Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Sistema di Monitoraggio Proattivo Per la Building Automation

A. Messina, P. Storniolo

Rapporto Tecnico N.:
RT-ICAR-PA-12-02 **Giugno 2012**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Sistema di Monitoraggio Proattivo Per la Building Automation

A. Messina[1], P. Storniolo[1]

Rapporto Tecnico N.:
RT-ICAR-PA-12-02

Data:
Giugno 2012

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo
Viale delle Scienze edificio 11 90128 Palermo

Sommario

1 INTRODUZIONE

Seguire i continui sviluppi del mercato ICT porta le aziende ad avere la necessità di dotarsi di sistemi hardware e software sempre più all'avanguardia, per supportare e gestire i propri sistemi informativi computerizzati. Di conseguenza, è in continuo aumento anche la complessità del sistema informativo delle aziende, indipendentemente dalle dimensioni che essa assume nel proprio mercato di riferimento.

Assistiamo, anno dopo anno, alla progressiva introduzione di nuovi apparati o soluzioni software per soddisfare esigenze crescenti e per agevolare comunicazione e rapporti verso clienti e fornitori.

Apparati di rete quali switch, router, firewall, server di dominio, posta elettronica e software applicativi sono diventati elementi chiave per ottimizzare la riuscita di ogni business.

Purtroppo però, spesso accade che l'amministratore dei sistemi informativi rilevi anomalie, guasti o disservizi solo al momento della segnalazione da parte degli utenti (dipendenti, collaboratori interni ed esterni, business partner, ecc): tutto questo può provocare disagio e ingenti rallentamenti nel lavoro, costituendo un costo "imprevisto" per l'azienda.

Poter disporre, quindi, in tempo reale di una visione d'insieme delle risorse aziendali e del loro stato, permette di intervenire in modo mirato in caso di anomalie o indisponibilità delle stesse, prevenendo in certi casi anche la stessa indisponibilità.

Le moderne aziende (grandi e piccole, con una o più sedi) hanno quindi la necessità di dotarsi di un sistema di monitoraggio per verificare "lo stato di salute" dei propri sistemi informativi.

In alcuni casi, invece, può verificarsi che le aziende dispongano di più sistemi di monitoraggio per ogni tipologia di sistema informatico integrato nella rete aziendale, e ciò comporta una frammentazione dei report dello stato di salute di tutti gli apparati e servizi, con la conseguente impossibilità di monitorare in maniera globale il funzionamento dell'infrastruttura complessiva dell'azienda.

E' quindi auspicabile poter disporre di una soluzione che migliori l'accessibilità di monitoraggio, concentrando i rilevamenti dello status dei diversi componenti del sistema informativo nell'ambito di un'unica centrale di controllo, che possa, secondo i casi, anche fungere da sistema di reazione, oltre che di informazione.

Se adesso, per la misurazione e il controllo dei dispositivi si utilizza un protocollo standard, che abbia la interoperabilità come sua caratteristica principale e che sia, inoltre, anche espandibile, si può facilmente intuire come sia possibile effettuare il monitoraggio non solo di apparati/sistemi (router, switch, firewall, server, ecc.) e servizi applicativi (web server, mail

server, database server, ecc.) tipicamente afferenti al mondo ICT, ma anche di qualunque altro dispositivo o servizio che sia "accessibile" da un'infrastruttura di rete informatica.

Un esempio di protocollo dotato di tali caratteristiche è il Simple Network Management Protocol (SNMP).

L'applicabilità di un tale tipo di soluzione potrebbe quindi estendersi anche ad ambiti applicativi diversi da quello ICT in senso stretto: è sufficiente, infatti, che in questi altri ambiti vi sia una qualche possibilità d'interfacciamento con il "mondo" ICT.

Nel campo della Domotica e Building Automation, ad esempio, sono disponibili dei convertitori studiati espressamente per interconnettere tra loro i più diffusi bus utilizzati in quell'ambito, consentendo quindi l'integrazione di mondi eterogenei quali climatizzazione, automazione, illuminazione, sistemi di misura, sistemi antincendio e sistemi di antintrusione. Essendo dotati di porte ethernet, tali convertitori possono anche essere connessi a sistemi di controllo e monitoraggio, con i quali colloquieranno tramite protocollo SNMP.

O ancora, ad esempio, esistono sul mercato dei dispositivi per il Monitoring Ambientale, a basso costo e basati su SNMP, che possono essere configurati per prevenire specifici casi di esposizione all'umidità, allagamenti, gas, vento, alte/basse temperature, sbalzi di tensione, movimento, suono.

A prescindere, quindi, dal particolare ambito, la piattaforma di monitoraggio sarà sempre e comunque costituita da un server centrale (eventualmente ridondato) cui sono collegate delle sonde, le quali raccolgono le informazioni dai vari apparati che compongono l'infrastruttura in oggetto.

Alla luce di tale generalizzazione e intendendo per servizio ciascun dispositivo o componente dell'infrastruttura, si potrà parlare di *Service Management*.

Scopo del presente lavoro sarà di verificare la fattibilità dell'integrazione di sistemi di acquisizione delle informazioni con sistemi di elaborazione e presentazione, attraverso l'uso di un protocollo comune.

In particolare, si estenderanno le funzionalità di base della piattaforma di management Open Source denominata Zabbix, che nasce come Network Management System (NMS) di classe enterprise, integrandovi le nuove funzionalità necessarie a consentirne l'interfacciamento nativo con i gateway SCS-Ethernet di BTicino (ad esempio il modello F453AV).

Si dimostrerà quindi come sia possibile monitorare, grazie al protocollo SNMP, il funzionamento di un impianto domotico, integrandone le relative informazioni nell'ambito di un'unica piattaforma di management e controllo, potendo anche associare, infine, delle azioni/reazioni personalizzate al verificarsi di un particolare evento o tipologia di eventi.

2 IL PROTOCOLLO SNMP

2.1 INTRODUZIONE

Il *Simple Network Management Protocol* (SNMP) è un protocollo di rete che permette di raccogliere dati, controllare e gestire il funzionamento degli elementi di una rete di tipo TCP/IP, attraverso la rete stessa. Lo sviluppo di questo strumento è gestito da sempre dall'organizzazione sopranazionale *Internet Engineering Task Force* (IETF) mediante l'emanazione di diverse *Request For Comment* (RFC).

Nel maggio del 1990, fu pubblicato l'RFC 1157, che definiva la prima versione di SNMP. Insieme ad un manuale sulle informazioni di gestione , SNMP definiva una maniera sistematica di controllare e gestire una rete di computer. Esso nasce come una evoluzione del protocollo *Simple Gateway Monitoring Protocol* (SGMP) progettato nel 1987 e limitato alla sola gestione di router e gateway.

Successivamente, nel 1995, al fine di superare le imperfezioni venute alla luce, venne definito l'SNMPv2 . Ancora, nel 1998, fu rilasciata l'ultima versione: l'SNMPv3. Fin dalla sua nascita questo protocollo si è imposto come "standard de facto" per la gestione delle reti, ed è, come ogni standard, sempre in via di miglioramento.

2.2 ARCHITETTURA

Il protocollo SNMP è stato creato per operare come protocollo di rete al livello applicativo dello stack TCP/IP e si basa, per la comunicazione, sul protocollo UDP.

L'architettura, di cui il protocollo SNMP fa parte, è detta *Internet Network Management Framework*.

Il management di una rete include la distribuzione, l'integrazione e la coordinazione di risorse hardware, software ed umane per monitorare, testare, configurare, analizzare, interrogare e controllare la rete e le sue risorse per valutarne le performance operative e la qualità.

L'architettura si basa sul concetto di Manager/Agent: infatti per svolgere tali funzionalità il "centro di gestione" interagisce con i "network element" da gestire attraverso un'infrastruttura di comunicazione dedicata al trasporto delle informazioni di gestione (rete di gestione sovrapposta alla rete gestita), oppure attraverso la stessa rete gestita.

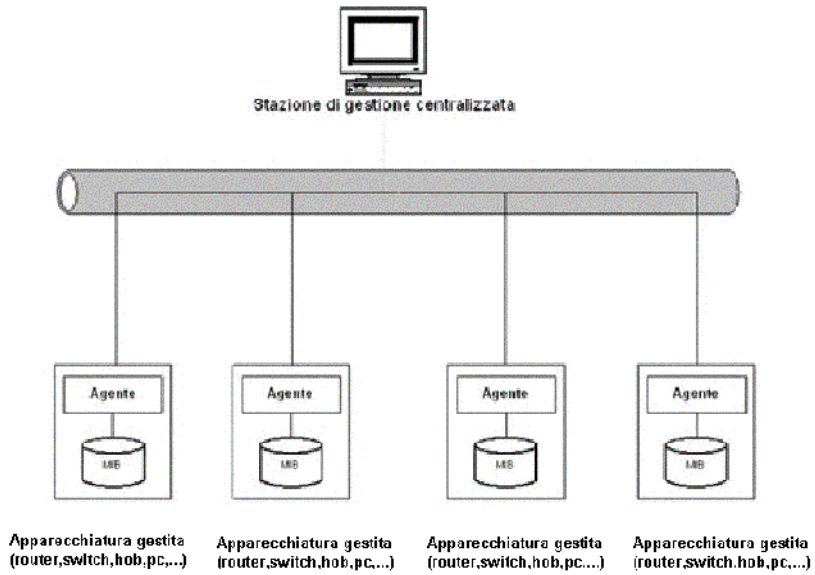
Tale colloquio si attua attraverso meccanismi di comunicazione che possono essere proprietari, cioè realizzati da un costruttore in modo specifico per la gestione dei propri apparati, oppure come nel nostro caso standardizzati attraverso un protocollo.

Alla base della gestione di rete c'è l'introduzione negli apparati di rete di una "strumentazione" sempre più completa e sofisticata, in grado di raccogliere una enorme quantità di dati dagli apparati: configurazione e parametri operativi dei singoli elementi che compongono ciascun dispositivo, dati di traffico, tassi di errore, ecc.. Il compito del gestore è quello di analizzare tale massa di informazioni, riconoscere eventuali stati di funzionamento anomalo ed effettuare le operazioni necessarie per ripristinare il corretto funzionamento della rete.

Il colloquio tra la stazione di gestione e l'apparato gestito si esplica in particolare tra due entità, realizzate per mezzo di processi software, denominate rispettivamente Manager, nel centro di gestione, ed Agent, nel nodo gestito.

Il trasferimento di informazioni tra Manager ed Agent avviene in accordo ad un insieme di regole, sintattiche e semantiche, che costituiscono il protocollo di gestione. Il protocollo di gestione è un protocollo di livello applicativo che si appoggia sulla pila protocollare sottostante.

Uno schema logico di tale architettura è riportato di seguito:



L'architettura per la gestione di rete che si basa sul protocollo SNMP comprende essenzialmente i seguenti elementi:

- Stazione di Gestione (NMS o Manager)
- Nodi Gestiti e Agente di Gestione (AGENT)
- Management Information Base (MIB)
- Protocollo per la gestione di rete.

2.3 STAZIONE DI GESTIONE (NMS)

E' tipicamente un dispositivo (o più) che funziona da interfaccia tra l'amministratore di rete ed il sistema da gestire. Sono nella maggioranza dei casi normali calcolatori che eseguono un software speciale di gestione.

Tale stazione sarà costituita dai seguenti blocchi funzionali:

- Uno o più processi che comunicano con gli agenti sulla rete (cfr. 1.4), inviando comandi e ricevendo risposte.
- Una interfaccia (testuale o grafica) attraverso la quale l'amministratore può controllare lo stato della rete ed intervenire quando necessario.
- Un database contenente le informazioni (o, meglio, il tipo delle informazioni) che la stazione può ottenere dai vari dispositivi (attraverso gli agent): il MIB (vedi 1.5).
- Un modulo con la capacità di "traslare" le richieste effettuate dal sistema sovrastante (applicazione) in formato standard comprensibile per i vari agenti remoti
- Un gestore delle trap, ovvero le segnalazioni di errore da parte degli agenti

Nel modello SNMP tutta la *business logic* è mantenuta nelle stazioni di gestione, in modo da tenere gli agenti il più possibile semplici e minimizzare l'effetto sui dispositivi in cui "girano".

Secondo la teoria delle applicazioni client/server, il client è la componente che ha un ruolo proattivo (ovvero si occupa di iniziare le transazioni), mentre il server è la componente passiva (ovvero rimane in attesa delle richieste da parte del client).

Una rete SNMP può essere vista come un sistema distribuito di tipo client/server atipico: i client (le stazioni di gestione) sono in numero molto minore rispetto ai server (agenti di gestione).

Inoltre, il ruolo client/server si capovolge solo nel caso di eccezioni (trap) che si verificano negli agenti: è l'agente stesso che notifica alle stazioni di gestione (server) un problema.

2.4 NODI GESTITI E AGENTE DI GESTIONE (AGENT)

I nodi gestiti sono dispositivi di rete (bridges, router, hubs, stampanti etc...), workstation, server, applicazioni software: qualunque entità o nodo in grado di eseguire un particolare applicativo, agent.

Quest'ultimo è un processo presente sui vari dispositivi della rete che risponde alle richieste di informazioni da parte della stazione di gestione, ed esegue operazioni imposte da quest'ultima. Inoltre può comunicare con essa in modo asincrono, generando eventi urgenti.

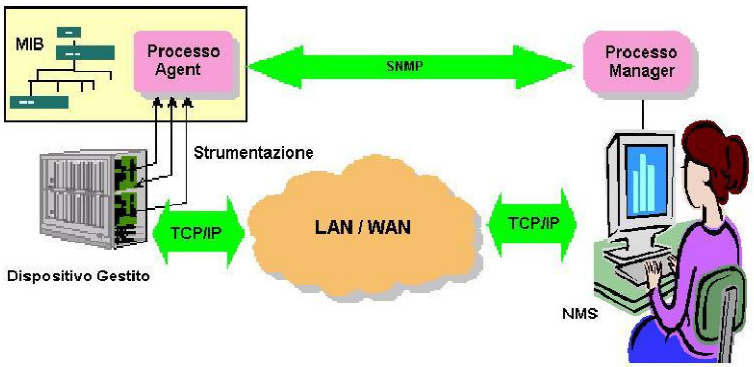
Ogni agent mantiene un database locale (MIB) di variabili, che descrive lo stato del dispositivo e che influenza le sue operazioni.

Scopi degli agenti sono:

- Mantenere un insieme di variabili organizzate gerarchicamente. Tali variabili contengono lo stato e la storia del dispositivo, oppure consentono, se modificate, di intervenire sul comportamento del nodo stesso (tenere traccia e modificare la configurazione del dispositivo).
- Consentire la letture o la modifica di queste variabili da parte dei processi autorizzati
- Rilevare malfunzionamenti del dispositivo o azioni significative che lo interessando, segnalandolo alla stazione di gestione tramite l'invio di eventi (TRAP).
- Rispondere ai comandi inviati dalla stazione di gestione

Questo modello assume che ogni nodo gestito possa eseguire al proprio interno un agente SNMP.

Ci sono alcuni dispositivi vecchi o che in origine non erano stati pensati per un utilizzo in rete che non hanno queste possibilità. Per gestirli SNMP definisce ciò che viene chiamato un agente proxy, di fatto un agente che controlla uno o più dispositivi non SNMP e che comunica con la stazione di gestione riguardo la loro affidabilità, e con dei dispositivi stessi mediante protocolli non standard.



2.5 MANAGEMENT INFORMATION BASE (MIB)

Come si è visto nei punti precedenti, di fondamentale importanza risulta essere la definizione di cosa e come queste informazioni di gestione devono essere scambiate.

Si è parlato genericamente di database, vediamo nello specifico questi due aspetti del protocollo.

La maggior parte delle reti nella realtà hanno componenti di provenienza diversa, con host di uno o più costruttori, bridge e router di altri produttori e stampanti di altri ancora.

Per permettere ad una stazione di gestione (potenzialmente ancora di un altro produttore) di colloquiare con tutti questi componenti diversi, bisogna che l'informazione mantenuta da essi sia rigidamente specificata.

E' inutile che una stazione di gestione chieda ad un router qual è il suo tasso di perdita di pacchetti, se il router non ne serba memoria.

Perciò SNMP descrive l'informazione precisa che ogni tipo di agente deve mantenere e il formato che gli deve essere fornito.

In breve, ogni dispositivo conserva una o più variabili che descrivono il suo stato. Nella letteratura SNMP queste variabili sono dette OGGETTI, ma il termine è impreciso in quanto non sono oggetti nel senso di un sistema orientato agli oggetti dato che hanno solamente lo stato e nessun metodo (eccetto che per leggerne e scriverne il valore).

Il database summenzionato è chiamato MIB ("Management Information Base"), e contiene la collezione di tutte le possibili informazioni (oggetti) prelevabili dalla rete da parte della stazione di gestione.

Nel "Manager" è quindi contenuto l'intero database MIB (o la parte afferente alla specifica rete).

Negli “Agent” invece, è presente esclusivamente quella parte di MIB per la quale ha senso per l’Agente mantenere lo stato.

E’ interessante notare che le informazioni utilizzate dalla stazione di gestione sono memorizzate all’interno dei MIB locali dei dispositivi gestiti, che costituiscono quindi un sistema distribuito.

E’ importante chiarire che, anche se nella terminologia comune si usa il termine database per definire il MIB, questo non è un database nel senso stretto del termine, poiché non contiene dati e non mantiene neanche i dati prelevati dai dispositivi monitorati.

Semplicemente il MIB è la descrizione di cosa il Manager può richiedere. Si può vedere come una specie di contratto fra entità coinvolte, in modo da far sapere a tutti cosa è ottenibile e come.

SMI (“Structured Management Information”) definisce le modalità con cui le informazioni di gestione sono strutturate internamente, si occupa quindi di definire le strutture o il formato dei dati di SNMP.

2.6 IL MODELLO MANAGER/AGENT/MANAGEMENT OBJECT

Le informazioni che il Manager richiede ad un Agent sono relative ad Oggetti logici che rappresentano la realtà fisica del dispositivo. Tali oggetti sono contenuti in un database denominato MIB (Management Information Base), che ha una struttura ad albero.

Le informazioni contenute nel MIB forniscono una rappresentazione logica del dispositivo e del suo stato; tale rappresentazione logica permette al Manager di accedere ai dati di un dispositivo in modo non ambiguo. Il Manager conosce infatti la struttura del MIB e nel colloquiare con l’agente fa riferimento agli oggetti indicandone la posizione sul MIB.

In tal modo il manager non è vincolato a conoscere la realtà fisica del dispositivo.

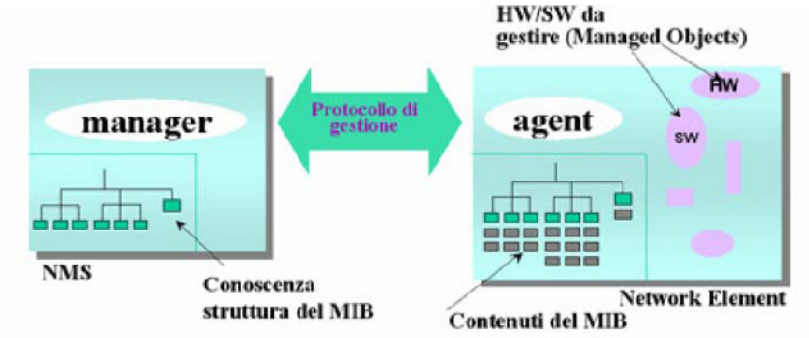
Se ad esempio il Manager volesse conoscere il numero di pacchetti che un dispositivo ha ricevuto su una sua interfaccia, esso chiede all’agente di restituirgli il valore della variabile corrispondente sul MIB. Sia Manager che Agente hanno una conoscenza della struttura del MIB; l’agente, oltre alla struttura, possiede anche i dati che la popolano, acquisendoli dalla strumentazione dell’apparato.

Essenzialmente il meccanismo di base su cui è impostata la gestione di rete consiste quindi nell’interrogazione del database la cui struttura è definitanel MIB.

Il Manager, tramite l’Agente, è anche in grado di modificare i dati presenti nel MIB, effettuando quindi operazioni di configurazione dell’apparato. Un Agente, in seguito alla

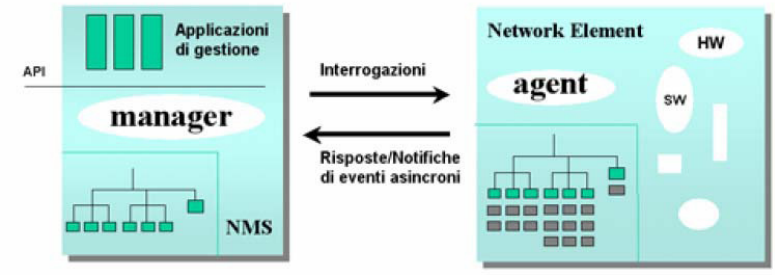
richiesta del Manager, è anche in grado di eseguire dei comandi (ad esempio eseguire una operazione di reinizializzazione dell'apparato).

I dispositivi possono anche inviare all'applicazione di gestione degli allarmi che indicano il verificarsi di eventi particolarmente significativi dell'apparato gestito.



Nel seguente modello:

- il manager effettua interrogazioni sui MIB dei diversi dispositivi di rete, riceve le risposte dagli agenti, riceve dagli agenti notifiche asincrone di particolari eventi verificatesi nel dispositivo, ecc.;
- l'agente mantiene aggiornato il MIB, risponde alle richieste del manager, esegue sugli oggetti gestiti le operazioni richieste dal manager, notifica al manager eventi asincroni (guasti alle interfacce ecc.);
- i managed object sono entità logiche che rappresentano il dispositivo fisico (stato delle interfacce, contatori di traffico, tassi di errore, indirizzi, ecc.).



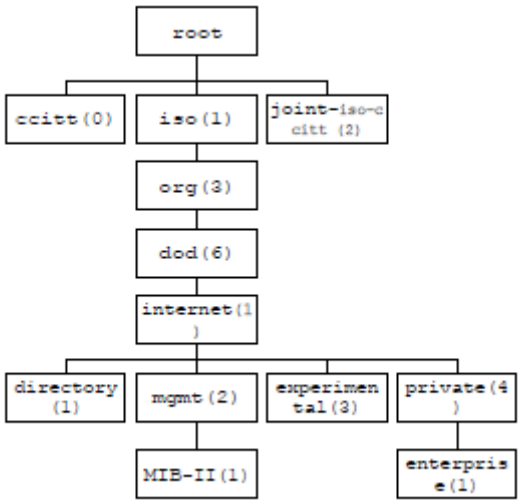
2.7 STRUTTURA DEI MIB

Tutti gli oggetti gestiti dal protocollo SNMP sono disposti in una struttura ad albero. Le parti terminali dell'albero (foglie) rappresentano gli oggetti gestiti, ognuno dei quali può rappresentare risorse o informazioni.

La stessa struttura ad albero definisce un raggruppamento di oggetti in insiemi logicamente correlati.

Tali oggetti sono caratterizzati da un nome e da un identificatore ASN.1 detto *OBJECT IDENTIFIER* (OID) che lo individua in modo univoco all'interno della struttura.

La struttura di ogni OID non è altro che una sequenza di interi che rappresenta il "cammino" dalla radice all'elemento.



Al livello più alto della gerarchia MIB sono presenti le organizzazioni standard a livello mondiale, tra le quali `ccitt`, `joint-iso-ccitt` e `iso` (che ha identificatore 1).

A sua volta l'`iso` riserva l'identificatore 3 per `org` (insieme delle organizzazioni riconosciute e che possono emanare standard) il quale riserva l'identificatore 6 per il `dod` ("Department of Defense" americano). Da questo si dirama il sottoalbero `internet` con identificatore 1.

In notazione ASN.1 quindi:

```
internet OBJECT IDENTIFIER ::= {iso(1) org(3) dod(6) 1}
```

ovvero:

```
OID = 1.3.6.1
```

Da quest'ultimo escono altri quattro nodi:

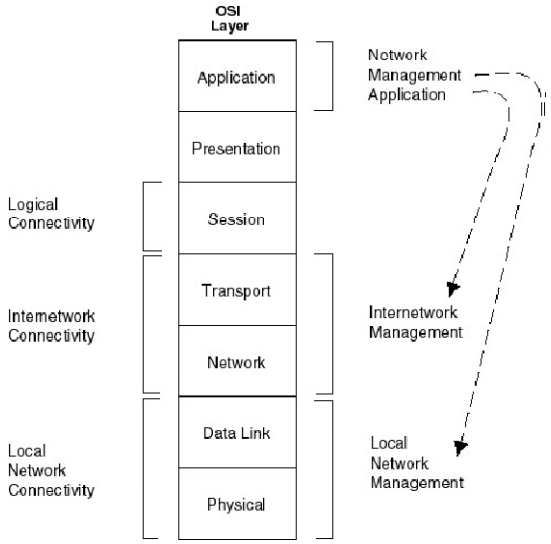
- Directory: riservato per usi futuri con osi;
- Mgmt: contiene tutte le MIB definite ufficialmente dall'Internet Architecture Board (IAB), un ente che si occupa dello sviluppo di Internet. Attualmente contiene due MIB: MIB-I e la MIB-II che rappresenta una estensione della prima;
- Experimental: contiene gli oggetti usati per gli esperimenti in Internet;
- Private: contiene le MIB definite unilateralmente senza nessuna verifica da parte dello standard. Sono utili per le necessità proprie di ciascuna azienda.

Entrambe MIB-I e MIB-II hanno uno stesso OID (1.3.6.1.2.1) in quanto solo una di esse è presente nell'albero. In futuro, se necessario, la MIB-II sarà ricostruita, espandendola e modificandola dove necessario, e, probabilmente chiamandola con il nome di modulo MIB-3 ma con lo stesso OID 1.3.6.1.2.1.

Per quanto riguarda il sotto-albero Private, attualmente contiene solo enterprise, dove i vari costruttori possono inserire moduli MIB proprietari per condividere informazioni riguardanti i loro dispositivi, cui viene assegnato un identificativo univoco dalla *Internet Assigned Number Authority* (IANA). Il modo in cui il nodo internet è suddiviso (4 sottoalberi) è un elemento fondamentale per la futura crescita del MIB, in modo da mantenere sempre aggiornato il protocollo su future necessità al momento attuale non preventivate.

2.8 IL PROTOCOLLO DI GESTIONE SNMP

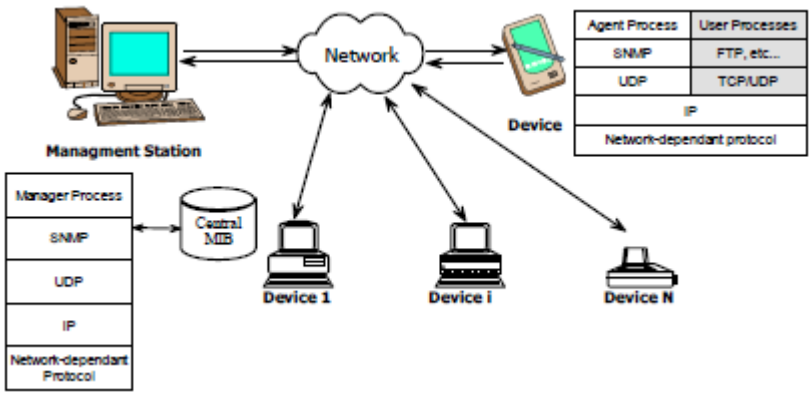
Si è già detto che l'architettura di un sistema SNMP comprende tipicamente una stazione di gestione (Manager) che ha il compito di monitorare i dispositivi collegati in rete, sui quali deve essere presente un Agente (bulk o proxy) che soddisfa le richieste provenienti da quest'ultima: la comunicazione tra queste entità avviene tramite il protocollo di comunicazione definito dall'SNMP.



Si tratta di un protocollo al livello applicazione dello stack ISO/OSI, basato su UDP, quindi intrinsecamente con caratteristiche di leggerezza ed inaffidabilità, in quanto non viene creato alcun canale di comunicazione tra i due interlocutori.

Più specificatamente, l'Agente ascolta richieste provenienti dal Manager sulla porta UDP 161, mentre il Manager invia richieste alla suddetta porta e ascolta eventi di notifica critici (trap) sulla porta UDP 162.

In generale il sistema può essere rappresentato nel seguente modo:



Nella stazione di gestione, il processo Manager controlla l'accesso alla MIB ivi presente e contenente tutte le informazioni reperibili dai dispositivi presenti sulla rete gestita.

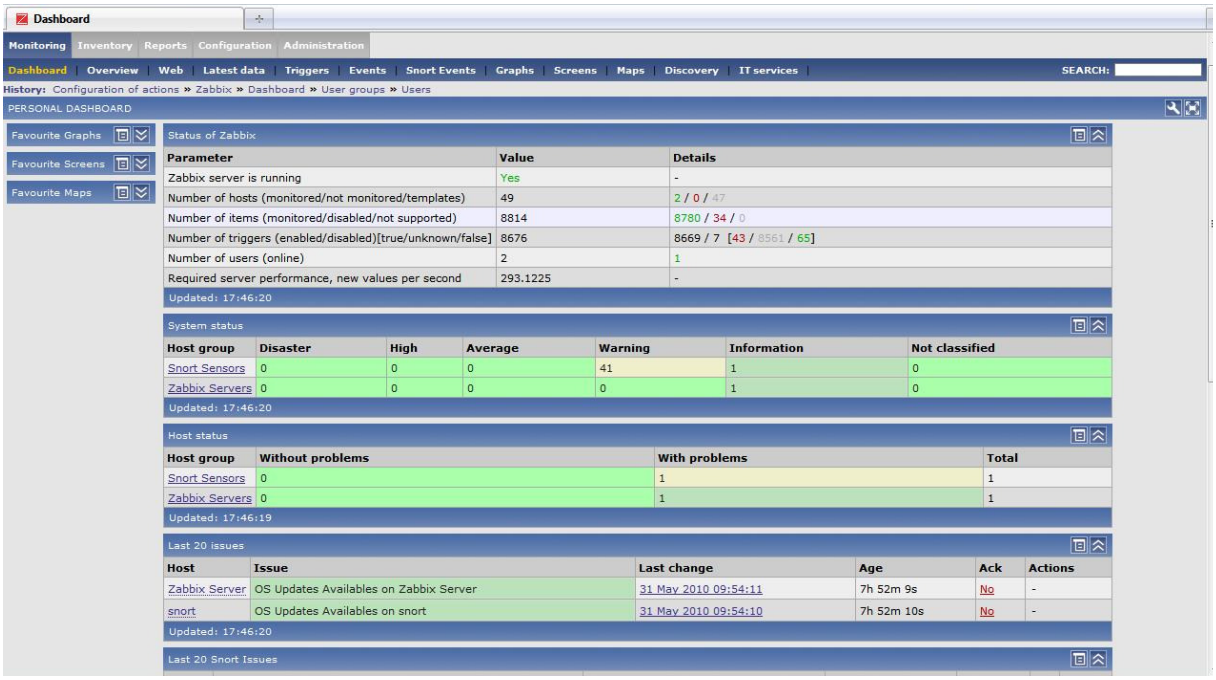
L'Agente presente sul nodo interpreta i messaggi SNMP, controlla le MIB del dispositivo sul quale è situato, e risponde fornendo i valori e le informazioni richieste.

3 ZABBIX

3.1 INTRODUZIONE

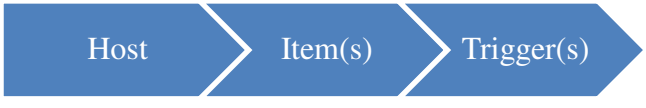
Zabbix è nato nel 1998 come progetto interno in una banca, a cura dello sviluppatore Alexei Vladishev. Nel 2001 è stato rilasciato in GPL e solo nel 2004 è venuta alla luce la prima versione stabile. L'ultima versione disponibile è la 1.8.2 ed è attualmente sviluppato da Zabbix SIA.

Zabbix è composto sostanzialmente di tre moduli distinti: server, agent, front-end. Server e agent sono scritti in linguaggio C, mentre il front-end è implementato in PHP e Javascript.

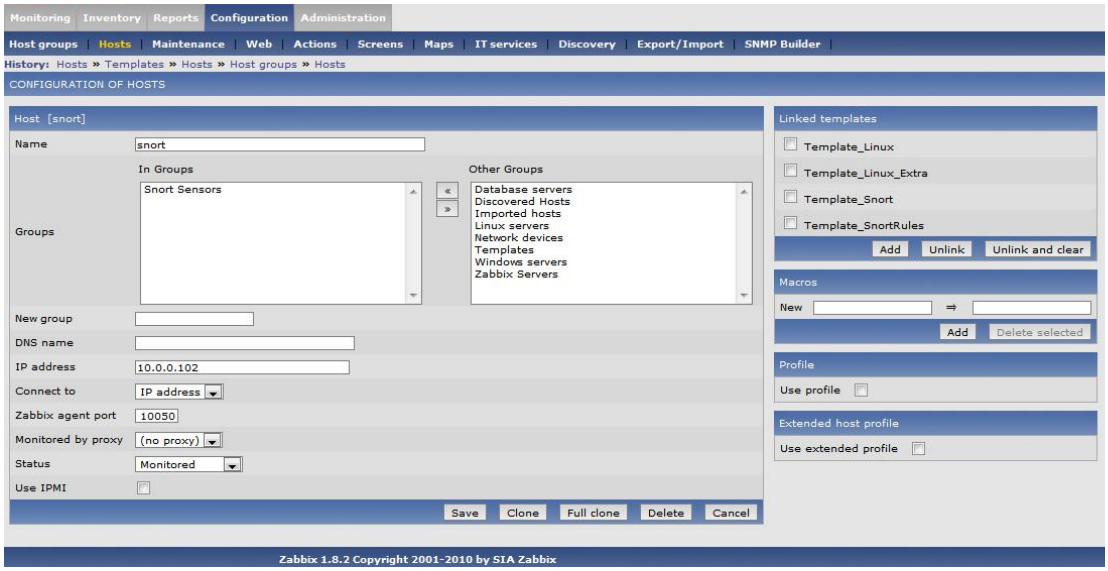


3.2 DISCOVERY

Zabbix è un sistema molto potente e complesso ma la sua documentazione, pur essendo chiara e dettagliata riguardo a tutte le caratteristiche e funzionalità, lascia un po' a desiderare in termini di setup iniziale del monitoraggio. Bisogna, infatti, tenere presente che esiste un ordine di precedenza tra i passi configurazione, che devono essere effettuati per attivare il monitoring di un host o di un servizio



La creazione di un host avviene premendo il bottone Create Host nella pagina “Configuration > Hosts”.

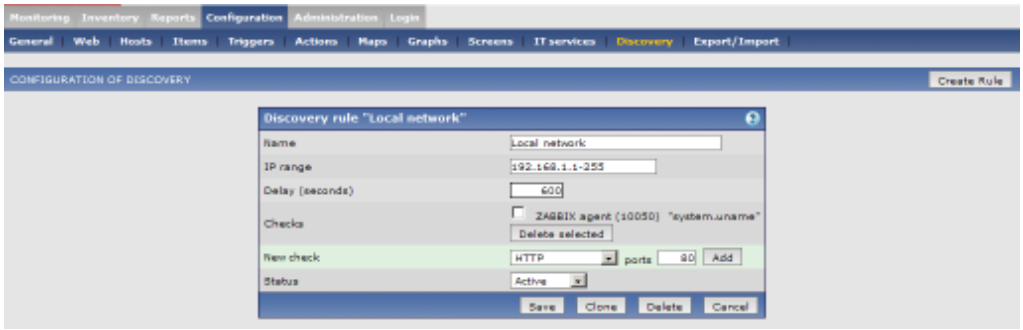


In questa fase si dovranno indicare il nome dell’host, gli eventuali gruppi cui dovrà appartenere, l’indirizzo IP, il numero di porta da interrogare ed infine l’associazione con uno o più template, dai quali l’host dovrà ereditare le caratteristiche (Items, Applications, Triggers, Graphs) ivi definite.

Riguardo al numero di porta, in caso di semplice controllo ICMPcheck si indicherà il valore “0”; nel caso di agent Zabbix si indicherà il valore “10050” (o eventualmente altro numero di porta, allineato con quello indicato in fase di configurazione dell’agent); nel caso di SNMP, infine, si utilizzerà il valore “161”.

Oltre alla procedura di creazione manuale, è disponibile una potente funzionalità di discovery automatica, basata sulle seguenti informazioni:

- Intervalli di indirizzi IP;
- Disponibilità di servizi (FTP, SSH, HTTP, POP3, IMAP, TCP, etc.);
- Informazioni ricevute da agent Zabbix;
- Informazioni ricevute da agent SNMP.



Ogni servizio e host (IP) verificato dal modulo di discovery genera degli eventi che possono essere utilizzati per creare regole per le seguenti azioni:

- Generazione di notifiche;
- Aggiunta e rimozione di host;
- Attivazione e disattivazione di host;
- Aggiunta e rimozione di un host ad/da un gruppo;
- Associazione host-template;
- Esecuzione di script remoti.

3.3 AVAILABILITY MONITORING

Vi sono diverse modalità per monitorare un servizio in esecuzione da qualche parte all'interno della propria rete, o anche all'esterno ed il metodo da utilizzare dipenderà dal livello di accesso che si ha sull'host e dal servizio.

Nel caso del controllo della risposta di un web server, ad esempio, per assicurarsi che il server stia rispondendo si potrà utilizzare una semplice richiesta sulla porta 80, senza alcuna necessità di installare agent sull'host. Nel caso di un router, si utilizzeranno query SNMP. Ma se di un server web che prende le pagine da un database si volesse controllare anche lo stato del database, allora sarà necessario installare sull'host l'agent.

Avendo definito un host, il modo più semplice per associarvi un test di disponibilità è di creare un item di tipo simplecheck, con associata la key icmpping, corrispondente a un semplice ping test all'indirizzo IP dell'host.

Le tipologie di item disponibili sono:

- Zabbix agent (anche con check attivi);
- Simple check;
- SNMP agent V1, V2 e V3;
- Zabbix trapper;
- Zabbix internal;
- Zabbix aggregate;
- External check;
- Database monitor;
- IPMI agent;

- SSH agent;
- Telnet agent;
- Calculated.

3.4 PROBLEM MANAGEMENT

Definiti gli Item per il collezionamento dei dati, a questi possono essere associati uno o più Trigger, cioè delle espressioni logiche che rappresentano lo stato del sistema in funzione dei valori dei dati stessi.

Lo stato di un Trigger (expression) viene ricalcolato ogni qualvolta Zabbix riceve dei nuovi valori per un Item, se tale Item fa parte dell'espressione. L'espressione può assumere i seguenti valori:

- **PROBLEM**: Indica normalmente che è successo qualcosa. Ad es. carico della cpu troppo elevato.
- **OK**: E' lo stato normale di un trigger.
- **UNKNOWN**: Indica che Zabbix non è stato in grado di valutare l'espressione. Ciò può accadere quando:
  - il server non è raggiungibile;
  - l'espressione del trigger non può essere valutata per mancanza di dati sufficienti;
  - l'espressione del trigger è stata modificata di recente.

In fase di creazione, per ciascun trigger possono anche essere definite delle relazioni di dipendenza e può essere impostato il livello di severità che dovrà caratterizzare l'allarme.

Il passo successivo potrà essere a questo punto quello di definire un'azione, per inviare ad esempio una notifica via email o sms.

Un'azione è definita da tre componenti principali:

- Configurazione generale: consente l'impostazione delle opzioni generali, come il subject della mail da inviare e il messaggio;
- Operazioni: specificano cosa deve essere esattamente fatto, compreso chi invia il messaggio, a chi deve essere inviato, e quale messaggio inviare;

- Condizioni: permettono di specificare quando deve essere attivata l'azione e quando eseguire le operazioni. E' possibile indicare, anche insieme, diversi tipi di condizioni, come host, host group, tempo, problemi specifici (trigger) e loro severità, e tanti altri.

3.5 FLUSSO DELLE INFORMAZIONI

Dati un host, un suo item e un trigger ad esso riferito, vediamo un attimo quale sia il flusso delle informazioni scambiate tra le varie entità in Zabbix:

- Quando, in base al valore dell'item, l'espressione di un trigger diventa vera, lo stato del trigger viene impostato a PROBLEM.
- Quando l'espressione ritorna ad essere falsa, lo stato torna ad essere OK.
- Ogni volta che avviene il cambio di stato di un trigger, viene generato un evento, che contiene i dettagli del cambiamento di stato del trigger, quando è successo e quale è lo stato attuale.
- Quando si configura un'azione, si possono impostare diverse condizioni, in modo da considerare solo alcuni degli eventi. Ad ogni azione possono inoltre essere associate delle operazioni, che indicano cosa esattamente debba essere fatto

4 BTICINO OPEN WEB NET

4.1 INTRODUZIONE

Nell'ultimo decennio BTicino ha investito nella ricerca e nello sviluppo di impianti domotici applicabili sia a realtà residenziali che industriali.

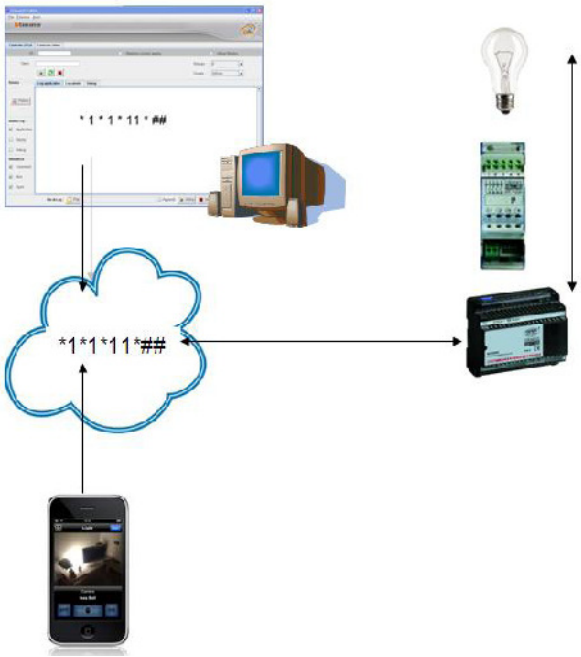
Per permettere a chiunque abbia conoscenze informatiche di linguaggi ad alto livello, di poter interagire con i sistemi e di poter definire funzioni innovative, è stato studiato e implementato un linguaggio di comunicazione: Open Web Net.

L'Open Web Net è un linguaggio grazie al quale è possibile scambiare dati ed inviare comandi tra un'unità remota e il sistema *My Home BTicino*.

Il protocollo è pensato per essere indipendente dal mezzo di comunicazione utilizzato, considerando come requisito minimo la possibilità di poter utilizzare toni DTMF sulla normale linea telefonica PSTN.

Attualmente, i dispositivi che utilizzano tale protocollo sono i web server, il comunicatore telefonico e l'attuatore telefonico.

Questo linguaggio è stato pensato anche per permettere l'integrazione con funzioni di altri produttori oppure per permettere a dispositivi come PC, Smartphones e tablet di comunicare con l'impianto My Home da remoto.



L'Open Web Net è stato introdotto per fornire un livello astratto che permette la supervisione e il controllo dei sistemi My Home concentrandosi sulle funzioni senza curarsi dei dettagli di dell'installazione e senza dover per forza conoscere la tecnologia SCS.

Conoscendo la sintassi del protocollo è possibile usare un proprio software per controllare il sistema My Home.

4.2 SINTASSI DI UN MESSAGGIO OPEN

Un messaggio OPEN è composto da caratteri appartenenti al seguente insieme:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #}; inizia con il carattere '*', e finisce con la coppia di caratteri '##'.

Il carattere '*' viene anche usato come separatore tra i tag.

Un messaggio OPEN è così strutturato:

***tag1*tag2*tag3*...*tagN##**

Un tag è composto da caratteri appartenenti all'insieme {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #}. Un tag non può contenere la coppia di caratteri '##'. Un tag può essere anche omesso, creando così messaggi open del tipo:

***tag1*..*..*tag4*...*tagN**##**

A seguito è presentata una tabella che mostra i diversi tipi di messaggi che possono essere inviati e ricevuti all'interno di una conversazione Client-Server Open.

ACK	*#*1##
NACK	*#*0##
NORMALE	*CHI*COSA*DOVE##
RICHIESTE STATO	*#CHI*DOVE##
RICHIESTA GRANDEZZA	*#CHI*DOVE*GRANDEZZA##
SCRITTURA GRANDEZZA	*#CHI*DOVE*#GRANDEZZA*VAL1*VAL2*...*VALn##

4.3 SESSIONI DI COMUNICAZIONE

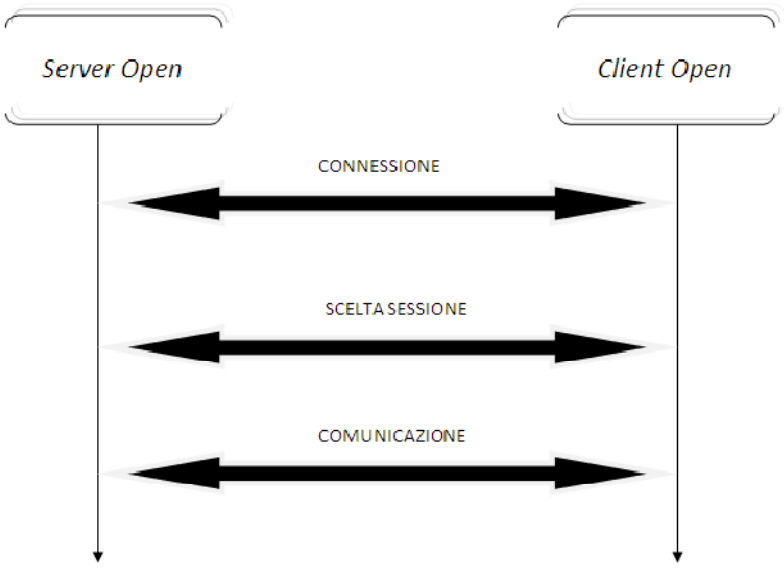
I gateway TCP-IP offrono il server Open Web Net sulla porta TCP 20000.

Tre sono le fasi individuabili per instaurare una sessione:

- connessione
- identificazione
- comunicazione

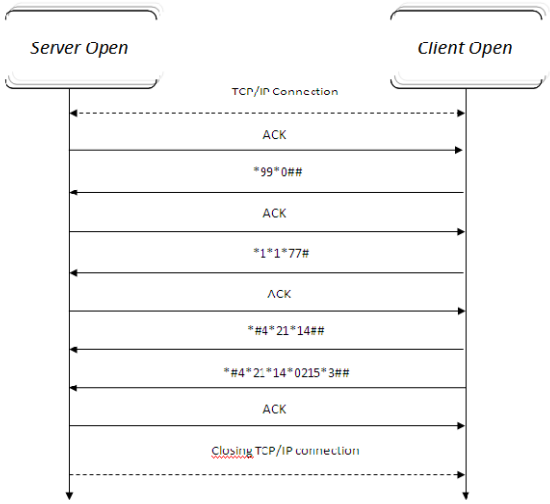
Il client OPEN può instaurare due tipologie di sessioni:

- **Sessione comandi (azioni)**: utilizzata per inviare comandi, richiedere lo stato; richiedere e impostare la dimensione.
- **Sessione eventi**: usata dal Client Open per leggere tutto quello che succede sul bus dell'impianto domotico in modo asincrono.



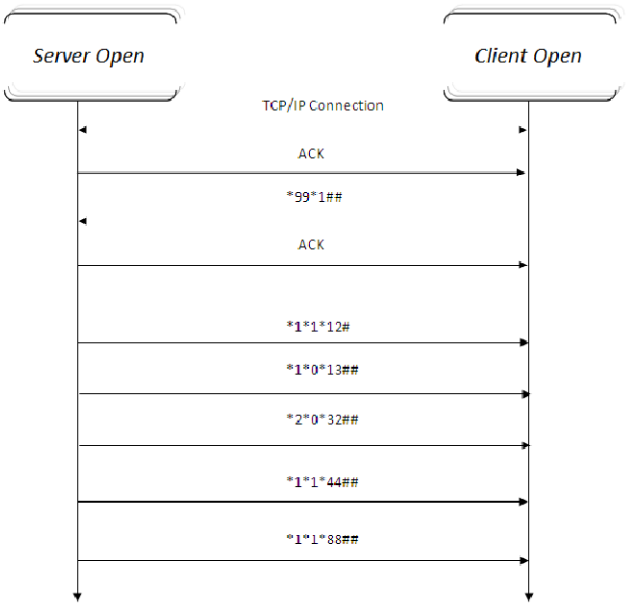
4.4 SESSIONI COMANDO/AZIONE

Subito dopo aver instaurato una connessione TCP-IP tra la macchina che richiede il servizio (il client) e la macchina che offre il servizio (il server), il flusso che si instaura è del tipo seguente:



4.5 SESSIONE EVENTI

Subito dopo aver instaurato una connessione TCP-IP tra la macchina che richiede il servizio (il client) e la macchina che offre il servizio (il server), il flusso che si instaura è del tipo seguente:



4.6 MESSAGGI OPEN PARTICOLARI

Oltre ai messaggi di comando esistono dei messaggi particolari che vengono trasmessi all'interno del flusso comunicativo come il messaggio di ACK e NACK.

4.6.1 MESSAGGIO DI ACK

Il messaggio Open di acknowledge ha la seguente sintassi:

***#*1##**

Questa frame indica che il messaggio Open, inviato dal Client e ricevuto dal Server, è sintatticamente e semanticamente corretto. Inoltre viene utilizzato come messaggio terminatore quando la risposta ad un messaggio Open preveda l'invio di uno o più messaggi in sequenza (richiesta stato o richiesta grandezze).

4.6.2 MESSAGGIO DI NACK

Il messaggio OPEN di Not-acknowledge (NACK) è

***#*0##**

Questa frame indica che il messaggio Open, inviato dal Client e ricevuto dal Server, è semanticamente o sintatticamente errato. Inoltre viene utilizzato come messaggio terminatore quando la risposta ad un messaggio Open preveda l'invio di uno o più messaggi in sequenza (richiesta stato o richiesta grandezze). In questo caso, il client deve considerare non validi i messaggi ricevuti prima del NACK.

4.7 OPEN DI COMANDO/STATO

Il messaggio Open, che ha tale funzione, è così strutturato:

***CHI*COSA*DOVE##**

Tale frame è utilizzata sia nella sessione comandi/azioni che in quella degli eventi.

- Sessione degli eventi: indica che un oggetto del sistema My Home ha cambiato il suo stato.

- Sessione comandi/azioni:
  - Messaggio inviato dal server al client in risposta ad una richiesta stato.
  - Messaggio inviato dal client al server per richiedere l'esecuzione di un'azione.

4.7.1 TAG CHI

Il tag CHI, individua la funzione dell'impianto domotico interessata al messaggio OPEN in questione.

4.7.2 TAG COSA

Il tag COSA, individua l'azione da compiere (ON luci, OFF luci, dimmer al 20%, tapparelle SU, tapparella GIU, imposta programma 1 in centrale di termoregolazione, etc...)

Per ogni CHI (e quindi per ogni funzione) viene specificata una tabella dei COSA.

Il tag COSA può anche contenere dei parametri (facoltativi) specificati in questo modo:

COSA#PAR1#PAR2...#PARn.

4.7.3 TAG DOVE

Il tag DOVE individua l'insieme di oggetti interessati al messaggio (zona, gruppo di oggetti, ambiente specifico, singolo oggetto, intero sistema).

Per ogni CHI (e quindi per ogni funzione) viene specificata una tabella dei DOVE.

Il tag DOVE può anche contenere dei parametri (facoltativi) specificati in questo modo:

DOVE#PAR1#PAR2...#PARn.

4.8 OPEN DI RICHIESTA STATO

Il messaggio Open di richiesta stato è così strutturato:

***#CHI*DOVE ##**

Viene inviato dal client nella sessione comandi/azioni per richiedere informazioni sullo stato di un singolo oggetto, di un insieme di oggetti o di un intero sistema. Il server risponde a questa richiesta inviando uno o più messaggi Open di stato.

La risposta deve terminare messaggio di ACK (o di NACK in caso di problemi o se l'oggetto/i di cui si richiede lo stato non è presente sul sistema).

Nel caso in cui il campo dove non è specificato, la richiesta di stato è generica a tutto il sistema.

4.9 OPEN RICHIESTA VALORE/GRANDEZZA

Il messaggio OPEN di richiesta grandezze è così strutturato:

***#CHI*DOVE*GRANDEZZA##**

Viene inviato dal client nelle sessioni di tipo comandi per richiedere informazioni sul valore di una grandezza di un singolo oggetto, di un insieme di oggetti o di un intero sistema.

Il server risponde ad una richiesta grandezza inviando uno o più messaggi OPEN di valore grandezza così strutturati:

***#CHI*DOVE*GRANDEZZA*VAL1*...*VALn##**

Il numero di campi VAL dipende dalla GRANDEZZA richiesta.

Il messaggio di risposta è seguito dal messaggio ACK.

Se non vi è seguito alla richiesta con una risposta o si verifica un errore alla frame di richiesta grandezza segue un NACK.

E' possibile che ad una richiesta di grandezza segua oltre ad una frame di valore grandezza uno o più messaggi OPEN di tipo STATO chiusi dal messaggio di ACK o NACK (in caso di errore).

Il messaggio di risposta viene generalmente inviato anche nelle connessioni eventi.

Il server OPEN invia il messaggio di valore grandezza a fronte di variazioni della grandezza o semplicemente se il dispositivo che lo invia deve segnalare periodicamente lo stato della

grandezza (ad esempio ogni 15 minuti le sonde della funzione di termoregolazione comunicano il valore della temperatura rilevata anche se questa non varia).

4.10 OPEN RICHIESTA VALORE/GRANDEZZA

Il messaggio OPEN di scrittura grandezze è così strutturato:

***#CHI*DOVE*#GRANDEZZA*VAL1*...*VALn##**

Il messaggio comporterà un'effettiva modifica solo per le grandezze abilitate alla scrittura.

Tale comando è inviato dal client nelle sessioni di tipo comandi per modificare i valori della grandezza di un singolo oggetto, di un insieme di oggetti o di un intero sistema. Il server risponde con il messaggio di ACK o di NACK.

5 BUILDING AUTOMATION MANAGEMENT SYSTEM

5.1 INTRODUZIONE

Avendo scelto Zabbix come sistema di management di riferimento, si vedrà adesso in che modo poterlo estendere ed utilizzare per renderlo una soluzione di Service Management, verificandone l'integrazione con sistemi di acquisizione delle informazioni diversi dai canonici dispositivi di rete o server, componenti tipici di un'infrastruttura ICT.

Si dimostrerà come sia effettivamente possibile integrare un qualunque oggetto/servizio in una piattaforma di management tradizionale, grazie all'utilizzo del protocollo SNMP.

Quindi, pur continuando a considerare in questa sede un ambito applicativo affine all'ICT, un semplice ed immediato processo di generalizzazione e astrazione ci consente di applicare potenzialmente gli stessi principi ad un qualunque altro ambito applicativo, con i soli vincoli che quest'ultimo sia in qualche modo interfacciabile con un'infrastruttura di rete informatica e che i suoi dispositivi/oggetti/servizi possano comunicare su tale rete mediante protocollo SNMP.

Qui, in particolare, ci si preoccuperà di realizzare una piattaforma per il management di sistemi domotici o di building automation.

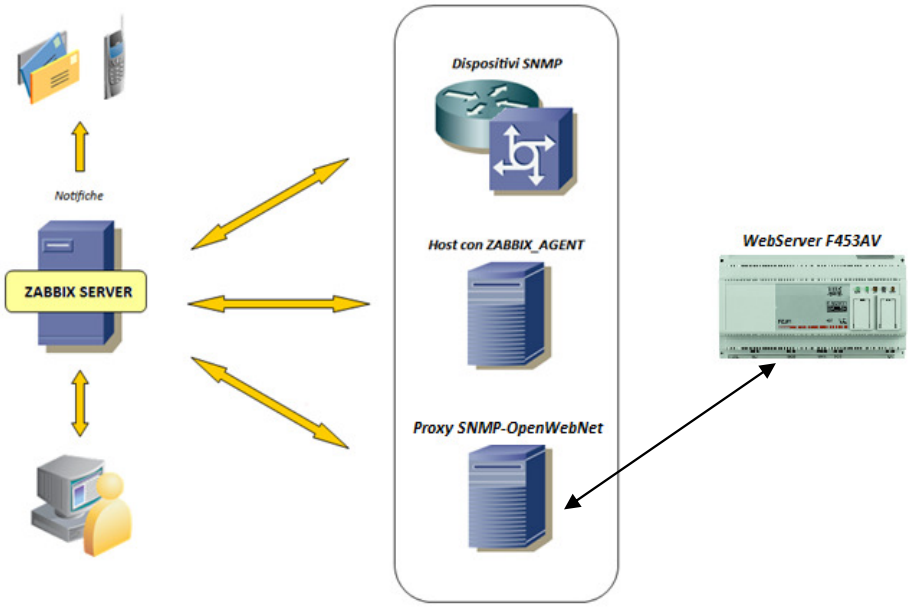
Nei paragrafi successivi si descriverà l'architettura implementata ed, infine, verranno descritte in dettaglio tutte le componenti del sistema, rivolgendo particolare attenzione a quelle funzionalità che sono state utilizzate nello sviluppo della parte sperimentale di questo lavoro.

5.2 ARCHITETTURA GENERALE E COMPONENTI

In termini generali, l'architettura della piattaforma implementata non si discosta da quelle in cui opera un tipico sistema di network e system management.

La particolarità principale consiste nella presenza di un nuovo servizio di proxy SNMP-OpenWebNet, che è l'elemento cruciale che consente l'integrazione di una infrastruttura domotica basata sui prodotti BTicino MyHome con il "mondo" ITC tradizionale.

Dal punto di vista dell'infrastruttura ICT questi risponderà come un tradizionale agent SNMP, e di fatto lo è. In più, verrà effettuata la traduzione delle richieste SNMP in comandi conformi al protocollo Open Web Net, che verranno indirizzati al gateway SCS-Ethernet BTicino.



Tale funzionalità è stata implementata mediante la realizzazione di un apposito modulo che estende il comportamento standard dell'agent SNMP.

La presenza del dispositivo F453AV è stata simulata mediante un modulo software creato ad-hoc, conforme alle specifiche Open Web Net.

Nei paragrafi successivi verranno presentati e descritti nel dettaglio i moduli funzionali che implementano l'architettura in oggetto.

5.3 NET-SNMP

Net-SNMP è una suite di applicazioni utilizzate per implementare i protocolli SNMP v1, SNMP v2c e SNMP v3, sia su IPv4 sia su IPv6.

La suite include:

- Applicazioni a linea di comando per:
  - leggere informazioni da un qualsivoglia dispositivo SNMP, utilizzando richieste singole (snmpget, snmpgetnext) o multiple (snmpwalk, snmptable, snmpdelta);
  - modificare informazioni di configurazione su dispositivi SNMP (snmpset);

    - estrarre una collezione prefissata di informazioni da dispositivi SNMP (snmpdf, snmpnetstat, snmpstatus);
    - convertire gli OID delle MIB tra le forme numerica e testuale e visualizzare il contenuto e la struttura delle MIB (snmptranslate).
- Un browser grafico per le MIB (tkmib) scritto in Tk/perl;
- Un'applicazione di tipo daemon per la ricezione delle notifiche SNMP (snmptrapd). Le notifiche possono essere registrate (syslog, Windows Event log, file di testo), inoltrate ad altro sistema di management SNMP o passate ad un'applicazione esterna;
- Un agente programmabile per rispondere a richieste SNMP (snmpd), capace di riconoscere un gran numero di moduli MIB, estendibile mediante l'utilizzo di moduli caricati dinamicamente, script e comandi esterni, e mediante l'utilizzo dei protocolli SNMP multiplexing (SMUX) e Agent Extensibility (AgentX);
- Una libreria per lo sviluppo di nuove applicazioni SNMP, utilizzando i linguaggi C e Perl.

In questo lavoro, la suite è stata utilizzata in particolare per l'implementazione dell'agent SNMP che viene interrogato dal sistema di monitoraggio.

5.4 MIB ICARCNR-ROOT E ICARCNR-OWN

Per la lettura delle informazioni mediante protocollo SNMP da parte del sistema di monitoraggio si sono sviluppati due file MIB con le definizioni degli OID necessari.

A tale scopo si è richiesto allo *IANA* l'assegnazione ufficiale di un Private Enterprise Number (PEN). All'ICAR-CNR è stato attribuito il valore **39840**:

```
.1.3.6.1.4.1.39840 = enterprises.39840
```

Successivamente, grazie al PEN che garantisce l'univocità a livello internazionale della numerazione, è stato realizzato il MIB fondamentale **ICARCNR-ROOT**, il cui contenuto è indicato di seguito:

```
ICARCNR-ROOT-SNMP-MIB DEFINITIONS ::= BEGIN

IMPORTS
        enterprises,
        MODULE-IDENTITY
                FROM SNMPv2-SMI;

icarcnr MODULE-IDENTITY
        LAST-UPDATED "201206040921Z"    -- Jun 4, 2012 9:21:00 AM
        ORGANIZATION "ICAR CNR Palermo"
        CONTACT-INFO
                "Antonio Messina <messina@pa.icar.cnr.it>
                Pietro Storniolo <storniolo@pa.icar.cnr.it>

                Viale Delle Scienze
                Edificio 11
                90128 Palermo"
        DESCRIPTION
                "Extension of the MIB sub-tree of ICAR CNR Palermo"
        REVISION "201206040921Z"        -- Jun 4, 2012 9:21:00 AM
        DESCRIPTION
                "Initial version."
        -- 1.3.6.1.4.1.39840
```

Da questo infine è stato derivato un ramo di OID da dedicare al progetto in questione: **ICARCNR-OWN**.

Nell'assegnazione degli OID di quest'ultimo MIB, al fine di creare e mantenere un certo parallellismo in termini di valori, si è utilizzata la numerazione seguita dal protocollo Open Web Net nella definizione dei CHI.

Quindi, data la seguente tabella dei CHI:

Codice	Descrizione
0	Scenari
1	Illuminazione
2	Automatismi
3	Controllo carichi
4	Gestione della temperatura
5	Antifurto
6	Videocitofonia base
13	Gestione del gateway

si è definito, ad esempio:

```
...
ownGatewayControl OBJECT IDENTIFIER
        -- 1.3.6.1.4.1.39840.1.13
        ::= { own 13 }
...
```

Analogo approccio è stato seguito per la numerazione degli OID da interrogare, ottenendo quindi, ad esempio:

```
...
ownGatewayMacAddress OBJECT-TYPE
        SYNTAX  MacAddress
        MAX-ACCESS read-only
        STATUS  current
        DESCRIPTION
                "The MAC Address of the OWN Gateway"
        -- 1.3.6.1.4.1.39840.1.13.12
        ::= { ownGatewayControl 12 }
...
```

5.5 OWNSNMPAGENT

Grazie alla direttiva *pass* prevista dal file di configurazione del demone `snmpd`, è possibile rendere esplicita l'esecuzione di un qualsivoglia file esterno allorquando viene richiesto un qualunque OID sottostante il ramo specificato.

Poiché nel nostro caso si richiede che l'agent SNMP debba rispondere adeguatamente agli OID con prefisso .1.3.6.1.4.1.39840.1, si è quindi inserita nel file `snmpd.conf` la seguente direttiva:

```
pass .1.3.6.1.4.1.39840.1 /bin/sh /usr/local/bin/ownSnmpAgent
```

ownSnmpAgent è uno script di shell che implementa la funzionalità di proxy SNMP-OpenWebNet.

Poiché ad ogni OID corrisponde un oggetto del mondo OpenWebNet, ad ogni richiesta lo script effettuerà un brevissima connessione al gateway inviando i comandi corrispondenti e conservandone la risposta.

Quest'ultima verrà quindi ripulita dei codici di controllo propri del linguaggio Open al fine di considerarne solo la parte rilevante contenente le informazioni di interesse, che saranno quindi restituite, a meno di trasformazioni formali, sotto forma di valore dell'OID richiesto.

Coerentemente con le finalità del presente lavoro, ci si è limitati alla integrazione di tools a linea di comando standard (*netcat*, *awk*) al fine di produrre i risultati desiderati.

Ad esempio, la richiesta dell'OID *ownGatewayTime* (1.3.6.1.4.1.39840.1.13.0) verrà evasa dal seguente codice:

```
1.3.6.1.4.1.39840.1.13.0)
  echo "string" ;
  echo '*#13**0##' | nc localhost 20000 |
  awk '{ STR=substr($0,7,length()-14) ; print gensub(/.*\*\*[0-9]+\*/,"",1,STR) }' |
  awk '{
    split($0, a, "*");
    PLUS=gensub(/^0/,"+",1,substr(a[4],1,1));
    DELTA=strtonum(substr(a[4],2));
    print sprintf("%d:%d:%d,%s%d", a[1],a[2],a[3],PLUS,DELTA);
  }'
  ;;
```

5.6 OWNSERVER

Essendo pubblicamente disponibili le specifiche complete del protocollo Open Web Net, a partire da queste è stato possibile realizzarne un simulatore, codificando un subset minimale delle funzionalità.

A tale scopo si è utilizzato il linguaggio Expect, particolarmente indicato per implementare script che interagiscono con programmi terzi.

Il server viene eseguito per mezzo di un wrapper che, aperto un socket TCP in ascolto sulla porta 20000 (standard OpenWebNet), provvede alla redirezione dello standard input e dello standard output del processo su quel socket:

```
#!/bin/bash
/usr/local/bin/tcpserver 127.0.0.1 20000 /usr/bin/expect ownServer.exp
```

Un esempio di implementazione di una funzionalità Open è dato dal seguente codice Expect:

```
#!/usr/bin/expect

proc ACK {} { send "*#*1##" }
proc EOC {} { send "##" ; ACK }
ACK

while {1} {
  expect {
    eof { break }
    .
    .
    .
    "\*#13\*\*0##" { send "*#13**0*"; send [exec ./ownCommand getTime] ; EOC }
    .
    .
    .
  }
}
```

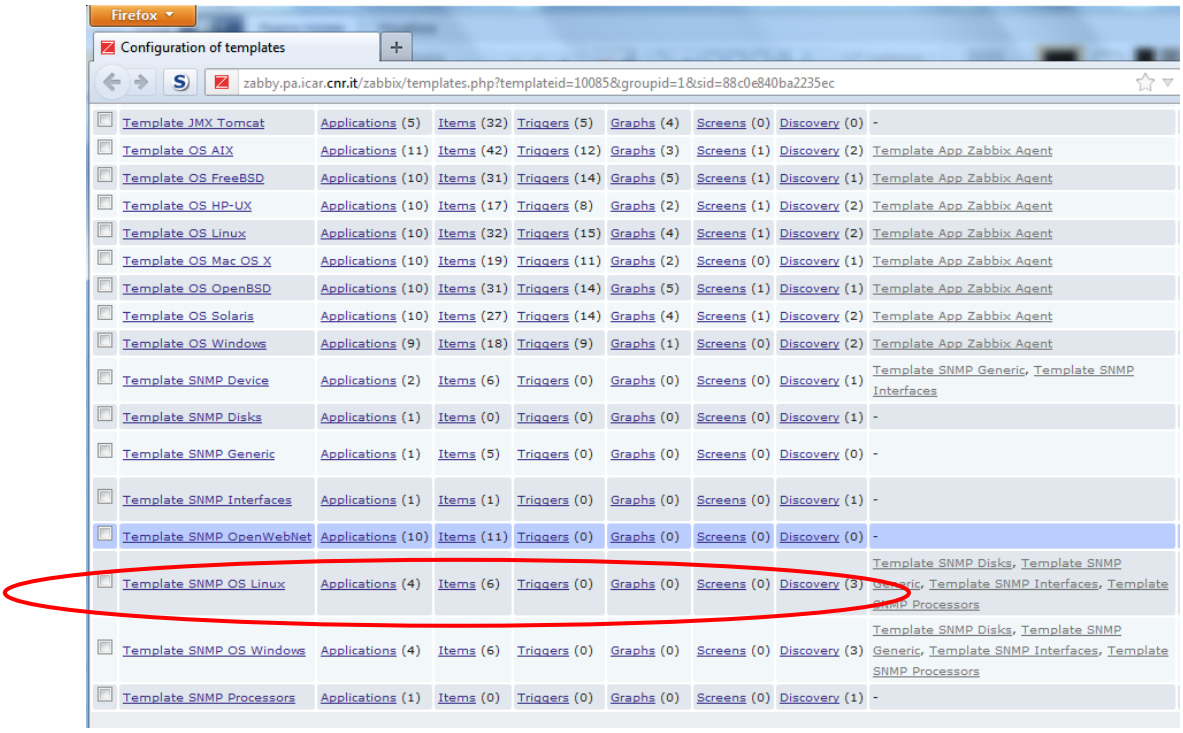
dove *ownCommand* è un ulteriore, piccolo, script di shell che implementa la specifica funzionalità richiesta:

```
#!/bin/bash
case $1 in
...
getTime)
        TZ=`date +%z |
        awk '{ if (/^\+[0-9]*/) printf 0; else printf 1; print substr($1,2,2) }'`
        date +%H*%M*%S*"$TZ"
        ;;
...
```

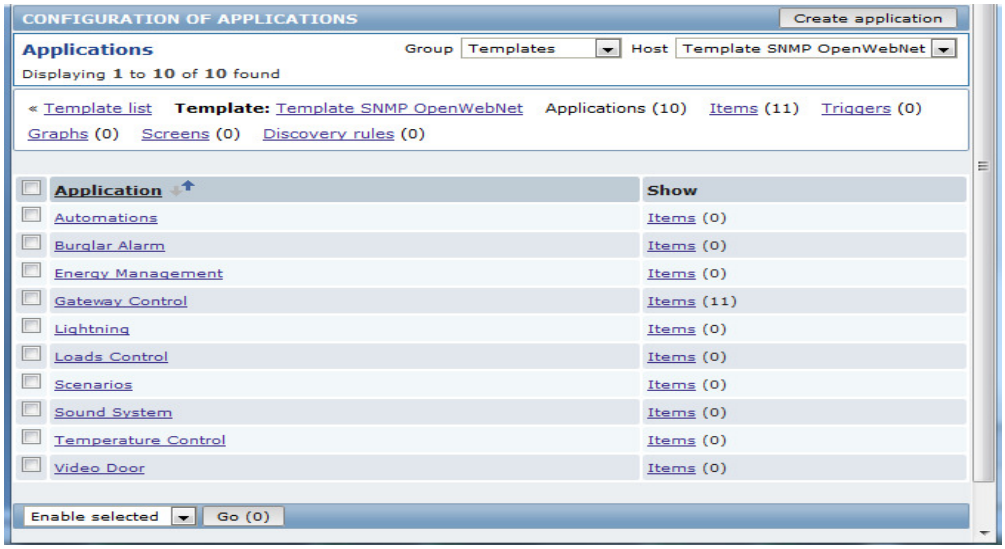
5.7 ZABBIX

Avendo a disposizione un gateway OpenWebNet (simulato), un servizio di agent SNMP attivo, le MIB realizzate ad-hoc e l'estensione dell'agent SNMP per tali MIB, è stato quindi possibile andare a definire su Zabbix il gateway stesso.

Inizialmente si è provveduto alla creazione di un template specifico per Open Web Net:

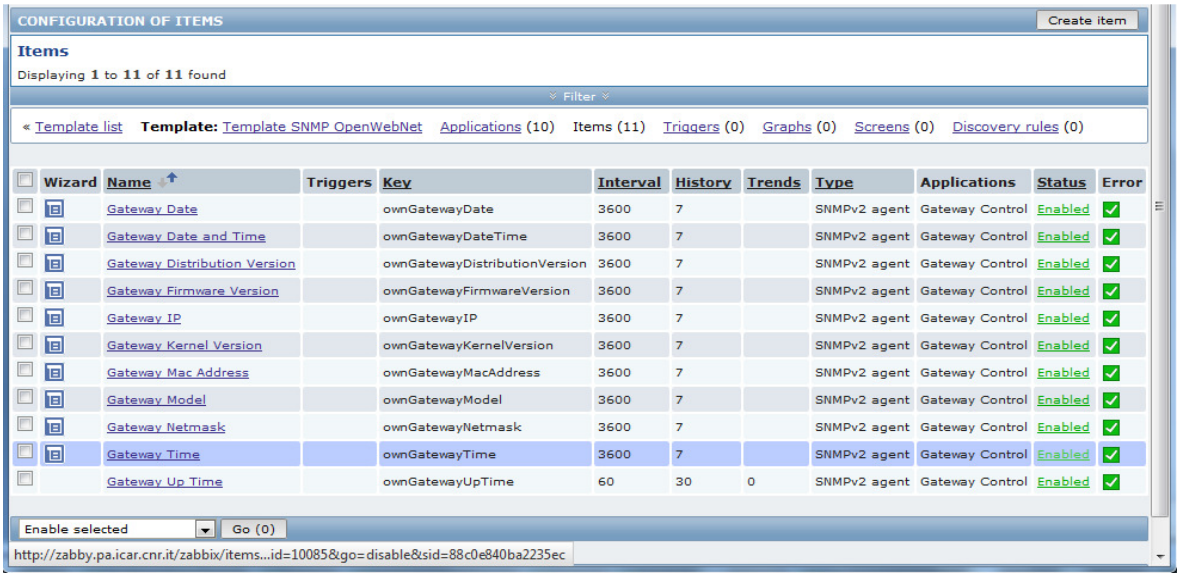


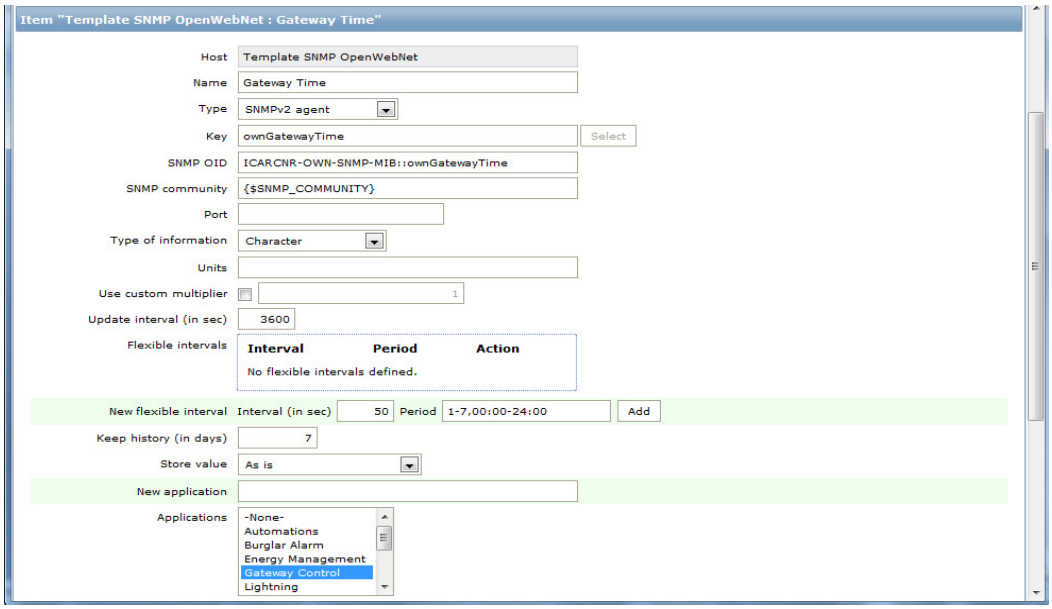
All'interno di questo è stato quindi definito un gruppo di *Application*, corrispondenti ai **CHI** del protocollo Open Web Net:



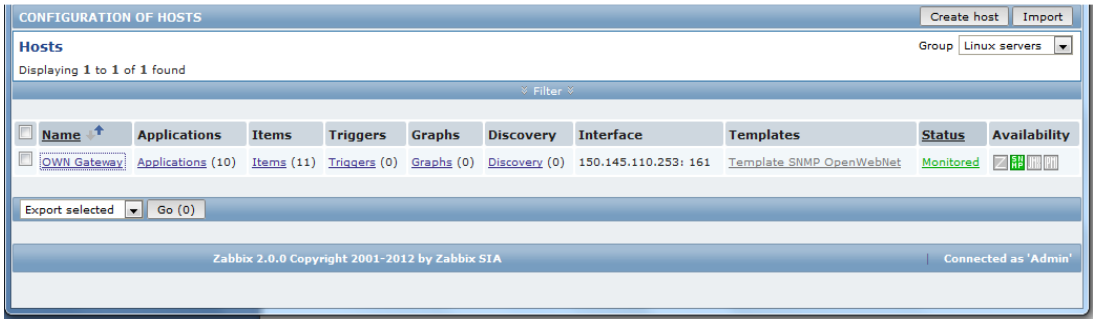
Infine, si sono creati gli item.

In questa sede sono stati presi in considerazione i soli item relativi alla *Application* Gateway Control, corrispondente al **CHI** 13, che restituisce informazioni sul gateway Open Web Net:

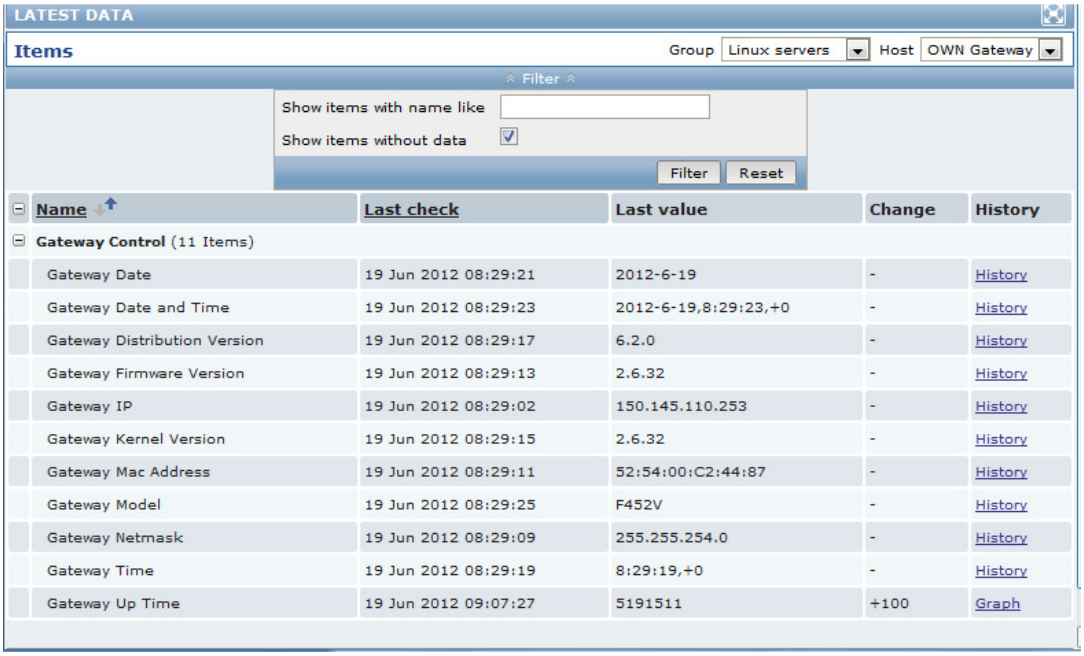




Una volta completata la definizione del template è stato creato in Zabbix un nuovo Host e vi si è associato il template SNMP OpenWebNet.



Subito dopo la creazione dell'host, il sistema di monitoraggio ha subito iniziato la raccolta dei dati mediante comandi SNMP standard:



LATEST DATA

Items — Group: Linux servers — Host: OWN Gateway

Filter

Show items with name like

Show items without data

Filter | Reset

Name	Last check	Last value	Change	History
Gateway Control (11 Items)				
Gateway Date	19 Jun 2012 08:29:21	2012-6-19	-	History
Gateway Date and Time	19 Jun 2012 08:29:23	2012-6-19,8:29:23,+0	-	History
Gateway Distribution Version	19 Jun 2012 08:29:17	6.2.0	-	History
Gateway Firmware Version	19 Jun 2012 08:29:13	2.6.32	-	History
Gateway IP	19 Jun 2012 08:29:02	150.145.110.253	-	History
Gateway Kernel Version	19 Jun 2012 08:29:15	2.6.32	-	History
Gateway Mac Address	19 Jun 2012 08:29:11	52:54:00:C2:44:87	-	History
Gateway Model	19 Jun 2012 08:29:25	F452V	-	History
Gateway Netmask	19 Jun 2012 08:29:09	255.255.254.0	-	History
Gateway Time	19 Jun 2012 08:29:19	8:29:19,+0	-	History
Gateway Up Time	19 Jun 2012 09:07:27	5191511	+100	Graph

6 CONCLUSIONI

6.1 INTRODUZIONE

I risultati ottenuti hanno dimostrato la possibile integrazione tra Zabbix ed il il gateway ethernet BTicino, utilizzando SNMP come protocollo di comunicazione tra le parti.

Questo risultato coincide esattamente con l'obiettivo che ci si era inizialmente prefissato: "Verificare la fattibilità dell'integrazione di sistemi di acquisizione delle informazioni con sistemi di elaborazione e presentazione, attraverso l'uso di un protocollo comune, al fine di pervenire a una soluzione completa per il Service Management".

Nell'architettura implementata in questa sede, quindi, le componenti oggetto dell'obiettivo possono essere definite nel seguente modo:

- Service Management: Building Automation Management
- Sistema di acquisizione delle informazioni: Gateway SCS-Ethernt F453AV;
- Sistema di elaborazione e presentazione: Zabbix, piattaforma Open Source per il network e system management di livello enterprise;
- Protocollo comune: SNMP.

6.2 ULTERIORI POSSIBILI SVILUPPI DELLA PIATTAFORMA

Considerato quanto realizzato, è chiaro che la piattaforma implementata avrebbe bisogno di ulteriori interventi ed estensioni prima di poter essere utilizzata come "vero" Building Automation Management System in ambiente di produzione.

E' stato infatti considerato ed implementato a scopo esemplificativo solo un subset minimo dell'insieme dei comandi di controllo previsti dal protocollo Open Web Net.

Completando il supporto ad Open Web Net, si arriverebbe a monitorare e controllare qualunque tipo di componente di un impianto domotico MyHome, e anche lo stato dei dispositivi di potenza a valle degli attuatori.

Inoltre, il sistema già di per se consentirebbe il monitoraggio e controllo in tempo reale di più gateway SCS-Ethernet, e quindi di più bus domotici diversificati e separati tra loro.

Unita alla possibilità di utilizzo della componente "reattiva" di Zabbix, tale flessibilità può consentire la definizione ed esecuzione di azioni complesse, che possano agire

contemporaneamente su impianti domotici anche distinti che quindi posso cooperare per ottenere effetti altrimenti non raggiungibili.

Inoltre, sviluppando gli appositi proxy SNMP, è possibile interfacciare e far interagire tra loro anche impianti domotici o di building automation utilizzanti standard e tecnologie differenti da SCS/Open Web Net utilizzata da BTicino, come ad esempio quelli del mondo Konnex (KNX).

6.3 APPLICABILITÀ DELLA SOLUZIONE AD ALTRI AMBITI OPERATIVI

La possibilità di realizzazione e utilizzo di nuove MIB SNMP ha consentito l'interfacciamento al sistema di management di una nuova categorie di "sensori/dispositivi": i gateway SCS-Ethernet di un sistema domotico MyHome di BTicino.

Quindi, il protocollo SNMP consente potenzialmente, grazie alle sue caratteristiche d'interoperabilità e di espandibilità, la misurazione e il controllo di qualunque tipo di dispositivo o servizio accessibile da un'infrastruttura di rete informatica, permettendo l'utilizzo di un tradizionale sistema di network e system management, eventualmente opportunamente esteso nelle sue funzionalità, anche in ambiti diversi dall'Information and Communication Technology, come appunto nel caso in oggetto.

Ulteriori campi di applicabilità potrebbero, ad esempio, anche essere:

- **Hospitality**, grazie al controllo e monitoraggio di switch VDSL (Very High Bit Rate Subscriber Line). VDSL è una tecnologia che consente di offrire agli utenti accessi a banda larga sfruttando il normale cablaggio telefonico in rame e viene utilizzata tipicamente all'interno delle strutture alberghiere, ma è applicabile anche a strutture quali ospedali, università, musei, caserme militari, supermercati, stabilimenti produttivi e logistici;
- **Telemedicina**, mediante, ad esempio, sistemi integrati di videocomunicazione e teleconsulto, che consentono di gestire i dati anagrafici e sanitari di pazienti provenienti da apparati elettromedicali. Questi sistemi, monitorabili anche via SNMP, sono progettati e sviluppati per supportare la consultazione medica remota e la formazione a distanza, gestiscono referti medici elettronici, elaborano e possono acquisire immagini mediche anche radiografiche, forniscono, infine, la semplice connessione a un'ampia gamma di periferiche mediche e

apparecchiature, inclusi: monitor per i segnali vitali, stetoscopi digitali, scanner radiografici, ECG e telecamere digitali;

- **Medicale**, come ad esempio per il controllo della qualità delle postazioni di lavoro PACS, in termini di verifica dello stato dei display (temperatura, durata della retroilluminazione, luminosità attuale), controllo dell'uniformità della luminanza, e tutte quelle altre informazioni utili per la garanzia della qualità.
- **Decision Support Systems** (DSS), ai quali piattaforme del tipo in esame potrebbero fornire dati "filtrati" mediante i meccanismi di azione/reazione. In tal modo, a partire da un qualunque tipo d'informazione misurabile mediante protocollo SNMP, è possibile ottenere un'integrazione tra un sistema deterministico ed uno non deterministico, quale appunto un DSS.

7 BIBLIOGRAFIA

Balsamo, S. *"Il protocollo SNMP" - Approfondimento per il corso di Reti di Calcolatori.* Università degli Studi Ca' Foscari di Venezia, 2004.

BTicino. *My Open Web Net Introduction.* 2011.

Curry, Jane. "Open Source Management Options." *Skills 1st consultancy: directory services, network and systems management.* 2008. http://www.skills-1st.co.uk.

Net-SNMP Community. *"Net-SNMP".* 2010. http://www.net-snmp.org/.

P. Storniolo, A. Messina. "*Progettazione e Realizzazione di un Intrusion Management System.*" Technical Report RT-ICAR-PA-10-02, ICAR CNR, Palermo, luglio 2010.

RFC-1155. *"Structure and Identification of Management Information for TCP/IP-based Internets".* 1990.

RFC-1157. *"A Simple Network Management Protocol (SNMP)".* 1990.

RFC-1212. *"Concise MIB Definitions".* 1991.

RFC-1213. *"Management Information Base for Network Management of TCP/IP-based internets: MIB-II".* 1991.

RFC-1215. *"Convention for defining traps for use with the SNMP".* 1991.

RFC-1514. *"Host Resources MIB".* 1993.

RFC-2263. *"SNMPv3 Applications".* 1998.

RFC-2578. *"Structure of Management Information Version 2 (SMIv2)".* 1999.

RFC-2579. *"Textual Conventions for SMIv2".* 1999.

RFC-2580. *"Conformance Statements for SMIv2".* 1999.

RFC-2790. *"Host Resources MIB".* 2000.

RFC-3411. *"An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks".* 2002.

RFC-3412. *"Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)".* 2002.

RFC-3413. *"Simple Network Management Protocol (SNMP) Applications".* 2002.

RFC-3414. *"User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)"*. 2002.

RFC-3415. *"View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)"*. 2002.

RFC-3416. *"Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)"*. 2002.

RFC-3417. *"Transport Mappings for the Simple Network Management Protocol (SNMP)"*. 2002.

RFC-3418. *"Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)"*. 2002.

Zabbix SIA. *"Zabbix :: An Enterprise-Class Open Source Distributed Monitoring Solution"*. 2010. http://www.zabbix.com/.