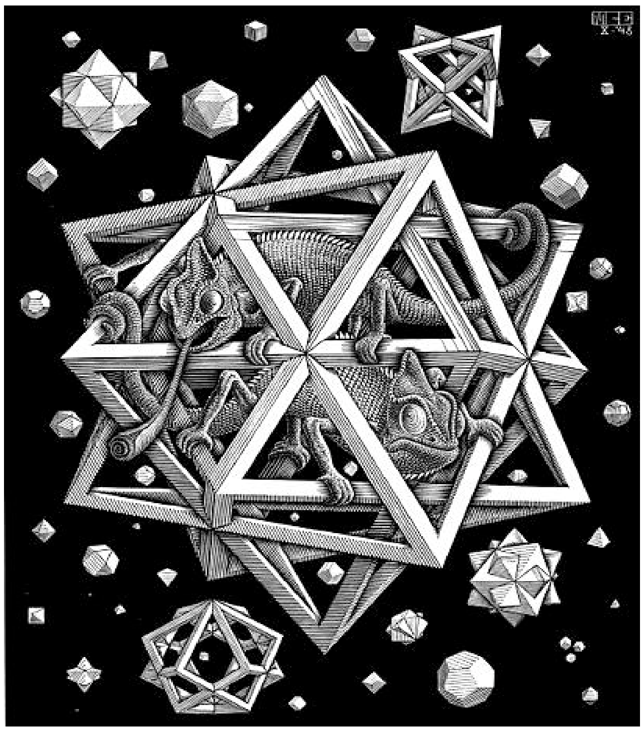
ICAR-CNR
Istituto di Calcolo e Reti ad Alte prestazioni
(CERE-CNR, CEntro di studio sulle Reti di Elaboratori)
Consiglio Nazionale delle Ricerche (Palermo, Italy)



(Stars, M. C. Escher)

A Management Architecture for Active Networks

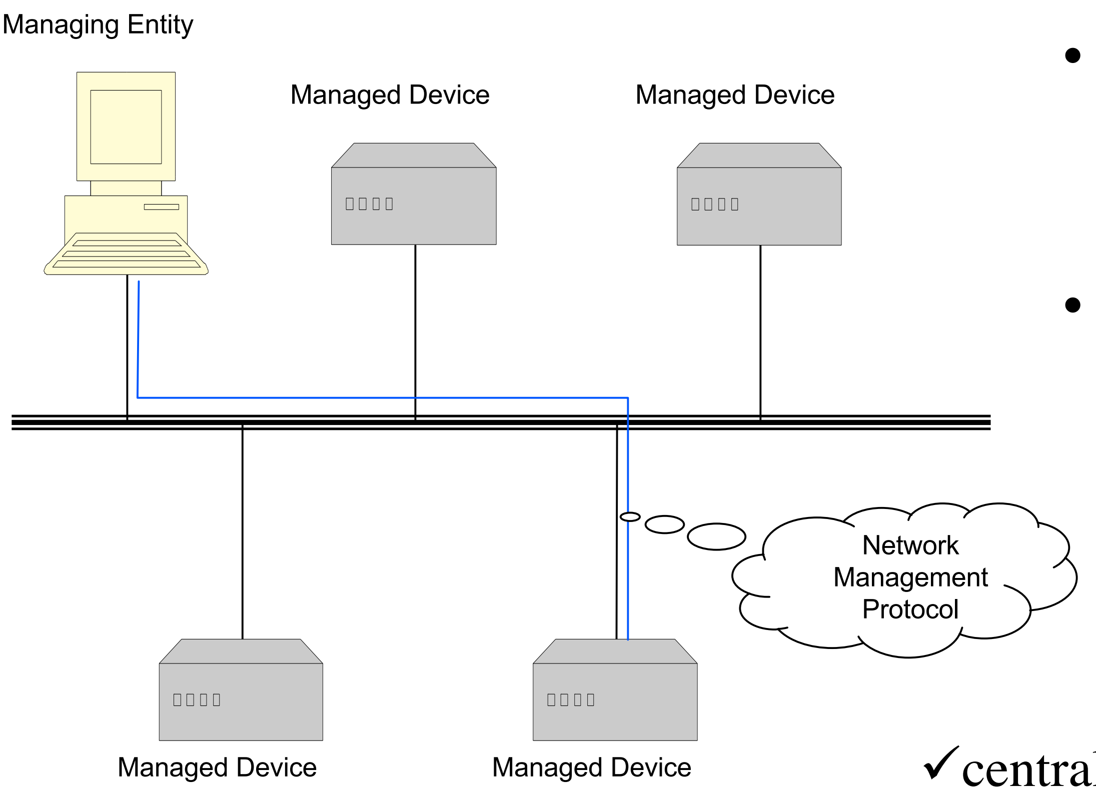
A. Barone, P. Chirco, **G. Di Fatta**, G. Lo Re

AMS2002, Active Middleware Service
Edinburgh (UK), 23 July 2002

Overview

- Network Management
  - SNMP framework
  - Active Networks for Network Management
- Active Networks Management
  - Active MIB and Active Local Agent
  - Active Routers Testbed and ANgate
    - Manager and Monitor tools
- Network Events Mining
- Conclusions

Network Management



- Managing Entity
- Managed Device
  - Local Agent
  - MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

✓centralized approach: polling

✓active local agent role: simple traps

✓issues: scalability and efficiency

Active Networks for Network Management

Active Networks benefits for Network Management are:

- availability of information in intermediate nodes
- data processing along the path
- distributed, cooperative, autonomous strategies

(e.g., SmartPackets)

Active Networks Management

AN allow and promote the **fast and dynamic introduction of new services and applications**. Active Applications are difficult to be debugged and monitored.

Active Networks require a more efficient approach for network and applications management.

Active Routers Testbed (ART)

A 40 nodes cluster at ICAR-CNR to study and test Active Networks

ART is a hardware platform which prototypes a network computational environment according to the **Active Networks** paradigm.

Aims of the **ART** project:

- testing new network services and protocols
- developing new active applications
- network resource optimisation

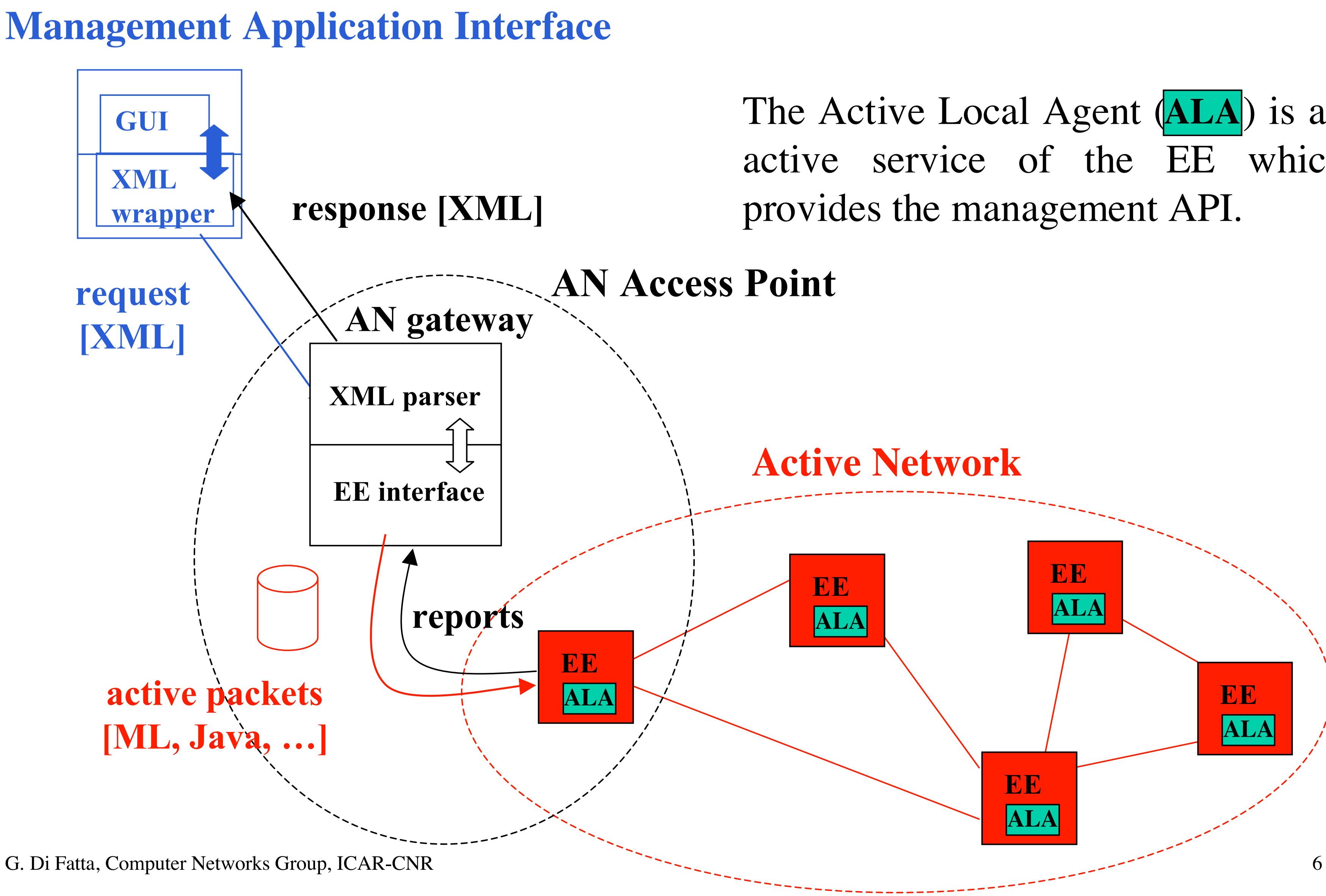
Practical need for an Active Management Architecture

<u>40 Active Nodes</u>

- Static Topology Configuration (80 peer-to-peer links)
- Dynamic Topology Configuration (240 port ethernet switch with management)

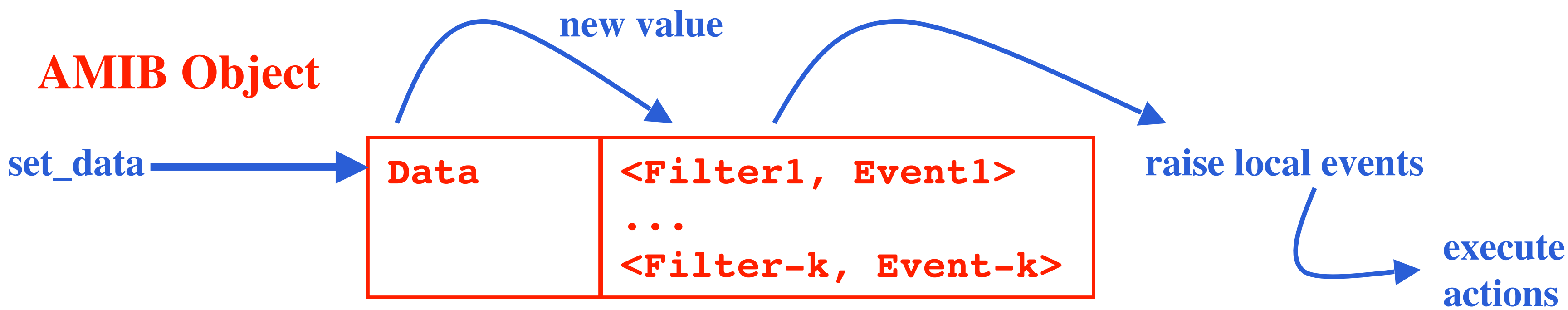
AN Management Framework

The Active Local Agent (ALA) is an active service of the EE which provides the management API.



Active Local Agent (ALA)

ALA is a SNMP-like management agent which exploits AN advantages: it provides the API for an object-oriented **Active MIB** and it is **programmable** according to a **Filter-Event-Action** model.

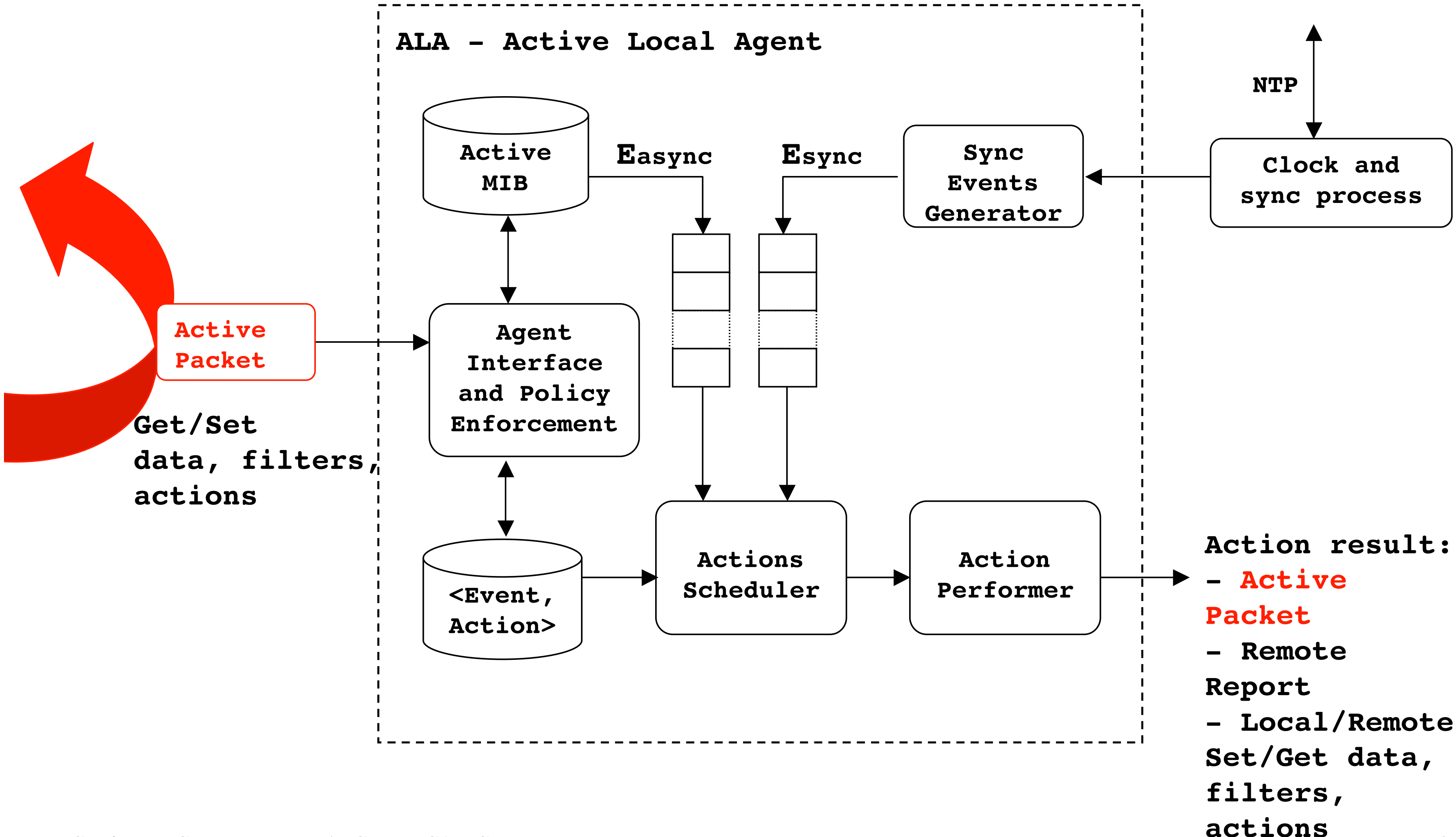


AMIB Objects are variables of Active Applications. AAs can explicitly submit data to ALA and associate Filters with data.

A **Filter** is a test executed when the data value changes. An **Event** is raised if the test succeeds.

An **Action** is a capsule associated with an Event and it is released when that Event is raised. Events have a local scope and a default Action is defined to simply report the Event.

ALA Architecture

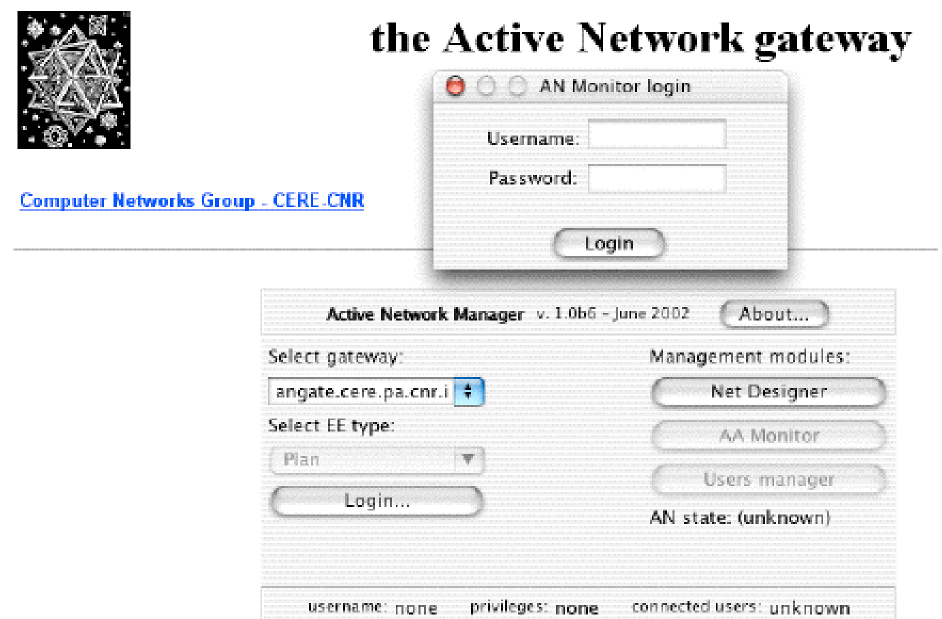


AN gateway

Services	*Implementation*	*Examples*
AN-level administrative	UNIX Shell scripts	setTopology, bootstrap_AN, shutdown_AN, …
AN-level public	packet-level (PLAN)	getTopology, ping, delivery_Path, delivery_Tree, …
AA-level administrative and public	ALA as PLAN service (OCaml)	setVarValue, getVarValue, getAppList, setFilterTest, …

ANgate - the Active Network gateway

- gateway server selection
- EE selection (only PLAN is supported in v1.0b6)
- user authentication (user, administrator)
- AN Manager module
- AA Monitor module



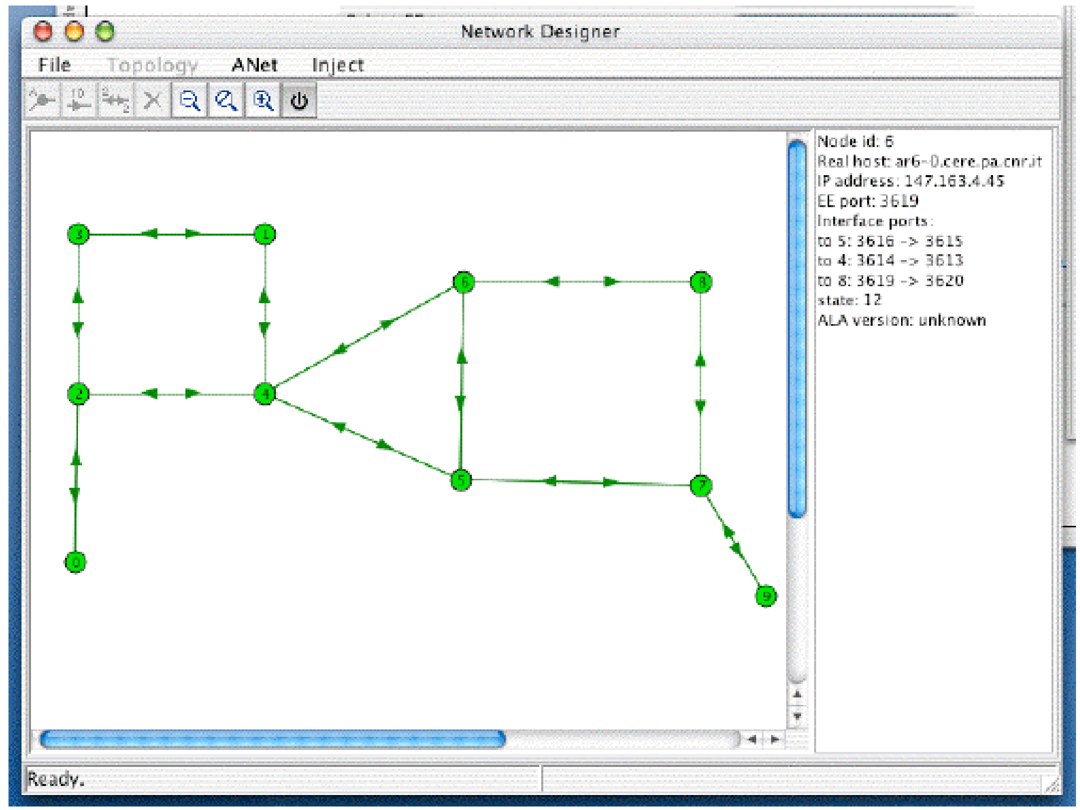
http://angate.cere.pa.cnr.it

username: guest

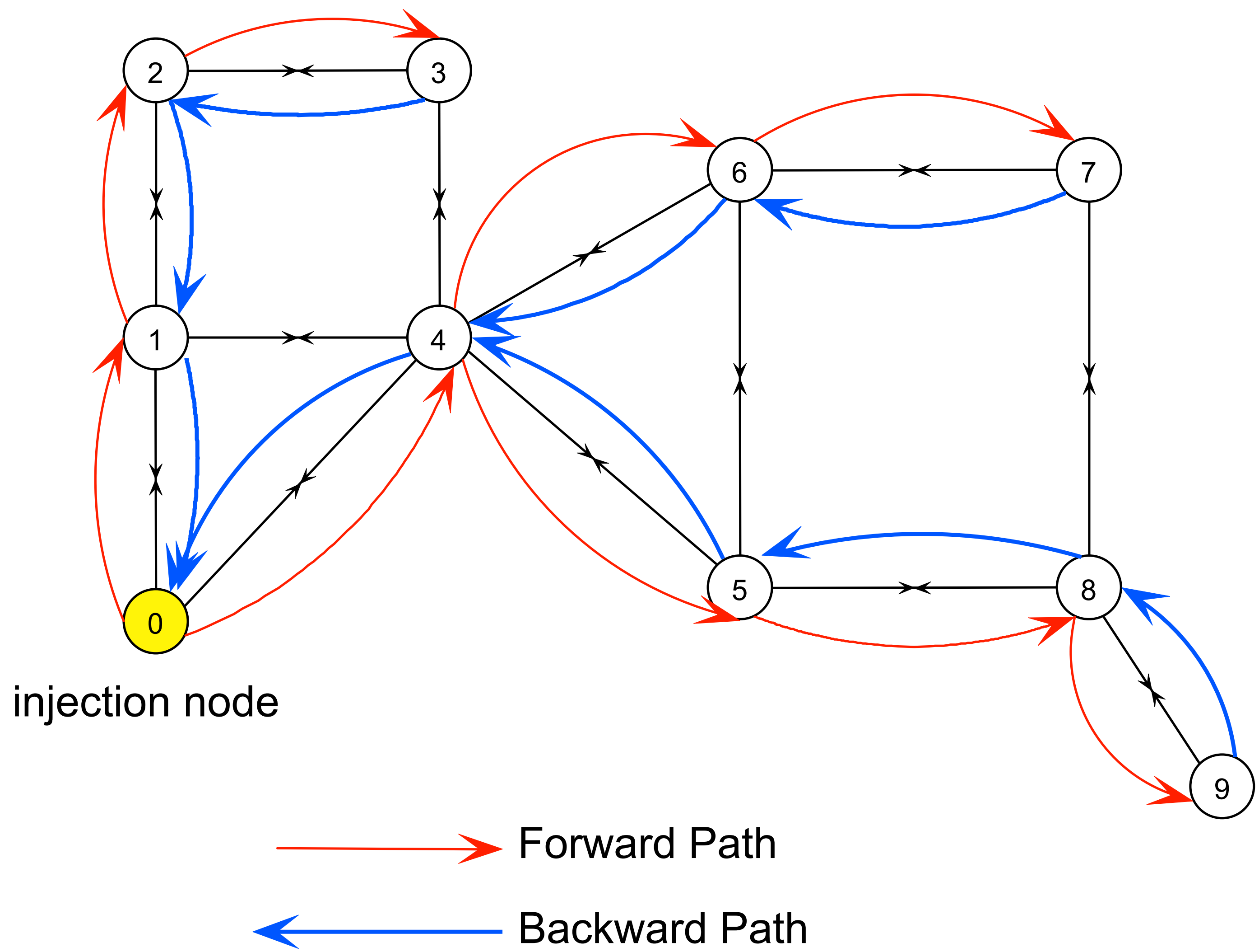
password: guest

Active Networks Manager

- Topology editor
- PLANet configuration and setup (only admin)
- Topology discovery
- Link and node status
- Capsule injection with navigation patterns
  - ✓ node
  - ✓ path
  - ✓ star
  - ✓ tree

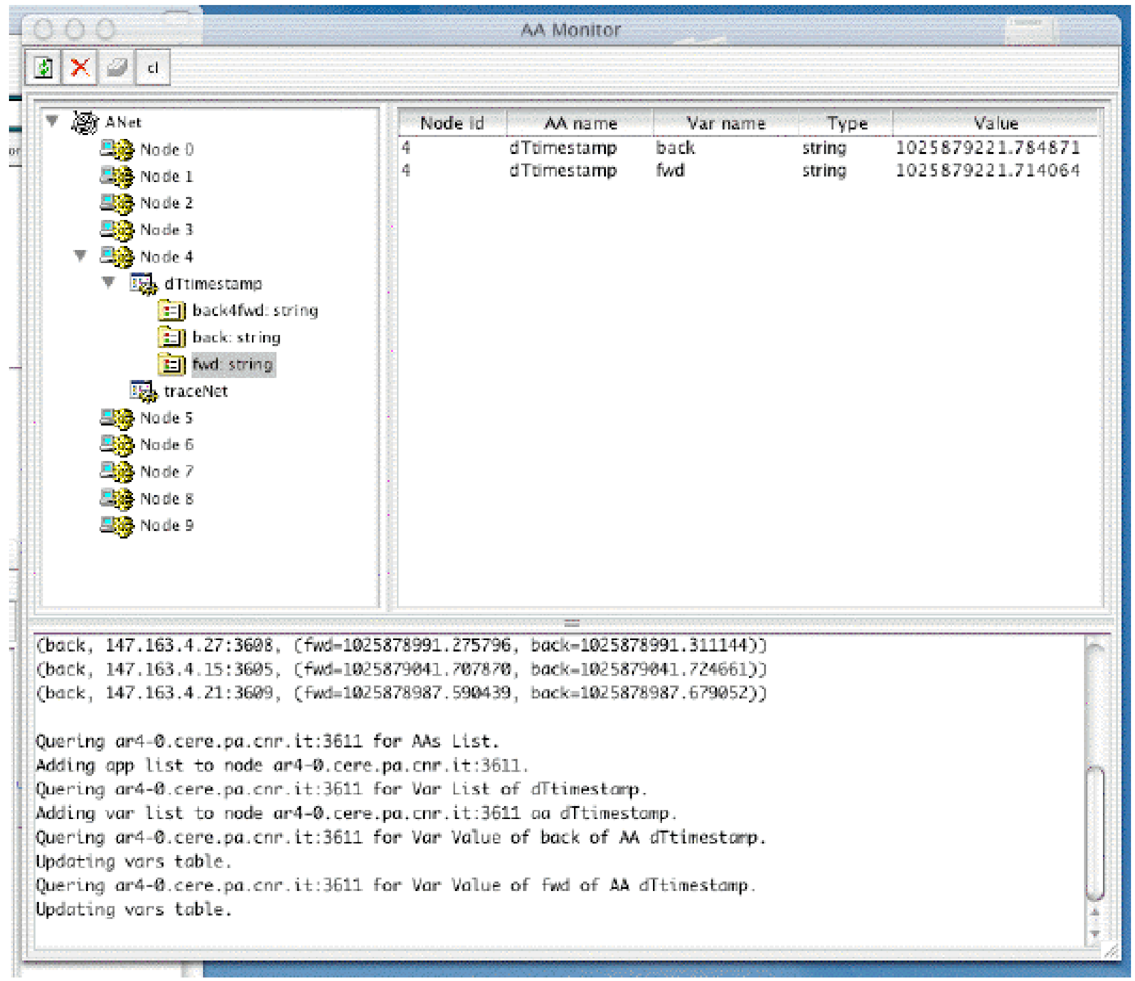


Navigation Pattern: Delivery_MST

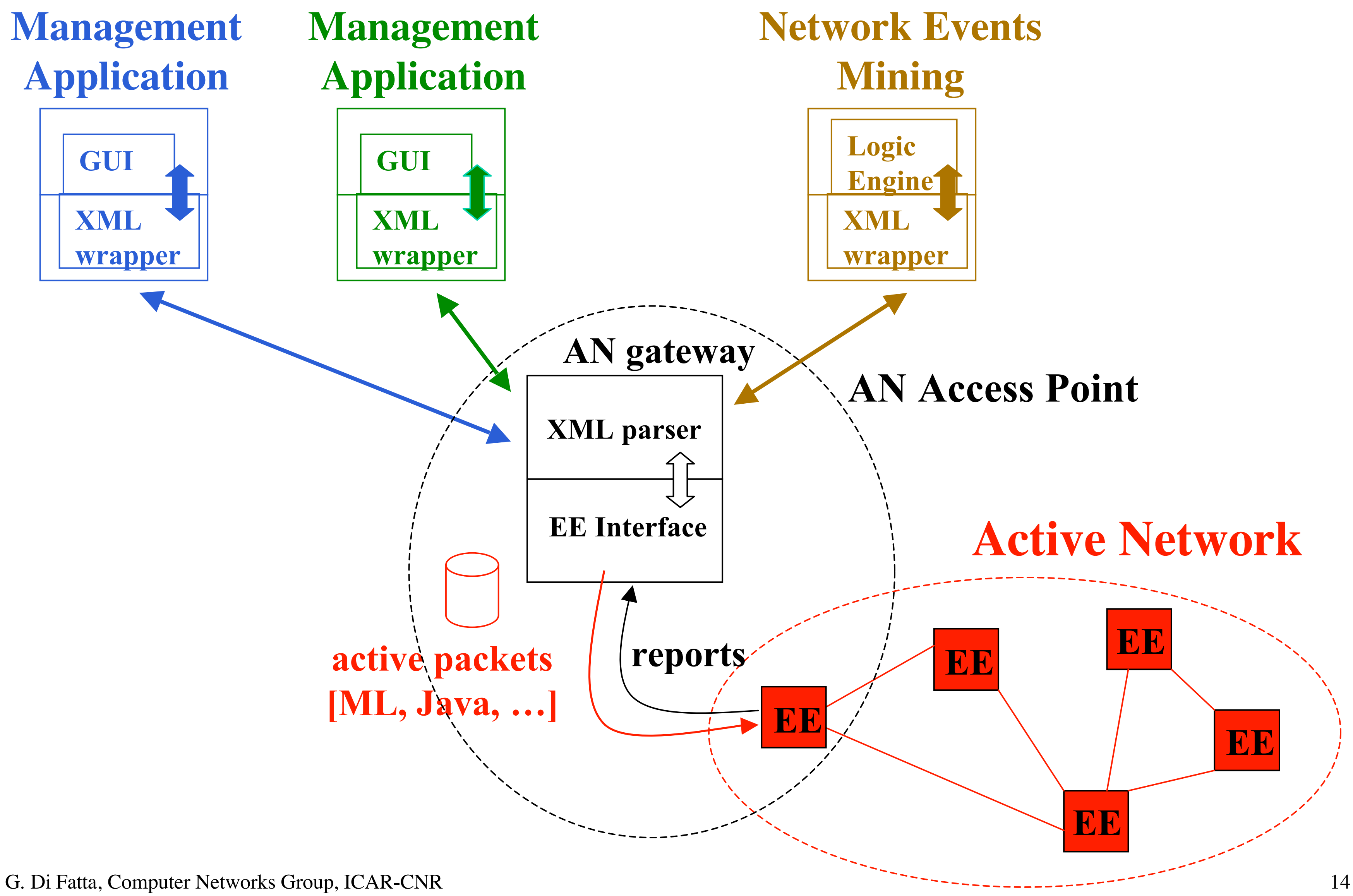


Active Applications Monitor

- ALA services GUI
- ANode/AA/AMIB-data tree
- AA tracing and debugging and performance measurements
- Message area for reports



New Applications



Network Events Mining (NEM)

NEM systems deals with **large archives of events** gathered from the network devices to extract useful information.

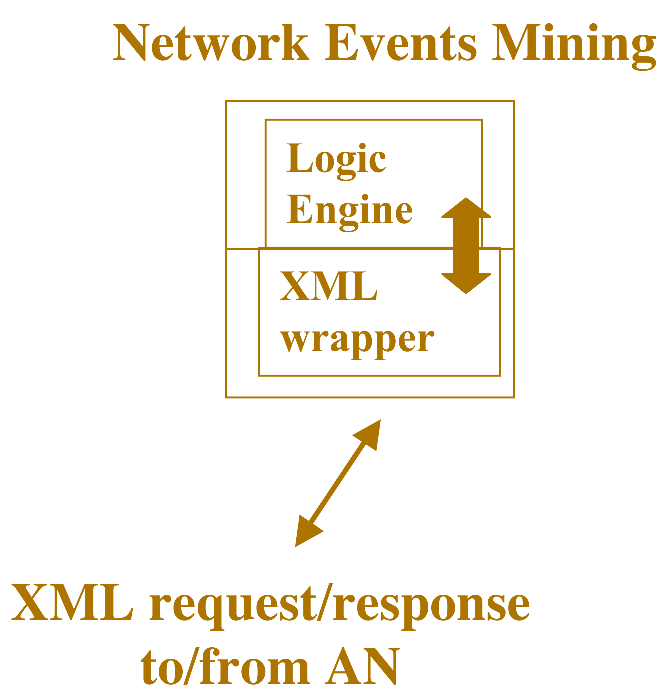
Typical **NEM goals** are:

- the diagnose of root causes of network faults and performance degradations by establishing relationships between network events;
- filtering event (alarm) flood by correlating several events into a single conceptual event.

NEM systems should provide **correctness** and **optimality**.

NEM Application

The Logical Managing Entity is based on the **Situation Calculus**, which is a dialect of the **First Order Logic** and allows the modeling of dynamic systems. It is implemented by the **Golog** logic programming language.



Situation Calculus

- entities
- events
- actions
- situations
- predicates
- fluents

The Logical System Design

1. **Ontological engineering process** provides
   - effective structured **representation** of all the network entities, events and actions.

     (predicates, fluents, actions, preconditions)

2. **Logical axiomatization of relationships** among the network events are to be defined.
   - The occurrence of an action makes the system move from a given situation s_i to the consequent situation s_{i+1}.

     (successor state axioms)

Ontological Analysis

Logical entities and relationships are expressed as **Predicates and Fluents**.

Predicates:

- `iface(I)`
- `address(I, <string>)`
- `node(N)`
- `flag_node(N)`
- `ala(A)`
- `ala_location(A, N)`
- `link(L)`

`connect(L, I1, I2) where link(L), iface(I1), iface(I2)`

- `belongs(I, N)`

. . .

Fluents

link_status(L, on\off, s)	In situation s the physical link L is on\off.
routing_table(N,<arrayTable>,s)	In situation s <arrayTable> is the routing table of node N.
neighbour(N1, N2, s)	In situation s nodes N1 and N2 are directly connected at network layer.
path(A, B, L, s)	**In situation s nodes A and B are connected at network layer by path L.**
iface_status(I, on\off, s)	In situation s interface I is on\off.
ala_status(A, on\off, s)	In situation s the Active Local Agent A is on\off.
node_status(N, on\off, s)	In situation s node N is on\off.
node_flag_status(N, on\off, s)	In situation s the flag of node N is on\off.

Primitive Actions

Link_up(L)	The network link L is set up (the semaphore is switched on in both the connected routers).
Link_down(L)	The network link L is set down (the semaphore of both the connected routers is switched on).
Update_send(R, upd)	Router R sends a topology change to all its neighbours.
Update_receive(R, upd)	Router R receives a topology change from a neighbour (the semaphore of the routing table is switched on).
Update_store(R, entry)	**Router R stores a routing table change (this allows to trace back events).**
SPT_execute(R, T)	Router R executes a Shortest Path Tree algorithm to determine the shortest paths table T toward all the destinations.
Forward(R, P, T)	Router R forwards a packet P according to its current routing table T.
Turn_flag_off(R)	The semaphore of the routing table in the router R is switched off.
...	...

Primitive Action Preconditions

- **Poss**(Link_up(L), s) ≡ link_status(L, off, s).
- **Poss**(Link_down(L), s) ≡ link_status(L, on, s).
- **Poss**(Update_send(R, upd), s) ≡ node_flag_status(R, on, s).
- **Poss**(SPT_execute(R, T), s) ≡ node_flag_status(R, on, s).
- **Poss**(Forward(R, PCK, T), s) ≡ Destination(PCK) = D
   Routing_entry(T,D,s) = I iface_status(I,on,s).
- **Poss**(Turn_flag_off(R), s) ≡ node_flag_status(R, on, s).

. . .

Successor State Axioms

- neighbour(N1,N2,**do**(a,s)) ≡ neighbour(N1,N2,s) connect(L,I1,I2)
   belongs(I1,N1) belongs(I2,N2) a ≠ Link_down(L).
- node_flag_status(N, on, **do**(a, s)) ≡ node_flag_status(N, on, s)
   a ≠ Turn_flag_off(N).

. . .

NEM goals

Finally, we can express goals of the logical engine such as:

```
:- path(node1, node2, X, s1).
yes X=[node1, node5, ..., node2]
```

or

```
:- path(node1, node2, X, s2).
no
```

Conclusions and ongoing work

The AN Management framework exploits network programmability to enable agent-based distributed strategies.

Main features are:

- extension of the traditional MIB objects into more powerful objects with user customizable code.
- agent programmability according to the Filter-Even-Action model

New management applications can be devised to take advantage of the distributed and autonomous agent control.

A Network Events Mining application is under development.