

# Knowledge Delivery Mechanism for Autonomic Overlay Network Management

Dominic Jones, John Keeney, David Lewis, Declan O'Sullivan

Knowledge & Data Engineering Group (KDEG), Trinity College Dublin (TCD), Ireland.

{ Dominic.Jones | John.Keeney | Dave.Lewis | Declan.OSullivan }@cs.tcd.ie

## ABSTRACT

The complexity of current computing networks and the required fluidity of their operation requires autonomy to be central to the design and implementation of a managed overlay network. This paper presents a system in which a Knowledge-based Network (KBN) is used as the delivery mechanism for management information. This allows multiple autonomous managers to co-exist, playing separate roles in the management of multiple parallel overlay networks using a single dynamic and flexible knowledge delivery mechanism.

## 1. INTRODUCTION

This paper addresses the development of a management mechanism for overlay networks. Specifically, we examine management tasks that relate to the *content* of communication being exchanged by applications. We manage a Knowledge-based Network, which is an extended content-based publish/subscribe middleware in which producers and consumers share knowledge across a network of message brokers based on the semantics of the content. Difficulty arises in managing such an overlay when the requirement for performance and managerial monitoring is distributed across multiple nodes. As a proposed solution we present a method of monitoring multiple, inter-connected and independent overlay entities which enforces technical actions based on incoming events. We achieve this through the use of Policy-based Network Management [1] in the form of event-triggered DROOLS rules [4]. Policy-based Network Management offers an event-condition-action (ECA) model in which policy decisions, located within a Policy Decision Point (PDP), are enforced across Policy Enforcement Points (PEP). Importantly, the events, which may or may not fire the policies, are filtered and matching to the policy systems using a Knowledge-based brokerage mechanism. This is interesting when applied to dynamic networks such as overlays, as it requires a more fluid level of event filtering. The knowledge-delivery mechanism is used and operated over multiple autonomous systems, thus operational change is much less predictable than in a relatively static network environment, such as an LAN. Policy systems, upon receiving an event, process the event and if conditions based on the system state are met, a management action is performed as specified by a set of event-condition-action policy rules. As the manager of the network, a human administrator can design policies according to high level goals or Service Level Agreements (SLAs). In a large, heterogeneous network with multiple managers both the collection of performance statistics

from the managed nodes and the enforcing of fired policy decisions across the same set of interconnected nodes becomes problematic. To provide a solution to this knowledge-delivery problem we can utilize the flexibility of the Knowledge-Based Network [3] itself in the communication between the overlay network being managed and the Policy-based Network Manager.

## 2. KNOWLEDGE-BASED NETWORKS

Knowledge-Based Networks [3] provide a subscription matching and event routing mechanism between a distributed collection of publishers (event producers) and subscribers (event consumers), conducted across a hierarchy of message brokers/routers. In this paper we view the policy systems, with its autonomic actions based on policy decisions, as the consuming entities in the publish/subscribe relationship. The collection of overlay nodes represent the producers, publishing management state at defined intervals using an agreed format.

The Knowledge-based Network matches the contents of publication messages to the filters, based around attribute, value, constraints held in users subscriptions. The KBN, in comparison to Content-based Networks (CBNs), utilizes semantics, in the form of ontologies, deep within the messages on which the network routes. Ontologies are seen as an electronic representation of agreed knowledge, and allow additional subscription filters to be applied to the KBN messages. CBNs allow a range of message types, including Strings, Integers and Booleans to be subscribed to using a number of operators, including =, !=, <, >, <=, >=, Prefix and Suffix (with regard to Strings). KBNs extend these types with Ontological classes, instances, properties and bags of all types using the More Specific, Less Specific, Instance of, Super and Sub Bag operators. KBNs support both semantic and non semantic message types offering a varied semantic subscription mechanism.

Linking the policy system and the overlay network is an extended KBN broker, termed the Trigger Broker. The Trigger Broker receives incoming subscriptions from the policy server (dynamically derived from its policy set) and stores these in a local subscription table. When management state (event) messages arrive from managed nodes in the overlay network, the Trigger Broker compares these to the stored subscriptions. If and when a match occurs the Trigger Broker forwards a notification to the relevant policy server, being the server that placed the original subscription. We see this as providing an initial level of event filtering and also acting as a portal from which multiple overlay networks and multiple policy systems can communicate. The Trigger Broker processes all incoming events as well as storing these events in a local database, which subsequently allows for the polling and retrieval of state information from the overall collection of brokers.

### 3. TECHNICAL ARCHITECTURE

The overlay network publishes performance statistics from a range of metrics to the trigger broker. The network administrator, being the same manager who designs and authors the networks policies, dictates to the nodes the values they are to publish and the interval at which publication should occur. This information is defined by the manager in a Managerial Ontology. This Managerial Ontology is built using classes, to represent the grouping of message types and instances to represent specific messages. For example a message class "Network\_Load", being a subclass of "Load" would both be represented ontologically by the two classes, but may be instantiated with multiple ontology individuals representing statistics the network management have interest in. The instances within a specific class represent the values the overlay node should publish, each instance within a class having a data type property which represents the interval at which the message should be published. This allows a manager to specify a bespoke extended ontology which is used by the overlay nodes to establish about what, and at what interval, they should publish their individual performance statistics. This ontology, once pushed to all of the nodes, is used by the overlay to start publishing information back to the Trigger Broker at the required interval using the ontologically represented message structure as defined by the manager. This process allows multiple managers with multiple interests, to define multiple metric messages based on the policies they design. The single ontology is reasoned over and used in the overlay nodes at startup and as and when an updated model is received, allowing an overlay to adapt as operational characteristics change.

The manager now has a method in which multiple overlay nodes can be updated to publish usage statistics using a flexible and adaptive mechanism where individual user requirements can be accommodated in a generic and flexible ontological model. This representation of required statistical publications are then utilized by the overlay nodes to publish performance metrics based on the defined interval, and a specific make-up, to a nominated Trigger Broker. Using the ontological and non-ontological operators and types available within the KBN, the network administrator can form very expressive subscriptions to certain conditions being met within the network. The networks manager registers this subscription with the Trigger Broker and awaits incoming notifications of matches from across the vast set of performance metrics arriving at the Trigger Broker from across the network.

When and if a notification does arrive at the policy server, it is parsed and the parts of the message used to update an individual Management Object, which represents an individual element in the overlay network, as part of an overall Management Information Base (MIB). Once the relevant managed object has been updated, using the received notification, the complete set of MIBs are passed to the policy engine. This engine processes each of the MIBs against each of the policies, such a process may, or may not, fire a policy (when the MIB of a particular broker matches the conditions outlined in the policy).

The Knowledge-delivery mechanism deals with event collection and condition matching as a pivotal part of the Policy-based Network Management process. The final stage of the operational design of the network is the creation of an overlay effector. The effector allows a manager, when designing policies, to prescribe the relevant actions to take when certain conditions are met. This

actioner completes the autonomic loop, in which overlay nodes publish to and policy systems subscribe against an individual Trigger Broker. With the Trigger Broker being formed around an adapted KBN broker node, we envisage future implementations of the Trigger Broker network to operate using multiple Trigger Brokers (as opposed to the single Broker discussed within). The introduction of multiple Trigger Brokers will occur in line with operational demand, as the number of managers and managed objects grow, and in a situation where incoming publications anywhere within the network are routed successfully towards the relevant and subscribed policy server.

### 4. CONCLUSIONS & FUTURE WORK

The managed distributed overlay, discussed within this paper, is in effect an additional Knowledge-based Network Deployment. The management mechanism developed and discussed here derives from a goal to apply a Policy-based Network Management approach to the clustering of KBN publishers and subscribers, around relevant brokers. The initial benefits and procedures seen as relevant to the management of such a clustered network are outlined in [2]. Having created an overlay management system in which flexibility and dynamism are central to the operation of an autonomic control loop, detailed experimentation will evaluate this control loop for the clustering of users in a publish/subscribe network. Having designed the management actions as used by the policy system effector and defined which metrics are relevant, detailed test policies will be designed and executed. Although an initial set-up cost is apparent within the system, this cost is seen as being acceptable when offset by the benefit of a single overlay being managed by multiple separate administrative domains. Additionally this cost is seen as negated once the full system is deployed and autonomy is achieved.

This paper presents the architectural overview of a system developed using ontological semantics in the delivery of events from a managed overlay network to policy servers. We see the extra filtering of event messages, both at the Trigger Broker and at the Policy Server as offering a scalable mechanism for the implementation and operation of a flexible and dynamic control loop moving towards subsequent autonomic management.

*This material is based upon works supported by the Science Foundation Ireland under Grant No 05/RFP/CMS014.*

### 5. REFERENCES

1. Boutaba, R. and Aib, I. Policy-based Management: A Historical Perspective. *The Journal of Network & Systems Management*, 15 (4). 447-480.
2. Jones, D., Keeney, J., Lewis, D. and O'Sullivan, D., Policy-based Management of Semantic Clustering. in *The 2nd International Conference on Distributed event-based systems (DEBS08)*, (Rome, Italy, 2008).
3. Keeney, J., Roblek, D., Jones, D., Lewis, D. and O'Sullivan, D., Extending Siena to support more expressive and flexible subscriptions. in *The 2nd International Conference on Distributed Event-Based Systems (DEBS 2008)*, (Rome, Italy., 2008).
4. Proctor, M., Neale, M., Frandsen, M., Griffith Jr., S., Tirelli, E., Meyer, F. and Verlaenen, K., DROOLS 4.0.7 <http://www.jboss.org/drools/> Accessed March 2009.