Evaluating Advanced Routing Algorithms for Content-Based Publish/Subscribe Systems

Gero Mühl Ludger Fiege Felix C. Gärtner Alejandro Buchmann
{gmuehl,fiege}@gkec.tu-darmstadt.de
{felix,buchmann}@dvs1.informatik.tu-darmstadt.de
Department of Computer Science
Darmstadt University of Technology
D-64283 Darmstadt, Germany

Abstract

We present an evaluation of advanced routing algorithms for content-based publish/subscribe systems that focuses on the inherent characteristics of routing algorithms (routing table sizes and filter forwarding overhead) instead of system-specific parameters (CPU load etc.). The evaluation is based on a working prototype instead of simulations and compares several routing algorithms to each other. Moreover, the effects of locality among the interests of the consumers are investigated. The results offer new insights into the behavior of content-based routing algorithms: Firstly, advanced routing algorithms can be considered mandatory *in large-scale publish/subscribe systems. Secondly, the use of advertisements considerably improves the scalability. Thirdly, advanced routing algorithms operate efficiently in more dynamic environments than was previously thought. Finally, the good behavior of the algorithms even improves if the interests of the consumers are not evenly distributed, which can be expected in practice.*

1. Introduction

A *publish/subscribe system* consists of a set of clients that exchange notifications indirectly with the help of a notification service that is interposed between them. *Clients* are producers, consumers, or both. *Producers* publish notifications (such as the latest sports results), and *consumers* subscribe to notifications (e.g., all NBA playoff results) by issuing *subscriptions*. Consumers can have multiple active subscriptions, and after a client has issued a subscription the notification service delivers all future matching notifications that are published by any producer until the client cancels the respective subscription. The benefits of publish/subscribe systems (loose coupling and asynchronous communication) are well-known [4] and make them first choice for implementing information-driven applications.

One of the ways to approach publish/subscribe systems is to use the concept of *content-based filtering* [9]. This allows subscriptions, which are boolean *filters*, to evaluate filtering predicates over the whole content of a notification, and so it provides a more powerful and flexible notification selection than other approaches (e.g., channel- or subject-based notification services) for which the content of a notification is opaque. Unfortunately, content-based filtering is not only more flexible, but also more complex to realize. It cannot be easily mapped to low-level communication mechanisms like IP-Multicast, and centralized solutions of content-based notification services do not scale. Hence, a scalable distributed notification infrastructure is needed.

Besides centralized implementations, a common way to implement distributed notification services is to use a set of cooperating *event brokers* that are arranged in a distributed topology. Each broker manages a set of *local clients* and forwards incoming notifications to its neighbors. Here, the technique of *flooding* is the simplest possible implementation: Every broker simply forwards a notification that is published by one of its local clients to all of its neighbors and if a broker receives a notification from a neighbor, it simply forwards it to all other neighbors. In consequence, each published notification is eventually processed by every broker. Hence, routing is simple, but obviously a lot of unnecessary messages (which have no consumers at the other end) may be exchanged among brokers.

An alternative is to use *content-based routing*. Here, each broker manages a *routing table* which consists of a set of *routing entries* that are comprised of a filter and a *destination*. A notification is forwarded to a destination if there is a routing entry regarding this destination whose filter matches the given notification. We distinguish among *local* routing entries which belong to clients of a broker and *remote* routing entries which belong to neighbor brokers. The routing tables are maintained by forwarding new and canceled subscriptions through the broker network.

Here, the simplest approach is to assume that each broker has global knowledge about all active subscriptions. This approach is called *simple routing*. In this case, a routing entry for every active subscription is incorporated into the routing tables of all brokers. To achieve this, new or canceled subscriptions are flooded into to broker network. Simple routing minimizes the notification traffic in the system but it enforces that *all* brokers have knowledge about *all* active subscriptions, which obviously is an undesirable feature in large systems. Therefore, there exists a tradeoff between the network resource waste caused by flooding and the maintenance overhead introduced by filtering.

In previous work, a set of advanced routing algorithms has been developed for content-based routing including identity-based, covering-based, and merging-based routing.

Identity-based routing [10] relies on identity-tests among subscriptions: A new subscription or unsubscription is not forwarded to a neighbor if previously a subscription that is identical to the former, i.e., which matches the same notifications, has been forwarded to that neighbor. This avoids routing entries for the same destination with identical filters and unnecessary forwarding of subscriptions.

Covering-based routing [3, 5] relies on covering-tests among subscriptions. A new subscription or unsubscription is not forwarded to a neighbor if previously a subscription that covers the former, i.e., which matches a superset of notifications, has been forwarded to that neighbor. Unfortunately, this implies that if an unsubscription is forwarded to a neighbor, it might be necessary to forward also some subscriptions to that neighbor that have been covered.

Merging-based routing [9, 10] goes even further. In this case, each broker can merge exiting routing entries to a broader, i.e., covering, subscription that is forwarded to some of its neighbors. While the idea of merging-based routing routing enables a large set of possible implementations, we have implemented a merging-based algorithm on top of covering-based routing. In this algorithm, each broker can replace routing entries that refer to the same destination by a single one whose filter covers all filters of the merged routing entries. The merged entries are then removed from the routing table and the generated merger is added instead and forwarded like a normal subscription. Similar, the merger is removed if there is an unsubscription for a constituting part of the merger or if a subscription arrives that covers a part or the whole merger.

Advertisements can be used as additional mechanism to further optimize content-based routing [3]. They are filters that are issued (and canceled) by producers to indicate (and revoke) their intention to publish certain kinds of notifications. For this purpose a second routing table is managed by every broker. This *advertisement-based* routing table is maintained by the same algorithms as the *subscription-based* routing table, i.e., by forwarding new and canceled advertisements through the broker network. While the subscription-based routing tables are used to route notifications from producers to consumers, the advertisement-based routing tables are used route subscriptions from consumers to producers. If advertisements are used, it is sufficient to forward (un)subscriptions only into those subnets in which a producer has issued an overlapping advertisements, i.e., where matching notifications can be produced. Moreover, if a new advertisement is issued, overlapping subscriptions are forwarded appropriately. Similarly, if an advertisement is revoked, remote subscriptions that can no longer be serviced are dropped. Advertisements can be combined with all routing algorithms discussed above.

All these ideas potentially reduce the size of the routing tables and they also ensure that new or canceled subscriptions are not forwarded unnecessarily. A common concern regarding the use of more complex routing strategies is that it is unclear how they perform in comparison with each other and in what settings which schemas are more preferable than others. These concerns have been addressed by a number of performance analyses of content-based routing algorithms. In the context of the SIENA system, Carzaniga, Rosenblum, and Wolf [3, 5] presented performance results of covering-based routing within a simulation framework. Unfortunately, they did not compare the results with other algorithms in the same framework and it is not easy to interpret their results because the setup of main parameters that influence the results are not described.

The current implementation of JEDI [6] exploits the well-known hierarchical version of covering-based routing [3]. Here, a notification is always propagated to a root broker regardless of the interests of the consumers. Bricconi *et al.* [2] describe an improved version that uses advertisements and present simulations based on an analytical model to compare the original with the improved version. The essential result of their work is that advertisements reduce the load that is induced on the brokers.

Work in the context of Gryphon has investigated several parameters of the implemented simple routing schema without advertisements. Banavar *et al.* [1] investigate the load caused at the individual brokers. The results presented show that flooding overloads at the same publishing rate regardless of the percentage of matches or the number of active subscriptions. Filtering-based routing on the other hand can handle much higher publication rates if subscriptions are highly selective or highly local which can be expected in large scale publish/subscribe systems. Other work in the Gryphon context was done by Opyrchal *et al.* [11] in which the authors focused on bandwidth utilization by comparing flooding to four multicast-enabled routing algorithms and ideal multicast. They state that filtering-based routing is superior to flooding under conditions of high selectivity and high locality of subscriptions. Nevertheless, it can be expected that their results are still too pessimistic because their work depends on simple routing, i.e., the routing algorithm does not exploit identity, covering, or merging.

Our work is distinguished from the above work in the combination of two factors:

1. We do *not* investigate processor load or network bandwidth utilization, but rather focus on the *fundamental characteristics of the routing algorithms*, namely their impact in terms of routing table size and notification forwarding overhead. The routing table size gives an indication of the *space complexity* of the algorithm while the notification forwarding overhead can be viewed as an approximation of the *message complexity*. We feel that new algorithms should be analyzed first with respect to inherent tradeoffs and parameter relations to gain more insight into their fundamental properties before measuring "real" run-time parameters like used network capacity.

2. We do *not* perform the above investigations in an abstract or analytic simulation environment but measure the behavior of the algorithms within an *implemented prototype system* called REBECA [7, 8, 9, 10] which incorporates all the mentioned routing strategies, including flooding, simple routing, identity-based, covering-based, and merging-based routing with or without advertisements.

Our results contribute four novel insights into the behavior of content-based routing algorithms:

1. Advanced routing algorithms can be considered *mandatory* in large-scale publish/subscribe systems.

2. Advertisements improves the scalability considerably, especially for systems with many brokers. In our settings, the improvement was at least 50%.

3. Simple routing requires that the rate with that subscriptions are issued and revoked, decreases with an increasing number of subscriptions to make filtering pay off in terms of saved messages. In contrast to that, advanced routing algorithms can "save more" if the dynamics of the system increases. This shows that, in contrast to current belief, increasing client dynamics does not automatically increase the message overhead of filtering algorithms.

4. The above findings hold if subscriptions are uniformly distributed. If this is not the case (i.e., if *locality* among the interests of the consumers exists and increases), the performance improves further.

The remainder of this paper is organized as follows: Section 2 explains the setup of the experiments. Section 3 and 4 discuss results related to the routing table sizes and the filter forwarding overhead, respectively. In Section 5 the effect of locality among the interests of the clients is investigated.

2. Setup of the Experiments

This section describes the general setup of the experiments which were performed in the context of a stock trading information system where clients can subscribe and unsubscribe to certain stock information and a central publisher of this information (e.g., a stock exchange) exists. Table 1 lists all symbols that are used throughout this paper. Besides the used routing algorithm, the main parameters that influence the results are the characteristics of the broker topology, the consumers, and the producers. To investigate the effects of *all* involved parameters goes beyond the scope of this paper. Therefore, the experiments assume a simple and meaningful scenario in which some of these parameters remain constant (see Table 2). In the following, the setup with respect to the mentioned parameters is described in more detail.

E	The set of all edges connecting brokers
m	The number of stocks
N_B	The set of neighbors of a broker B
S	The source of notifications
$\Sigma\|R\|$	Sum of all remote routing table entries
V	The set of all brokers
x	The number of active subscriptions

Table 1. Symbols used in this paper.

Number of consumers per local broker	$1 - 200$
Number of subscriptions per consumer	10
Number of stocks	1000
Number of notification sources	1
Number of event routers	40
Number of local event brokers	67
Number of neighbor broker	fix
Number of hierarchy Levels	5
Distribution of clients to brokers	fix
Distribution of stocks to clients	random
Degree of locality	none

Table 2. Fixed and varied parameters.

Broker Topology. The broker topology has a major impact on any experiment. Its main parameters are: the number of brokers, the number of neighbor brokers (which may

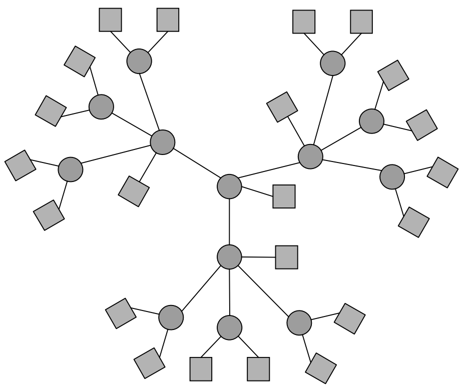

Figure 1. Broker topology with 4 levels.

be constant or vary), and the diameter of the network (which is the longest path connecting two arbitrary brokers).

In line with most previous work, this paper concentrates on a hierarchical and symmetrical, i.e, tree-like, topology with 5 levels of brokers. The use of a symmetrical topology facilitates the interpretation of the findings, and the hierarchical structure is justified by the hierarchical structure of real networks, like the Internet. Due to implementation reasons we distinguish among brokers which have local clients and those that have not. In the following the former are called *local brokers* while the latter are called *routers*. A local broker is connected to exactly one router. In the given topology, starting from a single router, called the *root router*, all routers except the leaves are connected to 3 subordinate routers. To each non-leaf router a single local broker is connected, while to each leaf router two local brokers are connected. Therefore, the topology consists of 40 routers and 67 local brokers. In Fig. 1 a topology of the same type with only 4 levels is shown. Note that the circles refer to routers while the squares refer to local brokers.

Characteristics of the consumers. The characteristics of its consumers has a large impact on a publish/subscribe system. The main parameters are: the number of consumers, the number, type, and distribution of the subscriptions, the assignment of the subscriptions to the consumers (e.g., locality of interests), the assignment of the consumers to the local brokers, and the rate of subscribing and unsubscribing.

In our experiments the consumers are equally distributed among the local brokers. We distinguish between *active* clients that have issued their subscriptions, and *inactive* clients that have not. If the size of the routing tables is investigated, all clients are active, while when investigating the filter forwarding overhead on average only half of them are active; this has to be kept in mind when interpreting the figures. In the latter case, a large number of iterations is carried out where, each time an arbitrary client is picked, its state is toggled from active to inactive or vice versa.

Each consumer has 10 random but distinct subscriptions. As the basis for the investigations quote subscriptions, which are borrowed from the implemented stock trading application, have been used. A *quote subscription* matches all quotes of a specific stock. For simplicity it is assumed that the number of different stocks m equals 1000. This choice may seem to be arbitrary, but indeed, the used subscriptions cover a wider range of subscriptions than it seems at first glance. For example, if m is different, findings can be derived by simple scaling. Moreover, if subscriptions have more than one attribute, results can be derived by assuming that m equals the product of the number of distinct values each attribute can have. For example, if subscriptions with three attributes are used where each attribute can have 10 possible values, this would lead to the same results. In order to compare covering-based with merging based-routing additionally *quote interval subscription* are applied which are slightly more complex. This type of subscription matches all quotes of a certain stock whose price is within a certain range, e.g., $\$10 - \$20$. The rationale for this distinction will be described later.

Characteristics of the producers. The main characteristics of a set of producers are: the absolute number of producers, the number, type and distribution of the advertisements, the assignment of the advertisements to the producers, the assignment of the producers to the local brokers, and the rate of advertising and unadvertising.

In our experiments the root router serves as a single source of notifications. At system start-up this ‘producer’ issues an advertisement that is never revoked and overlaps with all possible subscriptions. This simple scenario is sufficient to deduce the effects of static advertisements: Facts about the advertisement routing tables that arise if more than one producer is present can be inferred from the experiments because the advertisement routing tables are managed by the same algorithms as subscription routing tables. Of course, the factor by which the use of advertisements reduce the size of the subscription routing tables in such a scenario depends on the characteristics of the advertisements, e.g., their number and the degree of overlap. In the worst case, if at each local broker a producer issues an advertisement that overlaps with all subscriptions, advertisements would be completely useless. Fortunately, the probability for this case is very low. Instead, it can be expected that advertisements are only partly overlapping.

Scenarios with dynamic advertisements are out of the scope of this paper and are left for future work. Indeed, the filter forwarding overhead induced by dynamic advertisements is difficult to predict and depends, for example, on the rate of advertising and unadvertising, but it is reasonable to assume that compared to subscriptions this rate will be rather low.

3. Routing Tables Sizes

We now focus on the first key characteristic of any routing algorithm, namely the evolvement of the size of the routing tables with respect to the number of active subscriptions. This is an indication of the *space complexity* of the algorithms in the "size" of the system. We only count the remote routing entries here to concentrate on the behavior of filtering-based routing algorithms. The number of local routing entries equals the number of active subscriptions; they also exist if flooding is performed. In the figures, the abbreviation "RTS" stands for routing table size.

Simple Routing. If simple routing without advertisements is used, the size of a routing table equals the number of active subscriptions x. The characteristics of the subscriptions have no impact on the size of the routing tables. The sum of all remote routing table entries $\Sigma|R|$ is given by $(|V|-1)\cdot x$. Therefore, $\Sigma|R|$ grows linearly with both the number of active subscriptions and the number of brokers (see Fig. 2). This is because each subscription is stored on every broker. This indicates that the space requirement for simple routing generally grows unboundedly.

Simple Routing with Advertisements. The use of advertisements significantly reduces the size of the routing tables because in this case a subscription is only stored on brokers that are on a path from the respective consumer to the sources of the notifications of interest. This means that a local broker can only have remote routing entries in the subscription-based routing table if at least one of its clients has issued an advertisement. Therefore, in our scenario local brokers have no remote routing entries at all, while the size of the routing table of a router depends on the level to which it belongs (see Tab. 3). The root router might become overloaded first, because it has the largest routing table whose size is equal to the number of active subscriptions. The routing tables of the brokers on the lower levels are much smaller and their size corresponds to the number of subscriptions that are active in the respective subnet. For the investigated scenario the sum of all remote routing entries $\Sigma|R|$ is $P_{avg}\cdot x$ where P_{avg} is the average path length from a consumer to the notification source, i.e., the root router. This means that the routing tables still grow linearly with the number of active subscriptions (see Fig. 2), but logarithmically instead of linearly in the number of brokers. The average path length depends on the topology and the positioning of the notification source. For the given topology with the source at the root router P_{avg} is minimal and equals 3.73. Therefore, the use of advertisements reduces $\Sigma|R|$ by a factor of $(|V|-1)/P_{avg}=28.4$ which is independent of the number of active subscriptions (see Fig. 4). If the source is positioned at a local broker

Level	Number of routers	Size of routing table
1	1	$1.00\cdot x$
2	3	$0.33\cdot x$
3	9	$0.10\cdot x$
4	27	$0.03\cdot x$

Table 3. RTS for simple routing (with adv.).

of a leaf router (which is the worst case), P_{avg} would be equal to 6.78. Note that if the hierarchy has more levels, the factor by which advertisements reduce the routing table sizes would drastically increase because the number of brokers grows exponential in the number of levels. Therefore, advertisements increase the scalability of large publish/subscribe systems.

Routing based on Identity. Due to the identity-based routing algorithm routing table entries that have identical filters and destinations are diminished. In consequence, the size of the routing table of a broker B is limited by the number of stocks multiplied by the number of its neighbors, i.e., $m\cdot|N_B|$. Since no advertisements are used, this limit is approached for all brokers at equal speed. For relatively small numbers of active subscriptions the sum of all remote routing tables entries grows similar as in the case of simple routing, i.e., nearly linear, while for only slightly larger numbers the gradient monotonically decreases (Fig. 2). This is due to the fact that for larger numbers of subscriptions the probability increases that subscriptions are identical. In the investigated scenario $\Sigma|R|$ converges to $212,000$ for large numbers of subscriptions. More generally, if G is an arbitrary acyclic graph $\Sigma|R|$ converges to $\Sigma|R|=2\cdot m\cdot(|V|-1)$. The important thing to note here is that routing table sizes are bounded. This is a clear indication of scalability.

Routing based on Identity with Advertisements. The use of advertisements offers similar benefits for identity-based routing as in simple routing, e.g., only the routers have remote routing entries. For all routers B except the source the size of a routing table is limited by $m\cdot(|N_B|-1)$ while for the source it is limited by $m\cdot|N_B|$. These limits are reached for a small number of subscriptions for the topmost router, while for routers on the lower levels a larger number is necessary to reach saturation.

For relatively small numbers of subscriptions the sum of all remote routing entries approximates that of simple routing with advertisements, but the gradient decreases rapidly, and the sum finally converges to $106,000$. Indeed, if G is an arbitrary acyclic graph the sum of all remote routing entries is limited by $\Sigma|R|=m\cdot(|V|-1)$. Therefore, the use of advertisements halves the limit without advertisements. Interestingly, this limit is not dependent on the positioning of the source. The factor by which the use of advertisements reduces $\Sigma|R|$ depends on the number of issued subscriptions (see Fig. 4). The factor is near to $(|V|-1)/P_{avg} = 28.4$ for very small numbers of subscriptions but quickly decreases for larger numbers of subscriptions. Finally, it converges to 2 meaning a minimum improvement of 50%.

Routing based on Covering. Covering-based routing behaves exactly like identity-based routing if quote subscriptions are used. This is because a quote subscription covers another quote subscription iff they are identical, i.e., refer to the same stock. Therefore, the following type of quote interval subscriptions has been used to investigate the behavior of covering-based routing: $symbol = stock \wedge price \in [50 - t_1, 50 + t_2]$ with $t_1, t_2 \in [0, 50]$ randomly selected. This choice seems to be justified because in practice it can be expected that most quote interval subscriptions that refer to the same stock share a common price which is near to the current price of the stock. For two given quote interval subscriptions the probability that one of them covers the other is 50%. Of course, the benefits of covering-based routing would be less evident if this probability was lower.

Interestingly, the size of the routing tables and the sum thereof are larger and converge more slowly for covering-based routing with quote interval subscriptions than for identity-based routing with quote subscriptions (see Fig. 3). This is because, opposed to quote subscriptions, several quote interval subscriptions can exist that refer to the same stock and that do not cover each other. On the other hand, identity-based routing would nearly degrade to simple routing if it was applied to quote interval subscriptions. Therefore, the use of covering inherently improves the scalability if these more complex subscriptions are applied.

Routing based on Covering with Advertisements. Compared to identity-based routing with advertisements, the size of the routing tables and the sum thereof are larger and converge more slowly (see Fig. 3). The reason for this was presented in the preceding subsection. Interestingly, the difference is not as great as without advertisements. The curve of the factor by which advertisements reduce the sum of all remote routing entries is depicted in Fig. 4. It looks similar to that of identity-based routing but declines more slowly and therefore, the use of advertisements offers even more advantages if covering-based routing is used.

Routing based on Merging. The implemented merging-based routing algorithm leads to a routing table that has at most one remote entry for each neighbor if quote subscriptions are used. Hence, the sum of all remote routing entries $\Sigma|R|$ is limited by $2 \cdot (|V| - 1)$. Due to this fact, and in order to compare it with covering-based routing, quote interval subscriptions have been used to evaluate the behavior of merging-based routing. The resulting curve is shown in Fig. 3. It is identical to that of identity-based routing with quote subscriptions because two given quote interval subscriptions of the used form can always be merged if they refer to the same stock. Of course, the effect of filter merging would be smaller if the merging probability was lower than 100%. In any case, the use of merging improves the scalability if quote interval subscriptions are used.

Routing based on Merging with Advertisements. If merging with advertisements is used in conjunction with quote subscriptions, the routing table of the source contains at most one remote routing entry for each neighbor, while that of the other brokers contains at most the number of neighbors minus one. Hence, the sum of all remote routing entries $\Sigma|R|$ is limited by $|V| - 1$. Because of that and to achieve comparability, quote interval subscriptions have been used (see Fig. 3). Again, the curve is identical to that of identity-based routing for quote subscriptions because of the reasons that were stated in the preceding subsection.

4. Filter Forwarding Overhead

In this section, we turn to the second key characteristic of a routing algorithm, namely the filter forwarding overhead with respect to the number of active subscriptions. As a realistic measure for this overhead the number of control messages that are processed by the brokers has been chosen so the measurements offer an insight into the *message complexity* of the protocols. In the figures, the abbreviation “FFO” stands for filter forwarding overhead.

Simple Routing. If simple routing is used each new or canceled subscription is forwarded to every broker and hence, all routing tables are affected by an update. Therefore, the number of control messages that is necessary to update the routing tables is equal to the number of brokers minus one, i.e., $|V| - 1$, and independent of the number of active subscriptions (see Fig. 5).

Simple Routing with Advertisements. Interestingly, the use of advertisements reduces not only the size of routing tables but also the filter forwarding overhead. This is because a subscription is only stored in the routing tables of those brokers that lie on the path from the respective consumer to the notification source. In consequence, $P_{avg} = 3.73$ control messages are necessary on the average (see Fig. 5). This means that compared to simple routing without advertisements the number of necessary control messages is reduced by a constant factor of $(|V| - 1)/P_{avg} = 28.4$ (see Fig. 7). In consequence, advertisements reduce the filter forwarding overhead significantly, especially for large systems because for the investigated type of topology P_{avg} grows only logarithmically in the number of brokers V.

Routing based on Identity. Here, the filter forwarding overhead depends on the number of active subscriptions. From Fig. 5 it can be inferred that for small numbers of active subscriptions the majority of routing tables are affected by a new or canceled subscription. This number decreases very quickly at first and still nearly exponentially afterwards. This is due to the fact that a new or canceled subscription is only forwarded to a neighbor if no identical subscription that has not been canceled yet was forwarded to this neighbor before. The probability for this to occur decreases with the number of active subscriptions.

Routing based on Identity with Advertisements. Here, the average number of control messages necessary to update the routing tables starts at the level of simple routing with advertisements (see Fig. 5). Interestingly, the resulting curve quickly approximates that of identity-based routing without advertisements. This is caused by the uniform distribution of interests. Hence, the factor by which the number of control messages is reduced by using advertisements is quite large for small numbers of subscriptions, but for large numbers of subscriptions, advertisements do not reduce the filter forwarding overhead at all (see Fig. 7).

Routing based on Covering. The number of necessary control messages also starts at the level of simple routing and decreases for larger numbers of subscriptions if covering-based routing is used. Indeed, covering-based routing behaves exactly like identity-based routing if quote subscriptions are used. If quote interval subscriptions are used, the curve, i.e., the filter forwarding overhead, drops much more slowly (see Fig. 6). This is due to the fact that two quote interval subscriptions that refer to the same stock do not need to cover each another. This also implies that sometimes a subscription is being forwarded that does not cause any additional notifications to be received. This disadvantage is diminished by merging-based routing.

Routing based on Covering with Advertisements. If quote subscriptions are used, covering-based routing behaves exactly like identity-based routing. If quote interval subscriptions are used, the number of necessary control messages decreases more slowly (see Fig. 6). Interestingly, the curves with and without advertisements approach one another only slowly and therefore, advertisements remain useful for larger numbers of subscriptions (see Fig. 7). This fact becomes even more evident for larger systems.

Routing based on Merging. The filter forwarding overhead induced by merging-based routing is similar to that of covering-based routing but decreases more slowly (see Fig. 6). In fact, the gradient is smaller in all areas of the curve. The reasons for this still need to be investigated.

Routing based on Merging with Advertisements. Here, the filter forwarding overhead is very similar to that of covering-based routing with advertisements. In fact, the overhead induced by merging-based routing is only slightly higher. This is interesting because the difference is much larger if advertisements are not used. The filter forwarding overhead without advertisements is clearly larger than those with advertisements even for larger numbers of subscriptions. Hence, the use of advertisements offers an advantage in this case, too (see Fig. 7).

5. Effects of Varying Degree of Locality

In the experiments of Sections 3 and 4 we assumed a system without locality, i.e., the interests of the clients were uniformly distributed over the entire system. In this section we investigate the effects that are caused by varying the degree of locality among the interests of the consumers. The identity-based routing algorithm serves as basis for these investigations; the findings for the other routing algorithms would be similar. In order to capture the behavior, the degree of locality of the three subnets that are induced by the subordinate routers of the root router has been varied. The local broker that is connected to the root router has been left without any subscriptions.

The parameter d is a measure of the amount of locality among the sets of clients of the respective subnets. The resulting curves vary d stepwise from $1/3$ to 1. For $d = 1/3$ the interests are disjoint and for $d = 1.0$ they are identical.

From the Figures 8, 9, 10, and 11 it can be inferred that the degree of locality has a great impact on the size of the routing tables and the filter forwarding overhead. A uniform distribution of interests (no locality) results in the largest routing tables and the highest filter forwarding overhead so the measurements in Sections 3 and 4 can be considered conservative. For smaller values of d the routing table and filter forwarding overhead monotonically decreases. Interestingly, the sum of all remote routing entries is limited by $d \cdot m \cdot (|V| - 2)$ if advertisements are used, and by $2 \cdot \frac{2+d}{3} \cdot m \cdot (|V| - 2)$ if advertisements are not used. Here again the use of advertisements shows its advantages.

The degree of locality also influences the amount of traffic that is saved when filtering is compared to flooding (see Fig. 12). For routing algorithms which do not forward notifications unnecessarily, the amount of saved payload traffic is independent of the underlying routing algorithm; it only depends on the type and number of issued subscriptions. For $d = 1.0$ and large numbers of subscriptions the saved amount of traffic slowly converges to 0. Opposed to that, for $1/3 \leq d < 1.0$ a constant amount of traffic which equals $1 - d$ is saved for large numbers of subscriptions.

By combining the results about the saved traffic and the filter-forwarding overhead the scenarios in which filtering is advantageous can be estimated. In fact, it is possible to determine the ratio of the event production rate to the subscription change rate for which the number of payload messages per time unit that is saved (by applying filtering) equals the number of control messages (that are necessary to update the routing tables). This gives an indication on the "break even point", i.e., up to how much dynamic activity the routing schema still saves messages. Surprisingly, the minimum ratio for which filtering is advantageous decreases for increasing numbers of subscriptions and degrees of locality. This becomes evident by comparing, for example, Figures 13 and 15. In simple routing (Fig. 15), the system must be increasingly static in order for the routing scheme to still save messages. In identity-based routing (Fig. 13), the gradient is negative meaning that the dynamics of the system can even "get worse" (i.e., increase) while still saving messages through filter-based routing. In particular, the degree of locality determines the gradient with which the minimum ratio decreases (see Fig. 13 and 14). Apparently, the use of filtering is advantageous in more dynamic scenarios than was previously thought.

6. Conclusion and Future Work

The evaluation has shown that for the investigated scenarios the advanced routing algorithms are clearly superior to the simple routing algorithm. Both the size of the routing tables and the filter forwarding overhead are significantly reduced by applying the proposed routing optimizations. The use of advertisements further reduces these numbers by a large factor which depends on the average path length in the given topology. In general, the average path length increases logarithmically in the number of brokers and therefore, advertisements are especially advantageous in content-based publish/subscribe systems with many brokers.

Furthermore, it has been shown that the degree of locality among the interests of the consumers has a large impact on the routing table sizes and the filter forwarding overhead, too. While a uniform distribution is the worst case, increasing locality reduces these magnitudes substantially. Moreover, if locality exists, a constant percentage of payload traffic is saved even for very large numbers of subscriptions. Importantly, by combining the results about the filter forwarding overhead and the saved traffic it was possible to derive the ratio of the event production rate to the subscription change rate for that the number payload messages that is saved (by applying filtering) equals the number of control messages (that are necessary to update the routing tables). It showed up that by applying the proposed optimizations filtering is advantageous in even more dynamic environments than was previously thought.

It can be concluded that the implemented routing algorithms improve the scalability of content-based publish/subscribe systems. Future work can build upon these results, e.g., by considering other and also more general scenarios. For example, dynamic advertisements should be considered. Moreover, algorithms that can adapt to changing environments, e.g., imperfect merging algorithms based on statistical on-line adaptation, are especially interesting.

References

[1] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems*, pages 262–272, 1999.

[2] G. Bricconi, E. D. Nitto, A. Fuggetta, and E. Tracanella. Analyzing the behavior of event dispatching systems through simulation. In *In the Proceedings of the 7th International Conference on High Performance Computing IEEE*, 2000.

[3] A. Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks*. PhD thesis, Politecnico di Milano, Milano, Italy, Dec. 1998.

[4] A. Carzaniga, E. Di Nitto, D. S. Rosenblum, and A. L. Wolf. Issues in supporting event-based architectural styles. In *ISAW '98: Proceedings of the Third International Workshop on Software Architecture*, pages 17–20, 1998.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001.

[6] G. Cugola, E. Di Nitto, and A. Fuggetta. The JEDI event-based infrastructure and its application to the development of the opss wfms. *IEEE Transactions on Software Engineering*, 2001.

[7] L. Fiege, M. Mezini, G. Mühl, and A. P. Buchmann. Engineering event-based systems with scopes. In B. Magnusson, editor, *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, volume 2374 of *LNCS*, pages 309–333, Malagá, Spain, June 2002. Springer-Verlag.

[8] L. Fiege, G. Mühl, and F. C. Gärtner. A modular approach to build structured event-based systems. In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'02)*, pages 385–392, Madrid, Spain, 2002. ACM Press.

[9] G. Mühl. Generic constraints for content-based publish/subscribe systems. In *Proceedings of the 6th International Conference on Cooperative Information Systems (CoopIS '01)*, volume 2172 of *LNCS*, pages 211–225. Springer, 2001.

[10] G. Mühl, L. Fiege, and A. Buchmann. Filter similarities in content-based publish/subscribe systems. In H. Schmeck, T. Ungerer, and L. Wolf, editors, *International Conference on Architecture of Computing Systems (ARCS)*, volume 2299 of *Lecture Notes in Computer Science*, pages 224–238. Springer-Verlag, 2002.

[11] L. Opyrchal, M. Astley, J. Auerbach, G. Banavar, R. Strom, and D. Sturman. Exploiting IP multicast in content-based publish-subscribe systems. In J. Sventek and G. Coulson, editors, *Middleware 2000*, volume 1795 of *LNCS*, pages 185–207. Springer-Verlag, 2000.

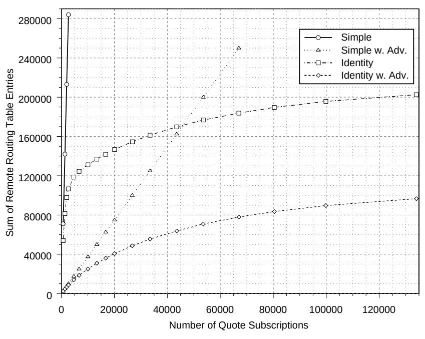

Figure 2. Simple vs. identity (RTS).

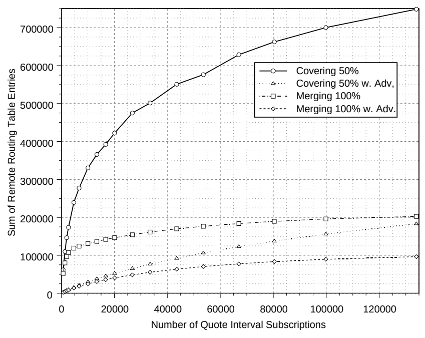

Figure 3. Covering vs. merging (RTS).

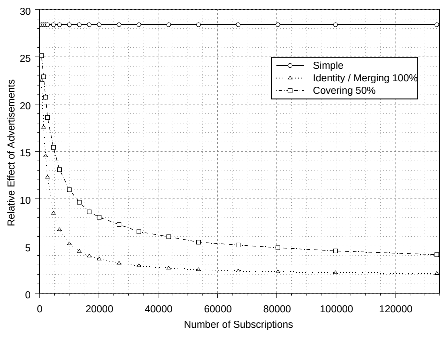

Figure 4. Effect of advertisements (RTS).

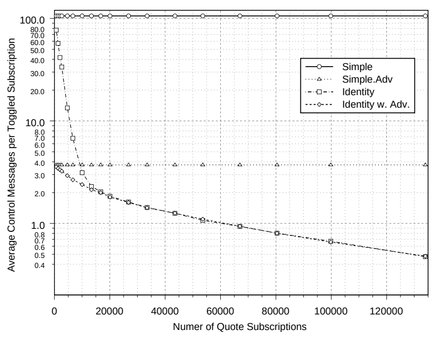

Figure 5. Simple vs. identity (FFO).

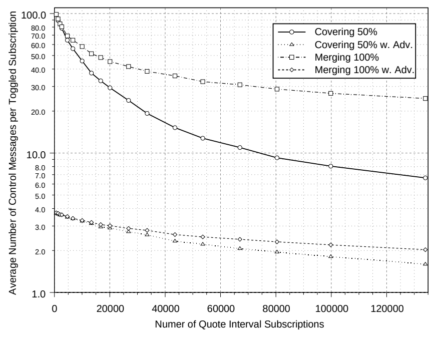

Figure 6. Covering vs. merging (FFO).

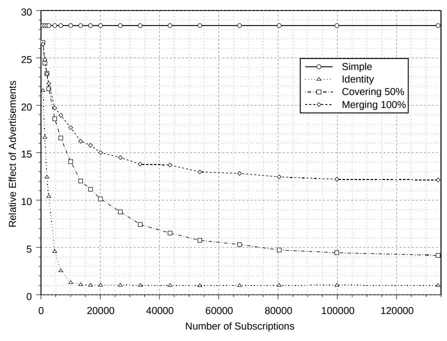

Figure 7. Effect of advertisements (FFO).

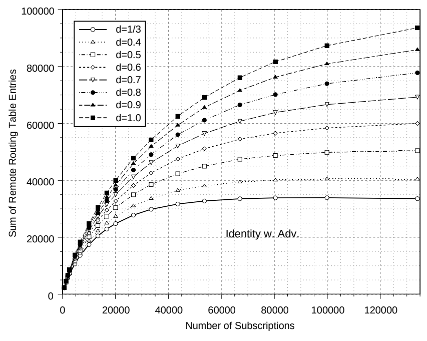

Figure 8. Identity (RTS with adv.).

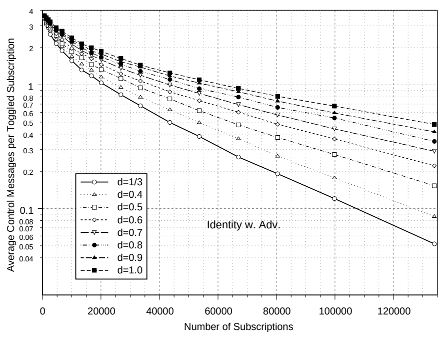

Figure 9. Identity (FFO with adv.).

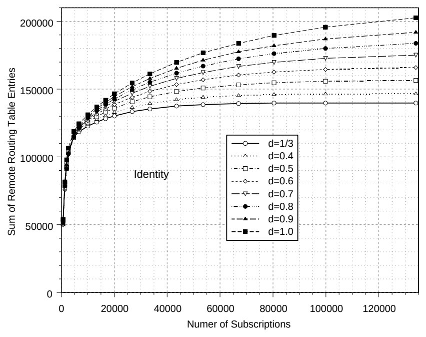

Figure 10. Identity (RTS without adv.).

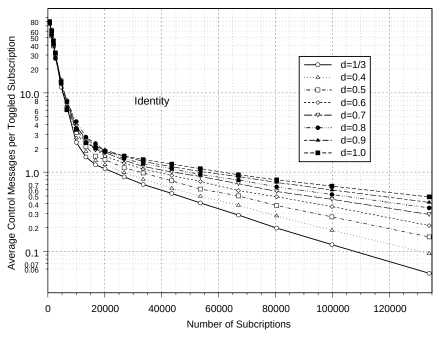

Figure 11. Identity (FFO without adv.).

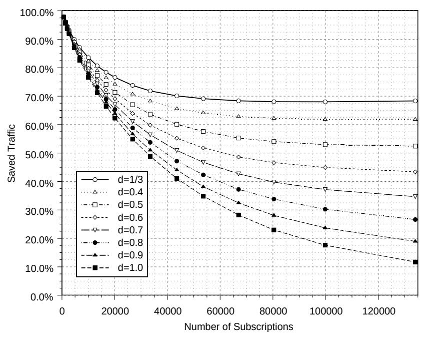

Figure 12. Saved payload cf. flooding.

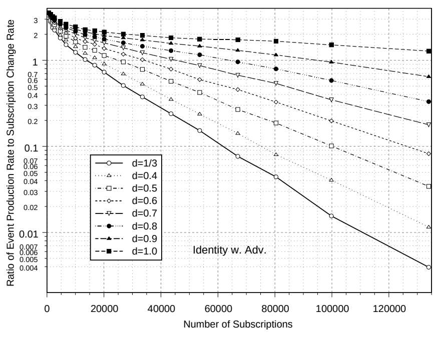

Figure 13. Dynamics ratio (with adv.).

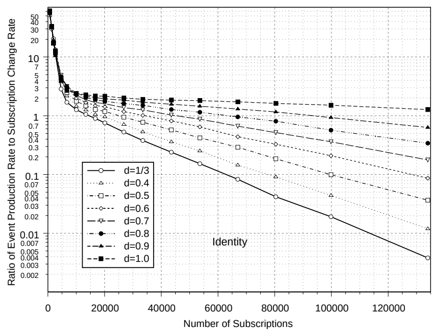

Figure 14. Dynamics ratio (without adv.).

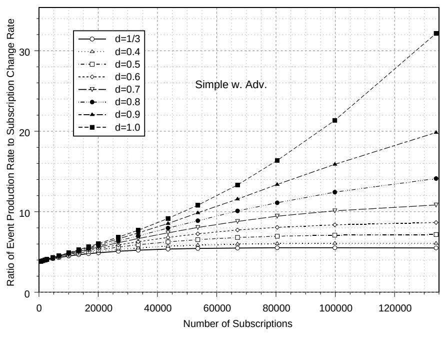

Figure 15. Dynamics ratio (with adv.).

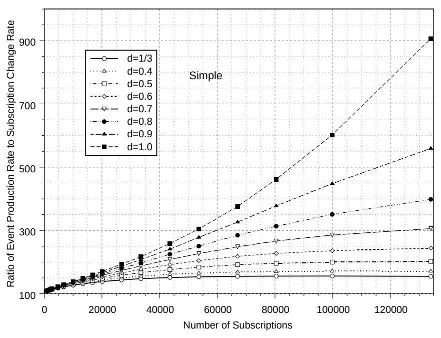

Figure 16. Dynamics ratio (without adv.).