Setup and Maintenance of Overlay Networks for Multimedia Services in Mobile Environments

Eskindir Asmare, Stefan Schmid, Marcus Brunner

Network Laboratories, NEC Europe Ltd.
69115 Heidelberg, Germany
eskindir@kth.se,{schmid|brunner}@netlab.nec.de

Abstract. A Service-specific Overlay Network (SSON) is a virtualization concept proposed for customized media delivery in the Ambient Networks architecture [1]. The service specific media delivery network has to be constructed dynamically without prior knowledge of the underlying physical network. This process must consider unique properties, such as routing the media flow through strategic locations that provide special media processing capabilities (for example, media transcoding, caching and synchronization) inside the network. In today's dynamic and mobile network environments establishing an optimal SSON with a reasonable time and message/traffic complexity is challenging. This paper proposes a pattern-based methodology to establish the SSONs. This scheme enables setting up SSONs on demand without prior knowledge of the network topology and where the media processing capabilities are located in the underlying network. A new pattern referred to as *path-directed* search pattern is devised and applied to search for potential overlay nodes and to configure their media processing functions accordingly. The scheme is implemented on a pattern simulation tool and the result shows that SSONs of high quality can be built with a reasonable time and message/traffic complexity.

1 Introduction

A Service-Specific Overlay Network (SSON) is an overlay network solution developed for media delivery in particular. Media delivery in mobile networks has some additional challenges, since the nodes are not fixed, the topology is not stable, and processing of media data within the network makes sense as bandwidth on the wireless links is still limited and expensive.

Establishing service-specific overlay networks involves discovering network-side nodes that support the required media processing capabilities, deciding which nodes should be included in the overlay network and finally configuring the overlay nodes. Traditionally the list of nodes and their processing capabilities (services) are stored in a registry, which is queried during the overlay network setup to selected suitable nodes. However, the use of a registry for this information has serious limitations. One problem is that in dynamic networks, keeping an up-to-date registry is difficult and costly. Nodes with special services can join and leave the network with a high frequency.

Even if it was possible to keep an up-to-date registry of special service nodes, scalability will be a problem in big networks. The single point of failure that will be introduced by the registry and the administrative overhead are also problems worth mentioning. Automated node selection and configuration are also difficult tasks when establishing service-specific overlay networks across dynamic network infrastructures.

The objective of this work is to develop a concept and implementation of a provisioning and maintenance system for service-specific overlay networks. In this system, any node will be able to initiate a service discovery process, perform selection, and initiate configuration of the selected nodes. The proposed solution is a decentralized network management system that is based on a pattern-based paradigm [2]. By using patterns, the service discovery (i.e. the search for potential overlay nodes with the required media processing capabilities) and configuration of the selected overlay nodes is decentralized.

The problem is formulated as follows: Given any source node and a set of destinations, the system must setup an appropriate SSON based on an overlay network specification. The specification will state the number and type of media processing functions required between the source and each destination node as well as the preferred position of the functions with respect to the source and destination. The service provided by a node is referred to as a function. In this work we mainly focus on media processing functions like caching, transcoding, and synchronization. However, the setup and maintenance system works also for any other type of overlay network.

The paper is organized as follows: first we give some background on the overall SMART architecture for overlay networks and the pattern-based management paradigm; second, the related work section studies various related overlay deployment systems. Then, the overlay provisioning system is described, and its scalability properties are evaluated through simulation. Finally, the paper discusses the results and concludes.

2 Background

2.1 SMART Architecture for Mobile Overlays

In today's Internet technologies, there is no common control layer which controls and manages the different resources and technologies of heterogeneous networks. The Ambient Networks architecture [1] defines the Ambient Control Space (ACS), which is a common control layer to all resources and technologies in networks. One function of the ACS is Media Delivery. The Smart Multimedia Routing and Transport Architecture (SMART) [14] aspires for guiding media flow through specialized network nodes to make use of their ability to cache, transcoded or synchronize multimedia data as appropriate. To achieve this functionality, SMART makes use of overlay networks. It defines the Overlay Control Space (OCS), which will take care of selecting the necessary media processing nodes and establishing an end-to- end overlay network for media delivery. This overlay network is referred as Service Specific Overlay Network (SSON). Figure 1 below shows the draft architecture for SMART [14], and how the service-specific overlay networks are used.

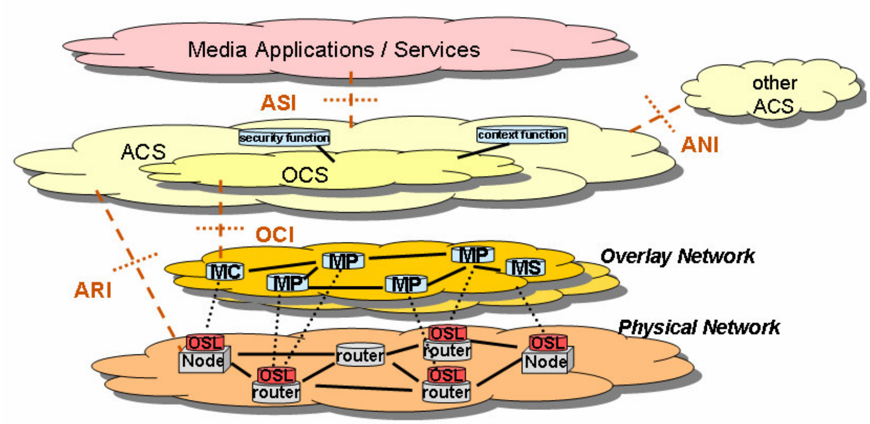
Some of the nodes in the physical network support special services which are used to enhance media delivery. These nodes are referred to as Media Ports (MP). The source of the media flow is referred as a Media Server (MS) and the receivers are referred as Media Clients (MC).

The SSON is an overlay network whose nodes are the Media Server, Media Ports and Media Clients. Specifically for mobile environments, certain network side multimedia processing functionality is helpful to get the best possible service to the mobile terminal. This paper proposes a scheme for dynamically establishing the media delivery SSONs by using a pattern-based network management paradigm [2].

2.2 Pattern-based Management

Pattern-based management [2] is an approach to distributed management that aims at over-

coming the limitations of centralized management. Its goal is to build scalable, robust and adaptable management systems.
Pattern-based applications map network-wide operations into local operations that will be performed by managed nodes. These operations are distributed using communications patterns, which create an execution graph. In the case of monitoring tasks, local operations typically include the collection of statistics and the incremental aggregation of the collected data. This aggregation is done in a parallel, asynchronous fashion across the network, whereby all nodes contribute to the computation. From the perspective of a network manager, a pattern provides the means to "diffuse" or spread the computational process over a large set of nodes.



OCS: Overlay Control Space
OCI: Overlay Control Interface
OSL: Overlay Support Layer
MS: MediaServer
MP: MediaPort
MC: MediaClient

Figure 1: The SMART SSON Architecture

The main benefits of pattern-based management are that it (i) separates the semantics of the task from the details of the distributed execution, (ii) enables building scalable management systems, (iii) facilitates management in dynamic environments, and (iv) does not require "a priori" knowledge of the network topology. Specifically the properties (iii) and (iv) are of major importance for mobile networks.

Previous work on pattern-based management [13] has identified a particularly useful pattern called the echo pattern. The defining characteristic of the echo pattern is its two-phase operation. In the expansion phase, the flow of control emanates from a node attached to the management station. A spanning tree is created to contact all managed nodes and request them to perform local management operations. After executing the local operation, the contraction phase starts. During this second phase, each node sends the result of the operations to its parent in the tree. The parent aggregates its result with its children's and forwards the aggregate to its parent. The global operation terminates when the root of the tree has received and aggregated the results from all its children.

The echo pattern dynamically adapts to changes in the network topology. It does not require global network information and thus scales well in very large networks like the Internet. Its time complexity increases linearly with the network diameter. The work presented in this paper improves the state-of-the-art of pattern-based management by proposing new patterns suitable for advanced overlay-network-management.

3 Related Work

Various structured peer-to-peer overlay network establishment schemes have been proposed recently. Also, several research activities have studied topology aware overlay construction. The overlay network proposed in this paper is different from the above networks in the way potential overlay node are discovered and the network graph is constructed. The network graph construction scheme is not the focus of this paper.

Touch et al. [8] define a Virtual Internet as a network of IP-tunneled links interconnecting virtual routers and virtual hosts providing full Internet capabilities at a virtual layer. One implementation of the Virtual Internet concept is the X-Bone [10]

Moreover, Bossardt et al. [9] have proposed a pattern based service deployment scheme for active networks, and Brunner et al. [15] have proposed a Virtual Active Network concept for dynamic service deployment, where the overlay setup is performed by a centralized management system in an operator environment.

The MBone [11] is a network layer overlay network running on top of the Internet. It is composed of networks that support multicasting. The purpose of this overlay network is to support audio/video multicasting.

Tapestry [4], Pastry [5], CAN (Content Addressable Network) [6] and Chord [7]: Although their implementation varies, these four application layer overlays pursue the same goal, namely to implement a Distributed Hash Table (DHT). The overlay is viewed as a distributed database. Because nodes are interconnected in a well defined manner the DHT based overlays are sometimes referred to as structured overlays.

4 Set-up and Maintenance of Overlay Networks for Multimedia Services in Mobile Environments

The setup and maintenance of overlay networks is typically not seen a big problem in fixed networks. However, in mobile and wireless network environments, topologies and network characteristics dynamically change all the time, making the setup and maintenance a tricky problem. Additionally, the concept of overlay networks is a means of implementing service-specific routing, caching, and adaptation functionality. The service-specific overlay networks are tailored towards the specific requirements of a media delivery service, and with it, the topology of the overlay network is also dependent on the service running within an overlay network.

In the following, we assume a specific type of overlay for the transport of multimedia data as described above in the SMART Architecture [14]. We assume that multimedia processing engines are specialized network elements (potentially running on dedicated hardware). The multimedia flows must be forced to pass through those nodes, potentially in a certain order, to improve the end-to-end media service. So, only a relatively small set of nodes in the network is capable of running these expensive multimedia processing functions. We call those nodes the potential overlay nodes.

The proposed system consists of the following functions: (1) detecting potential overlay nodes, (2) selecting suitable nodes, (3) setting up the overlay network, (4) adding client nodes to the overlay network, and (5) maintaining the overlay in order to adapt to the changing network context, and user or service requirements.

Even though we implemented all the functions based on the pattern-based approach, only the detection of potential overlay nodes and the clients joining an overlay are in the focus of this paper. The selection of suitable nodes is a matter of an optimization algorithm for service-specific requirements, such as using the cheapest overlay network, the overlay with the least number of hops, or an overlay network that is load-balanced, etc. The setup of the overlay is straight forward, once the nodes are known; they only need to be configured accordingly. The addressing and routing within the overlay are an orthogonal problem.

Concerning the detection of potential overlay nodes, we assume overlay nodes storing the required overlay node parameters locally in a standard format. For instance, a node stores the set of functions it can perform, for how many media flows this node can perform each function, and the cost for each function. (These are the primary parameters, depending on the service and future capabilities further parameters might be of relevance.)

Detecting potential overlay nodes involves probing each node for the required functionalities for the multimedia service, the available resources, and the cost. This means only nodes capable of hosting a virtual node with a certain function are found in that process. There are various ways to perform this search. Using a directory based approach, the node capabilities and resources would register with a well-known directory service. Clients/nodes that want to find a suitable resource or function will then query the directory service. The downside of this approach is that it is expensive to keep the directory up-to-date in the case of highly dynamic and/or mobile environments. For example, when the availability or other necessary information about a resource or function changes frequently, or when the node providing the service is mobile, the update messages needed to keep the directory service up-to-date would increases rapidly. Moreover, such directory-based services are typically not able to take the topological location of a resource or service into account, which prevents selecting resources based on the proximity of the end-to-end communication path.

We show that the pattern-based paradigm is a nice tool for discovering network-side functions/resources. A pattern will determine which nodes should be probed. The pattern will also initiate the probing process as well as the gathering of information. Different patterns can be used for this purpose. The most basic one is the echo pattern (see the section above for a detailed description). The echo pattern starts from the requesting node and then expands towards the leaves (or any defined boundary) of the network. On the way back, when the pattern contracts towards the initiating node, it aggregates the results of the discovery. This serves the purpose of both detection of the network topology as well as discovery of potential overlay nodes. But, irrespective of the number of destination nodes in the overlay network, the echo pattern will flood the probing request throughout the whole network. A Time-to-Live (TTL) can be used for the flooding in case only local resources/services are of interest.

A more suitable pattern is the path-directed search pattern proposed here. The basic idea of this pattern is to limit the scope of the search to a configurable area along the end-to-end path between the communicating peers. The search pattern uses a parameter that defines the "distance" (e.g. in number of hops, delay, or any other measure, etc.) from the routing path that should be searched. This distance is also referred to as '*sideway expansion*'. Depending on the type of resource or function that is searched, this parameter can be changed.

Our pattern assumes to know the source of a multimedia service and a number of destinations, or regions where one or more receivers of the multimedia services are located. The path-directed search pattern starts from the source node and expands along the end-to-end routing path towards the destination nodes with a sideway expansion of a given distance (e.g. based on the number of hops, delay, etc.). After visiting the nodes defined by the pattern scope, the pattern contracts towards the source node gathering the requested information (depending on the resources/service we are looking for). The sideway expansion parameter of the pattern controls the scope of the search and thus limits the number of nodes probed during the detection. Above all, it allows the discovery of network-side resources along a close approximation of the routing path.

Figure 2 shows a simple logical sequence of the pattern execution with sideway expansion of 1 hop on a small grid network. The pattern starts on the start node (Src). The state when the start node has sent an explorer to the next hop is called initialization state. Every hop along the routing path towards the destination (Dest) sends explorers to the next hop as well as sideways to all neighboring nodes (because the sideway expansion is greater than zero). The sideway expansion limits the reach of the explorers that are sent sideways. When the pattern reaches its sideway limit, it will start the contraction phase (back to the parent node along the routing path that started the sideway expansion). Nodes that have completed the pattern, but do not fulfill the requirements are colored blue (dark). Two of the nodes in the example are potential overlay nodes (PONs); this is indicated with the color pink (less dark). When parent nodes receive the results of their child nodes (contraction), they will start aggregating the information. Nodes in this state are shown with the color yellow (light gray). Finally, when the destination node (Dest) is reached, the pattern starts contraction from the destination towards the source.

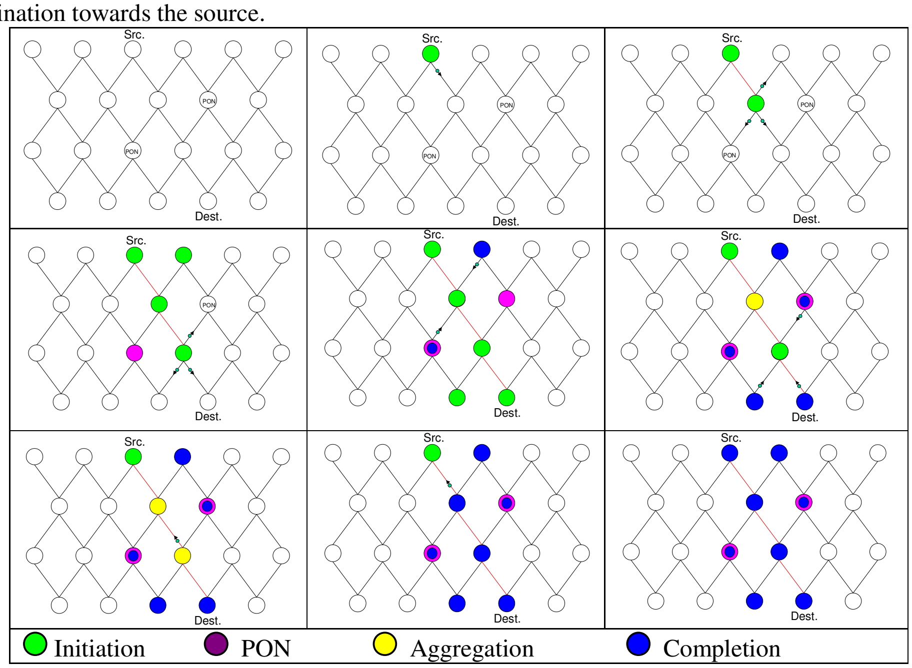


Figure 2: Path-directed Search Pattern Execution

Figure 3 depicts a scenario that illustrates the path-directed search scenario in a larger setting (as shown in the SIMPSON pattern simulator). The source node is node 110 and the destination nodes are nodes 209, 200, 11 and 20. The tested nodes are colored (dark). The potential overlay nodes are colored pink (light gray). The path-directed pattern assumes that the path from source towards the destination is known (according to traditional IP routing on all the nodes), but the multimedia processing functions are not necessarily directly on the path – they can be located anywhere 'along' the path. The path-directed search pattern is ideal to find appropriate network functions/resources (i.e. potential overlay nodes) along the end-to-end routing path from the source to the destination on-demand. I.e. whenever a particular network function/resource is required, the path-directed search pattern can be used to discover such functionality/resources.

The access of a new client to a given overlay for the first time is another problem, where patterns can help to reduce the amount of messages. Again, the echo pattern can be used with a certain diameter to search overlay nodes within a certain region around the client's network location. The path-directed search is helpful if the client knows the source, for example, the URL/DNS name of the source. In that case, the client can start searching towards the source with side way expansion in order to find another overlay node which might not be on the path, but nearby. If it finds an overlay of the desired type, a virtual link can be established and the client is then attached to the overlay. However, the point of attachment and the service received at the attachment point might be suboptimal. Therefore, the overlay topology and services should be regularly reconsidered. Overlay adaptation might also be triggered by the addition of a new client, but there we need to take into account the number of clients potentially attaching per time unit in order to find a scalable solution. This implies that the parameters are service specific and depend on the purpose and scale of the overlay network.

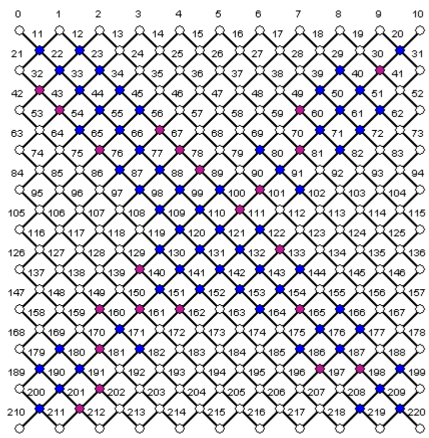


Figure 3: The path-directed search pattern with sideway expansion of 1 hop.

5 Evaluation of the Path Directed Search Pattern

The efficiency of the proposed pattern based scheme for the setup and adaptation of service-specific overlay networks are evaluated using the SIMple Pattern Simulator fOr Networks (SIMPSON) [3]. SIMPSON is a discrete event simulator used for implementing, testing, and evaluating pattern based systems.

Grid topologies with a node degree of 4 (except the nodes at the edges of the network) were used in the simulations. The protocol and operating system delay, propagation delay per hop, processing time on the nodes, and message size were taken to be 4ms, 5ms, 0.5s and 1024 bytes respectively. These values are justified to be reasonable estimates for the Internet in [12]. The link speed was taken as 10Mbs.

5.2 Single Destination

A set of simulations have been performed to observe the message complexity and the total number of nodes visited in a single destination scenario. The parameters used in the simulation were as follows: The simulation was done in the grid topology for different radiuses starting from radius = 5 hops to radius = 35 hops where the radius is the number of hops from the center of the grid topology to a corner. The number of nodes is given as $(r)^2 + (r+1)^2$ where r is the radius of the grid.

For each sideway value, the simulation was run 100 times with the middle node as the source and a randomly chosen node as a destination for each pass. The results of this simulation are illustrated below. Larger sideway expansion values were not used for smaller radiuses because they would visit all the nodes.

The percentage of visited nodes is computed out of the total number of nodes. The message complexity is plotted as compared to a pattern which will make a full search, i.e., probe all nodes. The echo pattern was used for the full search.

Both graphs show that the path-directed search pattern uses around 1 to 10 percent of the number of messages and visits only about 1-10 percent of the nodes. The impact of this enhancement for the quality of the overlay, by not visiting all the nodes, is shown below.

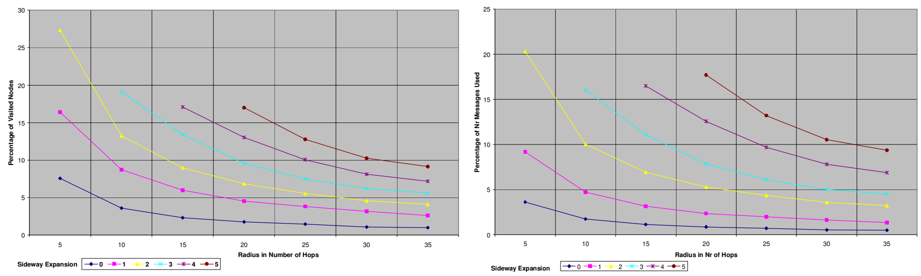


Figure 4: Single Destination: (a) Percentage of Nodes Visited; (b) Percentage of Messages Used Compared to Full Search

5.3 Multiple Destinations

This simulation was performed to study the behavior of the pattern in a multi destination scenario. Again a grid topology with radius of 30 hops (1861 nodes) was used and the number of destinations was varied from 5 to 25 in steps of 5. For each number of destinations, 100 sets of randomly selected destination positions were used and for each position the simulation was run for sideway expansions of 0 to 5 inclusive. The results of the simulation are shown below.

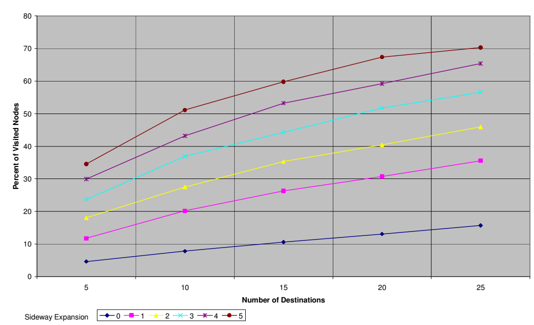


Figure 5: Multiple Destinations: Percent of Visited Nodes

5.4 Quality of the Service Specific Overlay Network

This set of simulations has been performed in order to evaluate the performance of the pattern for service-specific overlay network construction. When selecting overlay nodes, we consider the price of using the required function on that node and the distance of that node from the source and/or the destination. As described earlier the requirement specifies the type and number of functions needed in the path from source-to-destination as well as their preferred position (as near-to-source, middle, near-to-destination). The optimization is performed by using a weighted cost function. The following cost function was used in the simulations:

$$Cost = 0.5 \times price + (0.5 \times src \times dist_{src} + 0.5 \times dest \times dist_{dest} + 0.25 \times med \times dist_{src} + 0.25 \times med \times dist_{dest})$$
$$\times \max(price) / \max(dist);$$

whereby *src* = 0.9, *dest* = 0.1, *med* = 0, when the requirement is 'near-to-source'; *src* = 0.1, *dest* = 0.9, *med* = 0, when the requirement is 'near-to-destination'; and *src* = 0, *dest* = 0, *med* = 0.25, when the requirement is 'in-the-middle'.

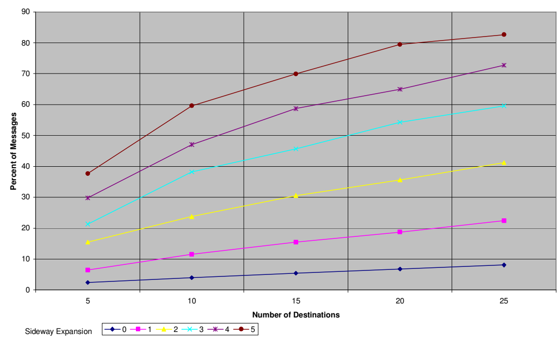


Figure 6: Multiple Destinations: Percent of Messages Used Compared to Full Search

To make the result of this cost function meaningful, the distance is linked with the price. We assume that traversing the maximum distance will cost us as much as paying the maximum price. The maximum distance max(*dist*) is the maximum of the maximum distance from the source and the maximum distance from the destination of potential overlay nodes. It is computed after the pattern completed the search. The price of each function ranges from 1 to 100 price units.

The above cost function gives the cost of each detected potential overlay node and the nodes with the least cost will be selected for each function. As a measure of performance, in constructing a service-specific overlay network using patterns, the quality of an overlay node is defined as follows: *Quality* = max*(price)/Cost*. Because we have set the maximum price to be 100, our formula reduces to *Quality = 100/Cost*.

The cost is a superposition of the price of using the node and the node's distance from the source and destination computed with weight based on the requirement. If a node has smaller cost then it will have higher quality. Also we define the quality of an overlay network as the average of the quality of its overlay nodes. We will later use this definition to interpret the result of the simulation. The following overlay specification was used in the simulation:

> *An overlay network of the least price with one transcoder (f1) preferably near to the source, two cache machines (f2) preferably in the middle and another transcoder (f3) preferably near to the destination.*

A grid topology of radius 30 hops (1861 nodes) was used. The source node was the center node and the destinations were the 4 corner nodes. The potential overlay node density was varied from 25 nodes to 200 nodes in steps of 25. For each density, a random placement with randomly chosen prices for each node was generated. Then the placement was randomly shuffled 100 times and for each placement the simulation was run for the sideway values of 0 to 5 inclusive.

The simulation was also run for sideway value of 31 to determine the quality for full coverage In this case only the cost of the selected nodes was registered. The other parameters are taken from a simulation of the echo pattern for the same radius. Except for technical simplicity for batch processing, the number of potential nodes detected in this case is the same as running the echo pattern and it could have been done that way also.

The quality of the overlay network constructed using the path directed pattern as compared to the one constructed using a full search is plotted below. The quantities used for full search, for comparison, are obtained by using the echo pattern to probe all nodes in search for potential overlay nodes. The quantities shown are hence the best that can be achieved both number of message and time complexity wise as far as full coverage is concerned. This is because the echo pattern is the efficient way both message and time complexity wise for full coverage [2]. The plot of the quality of the service-specific overlay network is shown below in Figure 7.

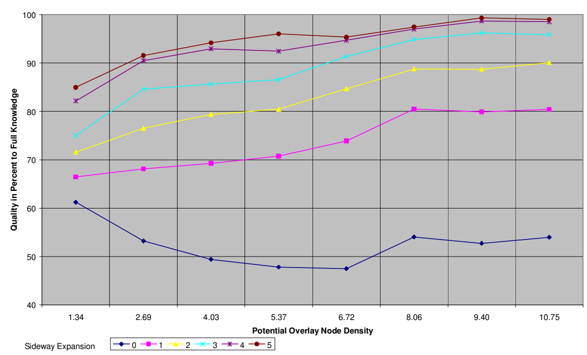


Figure 7: Quality of SSONs

6 Discussion

The evaluation of the path-directed search pattern with respect to the number of nodes visited, as well as the message and time complexity gives an insight of the performance of the path-directed search pattern. We measured the percentage of the number of nodes visited in the single-destination pattern as compared to a pattern, which makes a full search. The regressions of the curves show that the percentages of the number of nodes are proportional to the inverse of the radius of the network with a correlation coefficient of greater than 0.99. This evaluation depicts one of the main advantages of the path-directed pattern. Because the search is made only along the end-to-end routing path, the number of visited nodes does not considerably increase with the size of the network.

It is worth noting that the perfect inverse relation is in part accounted to the characteristics of the simulated networks. The number of nodes in the grid network is $(r)^2 + (r+1)^2$ where r is the radius of the network, and in a pattern which makes a full search, all the nodes will be visited. Hence the number of visited nodes in a full search is $O(r^2)$.

On the other hand, for a specified sideway expansion value, the number of nodes visited in the path-directed pattern is only directly proportional to the radius of the network, i.e., $O(r)$. The percentage of number of nodes visited in the single destination pattern as compared to a pattern which will make a full search will then be inversely proportional to the radius of the network.

The message complexity has the same property as the number of visited nodes. Because the message is generated by each visited node, the lesser the number of nodes visited the lesser the number of messages exchanged.

The time complexity of the pattern (not shown in a graph) increases linearly with the radius of the network. The time complexity increases only slightly with the increase in sideway expansion. This behavior is due to the parallel processing nature of the pattern. It has been proved analytically, in [12], that the echo pattern has a time complexity which is directly proportional to the radius of the network. Time-complexity-wise, the path-directed search pattern obeys the same rule.

The multiple-destinations pattern evaluation shows that using higher sideway expansion values results in a considerably higher number of visited nodes. For instance, when the number of destinations is greater than 20, more than half of the nodes in the network will be visited for sideway expansions of greater than 4 hops.

The quality of the service-specific overlay network with respect to the stretch introduced by the overlay and the price incurred are studied in a multiple destination scenario. The result shows that even with densities of potential overlay nodes as small as 1%, the quality of the network constructed using the pattern-based system can be more than 50% of the quality achievable if the overlay was constructed based on full information about the network.

The quality with larger densities of potential overlay nodes is comparable with the maximum possible quality. For instance, with 95% confidence, the average quality achieved with a sideway expansion of 2 hops and a potential overlay node density of greater than 7% is greater than 80%. The average quality will be even greater than 90% for this density with sideway expansion greater than 2 hops.

However, choosing the sideway expansion without prior knowledge of the density is not possible, but can be learnt while using the pattern several times from the same location (source). Given a certain estimated density, and an optimality goal, the algorithm can derive an approximation for the sideway expansion using the numbers in Figure 7. Note that the numbers are based on a certain topology and might not be easily extendable to any topology.

The issue of load balancing has not been further studied in this paper, but the choice of overlay nodes given a set of potential overlay nodes is an algorithm, where the load could be taken into account. So far we take only the availability of the resource and the price into account. Additionally, we can model the load by a flexible load-based pricing model on the nodes, and then our algorithm would immediately converge to a load balanced network.

7 Conclusion

The analysis of the pattern–based management approach and the evaluation of the path-directed search pattern in particular show that one can construct a service-specific overlay network of high quality on demand, with only a small traffic/message overhead. Its ability to construct the overlay network without prior knowledge of the existing physical network topology and the available network functions/resources makes the pattern-based approach most valuable.

There are still some issues to be addressed in using the path-directed pattern for service-specific overlay network construction. One open issue is security in the search, configuration, and reconfiguration process. Introducing authentication and authorization mechanisms is important to prevent an unauthorized node from initiating a search pattern, sending false configuration information, or requesting a false reconfiguration.

Additional further work includes the use of the approach across administrative boundaries, the porting of the pattern to a real pattern system like Weaver [13], and studying the possibility to run the optimization algorithm decentralized including appropriate management patterns.

Acknowledgment

This work is a product of the *Ambient Networks* project supported in part by the European Commission under its *Sixth Framework Programme*. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the *Ambient Networks* project or the European Commission.

References

[1] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, H. Karl, *"Ambient Networks: An Architecture for Communication Networks Beyond 3G"*, IEEE Wireless Communications, vol. 11, no. 2, Apr 2004, pp 14- 22.

[2] K. Lim, C. Adam, R. Stadler, *"Decentralizing Network Management",* IEEE electronic Transactions on Network and Service Management (eTNSM), Vol 1(2), 2004.

[3] "*SIMPSON – a SIMple Pattern Simulator fOr Networks*", http://comet.ctr.columbia.edu/adm/software.htm.

[4] B. Y. Zhao, LinHuang, Jeremy Stribling, S. C. Rhea, A. D. Joseph, J. D. Kubiatowicz, *"Tapestry: A Resilient Global-scale Overlay for Service Deployment"*, IEEE Journal on Selected Areas in Communications, 2004.

[5] Rowstron, P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems", International Conference on Distributed Systems Platforms(Middleware), pages 329- 350, Heidelberg, Germany, Nov 2001.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, *"A scalable content addressable network"*, in Proceedings of ACM SIGCOMM, 2001.

[7] I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, *"Chord: A scalable peer-to-peer lookup service for internet applications",* in Proceedings of SIGCOMM, 2001.

[8] J. D. Touch, Y. Wang, L. Eggert, *"Virtual Internets"*, USC/Information Science Institute, Jul. 1, 2002.

[9] M. Bossardt, A. Mühlemann, R. Zürcher, B. Plattner, *"Pattern Based Service Deployment for Active Networks",* in Proceedings of the Second International Workshop on Active Network Technologies and Applications, ANTA, Osaka, Japan, May 2003.

[10] J. Touch, S. Hotz, *"The X-Bone"*, In Proceedings of the third Global lnternet Mini-Conference, Sydney, Australia, Nov. 1998, pp. 75-83.

[11] H. Eriksson, *"MBONE: The Multicast Backbone"*, ACM Communications of the ACM 37, Aug. 1994, pp. 54-60.

[12] K. S. Lim, R. Stadler, *"A Navigation Pattern for Scalable Internet Management",* Seventh IFIP/IEEE International Symposium on Integrated Network Management (IM 2001), Seattle, USA, May 2001, pp. 405-420.

[13] K.S. Lim and R. Stadler; "Weaver: realizing a scalable management paradigm on commodity routers", 8th IFIP/IEEE International Symposium on Integrated Network Management, 24-28 March 2003, Colorado Springs, Colorado, March 2003, pp:409 – 424.

[14] S. Schmid, F. Hartung, M. Kampmann, St. Herborn, "SMART: Intelligent Multimedia Routing and Adaptation based on Service Specific Overlay Networks." To appear in Proc. of Eurescom Summit 2005, Heidelberg, Germany, 27–29 April, 2005.

[15] M. Brunner, B. Plattner, R. Stadler, "Service Creation and Management in Active Telecom Environments", Communications of the ACM, March 2001.