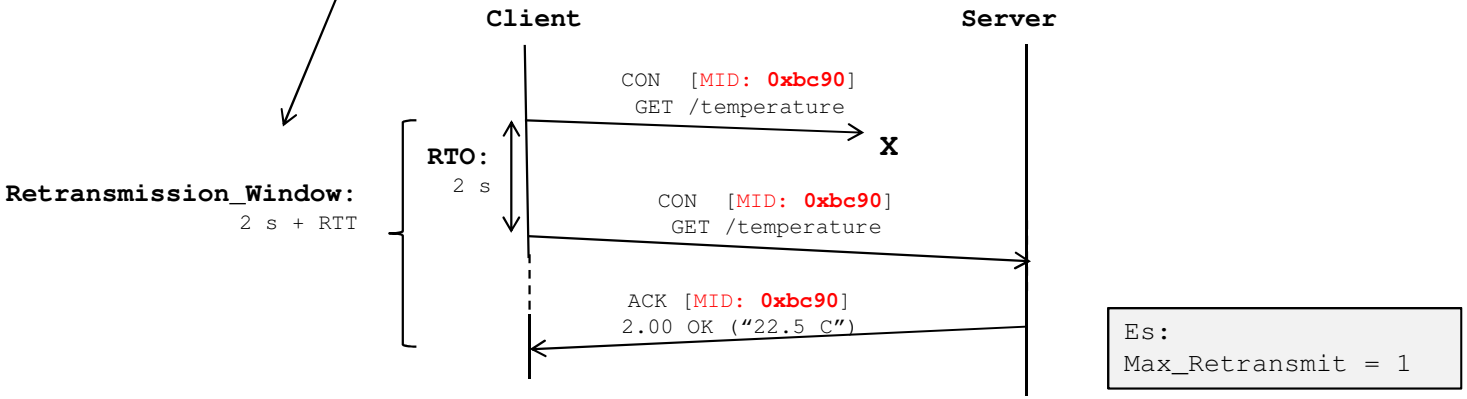
  VALID

Proprietà di safety:

- *" Il client non deve mai scambiare due risposte tra di loro "*
  NOT VALID

Nella realtà…

Retransmission window = $RTO \cdot (2^{Max\_Retransmit} - 1) + Max\_RTT$



Es:
Max_Retransmit = 1

CoAP

"Lo stesso Message ID (MID) non può essere riutilizzato all'interno della Retransmission_Window"

(circa 1 minuto)

ProMeLa

"Retransmission_Window ??"

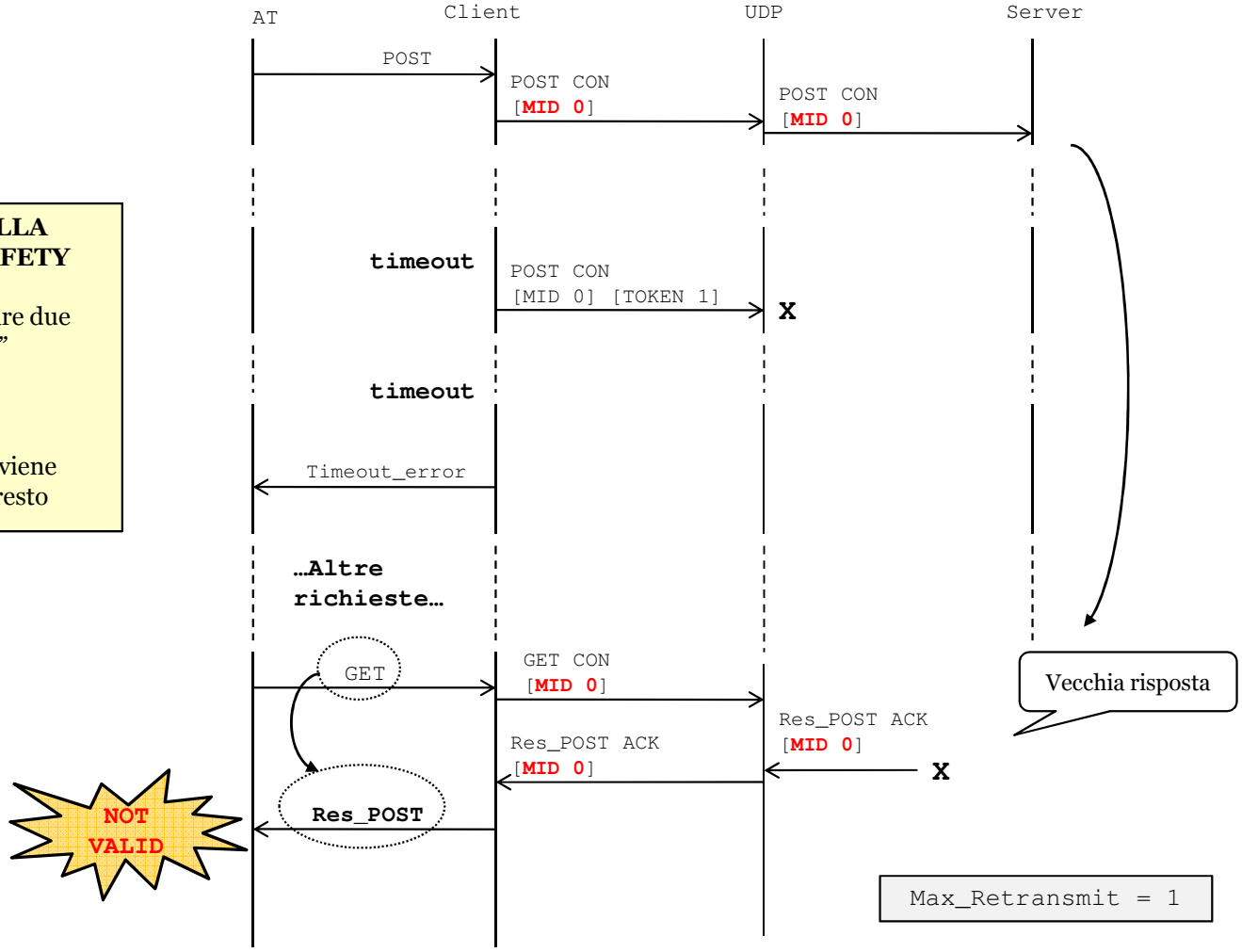
… nel modello ProMeLa

VIOLAZIONE DELLA PROPRIETA' DI SAFETY

" Il client può scambiare due risposte tra loro "

MOTIVO:

Il Message ID (MID) viene riutilizzato troppo presto



Validazione
di una relazione di sottoscrizione

Proprietà di liveness (di risposta)

- "*Ogni volta che il client invia un messaggio di refresh al server, se non esaurisce i tentativi di ritrasmissione per il refresh, dopo un po' riceve una notifica .*"
  VALID

- "*Ogni volta che il server invia una notifica di tipo CON al client, se non esaurisce i tentativi di ritrasmissione, dopo un po' riceve un ACK oppure un RST.*"
  VALID

- "*Ogni volta che il server decide di terminare la sottoscrizione, dopo un po' la sottoscrizione viene terminata sul client e AT ne riceve conferma.*"
  VALID

- "*Ogni volta che AT decide di terminare una sottoscrizione, dopo un po' la sottoscrizione viene terminata sul server e AT ne riceve conferma.*"
  VALID

Problema (e soluzione)

PROBLEMA:
Può succedere che la relazione venga terminata in modo non corretto

Il client considera una notifica **valida** in particolare se:
- Il *Token* è corretto
- La notifica è più recente di un'altra notifica che ha già ricevuto

Se il client riceve una notifica CON **non valida**
- La scarta
- Invia al server un messaggio di **RST**

Se il server riceve un messaggio di **RST**
Termina la sottoscrizione

SOLUZIONE

Se il client riceve una notifica CON **non valida**
- La scarta
- Invia al server un messaggio di **ACK**

Conclusioni

LAVORO SVOLTO

- E' stato analizzato lo stato dell'arte delle tecnologie per la comunicazione M2M in reti wireless a bassa potenza
- Sono stati analizzati i vantaggi che l'adozione del CoAP comporterebbe in svariati ambiti applicativi
- E' stato studiato il protocollo con l'ausilio di un programma di specificazione e validazione formale

- Sono state specificate le procedure di comunicazione del protocollo CoAP
  - Interazioni richiesta – risposta
  - Relazioni di sottoscrizione

- Sono state verificate delle proprietà logiche
  - di liveness (di risposta)
  - di safety (di identificazione delle risposte)

- La proprietà di safety è stata violata
  Motivo:
  - Inadeguatezza di ProMeLa nel descrivere i vincoli temporali

- E' stata individuata un'incompletezza nell'attuale bozza CoAP
  - Problema nella caratterizzazione delle relazioni di sottoscrizione

Sviluppi futuri

- Implementazione del CoAP su di un dispositivo M2M
  - Avvalendosi del fatto che ProMeLa ha una sintassi simile a quella del C
  - Sfruttando le interfacce verso AT e UDP che sono state definite nel modello
- Studio delle prestazioni del CoAP su di una rete wireless reale