Federated Advanced Directory Architecture

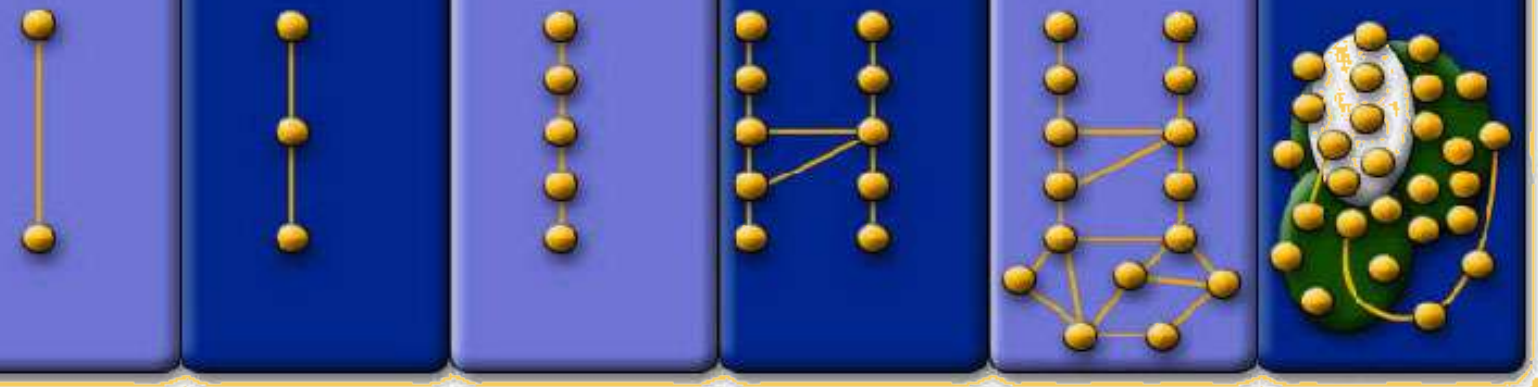
Miguel Vidal

Sun Microsystems, Inc

miguel.vidal@sun.com

Where we are?

Client-Server	3/N-Tier	Net Apps	Net Services	Next	After that
The Network Is the Computer	Objects	Legacy to the Web	The Computer is the Network	Network of Embedded Things	Network of Things
100s	1000s	1000000s	10000000s	100000000s	1000000000s
1984/1987	1990/1993	1996/1999	2001/2003	1998/2004	2004/2007
X	X	+HTTP (+JVM)	+XML, Portal	+RMI	Unknown
NIS, NIS+	+ CDS	+ LDAP (*)	+UDDI	+ Jini	+ ?
RPC, XDR	+CORBA	+CORBA, RMI	+ SOAP, XML	+ RMI/Jini	+ ?



*All things must change to something new, something strange (*H. Wadsworth Longfellow)

Fetish Network

The FETISH Network™ integrates information systems, applications and services from disparate tourism providers into a distributed, secure and scalable platform

It provides, through the use of Jini™ network technology, an adaptive and self configuring environment, delivering an integrated and distributed application framework over the Internet

From B2B model to S2S (Service to Service)

Fetish Challenges

- Develop a platform infrastructure to accommodate constantly changing environments
- Supporting spontaneous network of tourism-oriented Value added services.
- Provide basic open source software components to build value added services and join process through automatic composition and configuration
- Opportunity to offer of “high-level” complex services that are composed of basic tourism-related services (as Lego blocks) provided by different organizations

Fetish Components

- FETISH Repository
  - Service Definitions and Service Implementations (WSDL)
  - Business Data Types
  - Vendors, Contracts
- Service Modeler
- FADA Infrastructure
  - Cloud of Nodes (Virtual lookup server) and Proxies
- Ontology System
  - www.symontos.org
- Business Process Modeler and Executor
  - Proman

FADA framework goals

- Enable the system to scale
  - Over space
  - Over number of services
  - Over time
- Build reliable systems
  - On the network
  - From unreliable components
- Enable ad-hoc federations
  - No central point of control (or failure)

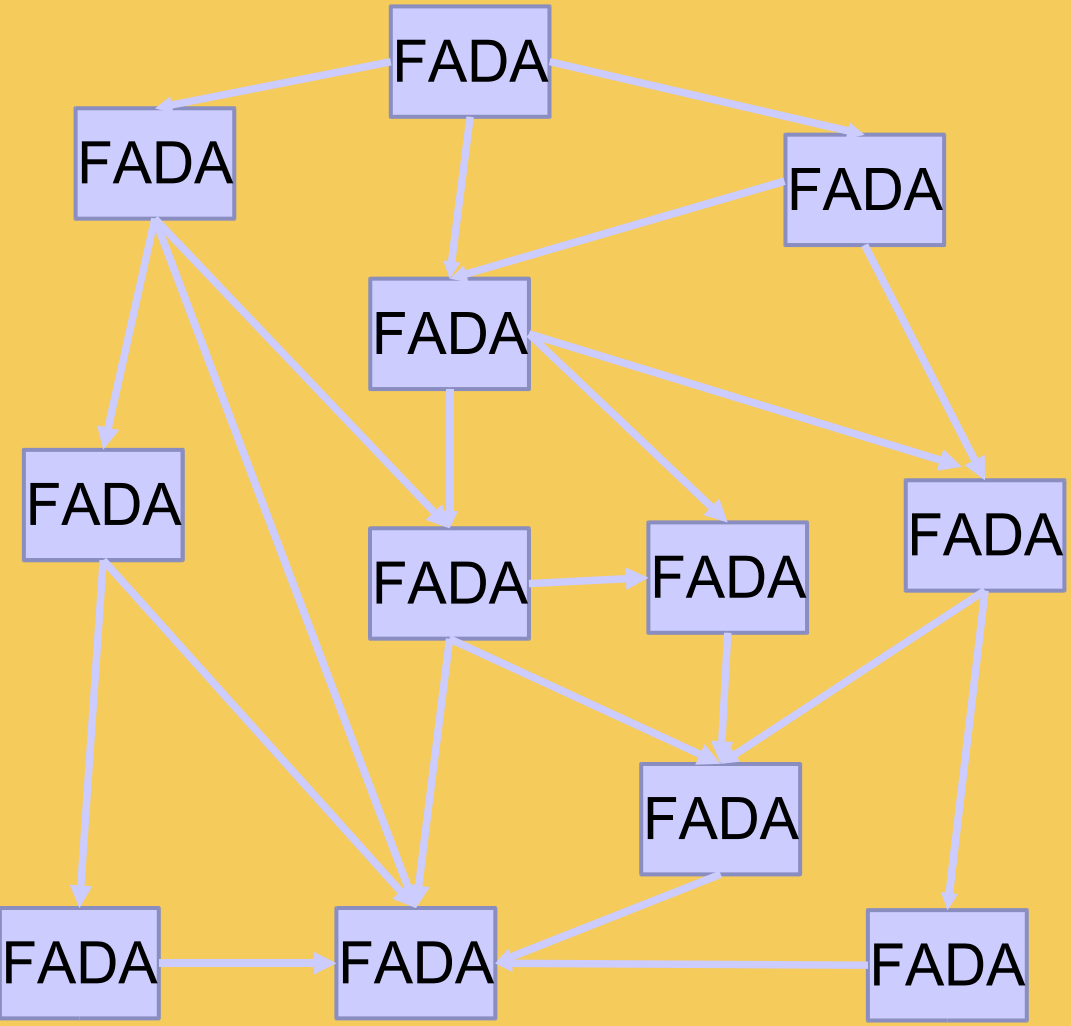
Jini/Reggie constraints

- Requires a broadcast medium
- Reggie implementation, uses RMI as the basic communication protocol (The firewall problem)
- It doesn't take into account variable net latency for lease renewal
- Discovery of Lookup services is based on broadcasting certain message types. This can NOT be achieved on the Internet

FADA nodes architecture

- An arbitrary graph of nodes is allowed, because:
  - As no central authority exists, everyone can manage the connections, and create cycles.
  - No single point of failure exists, except for the DNS server of domain fetishproject.com, which is replicated as-is.
- Broadcasting on an arbitrary (and potentially cyclic) graph must be achieved.
  - Several algorithms have been tested, but all of them require knowledge of the structure or a fast way to gain knowledge of the inmediate neighborhood.

FADA topology

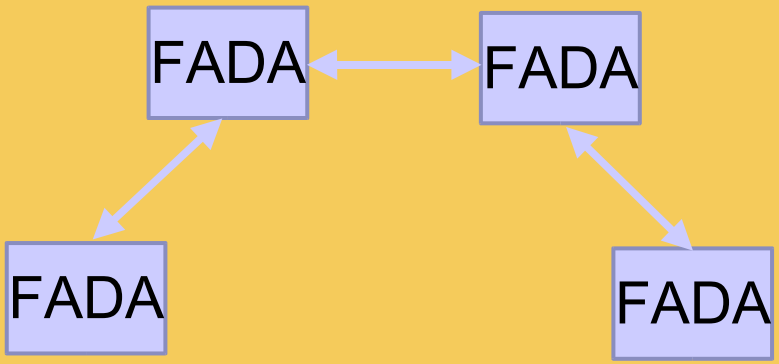


- Mesh instead of a tree
- Lookup using flooding algorithm
- Self healing
- Ad-hoc topology

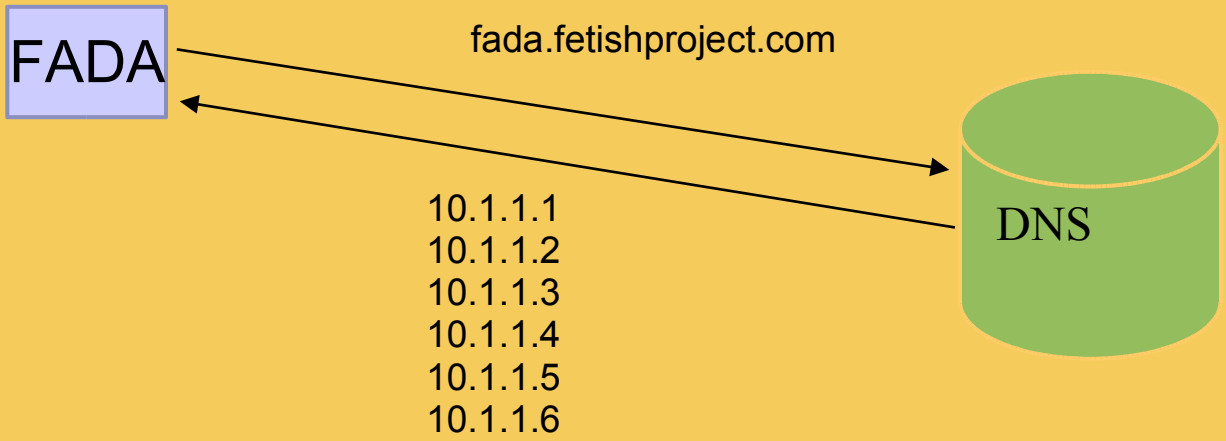
FADA lookup

- Solution: the good, old flooding algorithm.
- Main goal is not fast responsiveness, nor optimal paths, but AVAILABILITY.
- The flooding algorithm WILL find a path between two nodes if such path exists.
- It is reliable in front of connection failures: if two nodes are no longer connected directly, but a route through another one or more nodes exits, the flooding algorithm will find it.

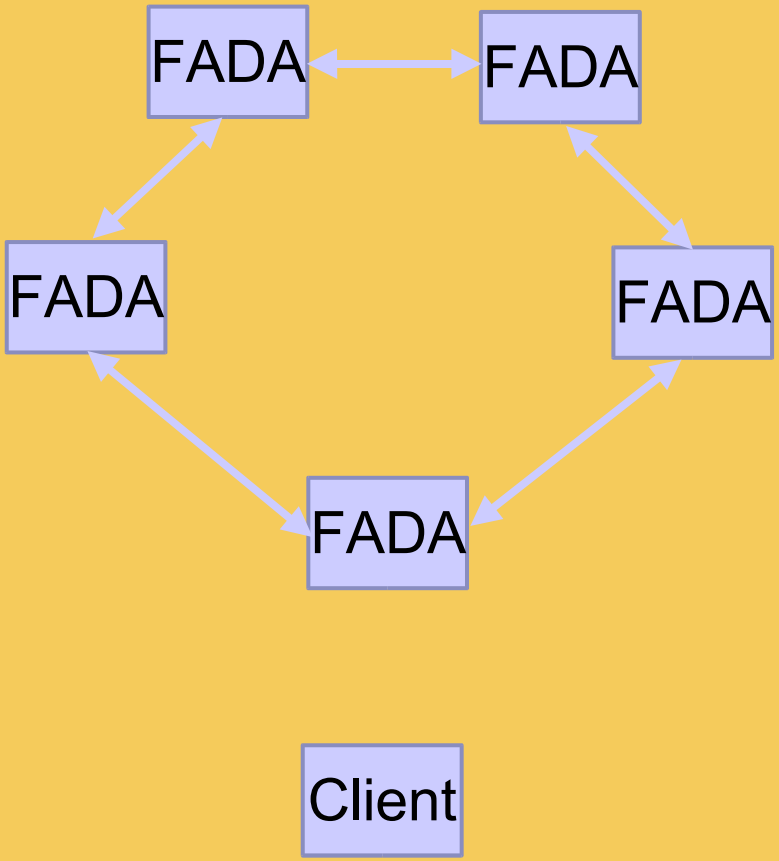
FADA Discovery



New FADA node requests a FADA node IP to its DNS in order to join the federation.

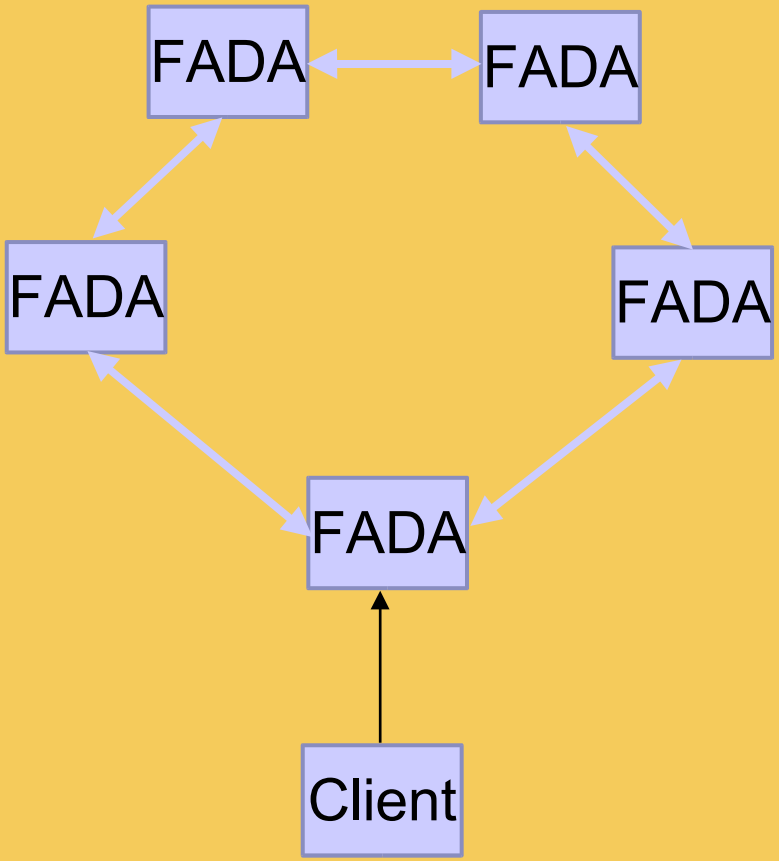


FADA Lookup example

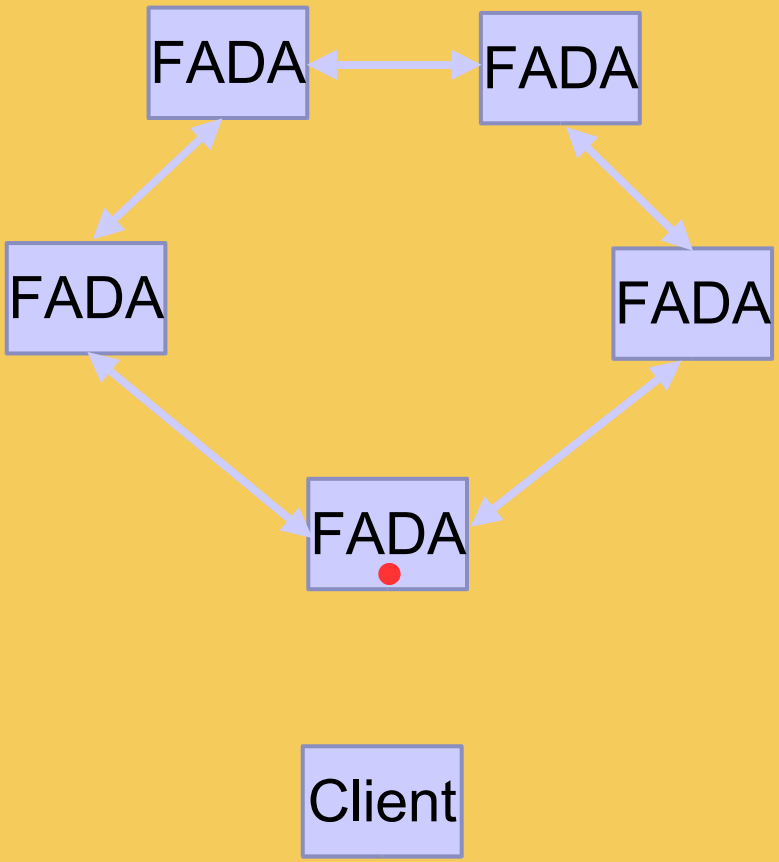


FADA Lookup example

Client performs lookup

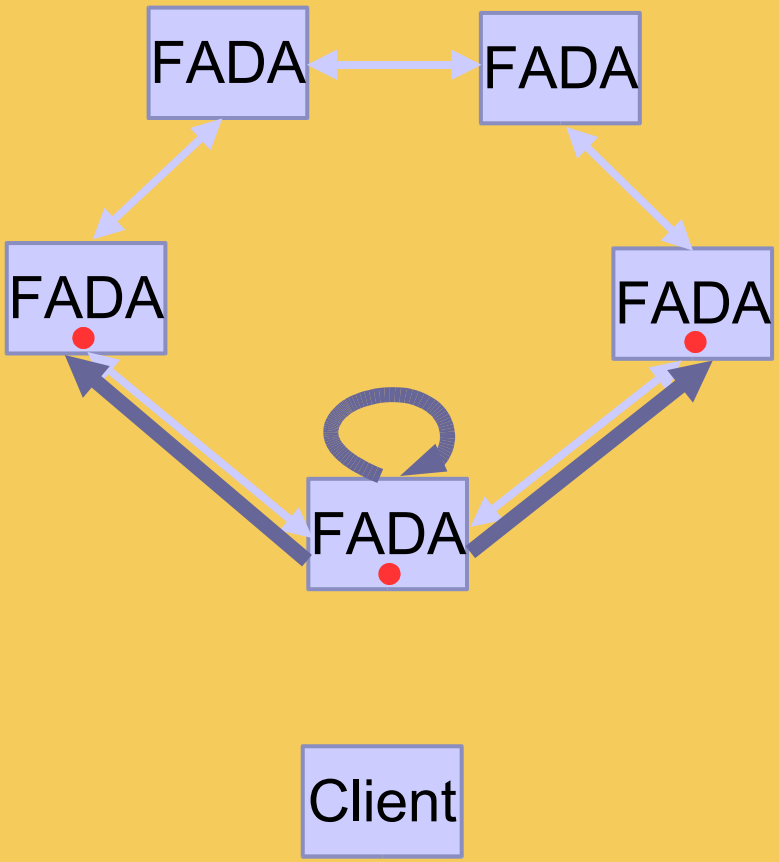


FADA Lookup example



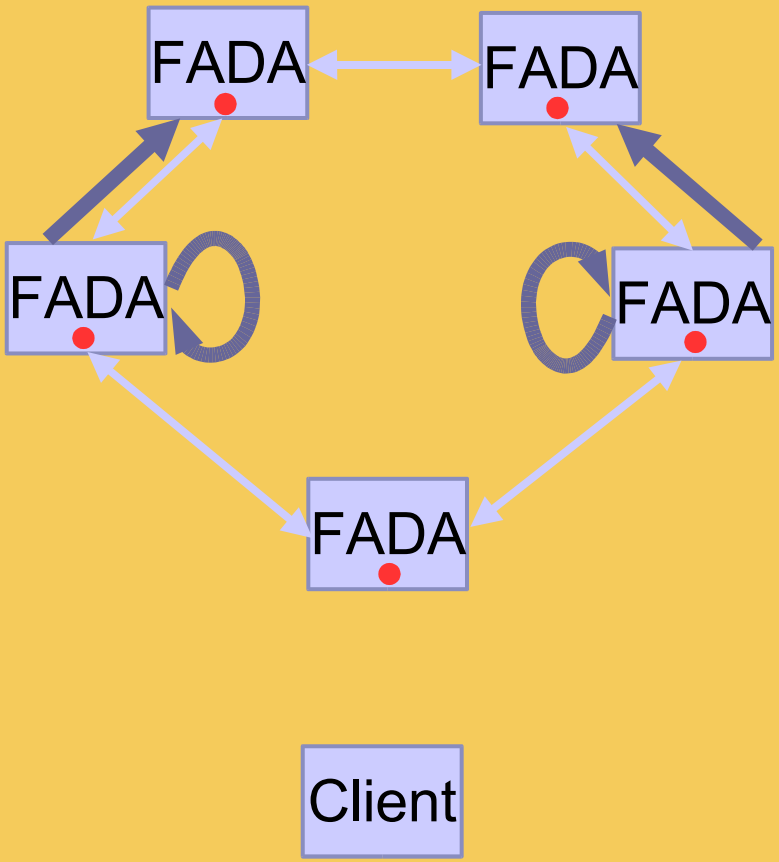
New globally unique SearchID is created

FADA Lookup example



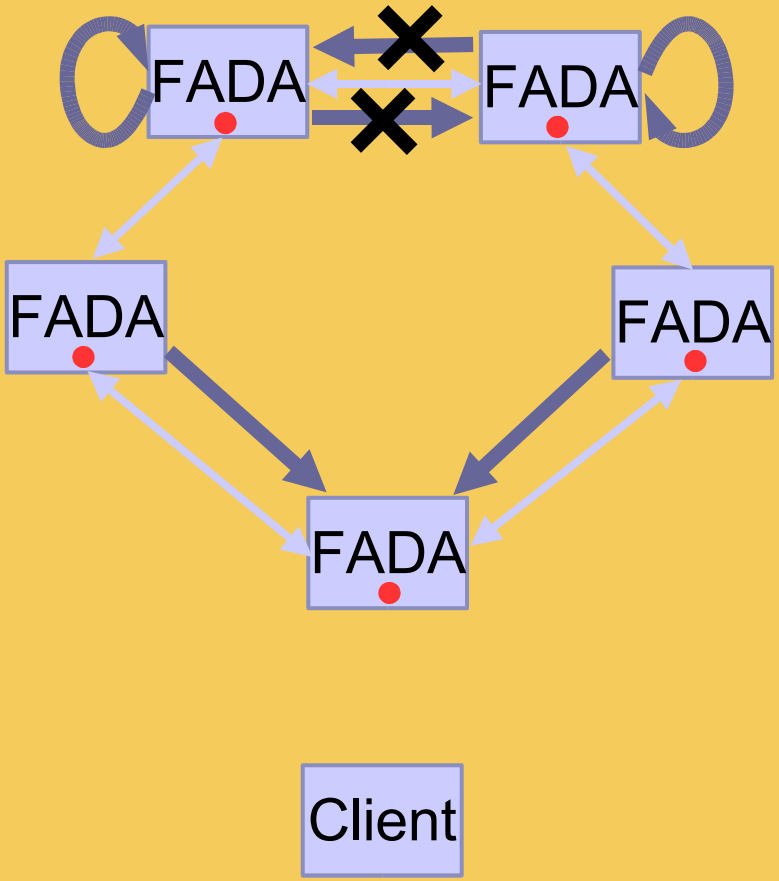
Local lookup is performed. Meanwhile, search is expanded.

FADA Lookup example



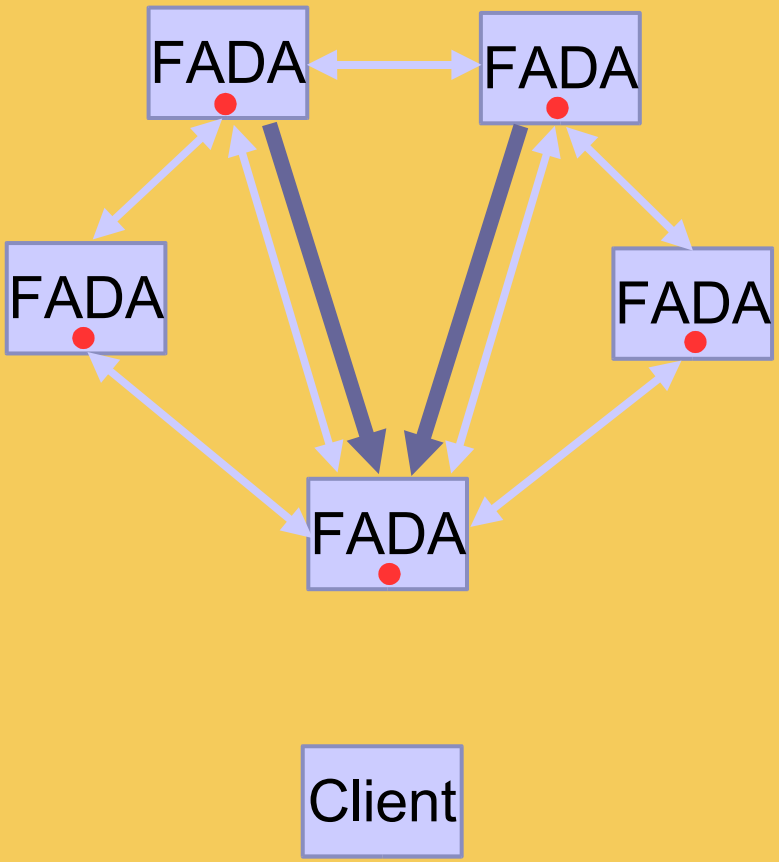
The above step is repeated by neighbors. Meanwhile, the first FADA node is waiting for expiration of a timeout

FADA Lookup example



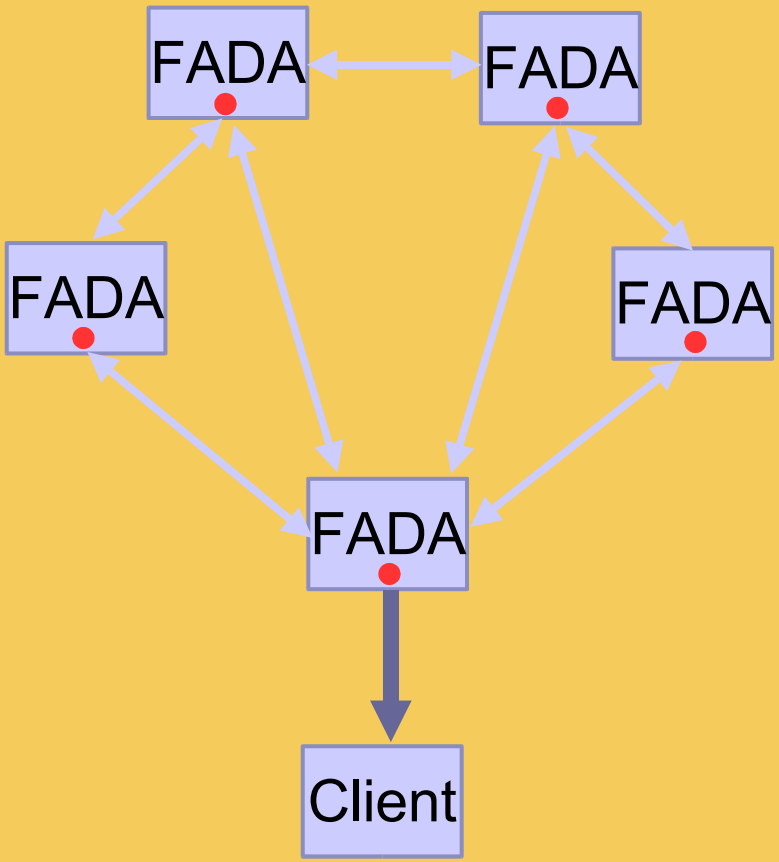
The direct neighbors notify their results to the first FADA node. Meanwhile, the neighbors in the second degree start local lookup and attempt search expansion. Both of them fail to accomplish the latter, because the node they attempt to contact already has the SearchID.

FADA Lookup example



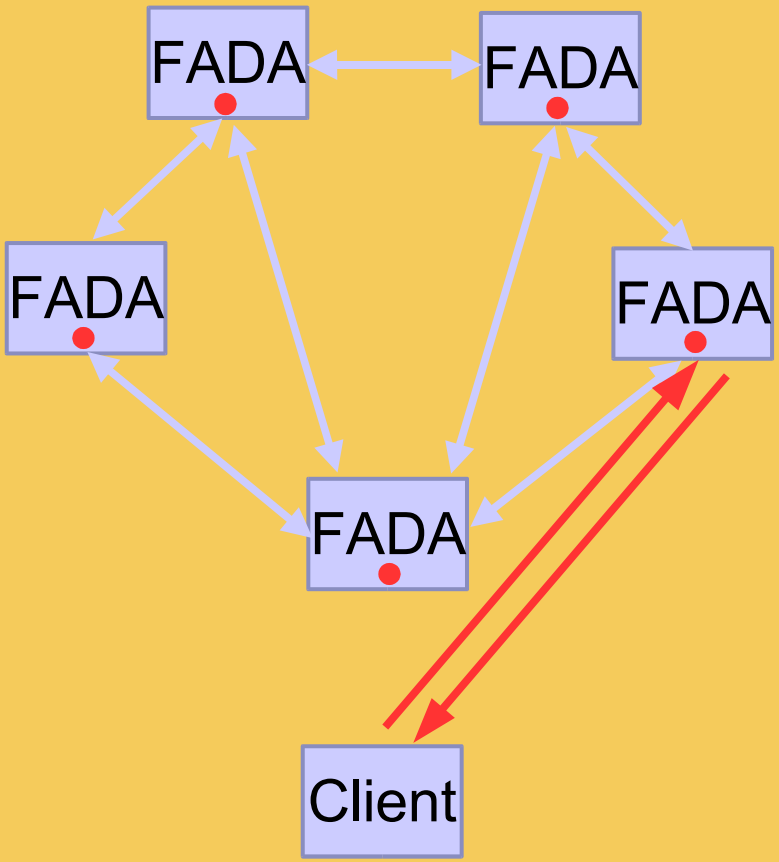
The latter neighbors notify their results to the first FADA node.
As they weren't previously connected to the first FADA node, they connect inmediately.

FADA Lookup example



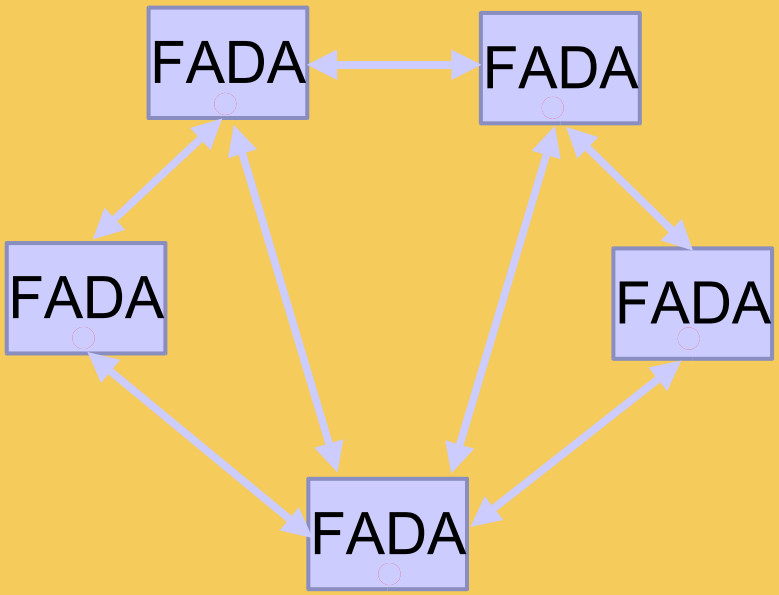
After the search timeout occurs, the first FADA node returns the total lookup results to the client.

FADA Lookup example



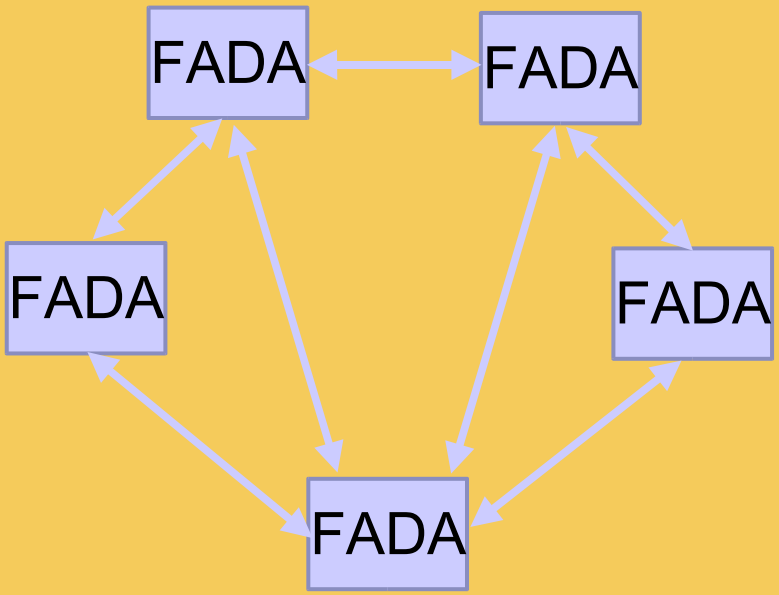
The service proxy is retrieved directly by the client, based on the Service locator delivered by the FADA node.

FADA Lookup example



The SearchIDs cached in every node will be deleted after 30 sec.

FADA Lookup example



Resulting state of the architecture: more redundacy, improved availability.

Reggie constraints

- Requires a broadcast medium
- Uses RMI as the basic communication protocol
- It doesn't take into account variable net latency for lease renewal

RMI

- Due to the fact that Reggie uses RMI as a communication protocol, it opens sockets on arbitrary ports. Firewalls don't allow this behaviour, and clients behind them are not allowed to open a connection to an arbitrary outside port.

PROTOCOL vs IMPLEMENTATION

- Instead of a particular implementation, the FADA architecture forces messages to be exchanged amongst nodes and clients.
- The use of XML documents allows clients residing behind firewalls to work with the FADA network without requiring the system administrator to open new types of outside connections.
- The already existing setup of proxies and open ports for http communication is used by the FADA to exchange XML documents.

XML-RPC

- We get rid of the RMI constraint by switching to another communication protocol: XML-RPC over HTTP

XML-RPC over HTTP

- Every FADA node opens a socket on a fixed port (chosen at startup)
- It receives HTTP POST requests from other FADA nodes or clients
- These POST requests contain XML documents that follow the XML-RPC standard

Benefits of XML-RPC / HTTP

- Clients behind firewalls can safely use the http proxies they may have available
- FADA nodes can be behind a firewall: the network administrator need only open a single configurable port

Benefits of XML-RPC / HTTP

- Paranoid network administrators can trace the traffic exchanges: it is only HTTP messages.
- FADA nodes can cooperate with other implementations: only the protocol is important

Reggie constraints

- Requires a broadcast medium
- Uses RMI as the basic communication protocol
- It doesn't take into account variable net latency for lease renewal

Lease renewal on the Internet

- Delays on the internet are much larger than those on a LAN.
- Delays on the internet vary much more than those on a LAN.
- A method is needed to estimate the probable value of the delay for a given instant of time.
- Delay depends on many unknown factors, beyond the observer's reach.
- Lease renewal times depend on the delay between the renewal request and the renewal operation.

Lease renewal on the Internet

-

- Hypothesis:
  It is possible to do a network latency forecast

- Solution: Kalman filter

Internet Delay Experiments

RFC-889

- The Internet delay distribution is more-or-less Poissonly distributed about arelatively narrow range with some important exceptions.
- The exceptions are characterized by occasional spasms where one or more packets can be delayed many times the typical value.
- The specified retransmission-timeout algorithm, is characterized by two parameters, a weighting factor F and a threshold factor G. For each measured delay sample R the delay estimator E is updated:

$$E = F*E + (1 - F)*R$$

- In the current TCP specification, if an interval equal to G*E expires after transmitting a packet, the packet is retransmitted, and suggests values in the range 0.8 to 0.9 for F and 1.5 to 2.0 for G. These values have been believed reasonable up to now over ARPANET and SATNET paths.

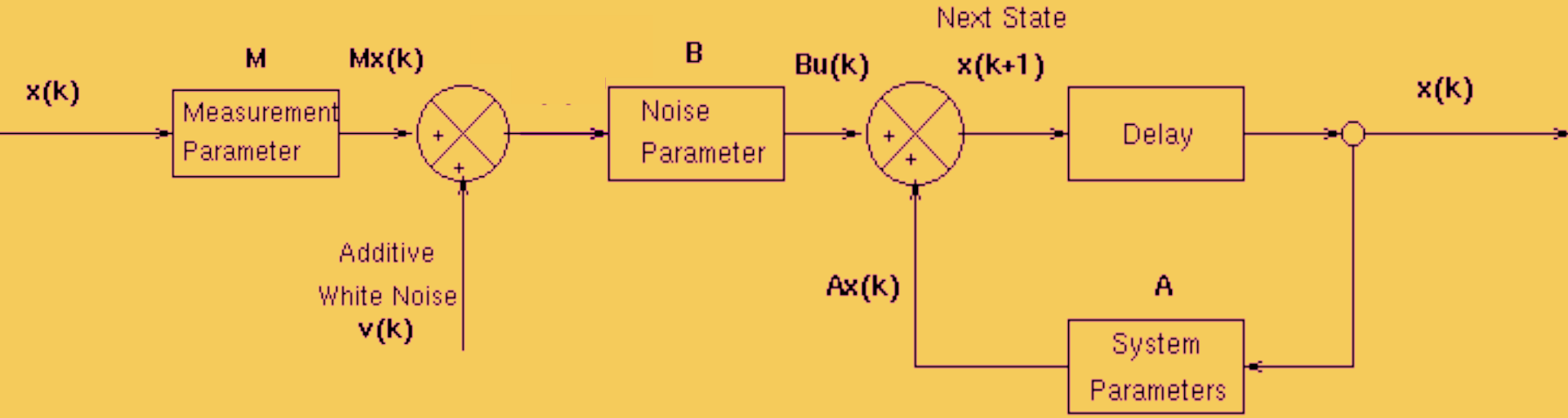
Kalman Filter (1960)

- In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem.
- How do you get accurate information out of inaccurate data? More pressingly, How do you update a "best" estimate for the state of a system as new, but still inaccurate, data pour in?
- The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.
- Kalman filtering has proved useful in navigational and guidance systems, radar tracking, sonar ranging, and satellite orbit determination.

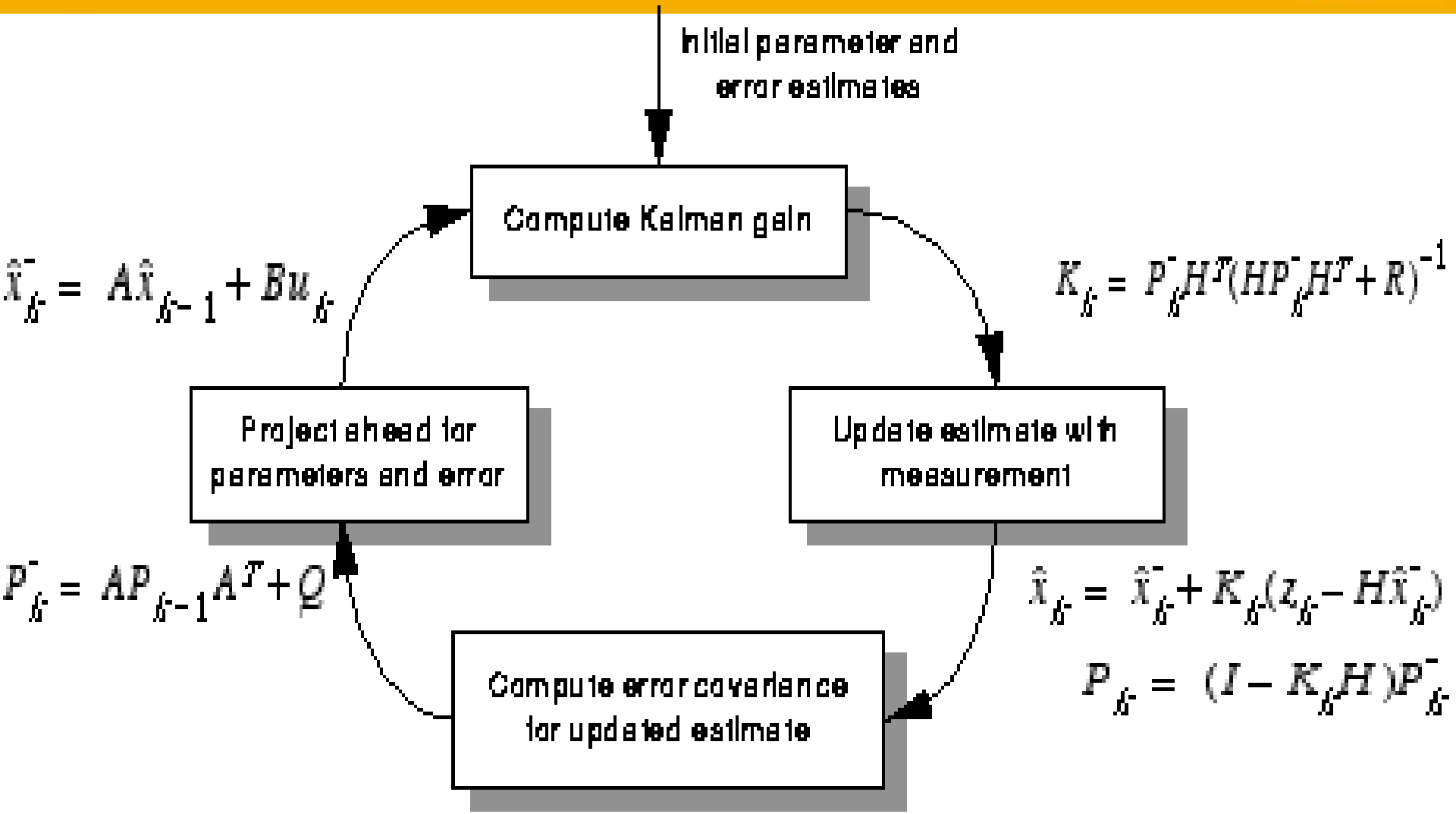
Why Kalman filter?

- The Kalman filter provides a set of equations that give the BEST estimate for the quantity under study.
- It can keep a history of past events without the need to store all previous values.
- It can cope with noisy measurement procedures.
- It can project the state in the future, and give the BEST estimate for a FUTURE value, plus the BEST estimate for its VARIANCE.
- In that way, it is possible to get the WORST value for the delay with a high probability of success.

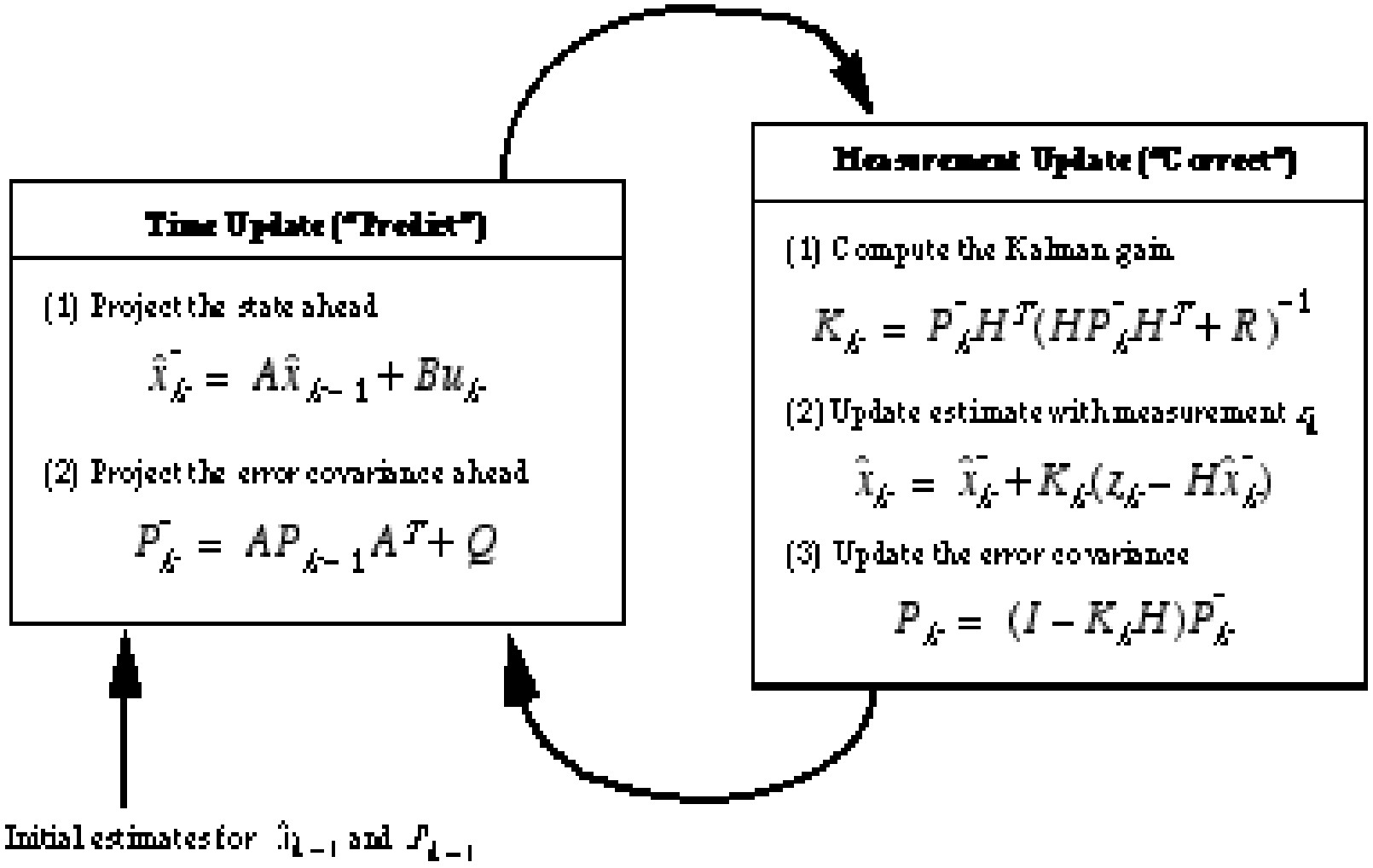
Kalman Filter Block Diagram



Kalman Filter States



Kalman Filter States



The Kalman assumptions

- The linear Kalman filter supposes the parameter to be estimated follows a linear function of its previous state plus an optional control signal affected by a white gaussian noise component.

- Non-linear Kalman filters still suppose a definite model. For instance, the movement of a car on a track, the movement of currents of hot/cold air, the movement of a rocket.

- We have no such model: delay may vary arbitrarily, grow or shrink, by a huge or a small amount

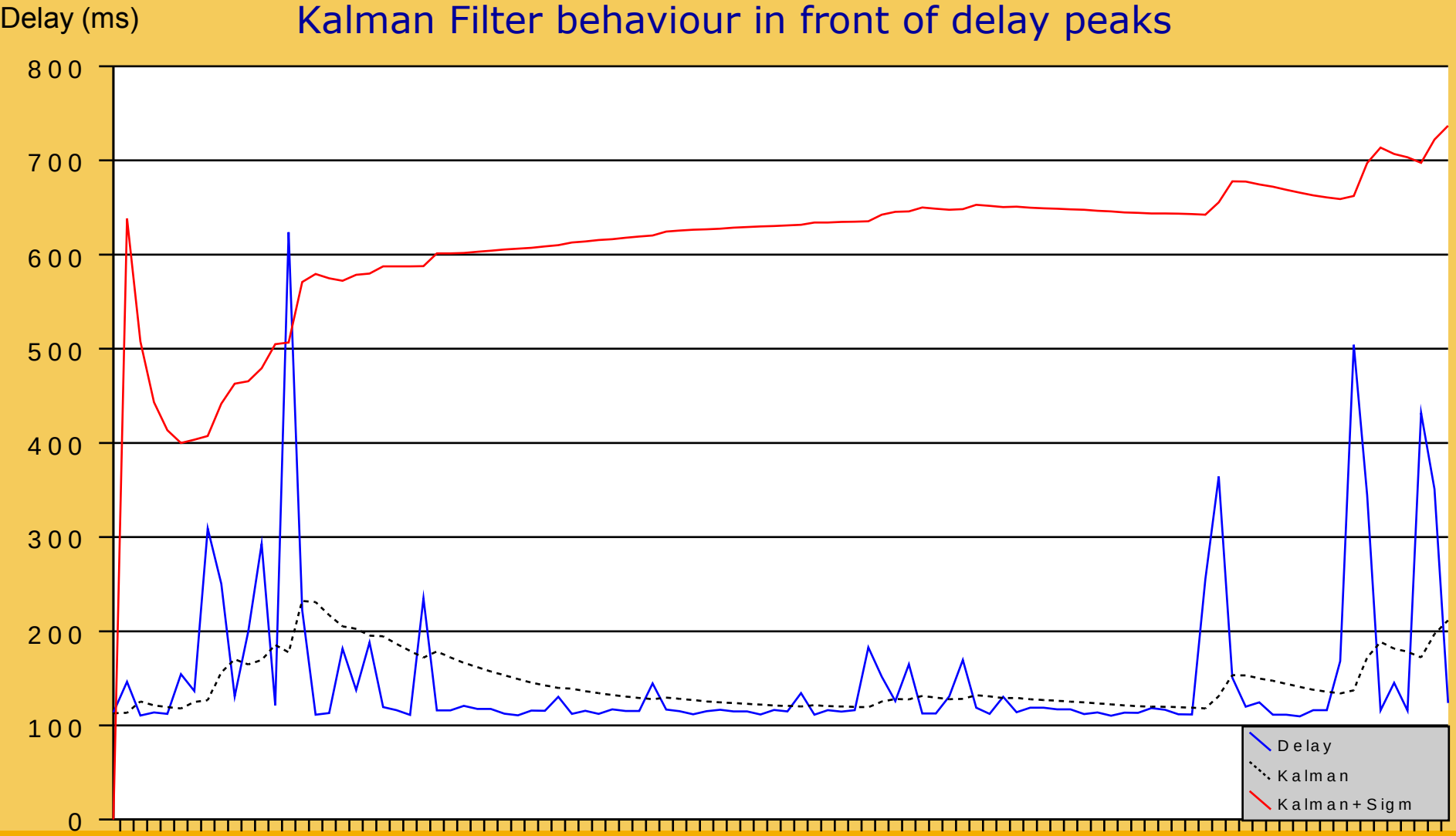
Our Kalman model

- Our approach is to consider the latency as a constant value affected by white gaussian noise.

- This approach will give us the best estimate for a parameter which is taken as a constant, however
  we know for sure it's not a constant.

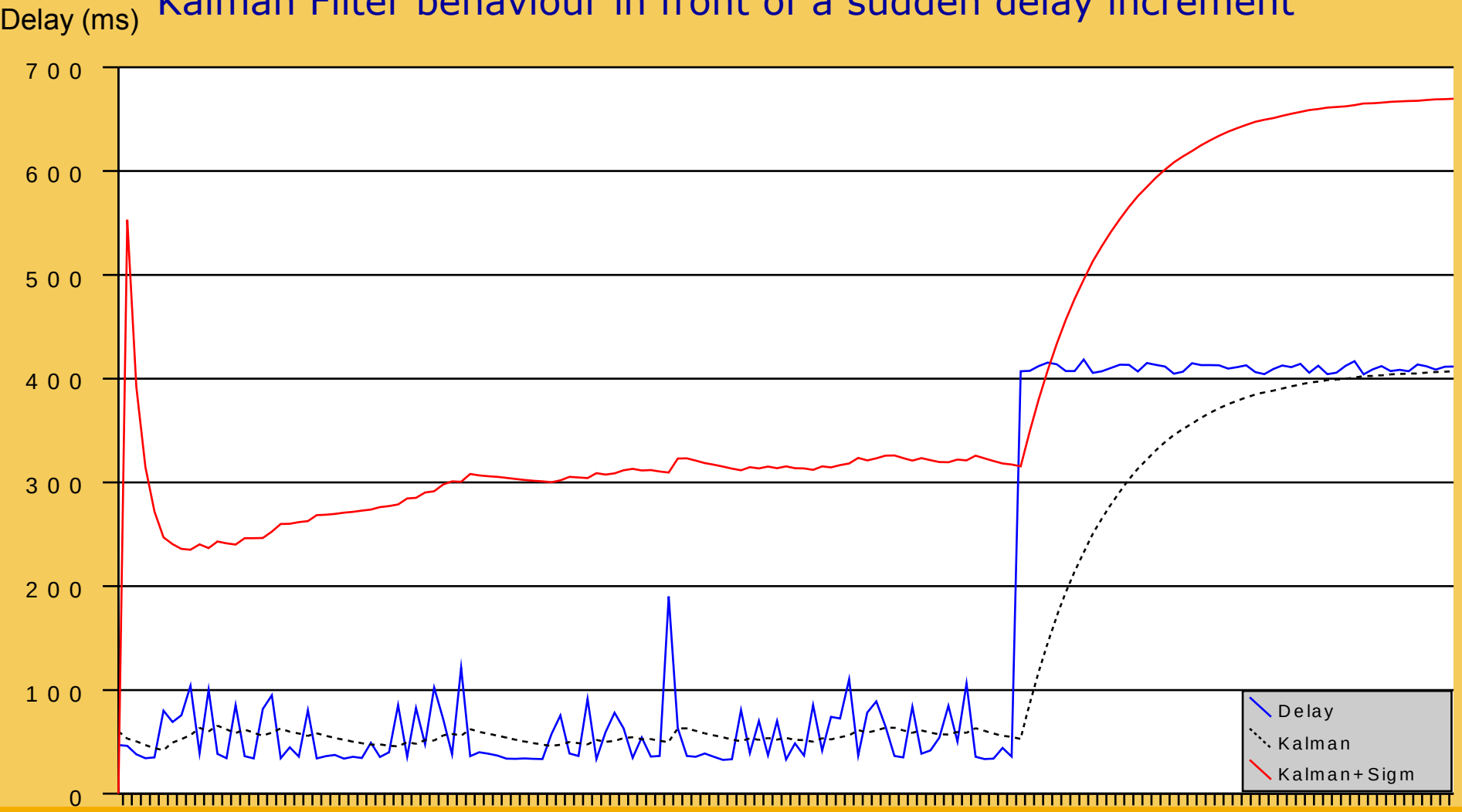
- We are interested in the variations of this parameter, but the Kalman filter will strip out these variations and give us the best estimate for an hypothetical central value.

FETISH Kalman Filter



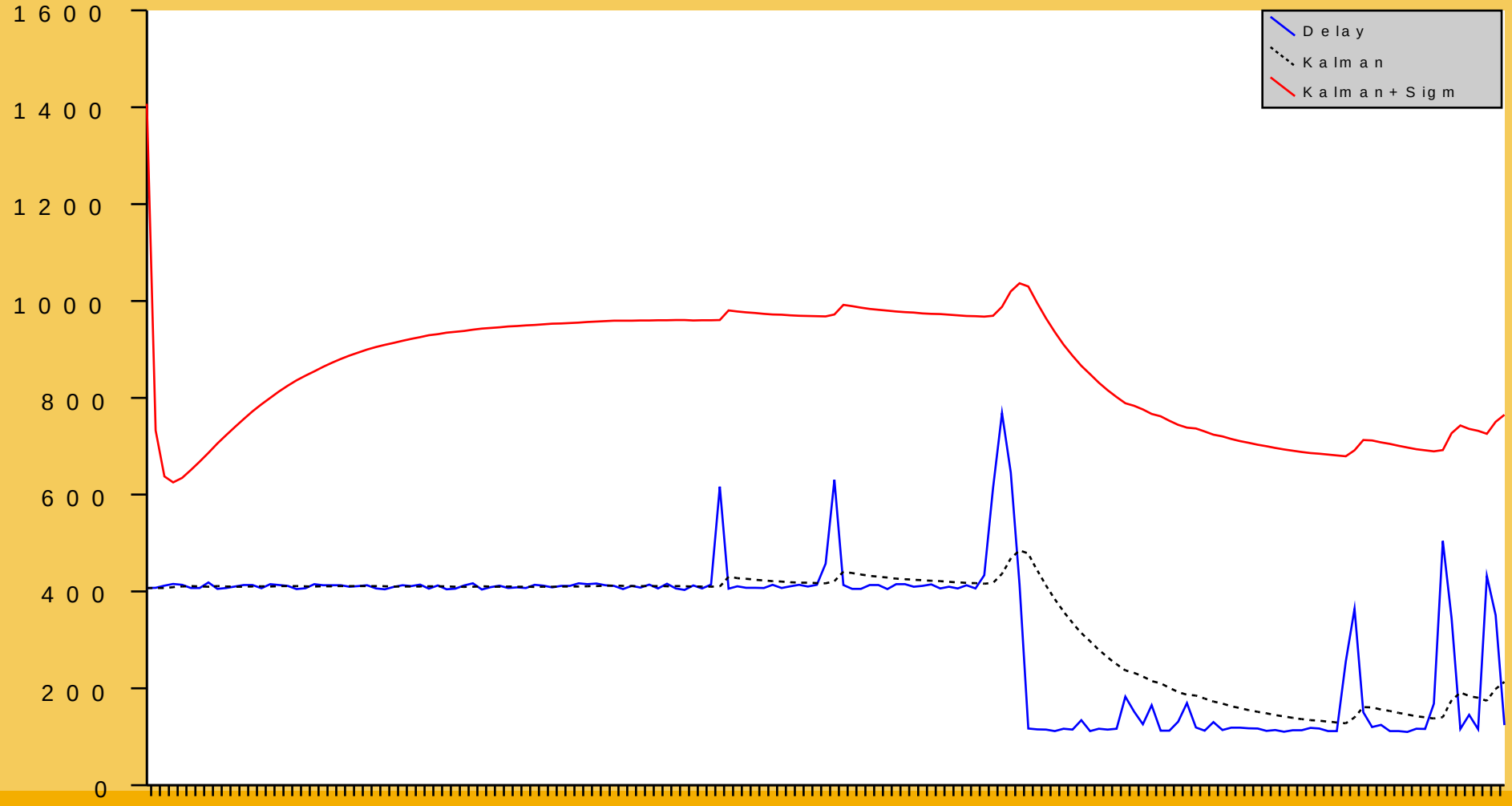
FETISH Kalman Filter

Kalman Filter behaviour in front of a sudden delay increment



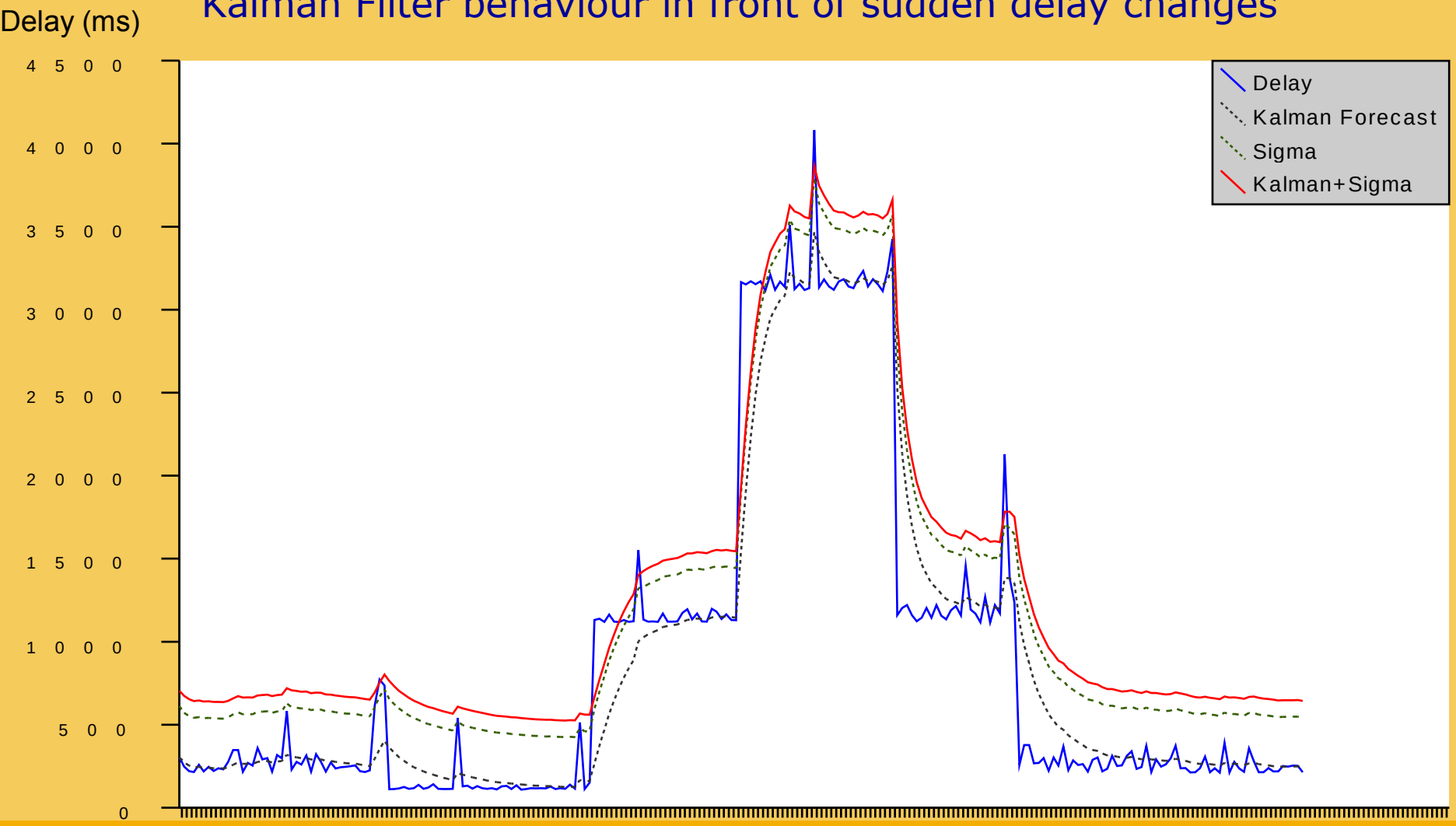
FETISH Kalman Filter

Kalman Filter behaviour in front of a sudden delay decrement



FETISH Kalman Filter

Kalman Filter behaviour in front of sudden delay changes



Kalman Filter: conclusion

- Network latency forecast >95% accuracy
- Get accurate information out of inaccurate data
- It supports estimations of future states, not knowing the nature of the modeled system

Use of the FADA toolkit: register

- The register method requires the existance of an instance of the class FadaLeaseRenewer, passed onto the constructor of FadaHelper. Then it's possible to use the FadaHelper instance's register method.
- Parameters:
  - The URL of the FADA node to register the item.
  - The item itself
  - The array of entries that further define the item.
  - The lease period, that is, the lease that will be asked by the FadaLeaseRenewer to extend the lease lifetime.
- Returns: the FadaServiceID of the registered item

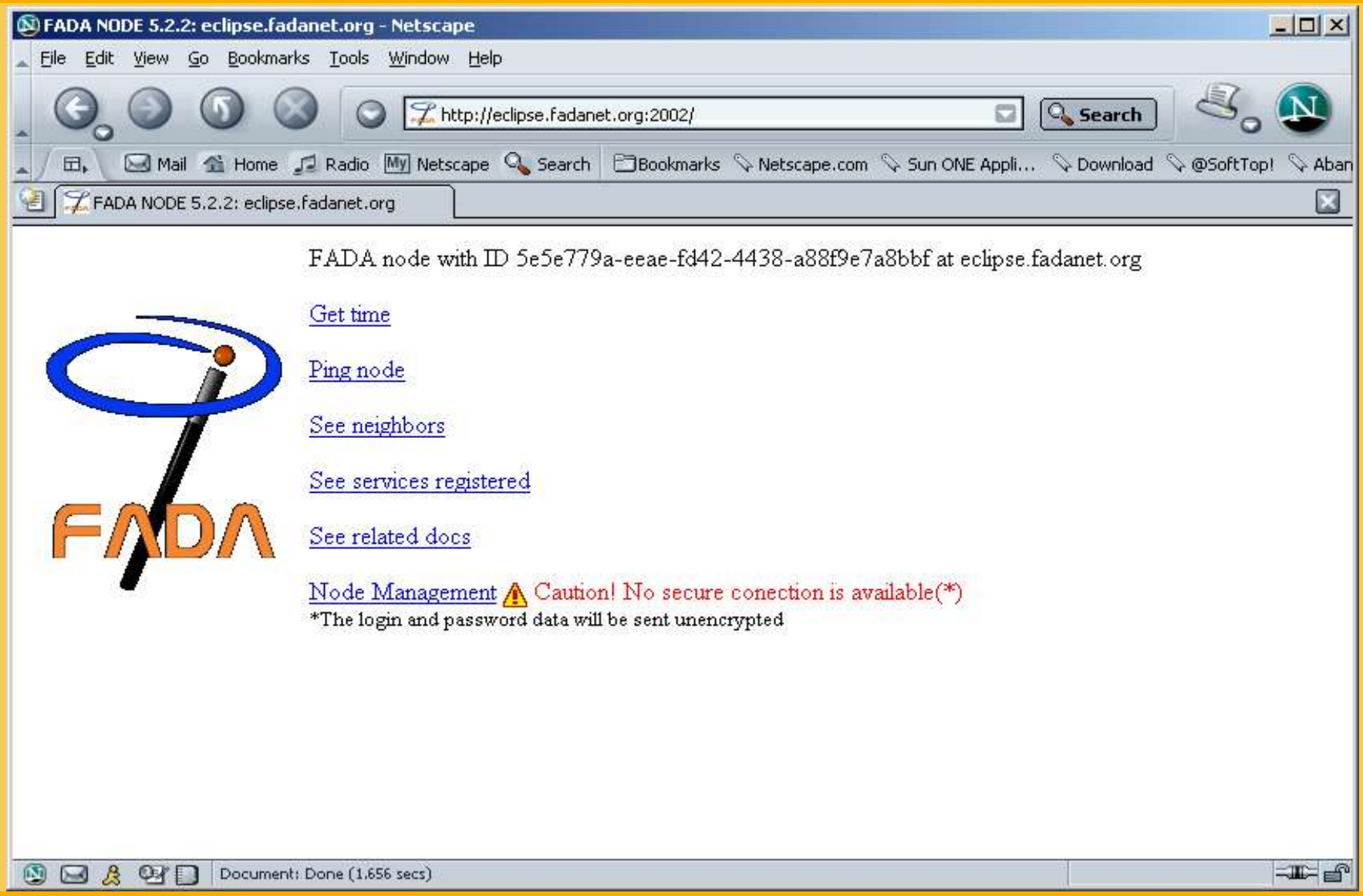
Use of the FADA toolkit: lookup

- To search for an item in the FADA using the FADA toolkit, you just need to use the method lookup() from the class com.fetish.toolkit.FadaHelper.
- Parameters:
  - The URL of the FADA node where the lookup must be started
  - The FadaServiceID of the item that's wanted (may be null, meaning that any FadaServiceID will match)
  - The array of entries that further define the item
  - The array of interfaces the item must implement.
- Returns: an Object, matching the lookup query criteria.

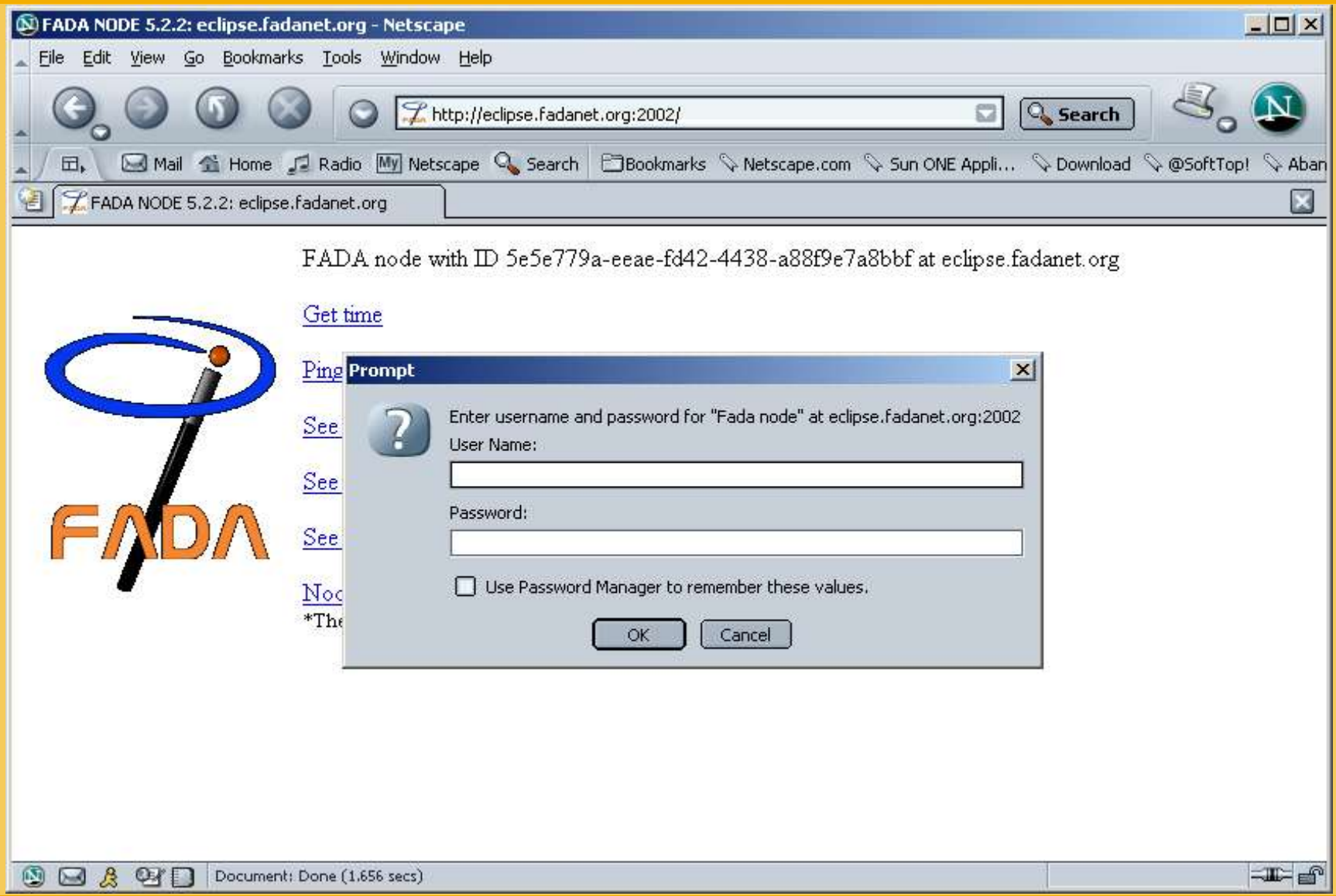
Use of the FADA toolkit: lookup (cont)

- Alternatively, you can use the lookupMany method.
- It returns an array of matching Objects, up to a maximum number of matches.
- Parameters: the same as in the lookup method, plus an integer with the maximum number of matching items.

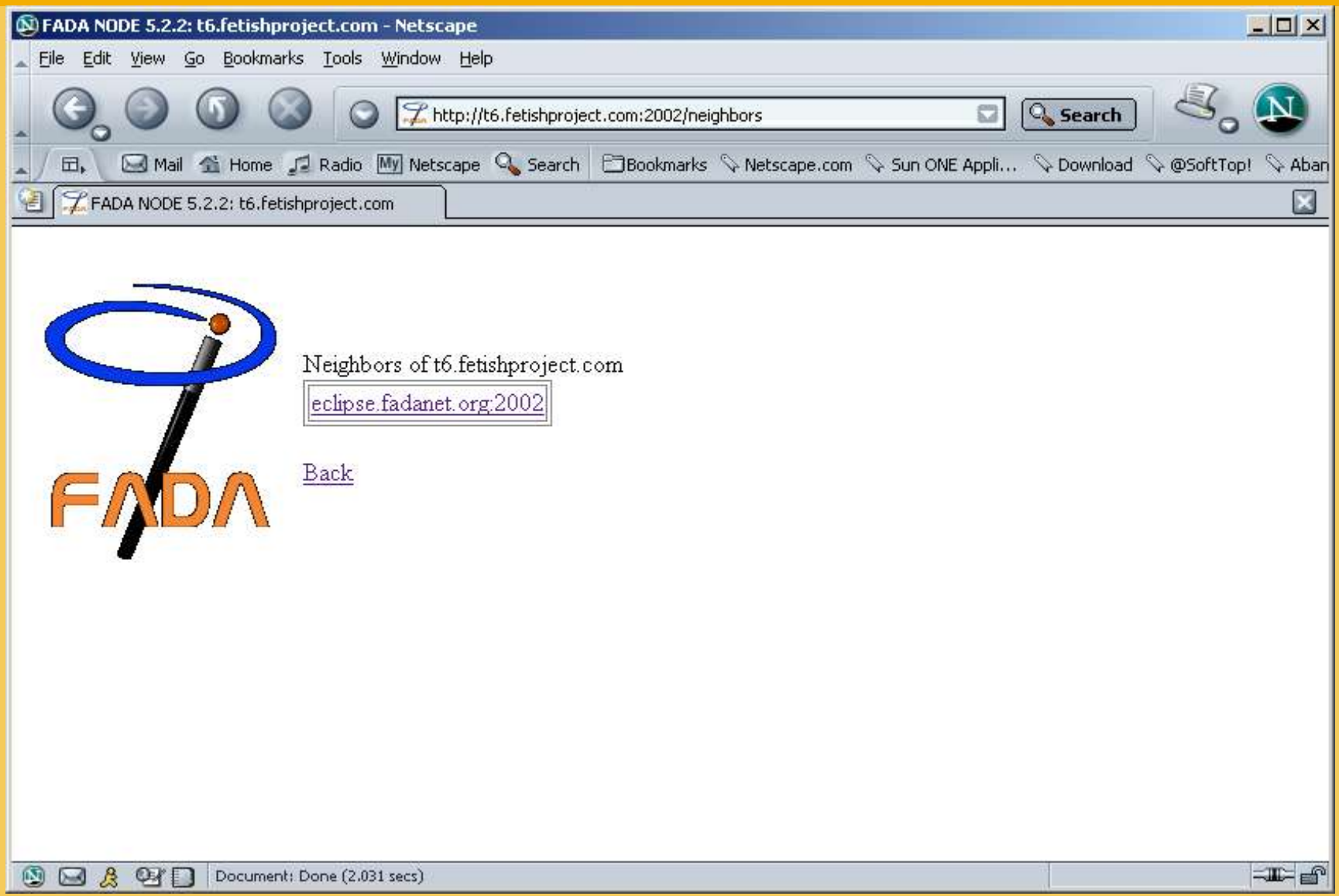
FADA Monitoring



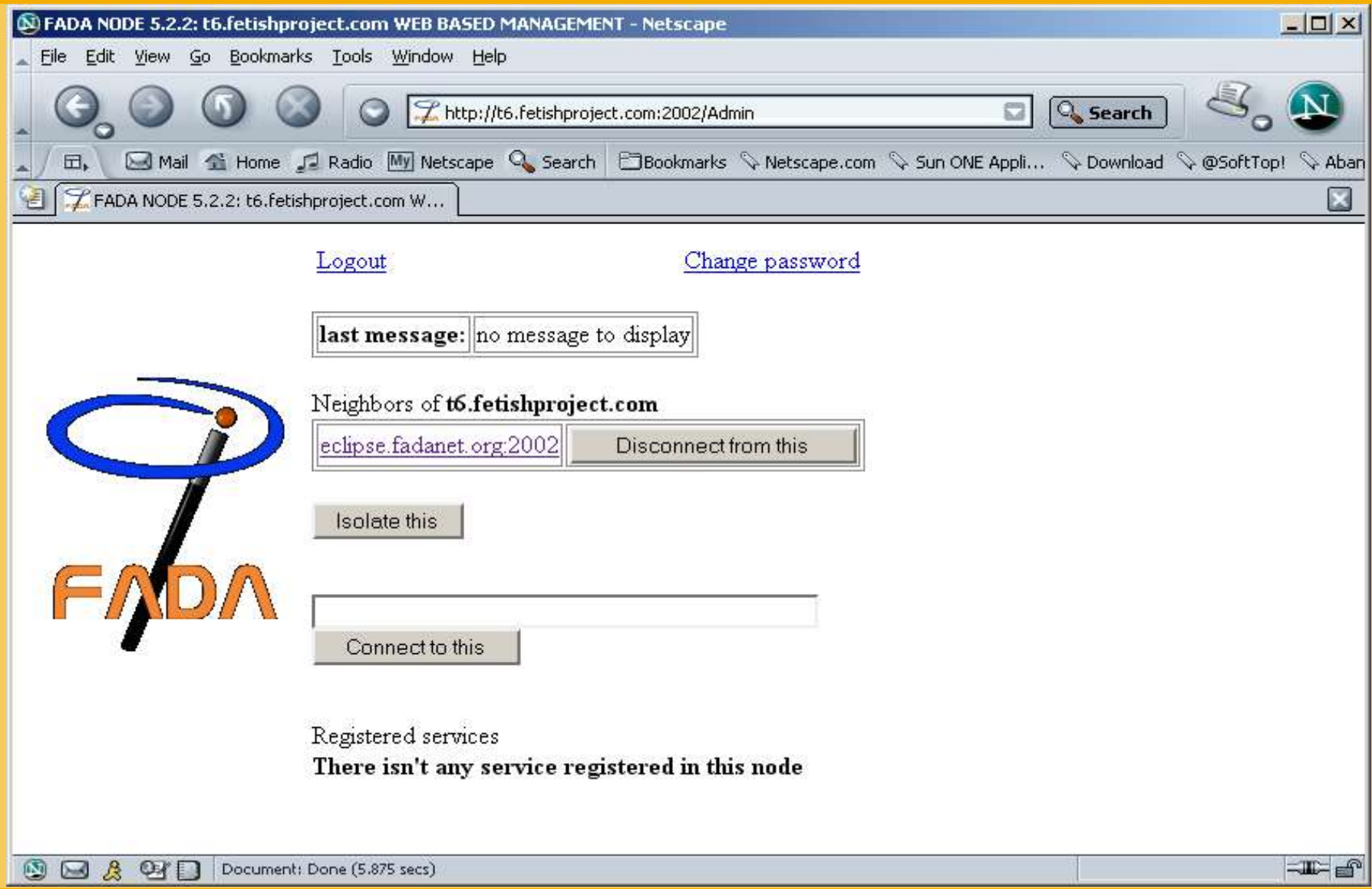
FADA Monitoring



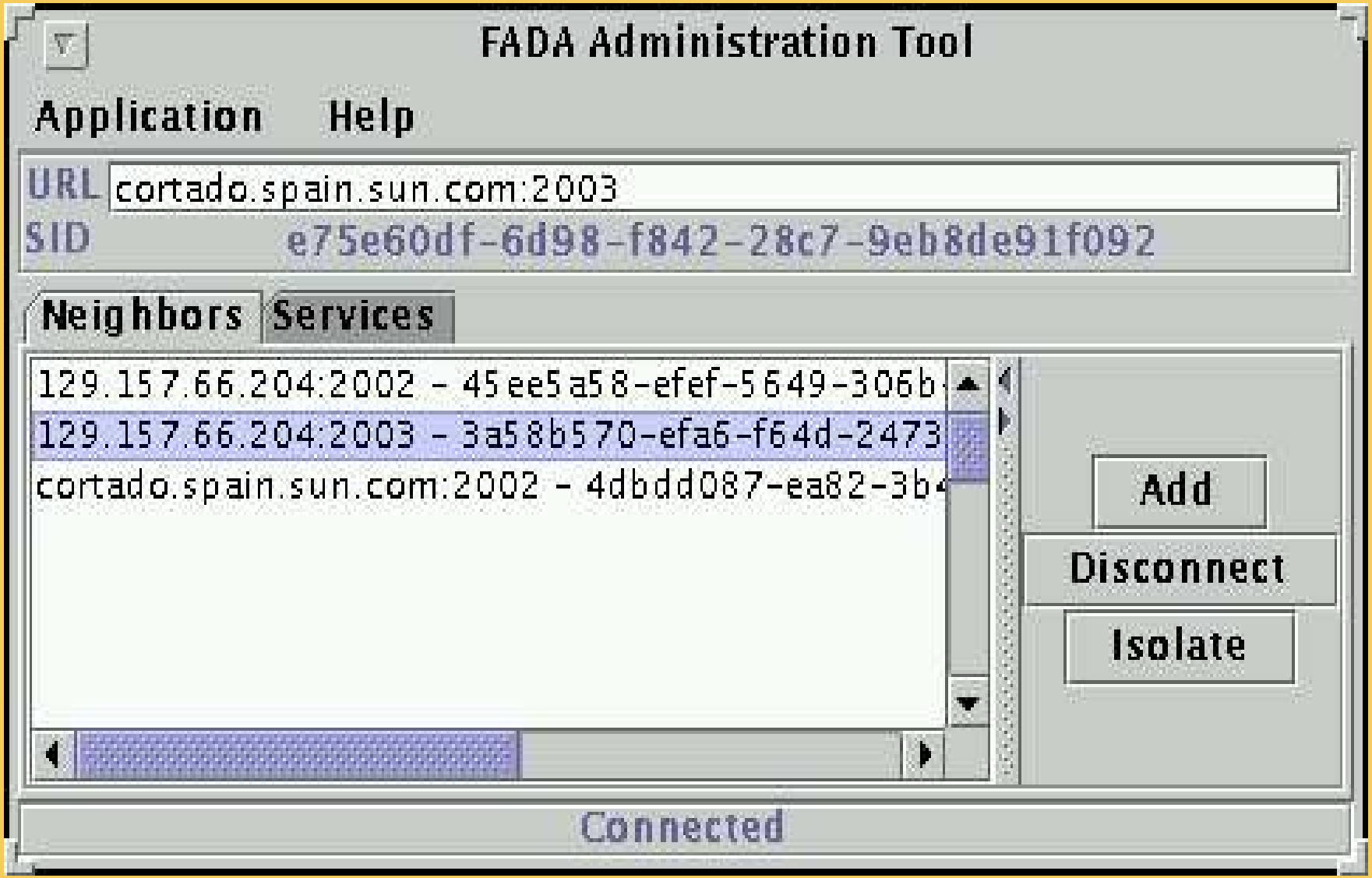
FADA Monitoring



FADA Administration

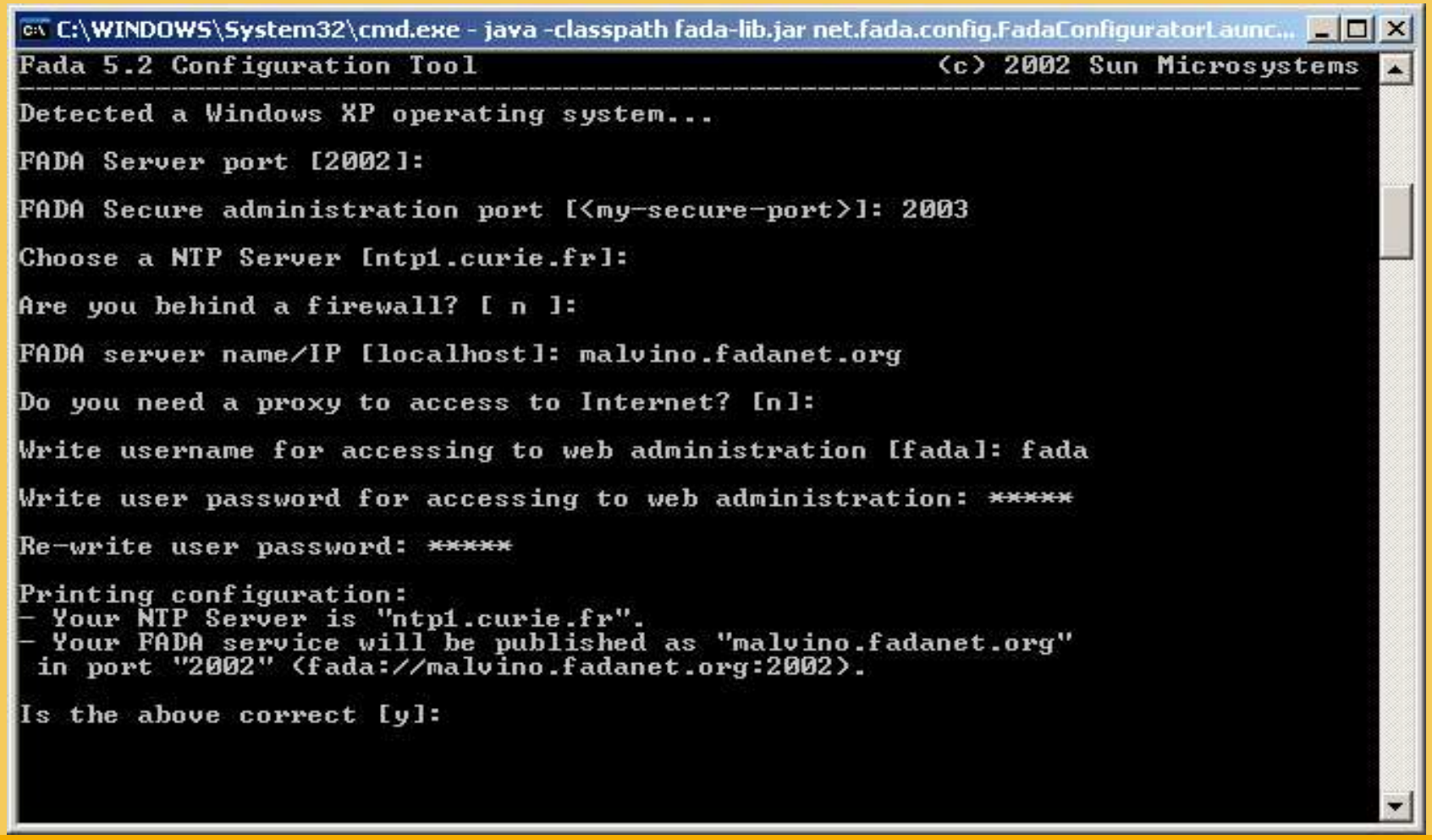


FADA Administration



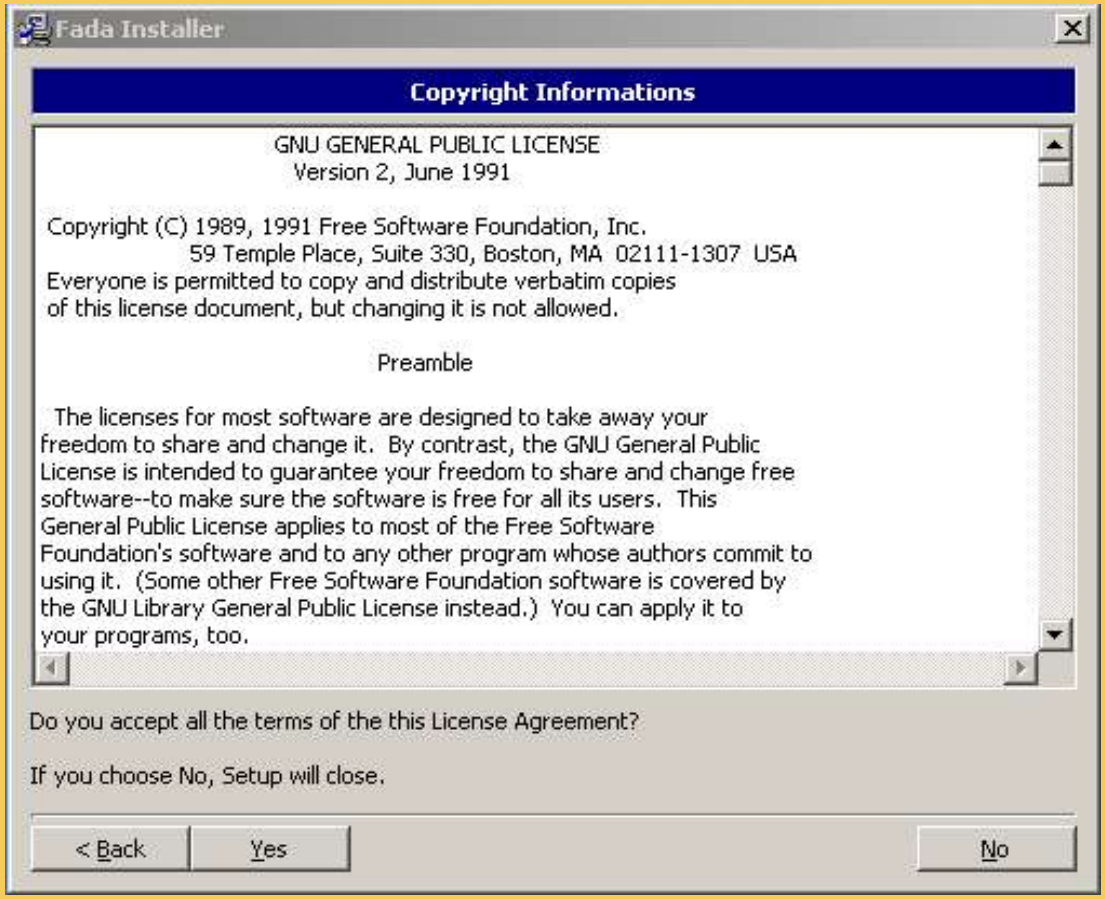
FADA Installation

- UNIX based installation process



FADA Installation

- Windows based installation process



FADA Releases

Version	Released
FADA 3.0	04/09/2001
FADA 3.1	04/10/2001
FADA 4.0	24/01/2002
FADA 4.1.0	12/03/2002
FADA 4.1.1	08/04/2002
FADA 4.1.2	28/05/2002
FADA 5.0	30/09/2002
FADA 5.0.3	04/11/2002
FADA 5.2	31/01/2003
FADA 5.2.1	03/03/2003

FADA Evolution

Footprint	1 Mbyte	280 Kbytes	500Kbytes
Speed Up	1 x	10 x	15x
Administration	Client/Server	+ Browser based	Browser based
Installation	Ad-hoc	Wizards (Unix/Win)	Wizards (Unix/Win)
Cross Firewalls	No	Yes	Yes

Coming soon...(FADA 5.0)

- Jini API convergence
- Administration autentification
- Jini proxy caching
- Hot pluggable transport protocol (RMI, JXTA, SOAP, XML-RPC...)
- SNMP protocol support
- MMAE (Multiple Model Adaptative Estimation) Kalman filter approach

Project Curriculum Vitae

- European IT Tourism Conference (Brussels 12/11/1999)
- IST Conference (Helsinki 24/11/1999)
- Jini Community Meeting (Amsterdam 10/12/2000)
- World Tourism (Lisbon 19/01/2001)
- Enter 2001 (Montreal 25/04/2001)
- Java One 2001 (San Francisco 07/06/2001)
- Intelligent Transport System World Congress (Torino 7/11/2001)
- European Commission Tourism Confecence (Brussels 9/7/2001)
- IST Conference (Dusseldorf 3-5/12/2001)
- Tourism exchange exibition BIT 2002 (Milan 22/2/2002)
- Java One 2002 (San Francisco 27/3/2002)
- 6th Jini Community Meeting (Boston 17-20/6/2002)