Knowledge-based Semantic Clustering

John Keeney, Dominic Jones, Dominik Roblek, David Lewis, Declan O'Sullivan
Knowledge & Data Engineering Group (KDEG) & CTVR
Trinity College Dublin, Ireland
{ John.Keeney | jonesdh | roblekd | Dave.Lewis | Declan.OSullivan }@cs.tcd.ie

ABSTRACT
Users of the web are increasingly interested in tracking the appearance of new postings rather than locating existing knowledge. Coupled with this is the emergence of the Web 2.0 movement (where everyone effectively publishes and subscribes), and the concept of the "Internet of Things". These trends bring into sharp focus the need for efficient distribution of information. However to date there has been few examples of applying ontology-based techniques to achieve this. Knowledge-based networking (KBN) involves the forwarding of messages across a network based not just on the contents of the messages but also on the semantics of the associated metadata. In this paper we examine the scalability problems of such a network that would meet the needs of Internet-scale semantic-based event feeds. This examination is conducted by evaluating an implemented extension to an existing pub-sub content-based networking (CBN) algorithm to support matching of notification messages to client subscription filters using ontology-based reasoning. We also demonstrate how the clustering of ontologies leads to increased efficiencies in the subscription forwarding tables used, which in turn results in increased scalability of the network.

Categories and Subject Descriptors
C.2.2 Network Protocols: Routing protocols, I.2.4 Knowledge Representation Formalisms and Methods: Semantic Networks

General Terms
Performance, Experimentation

Keywords
Publish-subscribe, content-based networking, ontologies

1 INTRODUCTION
Establishing a global event service at Internet scales presents a major challenge for existing networking technologies. Such an event service is crucial in the support of the explosion of dynamic interactivity expected through the increased use of Web 2.0 technologies where diverse and an increasing numbers of publishers and subscribers of content will be more mobile and dynamic [1]. The time at which items are posted is increasing in importance relative to the content of the post, e.g. blog postings rapidly fade in importance as time passes. The web has responded to this need with RSS feeds which allow event postings, to quickly be notified to interested users. However, this system relies on users subscribing to feeds of pages they have already located, whilst feed aggregators offer only rudimentary searches or simple classifications of feeds. This is partly because the near-real time events present in feeds are disassociated from the system of user-defined hyperlinks required by search engines which also introduces a discovery latency that is unacceptable to feed users. Though we can search for the static pages we are unable to search the body of feed events active at any point in time. As we look forward to future uses of the Web, in support of the 'Internet of Things,' searching for events becomes more important as devices and sensors become sources of high frequency postings of interest. In [27] it is suggested that that an Internet-wide event service may need to scale to 10^9 events per second, a similar order of event producers and huge variability in the proportion of consumers subscribing to an event. Current event-based publish-subscribe systems offer a networking model that is well suited to such applications, but they are typically limited to relatively static characterisations of events. Elements of this are being addressed by developments in Content Based Networks (CBN), a specialisation of the pub-sub paradigm where message forwarding is based on message attributes and their values. Extensive research is ongoing into finding a balance between restricting the expressiveness of event attribute types and subscription filters, their efficient matching at CBN nodes and efficient maintenance of routing tables [11, 21, 22, 33]. Currently user subscriptions are limited to simple syntactic matches (typically integers, strings and Booleans). In [16, 29, 31] the concept of Knowledge Based Networking is introduced, as a semantically enhanced publish-subscribe model extending content-based networking (CBN). This novel integration of semantics within the pub-sub routers themselves allows messages to be matched to subscriptions based not only on their contents, but also their semantics. Producers of knowledge express the semantics of their available information based on an ontological representation of that information, and publish semantically enriched messages as required. Consumers express subscriptions upon that information as long-lived semantic queries, in response to which they continually receive suitable matching messages. A Knowledge-based Network (KBN) is therefore more flexible, open and reusable to new applications. However, the scalability of a KBN to Internet scale requires a routing mechanism that minimizes both the size of routing state held in KBN nodes and the overhead of ontological reasoning in nodes. To address this, [29] proposes the efficient partitioning of the routing space based on clustering related to the semantics of message contents, rather than grouping within the hierarchies of network addresses. In this paper we describe some empirical evaluation into the performance of semantic-based clustering within a deployed KBN using realistic distribution of subscriptions, notifications and their semantics based on characteristics of existing RSS feeds.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

2 KBN IMPLEMENTATION

The particular flavour of KBN investigated in this paper is an extension of the Siena CBN [12]. A Siena notification is a set of typed attributes, each attribute is comprised of a name, a type and a value. The current version of Siena supports the following basic types: string, long, integer, double and Boolean. Siena subscriptions are a conjunction of filtering constraints, where constraint is comprised of the attribute name, an operator and a value. The subscription operators currently supported are equality and less/greater than etc., and for strings, substring, suffix and prefix. Each Siena router maintains its own set of subscriptions (routing table), which is dynamically built from the specific subscriptions it receives. A subscription "covers" a notification, if the event matches to all filtering constraints of a filter. Notifications are delivered to a client, if the client has submitted a subscription where the conjunction of the subscription's filters covers that notification. Siena also discovers coverings between subscription filters to optimize subscription routing. A filter covers another filter, if all notifications covered by the latter are also covered by the former. The Siena covering relationships, defined in [12, 20], allow each router's subscription set to be dynamically arranged into a hierarchical tree structure (routing table), with more general subscriptions towards the top, and more specific subscriptions towards the bottom. This structure allows subscriptions to be efficiently and correctly aggregated together to reduce the subscription tree size and efficiently match each publication to subscriptions as it passes through each router.

In [16, 29, 31] a KBN implementation is presented that extends Siena by providing three additional ontological base types: properties, concepts and individuals. It also supports subsumptive subscription operators, i.e. sub-class/property (more specific), super-class/property (less specific), and equivalence. E.g., a subscriber can subscribe to all KBN messages that contain an attribute whose value is a concept more/less specific than the named concept in the subscription. To achieve this, each KBN router holds a copy of a shared OWL ontology, within which each ontological class, property and individual used is described and reasoned upon. These new ontological types are first class KBN types, and can be used in any KBN subscription or notification, along-side the standard Siena types and operators. Due to the transitive nature of the sub-property/class and super-property/class properties, covering relationship for these operators were defined in [31], maintaining Siena's subscription aggregation efficiencies. A further fully-implemented extension, presented in [34], introduces a new "bag" type and associated bag operators. A bag (also called a multiset) is a set-like object . The bags of integers {1,2,3} and {2,1,3} are equivalent, but bags {1,1,2,3} and {1,2,3} differ. The bag of integers {1,1,2,3,4} is a super-bag of {2,4,3} and so on. A bag can contain any valid Siena values, including other bags. Bags are first order members of the Siena KBN type set so can appear in notifications, as well as in subscription filters.

The bag operators can also be combined with other Siena KBN operators to produce composite bag operators. The composite bag relation is a binary relation over bags composed of (i) another binary bag relation over bags and of (ii) a sub-relation over the bag elements. The bag of integers {1, 1, 2, 3, 4} is a super-bag of {2, 4, 3} using the default "equals" (==) sub-relation. The bag of integers {1, 2, 3} is an equal-bag of {2, 3, 4} using the "less than" (<) sub-relation. *(i.e. for every element in the second bag, there exists an element in the first bag that is* less than *the element, with no unused elements in either bag)*. A full description and logical proofs of KBN bags, and the simple and composite bag operators are outside of the scope of this paper, but are provided in [34].

These bag type and operator extensions greatly extend the expressiveness of the Siena KBN subscription mechanism, especially when combined with the ontological operators. Again, due to the transitive nature of the sub/super bag operators, when combined with the covering relationships for the other Siena KBN operators, covering relationships for the bag operators can also be defined [34]. This maintains the efficiencies of Siena, allowing a single homogenous KBN to scale to moderate sizes.

3 BENCHMARKING KBN PERFORMANCE

Many of the parameters that affect the performance of a KBN's routing scheme are largely application specific. Therefore a KBN can only be evaluated through its use in supporting diverse applications in a variety of scenarios. A benchmark, specifically for KBNs, is presented in [25], based on a synthetic benchmark for Content-based Networks in [26]. It defines the set of parameters that must be defined before an application of a specified KBN can be evaluated in either a qualitative or quantitative manner. These are summarised below:

Message generation: publication rate; subscription rate; active / inactive subscriptions cycle durations.

Publication generation parameters: number of fields in publication; publishers' ontologies (defined in terms of content, size, complexity, expressiveness, bushiness etc).; names of attributes in publications, which may be drawn from publishers' ontologies; type of each attribute; value space for each attribute, which may be drawn from publishers' ontologies.

Subscription generation parameters: number of subscriptions per subscriber; number of filters per subscription; subscribers' ontologies (defined in terms of content, size, complexity, expressiveness, bushiness etc. and its similarity to the publishers' ontologies); names of attributes used in each filter, which may be drawn from the publishers' or subscribers' ontologies; type of each attribute used in the filter; attribute values used in filters, which may be drawn from the publishers' or subscribers' ontologies; operators used in filters.

KBN routers' ontologies (defined in terms of content, size, complexity, expressiveness, bushiness etc. and their similarity to each other, the publishers' ontologies and the subscribers' ontologies).Only once the parameters listed above have been made explicit for each application running on top of a KBN, the performance can then be effectively and accurately compared.

4 PODCASTING – A REAL WORLD-BASED PUB - SUB USAGE SCENARIOS

In order to undertake empirical evaluation into the performance of a KBN using the benchmark identified in section 3, it was necessary to identify realistic distributions of subscriptions, publications and their semantics. Despite the increasing adoption of semantic-based metadata within the Web 2.0 community, there remains few sources of information to define distributions of subscriptions, messages and their semantics for different applications. In order to identify some realistic benchmark values we examined the distribution of podcast update feeds.

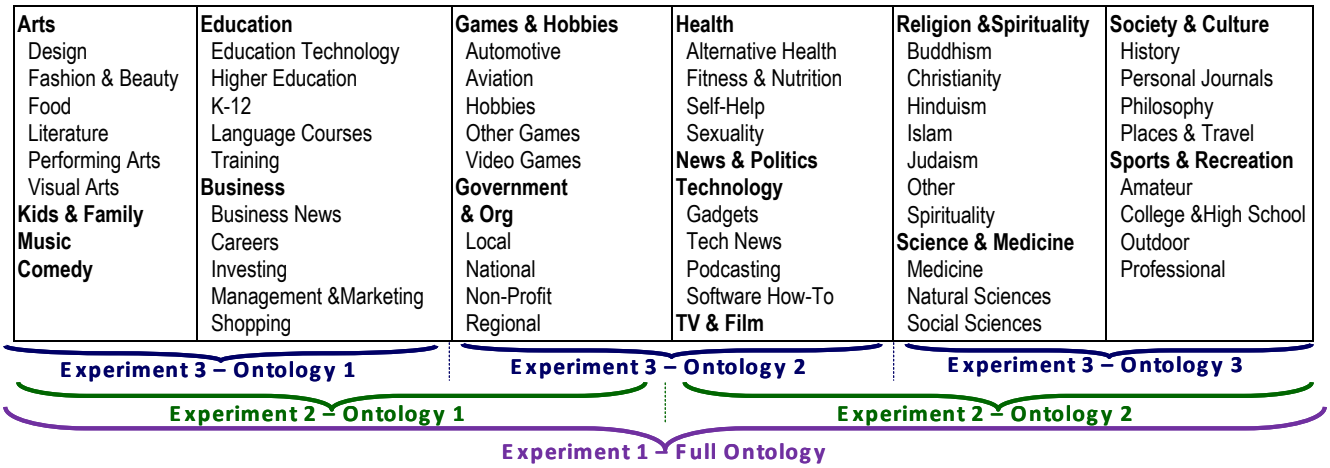

Figure 1: Categories in the Apple iTunes Podcast schema (Ontology)

4.1 PODCASTING

The application area chosen for study was podcasting, due to its popularity and the availability of a semantically rich de-facto metadata standard, the iTunes XML schema [7]. The iTunes schema allows for basic descriptors to be combined with semantically rich tags such as descriptive hierarchical categories, keywords, and ownership. Of particular interest is the set of hierarchical categories defined in the schema, as shown in Figure 1 which are used to annotate messages and aid in the searching for relevant podcasts. Firstly, it was necessary for the authors to establish exactly how many podcasts were actually being produced and consumed, or in terms of this paper, published and subscribed to. In 2006 a number of pivotal net analysts, including Neilson and Pew, aimed to answer this question. The survey by the "Pew Internet and American Life project" [3] in November 2006, through a telephone survey of 2,928 adults within the Continental US, showed 12% of Internet users had downloaded a podcast, averaging an estimated 65 million podcast listeners within the U.S.A. This is in contrast with the 7% of users who reported downloading a podcast in their April 2006 survey showing a growth of around 5% in as many months. The Neilson analysis [4] shows varying yet similar results to the data collected by Pew. Based on a phone interview of 1700 participants, 6% of the respondents were "regular podcast downloader's" leading to an estimate that 6% of US Adults, or around 9 million Web users had downloaded podcasts in the July-August period of 2006. Neilson also estimates that 72% of the respondents who regularly download podcasts download an average of 1-3 podcasts per week, 10% of whom download 8 or more podcasts per week. Neilson concludes approximately five podcast downloads per week was a fair estimate of average consumption. These studies provided us with a good indication of type of growth in podcast subscription that we should model in our distribution. From data donated by the administrators of FeedBurner.com (http://www.feedburner.com), one of the most popular and established podcast syndication websites, we determined the following with respect to characterising the distribution model for publications and subscriptions. In 2006, the number of podcast feeds grew from 31,167 feeds to 83,743 feeds, resulting in a growth of 52,576 new feeds over that 1 year period. When distributed equally over the year, this equates to a new podcast producer publishing approximately every 10 minutes. From a survey of the most 30 of the popular feeds, the average update period for each feed averaged one update per week. Where each feed is represented by one publisher, and each publisher publishes a new notification every week, this means that a new publisher starts on average every 600 seconds, with an average continuous publication rate of one notification every 604,800 seconds per publisher. The data from FeedBurner.com also shows a growth in subscriptions from 915,277 to 6,434,758, resulting in 5,519,481 new subscriptions in 2006 alone. This can be approximated to 836,285 new subscribers over the year or one every 37.73 seconds. Based on the data in a Yahoo White paper on RSS feeds [9], each subscriber maintains an average of 6.6 subscriptions. It is estimated that podcast users very rarely change their subscriptions once they have found feeds that they like, and they rarely subscribe to feeds that they do not like. For these reasons this scenario estimates that each subscriber takes one week to subscribe to only their favourite 6.6 feeds, and then never unsubscribe. These approximations mean that a new subscriber is created on average every 37.73 seconds, which each creates an average of 6.6 subscriptions over a week (i.e. one every approx. 91,636 seconds or 25.5 hours), and then continuously waits for messages, the ratio shown in Figure 2:

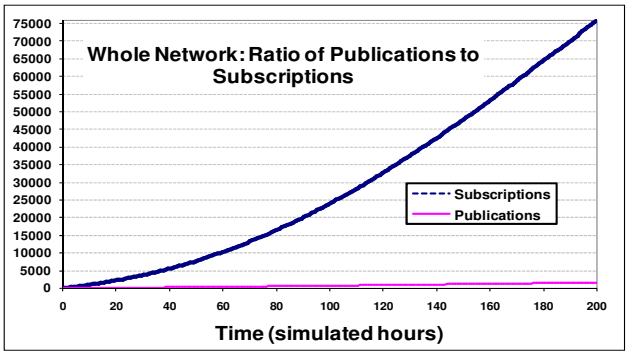

Figure 2: Ratio of Publications to Subscribers

4.2 DEPLOYING THE EXPERIMENT

In striving to replicate a real world distributed deployment, the PlanetLab network [10] was used to deploy a KBN network and exercise it according to the scenario above. The PlanetLab network comprises 756 nodes, distributed across 368 sites worldwide. Whereas traditional network simulations are evaluated on either a local or virtual test-bed, the PlanetLab environment allowed us to experiment across a physical Internet infrastructure of 77 random machines distributed across Europe, North America,

South America, Asia and Australia. The experimental setup consisted of 37 nodes running as dedicated KBN routers, 15 nodes running as dedicated publication creators and a further 25 were used as dedicated subscription generators. The 37 KBN routers form the hierarchical overlay as shown in Figure 3.

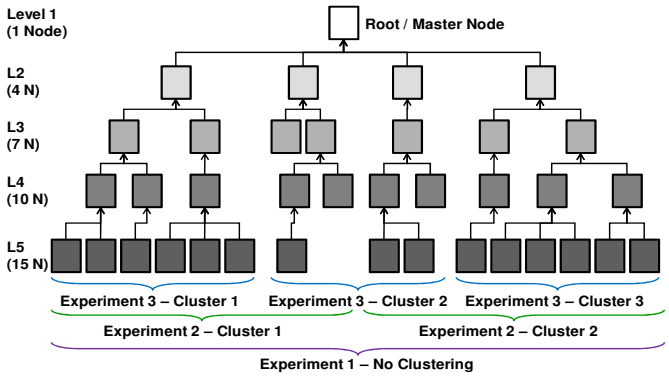

Figure 3: The KBN overlay network

4.3 THE PODCASTING BENCHMARK

To evaluate the performance of the KBN we simulated the distribution of podcast feed updates according to the traffic characteristics discussed in section 4.1. In this scenario a KBN publishing client is created for each feed, and the client generates KBN publications for every update announcing a new podcast episode for that feed. A KBN client was also created for each feed subscriber, and that client created a separate KBN subscription for each of its feed subscriptions. To speed up the gathering of data it was decided to speed-up the experiment by a factor of 365, i.e. model a full year's traffic in a single day.

Message generation rates: These are sourced from section 4.1: a new publisher was started on average every 600 seconds, with an average continuous publication rate of one notification every 604,800 seconds per publisher. a new subscriber was created on average every 37.73 seconds, which each created an average of 6.6 subscriptions over a week (i.e. one every approx. 91,636 seconds or 25.5 hours), and then continuously waited for messages.

Publication generation parameters: The publishers' relatively shallow and simple ontology was hand-crafted from the item categories as defined in the Apple iTunes podcast schema shown in Figure 1. This ontology is relatively small, at 38 kilobytes, with 67 classes, no properties and no individuals. Each publication message contained 15 named attributes, as defined in the Apple iTunes podcast schema. (*title, link, copyright, pubDate, itunes_Author itunes_Block, itunes_image, itunes_duration, itunes_explicit ,itunes_newFeedUrl, itunes_owner, itunes_subtitle, itunes_summary, itunes_category, itunes_keywords*). All of the named attributes except *itunes_category* and *itunes_keywords* were of type String. *itunes_category* was defined as a bag of ontological classes, and *itunes_keywords* was defined as a bag of Strings. Following a survey of the 30 most popular feeds hosted by FeedBurner.com, the average number of keywords in each podcast item/episode was calculated as 4, therefore the *itunes_keywords* bag of each publication contained 4 keyword strings. These keywords were randomly drawn from a dictionary of 80 popular keywords. In the same survey, an average of 3 categories were attached to each publication, so the *itunes_category* bag of each publication contained on average 2-4 classes, drawn randomly from the publishers' ontology described above and shown in Figure 1. The values for all of the attributes except *itunes_category* and *itunes_keywords* were hard-coded as static strings.

Subscription generation parameters: Despite extensive searches we were unable to locate any information about how subscribers search for and select podcasts. For this reason, we decided to base the subscription format on what we considered the most useful and important semantic attributes of published podcast update notifications, i.e. the *itunes_keywords* and *itunes_category* attributes. When searching for actual podcasts the user would most likely use a search engine. The most popular search engines, including www.podcast-search.info, www.google.com, podcasts.yahoo.com and so on, all implement searches using a conjunction of keywords. In this scenario we decided to implement this subscription using the bag subscription mechanism discussed in section 2 by requiring that any matching subscription's *itunes_category* bag of keywords must be a *super-bag* of the keywords requested by the searcher. In this scenario, each subscriber subscribes using between 0 and 3 keywords randomly drawn from the same dictionary used by the publisher. When searching, a user would most likely select a single category, which would include all sub-categories if it was a parent category. Using the compound bag operator and described in section 2, this search can be implemented by requiring that the *itunes_category* attribute of any message must contain a bag of categories that is a super-bag of the required category, where one of the elements in the published *itunes_category* bag must be "more specific" than the required category, i.e. subsumed by the requested category. (*i.e. a super-bag using the "more specific" sub-operator*). In this scenario each subscriber specifies one category class drawn from the same ontology as the publisher. Therefore, each subscription is a conjunction of 2 constraints, 0-3 keywords and a category class. This is based on the experimental assumption that a user when searching for content is typically less specific than a user posting content. In this scenario only the *itunes_category* and *itunes_keywords* attributes were used in the subscription filters so it was acceptable to have the other unused attributes in the publications hard-coded as representative static strings.

Network topology: Combined with the goal of testing the KBN in a physically distributed environment, due to the resource requirements of this very large scale KBN deployment scenario (in particular the memory requirements of the multi-threaded publishing and subscribing clients rather than that of the routers), it was not possible to test the KBN's operation for this scenario without widely distributing the workload of the clients. As discussed in the section 4.2, the experiment was deployed across 77 distributed PlanetLab nodes, with 37 randomly selected nodes acting as KBN routers deployed in a hierarchy shown in figure 3, 15 randomly selected dedicated publishing nodes, and 25 randomly selected dedicated subscribing nodes. This workload distribution took into account the high subscription rate and relatively low publication rate. Considering the hierarchical nature of the network, and envisioning that the Root/Master node would suffer the highest loading, no publisher or subscriber sent messages directly to the Root Node.

KBN routers' ontologies: The KBN routers each used a copy of the same podcast categories ontology as used by the subscribers and the publishers. The hypothesis of this research is that the

clustering of KBN nodes according to the semantics of the knowledge they present or request will have a positive effect on the performance of the KBN and improve its scalability. To evaluate this hypothesis the operation of the KBN was evaluated in 3 experiments. The scenario described was evaluated by crudely dividing the KBN hierarchical overlay into clusters of approximately equal size, as shown in Figure 3. In the first experiment the network was not divided (one cluster). In the second experiment the same logical network hierarchy was divided into two clusters. In the third experiment the same network was divided into three clusters. When divided, each cluster was tasked with focusing on only a proportion of the ontology, as shown in Figure 1.

To demonstrate the expected difference in performance due to clustering, the KBN's operation was measured. In the first "unclustered" experiment, each publisher and subscriber could send their subscriptions and publications messages to any random node in the network (except the Root node). In the second and third "clustered" scenarios, depending on the semantics of the subscription or publication to be sent, the client selected a random node from whichever cluster was most suited to receive that message (*i.e. the cluster that focussed on the portion of the ontology which contained the majority of their referenced ontological concepts*). If the message referenced the same number of concepts from all portions of the ontology then the message could be sent to any node in the network. In each of the experiments the same volume of publications and subscriptions were created, according to the same message generation distributions described above. It is important to note however that although the cluster to receive the message was calculated, the particular node within the cluster that would receive the message was randomly selected from the cluster's members. This approach was taken to maintain the idea that in a hierarchical overlay pub-sub network, where the logical hierarchy may be very different from the physical network's topology, clients are not restricted by which router (or *broker*) they should connect to. In many cases clients may connect to their closest router, but this is not a requirement, hence random node selection could be considered a worst-case scenario.

5 RESULTS AND FINDINGS

The primary metrics used in this paper were to compare the performance of the KBN's operation where the characteristics of the subscription tree / routing table stored at each KBN router. This was due the end-to-end nature of the network, the heterogeneity of the machines in the PlanetLab network, and the variability of dynamic resource availability on the PlanetLab nodes as they ran other experiments concurrently. The authors feel that this is a fair metric to objectively compare the experiments for two reasons. Firstly, by the nature of using content-based subscription filter matching, for each notification potentially all subscriptions filters may need to be evaluated against the notification. For each subscription the subscription tree needs to be searched to find the optimal position to insert the subscription. Therefore, a smaller and more ordered subscription tree is more efficient. Secondly, the hierarchical logical topology of the KBN overlay were randomly created, and clients connect to random nodes (within a cluster), so the aggregate network traffic across each of the experiments should remain similar (especially for larger networks.)

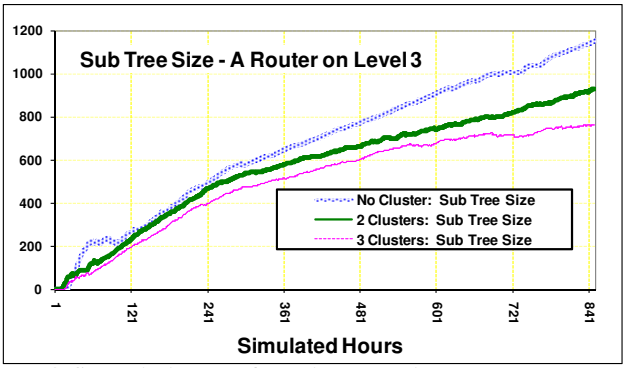

Figure 4: Subscription tree / Routing table size on one KBN router

Figure 4 shows the sizes of the subscription tree / routing table for a node at level 3 in the middle of the KBN hierarchy. The graphs are truncated to approximately 850 simulated hours since soon after this point the resources requirements of the subscribers began to exceed the conservative fair-use resource allocation of the oldest PlanetLab nodes, mainly due to the 365X speedup factor used in these experiments. Resource throttling at these weaker nodes meant that the results became unreliable after approx 1000 simulated hours (40+ simulated days). The graphs show how the total subscription tree size is smaller when semantic clustering is employed, despite a very similar number of subscriptions arriving at the node for each experiment. Despite the fact that the rate at which subscriptions were arriving continued to increase according the distribution in figure 1, the total subscription tree size was starting to level out. The results therefore show that for a given number of received subscriptions, the subscription tree is smaller when semantic clustering is employed. Similar graphs were generated for nodes at each level in the hierarchy, and all show very similar results (except the Root node), and so are not presented here. The main optimisation feature of the Siena Hierarchical CBN, upon which the evaluated implementation of the KBN is based, is the use of subscription aggregation / covering to merge and order similar subscriptions. In the Siena subscription tree, subscriptions which cannot be merged with other subscriptions, or grouped under more general (covering) subscriptions, form "root" subscriptions in the tree. The count of "root" subscriptions, when considered with the size of the subscription tree, gives an overview of the searchability and optimality of the subscription tree at each node. Therefore the number of Siena "root" subscriptions at each node was also measured, as shown in Figures 5-8.The "root" subscriptions at each node are the most general subscriptions of each node so it is only these root subscriptions that need to be passed up to a router's parent node. To a parent router a child router appears as a subscribing client using only the child's "root" subscriptions.

This is done iteratively up the tree, so subscriptions become more general in the nodes towards the top of the network, and more specific towards the bottom. This explains why the more general subscriptions flowing up the hierarchy cause the subscription tree to reach optimality quicker as more general subscriptions reduce the number of "root" subscriptions. In addition, with respect to how child nodes send their "root" subscriptions to their parent node, it can be seen that semantic clustering greatly reduces this traffic and associated routing table "churn". This is particularly apparent towards the bottom of the network. Again similar graphs

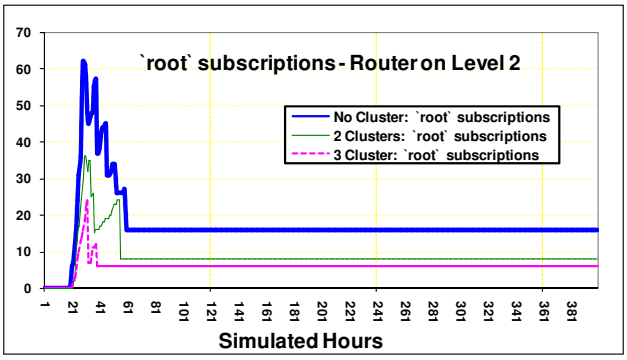

Figure 5: "Root" subscriptions in a level 2 router

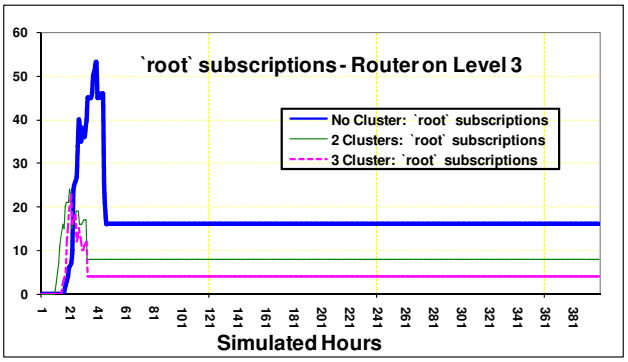

Figure 6: "Root" subscriptions in a level 3 router

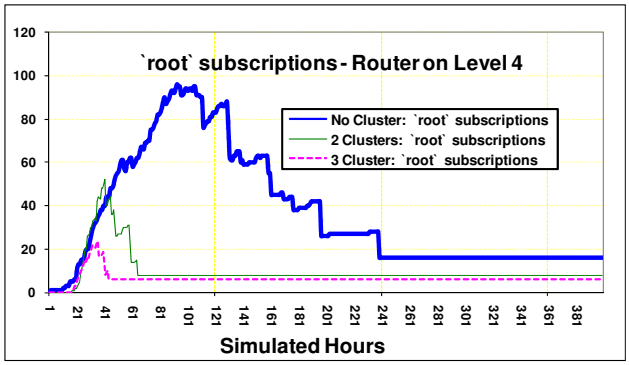

Figure 7: "Root" subscriptions in a level 4 router

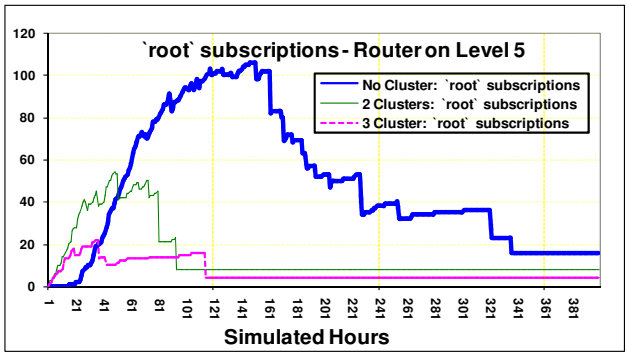

Figure 8: "Root" subscriptions in a level 5 router

were generated for other nodes at each level in the hierarchy, and all show similar results except the Root node. In a purely hierarchical network, where messages may be routed from one side of the network to the other side, it is clear that the Root/Master node can form a bottleneck in terms of the scalability of the network. As all the traffic travelling from one branch (and cluster) to another grows, the routing overhead in the Root node also grows. For this reason, to maximise scalability, it is necessary to minimise the size of the routing table at the root node, and optimise its searchability.

Figures 9 and 10 clearly show that for the application characteristics, KBN configurations, and scenarios introduced above, the subscription tree / routing table in the root node is reduced and converges to smaller size when even rudimentary semantic clustering is performed. In addition, since the subscribing clients do not send subscriptions directly to the Root node, the only subscriptions reaching the Root node come from the "root" subscriptions of the nodes on level 2 of the hierarchy. As a result of only receiving more general subscriptions, there is a much smaller difference between the total subscription tree size and the number of "root" subscriptions in the node. Shown in, specifically Figure 10, are the root subscriptions which are the covering subscriptions. The figures show the roots subs reaching maximum capacity, which with the introduction of new subscription content would begin a reversal of this trend.

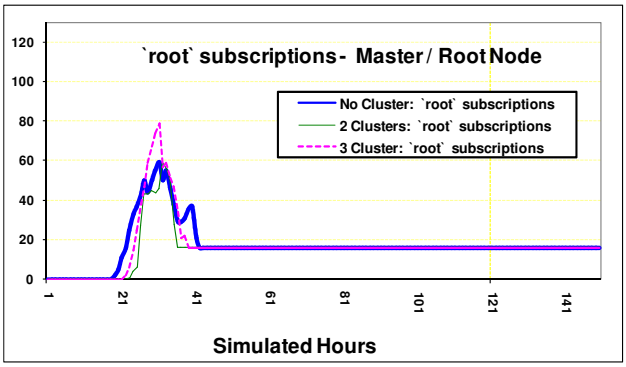

Figure 9: "Root" subscriptions in the Master/Root Node

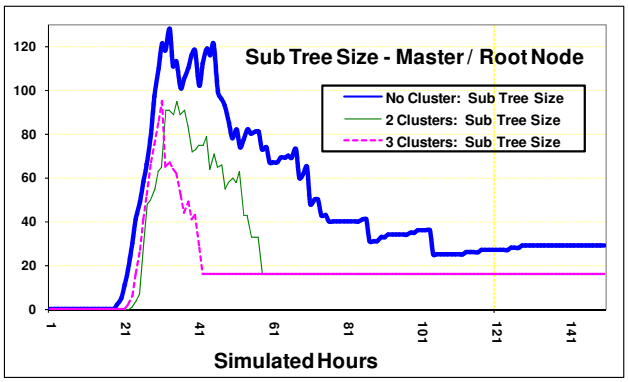

Figure 10: Subscription tree size in the Master/ Root node.

6 RELATED WORK

There has been little examination of the use of ontology-based semantics in content-based networking in the scientific literature. In [17] a semantic publish-subscribe system is presented, but it is based on a centralized (pub-sub bus) implementation and thus is

limited to enterprise scale and does not offer true CBN capabilities. In [35], semantics can be used in messages in a pub-sub middleware; however, the semantics are used only at the edge of the network in a manner similar to a small scale study presented in [30]. The KBN presented in this paper uses semantics deep in the forwarding algorithm of each message router within the network. Much work to date on content-based networks has focused on how efficiency in routing can be gained through subscription aggregation and merging. Recent progress with the XNET CBN has shown that perfect routing can be achieved in a scaleable manner independently of subscriber joins and leave rates though subscription aggregation [33]. The HERMES CBN [22], ToPSS [18] and the REBECCA CBN [24] have all applied peer to peer distributed hash table (P2P DHT) mechanisms to the formation of routing tables in CBN nodes. This is interesting in that it may form the basis for a flexible and robust clustering mechanism for routing in KBNs. It should be noted that though P2P systems themselves are concerned with efficiently routing queries to matching information sources in a query-response manner, they do not address the CBN concern of optimally routing a sequence of asynchronous replies back to the set of querying, or in CBN terms, subscribing clients. P2P DHTs provide efficient routing by using a cost metric keyed to the physical topology of the network resulting in average hop-counts for a route in the order of the log of the number of nodes in the network i.e. **O**(log(N)). However a difficulty remains in the mapping of content based subs to a key space suitable for DHTs.

There are several attempts at applying P2P DHT techniques to the retrieval of distributed ontology encoded knowledge information, e.g. in RDF, in semantic overlay networks [11, 15, 23]. In supporting an ontology-driven DHT-based P2P routing mechanism for the KBN, the approach outlined in [15] seems most promising due to its support for peer clustering. Used in this way, peer clustering introduces a hierarchy of peer groups based on policies. Such policies could admit nodes based on semantic closeness, recorded performance, administrative domains or indeed reasoning capabilities. It therefore provides a mechanism for these different routing configuration strategies to co-exist, serving different application domains or user communities in a way that supports incremental deployment and innovation.

Like the preliminary approach taken in this paper, the design presented in [5] uses the semantics of the message and knowledge of the entire network to decide where a subscription should be inserted into the network to minimise the routing table at individual nodes. A slightly different approach in [2] requires the entire network to be searched for a cluster before a subscription is submitted. However, these systems remain CBNs rather than KBNs because the semantics of the message cannot be used in subscriptions and so lacks the expressiveness of the system physically evaluated in this paper. In the presented KBN, it is planned to employ more sophisticated and dynamically reconfigurable clustering schemes, that can be without the need for the complete knowledge of the semantics of all of the network, either before hand or by searching, so improving scalability [29].

7 CONCLUSIONS AND FUTURE WORK

This paper raises some of the scalability issues involved in building a global Knowledge-Based Network. Fundamental to this is the need to support a large array of heterogeneous types in notification messages to accommodate the global variety in message sources and in the subscriptions to those messages. The performance of a KBN implementation which extends the Siena CBN has been explored. One part of the extension provides ontological concepts as an additional message attribute type, onto which subsumption relationships can be applied. The other part provides for a bag type to be used that allows bag equivalence, sub-bag and super-bag relationships to be used in subscription filters, composed with different bag element comparators. These two extensions augment the expressiveness of CBNs to directly support two major evolutions in the typing of data on the web, the use of ontologies in the Semantics Web and the use of string based tagging and folksonomies in Web 2.0. These evolutions allow the WWW to cope with a dramatic increase in the number of sources of information by providing richer meta-data about content; however the widespread use of rich semantics in meta-data is still not in evidence.

One of the main questions that surround the use of ontologies deep in the network at the routing layer remains the evaluation of the resulting performance overhead. Previous small scale studies in this area [29, 30, 31] show a definite performance penalty, but this may be acceptable when offset against the increased flexibility and expressiveness of the KBN subscription mechanism. Further research is required to evaluate how the performance of "off-the-shelf" ontology tools will affect the scalability of KBNs within larger scales. These results point to the potential importance of semantic clustering for efficient network and performance scalability. It is acknowledged that the experiments in this paper demonstrate only rudimentary semantic clustering. However, the experiments in this paper clearly demonstrate how even inflexible and static clustering can have a substantial positive effect. Ongoing research will focus on how clustering can be performed dynamically as the semantics of the data within the network changes.

Work is also focusing on integrating policy-based cluster management for the KBN [29] to support much more sophisticated cluster schemes, e.g. overlapping clusters and hierarchies of clusters under separate administrative control. Policy-driven clustering enables the size and granularity of peer clusters to reflect different application domains. For example, the clustering policy may be specified in terms of accuracy, latency or reasoning resources as well as the semantic spread of the query-able knowledge-base, or in terms of queries across a peer population and of the querying load across that population. In addition, the effect of semantic interoperability in node matching functions and in inter-cluster communications will be assessed. This requires evaluation of different schemes for injecting newly discovered semantic interoperability mappings into the ontological corpus held by any given cluster, as well as how these mapping are shared between clusters. We expect that any practical system will need to adapt its clustering to reflect the constantly changing profile of semantics being sent and subscribed to via the KBN, thus creating a network environment in which messages are passed from node-to-node, cluster-to-cluster based not on the data's destination but based on the messages semantic data.

Acknowledgement. This work is funded by Science Foundation Ireland under Grant No 05/RFP/CMS014.

REFERENCES

[1] Web 2.0: Predictions and Pithy Analysis Charles Buchwalter, VP Industry Solutions, Nielsen//NetRatings – Nov 2006

[2] Anceaume, E., Gradinariu, M., Datta, A. K., Simon, G., Virgillito, A., "A Semantic Overlay for Self- Peer-to-Peer Publish/Subscribe", Int'l Conf. on Distributed Computing Systems (ICDCS'06), 4-7 July 2006, Lisboa, Portugal.

[3] "Pew Internet Project Data memo", Mary Madden, Pew Internet and American Life Project, Nov 2006

[4] "The Economics of Podcasting" ,Nielson Media Research, July 2006

[5] Chand, R., Felber, P., "Semantic Peer-to-Peer Overlays for Publish/Subscribe Networks". EuroPar 2005, European Conference on Parallel Processing, Lisboa, Portugal, 2005.

[6] Chand, R., Felber, P., Garofalakis, M., "Tree-Pattern Similarity Estimation for Scalable content-based Routing". ICDE 2007, Int'l Conf. on Data Engineering, Istanbul, Turkey, 16-20 April, 2007.

[7] Apple - iTunes - iTunes Store - Podcasts - Technical Specification. Retrieved March 18th, 2007 from: www.apple.com/itunes/podcasts/techspecs.html

[8] Swoogle's Statistics of the Semantic Web Retrieved March 18th, 2007 from: http://swoogle.umbc.edu/index.php?option=com_swoogle_stats&Itemid=8

[9] "RSS - Crossing Into the Mainstream", Joshua Grossnickle, Yahoo! White Paper, Oct 2005

[10] Peterson, L., et al, "Experiences Building PlanetLab", Symposium on Operating System Design and Implementation (OSDI '06) - Nov 2006

[11] Cai, M., Frank, M., "RDFPeers: A scalable distributed RDF repository based on a structured peer-to-peer network", WWW conference, May 2004, NewYork, USA.

[12] Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. (2001). Design and Evaluation of a Wide-Area Event Notification Service. ACM Transactions on Computer Systems, 19(3).

[13] Weisstein, E. W. (2002). Multiset. MathWorld – A Wolfram Web Resource. Retrieved July 19, 2006, from http://mathworld.wolfram.com/Multiset.html.

[14] Jiang J. Conrath D., "Semantic Similarity based on corpus statistics and lexical taxonomy", Intl Conference on Research in Computational Linguistics, 1997.

[15] Loser, A., Naumann, F., Siberski, W., Nejdl, W., Thaden, U., "Semantic overlay clusters within super-peer networks", Int'l Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 2003

[16] Lynch, D., Keeney, J., Lewis, D., O'Sullivan, D., "A Proactive Approach to Semantically Oriented Service Discovery". Innovations in Web Infrastructure (IWI 2006). at World-Wide Web Conf., Edinburgh, Scotland. May 2006.

[17] Li, H., Jiang, G., "Semantic Message Oriented Middleware for Publish/Subscribe Networks", in proc of SPIE, Volume 5403, pp 124-133, 2004

[18] Muthusamy, V., Jacobsen, H.A., "Small–scale peer-to-peer publish/subscribe" in proc Workshop on Peer-to-Peer Knowledge Management, San Diego, USA, July 2005

[19] Rada R., Mili H., Bicknell E., Blettner M., "Development and application of a metric on semantic nets", IEEE Transactions on Systems, Man, and Cybernetics 19, 1989.

[20] Rutherford, M. J. (2004). "Siena Simplification Library Documentation 1.1.4." University of Colorado – Web Resource. Retrieved August 13, 2006, from http://serl.cs.colorado.edu/carzanig/siena/forwarding/ssimp/namespacesiena.html

[21] Segall, B. et al, "Content-Based Routing in Elvin4", In proc AUUG2K, Canberra 2000.

[22] Pietzuch, P., Bacon, J., "Peer-to-Peer Overlay Broker Networks in an Event-Based Middleware". Distributed Event-Based Systems (DEBS'03). At the ACM SIGMOD/PODS Conference, San Diego, California, June 2003

[23] Tempich, C., Staab, S., Wranik, A., "REMINDIN': semantic query routing in peer-to-peer networks based on social metaphors" WWW 2004, New York, USA, 2004.

[24] Terpstra, W.W., Behnel, S., Fiege, L., Zeidler, A., Buchmann, A.P., "A peer-to-peer approach to content-based publish/subscribe", in proc of DEBS 2003, ACM Press 2003

[25] Keeney, J., Lewis, D., O'Sullivan, D., "Benchmarking Knowledge-based Context Delivery Systems", in proc of ICAS 06, Silicon Valley, USA, July 19-21, 2006.

[26] Carzaniga, A., Wolf, A. L., "A Benchmark Suite for Distributed Publish/Subscribe Systems", Technical Report CU-CS-927-02, Dept. of Computer Science, University of Colorado. Apr 2002, http://serl.cs.colorado.edu/~carzanig/papers/

[27] Crowcroft, J., Bacon, J., Pietzuch, P., Coulouris, G., Naguib. H., "Channel Islands in a Reflective Ocean: Large-Scale Event Distribution in Heterogeneous Networks", IEEE Communications, Vol 40 No. 9, Sept 2002.

[28] Petrovic, M., Burceaa, I., Jacobsen, H.A. "S-ToPSS – a semantic publish/subscribe system" in proc VLDB, Berlin, Germany, September 2003

[29] Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., "Towards a Managed Extensible Control Plane for Knowledge-Based Networking", Distributed Systems: Operations and Management Large Scale Management, (DSOM 2006), at Manweek 2006, Dublin, Ireland, 23-25 October 2006

[30] Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Boran, A., Richardson, R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context", Network Operations and Management Symposium (NOMS 2006), Vancouver, Canada. 3-7 April 2006.

[31] Keeney, J., Lewis, D., O'Sullivan, D., "Ontological Semantics for Distributing Contextual Knowledge in Highly Distributed Autonomic Systems", Journal of Network and System Management, Vol 15, March 2007

[32] Muhl, G., Fiege, L., Gartner, F., Buchman, A., "Evaluating Advanced Routing Algorithms for Content-Based Publish/Subscribe Systems", Int'l Symp. On Modelling, Analysis and Simulation of Computer Telecommunications Systems (MASCOT'02), 2002

[33] Chand, R., Felber, P., "XNet: a Reliable Content Based Publish Subscribe System". SRDS 2004, Symp. on Reliable Distributed Systems, Florianopolis, Brazil, October 2004.

[34] Roblek, D., "Decentralized Discovery and Execution for Composite Semantic Web Services", M.Sc. Thesis, Computer Science, Trinity College Dublin, Ireland, December 2006.

[35] Cilia, M., Bornhövd, C., Buchmann, A. P., "CREAM: An Infrastructure for Distributed, Heterogeneous Event-Based Applications". CoopIS 2003, Catania, Sicily, Italy, Nov2003

[36] Carzaniga, A., Wolf, A. L., "Forwarding in a Content-Based Network" SIGCOMM'03, Kaelsruhe, Germany. August 2003