Leveraging caching for Internet-scale content-based publish/subscribe networks

Mohamed Diallo*, Serge Fdida*, Vasilis Sourlas†‡, Paris Flegkas†‡ and Leandros Tassiulas†‡
* UPMC Sorbonne Universités, Paris, France.
†Department of Computer & Communication Engineering,
University of Thessaly, Greece. ‡CERTH-ITI
Email: {mohamed.diallo, serge.fdida}@lip6.fr, {vsourlas, pflegkas, leandros}@inf.uth.gr

***Abstract*—This work is concerned with scaling decentralized content-based publish/subscribe (CBPS) networks for Internet-wide content distribution. A fundamental step for CBPS networks to reach the Internet-scale is to move from the exhaustive filtering service model, where a subscription selects every relevant publication, to a service model capturing the quantitative and qualitative heterogeneity of information consumers' requirements. In previous work [1], we described a service model allowing information consumers to express the maximum number of publications they would like to receive per service period and how to take advantage of such knowledge to pace the dissemination process. This paper extends [1] by introducing a generic service model that seamlessly supports content-based information retrieval and dissemination and investigates through extensive simulations the performances of six caching policies in terms of consumers satisfaction and bandwidth usage.**

***Index Terms*—Publish/subscribe, Content distribution networks.**

I. Introduction

As the volume of information produced at the Internet-scale blows up, the publish/subscribe (pub/sub) paradigm becomes increasingly popular for content access and dissemination. Alerting, syndication and podcasting services are omnipresent and broadly adopted by Internet users. Pub/sub enables information consumers to register their information interests to a mediation entity that will prospectively retrieve content relevant to those interests. Different pub/sub models exist and they are typically classified according to the semantic of the subscription language. *Channel-based* pub/sub allows information consumers to subscribe to publications originating from specific channels or feeds. The popular *ATOM* and *RSS* standards, comply with this model. *Topic-based* pub/sub enables information consumers to register to a set of predefined topics organized into a hierarchy, while *content-based* pub/sub (CBPS) typically supports subscriptions following an attribute/value schema. Applications of CBPS include notification systems [6], enterprise service buses [11] and information filtering [10]. Among the pub/sub models, CBPS is the most challenging to scale, as expressiveness is to be traded against scalability [7].

This paper is concerned with scaling decentralized content-based pub/sub for Internet-scale content distribution, which requires addressing several issues. First, information consumers have different requirements. In fact, most consumers are likely to be overwhelmed by the amount of relevant information available at the Internet-scale and would like the system to deliver matching publications at a maximum rate, while some would be interested in every publication matching their interests. Moreover, CBPS should enable consumers to select publications published in the past. Thus, Internet-scale CBPS services should be able to capture the heterogeneity of consumers' needs. Second, bandwidth is a critical resource, which should be efficiently used, especially when communications involve the middle-mile of the Internet [12]. In fact, the design of bandwidth-thrifty schemes remains one of the major challenges of CBPS research.

We envision a mediation network providing the mediation between information providers and consumers, and involving different service providers and administrative entities. The mediation network is a distributed infrastructure supported by a set of interoperating mediation routers. We assume that mediation routers have important storage resources available and support a store-and-forward model [3]. Combining caching and pub/sub has previously been explored in order to enable historic data retrieval [4], [9], or to support mobility [5]. Also, Chen et al. [8] studied how to pace the dissemination of information between a publisher and a proxy server when usage-patterns and subscription information are available. However, their solution is centralized and thus does not apply here. This work departs from previous efforts by leveraging caching in order to meet the above stated requirements of decentralized and efficient Internet-scale content distribution.

Our contributions are threefold. First, we describe a service model that captures the heterogeneity of information consumers' requirements. This service model which generalizes our previous work [1], [2], seamlessly supports content-based information retrieval and dissemination and enables information consumers to express their actual information needs by the maximum number of publications desired per service period. Second, we describe the protocols required to efficiently implement the service. Finally, we investigate and evaluate different caching policies in order to regulate the trade-off between caching efficiency and the QoS offered.

Interest message
predicate (filter)
max
lifetime
age
id
refresh flag

Publication message
meta-data
id
Data/content

Fig. 1. Interest/Publication message format.

II. The Content-Centric Mediation Network

A. Service model

A set of distributed entities called mediation routers (MR) cooperate to support the mediation. Information providers upload their publications to MRs in order to satisfy information consumers. A publication is composed of a content object and descriptive metadata. Information consumers register their content interests to MRs, as `I(`*predicate*`, ` *max*`, ` *lifetime*`, ` *age*`)`, where *predicate* is a list of constraints, usually in the form of name-value pairs of properties and basic comparison operators, *max* is the maximum number of responses that the requesting client is willing to retrieve during its *lifetime* and *age* is the maximum age of the publications it is interested in. A publication matches an interest, if the interest's *predicate* matches the publication meta-data and if the publication has been available in the system for less than *age*. The MR to which an interest is initially registered acts as a proxy for this interest, and is called its home mediation router (HMR). Each HMR advertises its interests to the other MRs that reply with matching publications. When *age* equals zero, the interest has the conventional semantic of a subscription, *i.e.* the interest selects only future publications. When *age* and *lifetime* are both positive, the interest is a loose subscription which differs from a subscription by the fact that requesters are only interested in publications that they did not consume previously. Loose subscriptions can be refreshed after *lifetime* expires. Then, the system guarantees to the requester that the refreshed subscription is not satisfied with previously consumed publications. In the case where *lifetime* equals zero, then the interest has the semantic of a non-persistent request asking the mediation network to retrieve up to *max* published messages currently available. In the case of non-persistent requests ($lifetime = 0$), the retrieved publications are directly forwarded to the requesting clients. In the case of persistent requests ($age = 0$ or $lifetime > 0$), HMRs cache the answered publications until consumers poll them.

B. Mediation Routers model

MRs are provisioned with a cache which stores publications uploaded from local publishers, publications retrieved for local subscribers and also publications opportunistically cached as they transited through them. Each MR maintains a set of data structures for its operation. The pending interest table (`PIT`) references interests that have been forwarded and which are waiting for matching publications. The publications available in the cache of a MR are referenced in two distinct tables. The pending publication table (`PPT`) references publications uploaded at a particular MR and waiting for opportunities to be disseminated towards other MRs. The dispatched publication table (`DPT`) references cached publications which have been used to satisfy originating or transiting interests at least once. The `DPT` covers publications waiting for consumption by local subscribers and publications opportunistically cached to serve future interests. Once a pending publication is selected, it is referenced through the `DPT` instead of the `PPT`. When a publication is discarded from the cache, as decided by the cache replacement strategy, the corresponding entries are removed either from the `DPT` or the `PPT`.

C. Protocols and basic mechanisms

1) Publication forwarding: Interests are propagated throughout the mediation network so as to form a tree connecting, in the worst-case, each publishing router to all the routers interested in a publication. These trees are materialized by the states installed in the `PIT` by the interest forwarding strategy. Fig. 1 describes the format of publication messages used within the mediation network. The *id* is useful to identify loops on cyclic topologies. When an information provider uploads a publication at a router, it is forwarded towards the routers that advertised an interest for that publication in the `PIT`. Otherwise, there is no matching interest in the `PIT` and the publication is registered in the `PPT`. Moreover, a router does not forward a publication originating from neighboring routers, but already available in the cache. In fact, the interest that selected the publication was forwarded through that router and it might have already replied with that publication.

2) Interest forwarding: Fig. 1 describes interests message format. The *id* included in interest messages is, among other purposes, useful to prevent loops. When a consumer registers a new interest `I` to its HMR, this latter searches in its cache for publications matching the interest. If *max* or more publications are found, then the interest is dropped and the `top-max` matching publications are selected to serve the interest. If there are less than *max* publications at the HMR, the interest is advertised in the `PIT` and forwarded to the HMR's neighbors with the *max* parameter decremented by the amount of matching publications found at the HMR. If a MR receives an interest message originating from a neighboring router such that its *id* and the corresponding *filter* are already in the `PIT`, it stops forwarding the interest without adding an entry in the `PIT`. This procedure is useful to prevent interest messages loops. Otherwise, it matches the interests with its cached publications. Then, the interest is processed as described previously according to the number of matching publications found. When a persistent request is satisfied, the responsible HMR advertises an `unsubscribe` message in order to remove the corresponding states from mediation routers' tables.

3) Interest refreshing: Once an interest's *lifetime* expires, the consumer can request more publications by refreshing its interest. The *refresh flag* is set to indicate that it is a refresh message. If a router receives a "refresh" interest message (from another MR), with an *id* and the corresponding $filter$ already present in the PIT, then it updates the previous entry from the PIT and forwards the refresh message similarly to the interest forwarding strategy, except that it selects only pending publications. This way, the system guarantees that over an interest's successive lifetimes, it is not satisfied more than once with the same publication, without having to track an exhaustive history of all the publications previously dispatched to it, or of all the interests already satisfied with a publication.

III. Caching Strategies

Given the storage available at MRs, we aim at designing caching strategies that maximizes the number of satisfied interests and minimizes the publications/interests message traffic. We assume that information consumers privilege fresh information and that information providers want their publications to reach the maximum number of information consumers.

A. Cache organization policies

The cache organization policies explore the trade-off between privileging refreshed interests and new ones. Refreshed interests benefit only from pending publications, which correspond to publications that are not popular (no matching subscription yet) or that are redundant. On the other hand, new interests benefit both from pending and dispatched publications. So, we investigate the following cache organization policies:

1) Pending Publication First policy (PPF): Pending publications are cached first.
2) Dispatched Publication First policy (DPF): Dispatched publications are cached first.
3) RAW: neither PPF nor DPF apply.

In the PPF and DPF cases, the cache can be represented as a stack, shared by both dispatched and pending publications and organized in two sub-stacks with a moving border between them, one associated to dispatched publications and the other to pending ones.

B. Selection policies

The selection policy indicates which publications are returned first upon the arrival of an interest message. In other words, the selection policy defines the publications with the highest priority to be selected in each case. So, we study the following selection policies.

1) MF: The highest priority corresponds to the freshest publications (according to the age parameter).
2) LF: The highest priority corresponds to the less fresh publication.
3) MRU: The highest priority corresponds to the most recently used publications.
4) MFU: The highest priority corresponds to the most frequently used publications.
5) LFU: The highest priority corresponds to the least frequently used publications.

C. Dropping policies

In presence of cache overflow, MRs have to decide which publication to replace upon the arrival of a new publication message. Similarly to the selection policies the dropping polices define the publications with the highest priority to be dropped. We consider four meaningful policies, namely LF, LRU (Least Recently Used), MFU and LFU. The LRU, as well as the two frequency based (MFU and LFU) dropping policies differentiate publications in a cache based on their usability as well as to provide better performance of the whole system. The most popular publications, those with multiple interests survive more in a cache while the publications with no interest are closer to be dropped.

D. Opportunistic caching

By default, a publication is cached in the HMR where it is initially uploaded and to the MRs that requested that publication. Also, when caching resources are available and the En-route policy is enabled, a publication can be opportunistically cached in every MR along the path from the publisher towards the requesting HMR.

IV. Performance Evaluation

Given the storage available at MRs, we investigate the performances of the six caching strategies presented in Table I in terms of communication costs and consumers' satisfaction.

Selection	DPF/MF	PPF/MF	MF	MRU	MFU	LFU
Dropping	PPF/LF	DPF/LF	LF	LRU	LFU	MFU

TABLE I
Evaluated caching strategies (where not explicitly denoted is the RAW caching policy).

In particular, we consider the following metrics:

- **Interest and publication message traffic**. It corresponds to the total number of interests and publication messages forwarded in the mediation network.
- **Starvation**. It is measured as the percentage of non-satisfied interests. An interest is not satisfied, only if the number of matching items available at the end of the service period at the responsible HMR is less than the interests' selectivity (*max*), while there were opportunities to retrieve enough publications.

A. Methodology

It is widely acknowledged that content-based pub/sub research lacks public data sets for meaningful evaluation. Thus, synthetic workload generation is largely accepted in the field, under reserve that the workload verifies a set of realistic assumptions. First, we assume a generic peer-to-peer MR topology of average degree 2 following a power-law distribution with few routers of high degree and many routers of small degree. We consider that a set of events drive publications and interests generation. Three parameters characterize each event: *popularity*, *locality* and *volume*. The popularity of an event refers to the number of interests related to it, its volume to the number of related publications and its locality to the regions of the topology likely to originate related interests and publications. In the generation of our workload, we assume that the more popular events are, the more related publications are generated and the more widely related interests are spread in the topology. In other terms, popular events are characterized by a larger and broader audience, and are also likely to trigger the larger volume of publications.

More precisely, let us consider e_i ($1 \leq i \leq E$), an event of popularity p_i, volume v_i and locality l_i. p_i is sampled from a power-law distribution of exponent 0.7 and the volume v_i is such that $v_i = P * p_i$. Publications associated to e_i can be issued by any of the N MRs, while interests only from a set of routers computed using l_i. We define l_i such that $l_i = p_i$ and such that $\lceil l_i * N \rceil$ MRs are potential issuers (hosting interested subscribers) of interests related to e_i. This set of MRs is computed by choosing a random root router and $\lceil l_i * N \rceil - 1$ additional routers among the most closed MRs to the root. Interests (respectively publications) arrival rate follows a Poisson distribution of average $r_s = S/T$ (respectively $r_p = P/T$). For each publication, we randomly select an event and a location among all MRs. When the volume associated to an event is reached, it is removed from the set of events that can be used to generate new publications. The locations of interests are selected among the set of MRs related to the event. Interests are generated using the *Max*, *Freshness* and $Lifetime$ parameters defined in the Table II. We model only loose subscriptions which are representative, and assume that each interest is renewed at the end of its lifetime with a probability P_r.

We use our PEERSIM [13] implementation of the Content Centric Mediation Network to simulate the performances of the policies listed in Table I, deployed with the `En-route` policy enabled. In particular, we study the impact of the cache size and the refresh probability on the performance metrics. Except explicit mention, parameters values are defined by Table II. The results are averaged over several simulation runs.

Parameters	**Definition**	**Value(s)**
T	Simulation length	1000
N	Network Size	100
E	number of events	20
S	number of basic interests	$10k$
Max	Maximum selectivity	3
P	number of publications	$500k$
$Freshness$	Maximum freshness (age)	1000
$Lifetime$	Maximum lifetime	50
$Slots$	Cache size of each MR (default)	100
P_r	Refresh prob. (default)	0.2
Exponent of the popularity distr.		0.7

TABLE II
PARAMETERS USED FOR THE PERFORMANCE EVALUATION

B. Impact of the cache size

The upper row in Fig. 2 depicts the starvation as well as the total interests and publications messages forwarded in the mediation network for various cache sizes, when $P_r = 0.2$. With $P_r = 0.2$ the ratio of refreshed interests to new interests is low. In this case, it is not surprising to observe that all the proposed caching strategies can satisfy the interests regardless of the cache size. Particularly, all the caching strategies except `DPF`, present almost no starvation. With the `DPF` caching strategy, only 8% of the subscriptions are not satisfied for small cache sizes. In fact, new interests are easier to satisfy than refreshed interests which request non-previously consumed publications. It is also noticeable that with such a low refresh probability, the starvation metric is not affected by the size of each MR's cache, and even cache sizes of $50 - 100$ slots are sufficient to successfully serve almost all interests. Regarding the interest and publication traffic, it is obvious that only a very small portion of this traffic (less than 0.1%) was forwarded in the mediation network. This means that the majority of the interests were satisfied locally, which implies that the proposed content centric mediation network can scale to Internet-scale content distribution.

The lower row in Fig. 2 also depicts the starvation and the interests and messages forwarded in the mediation network, when refreshed interests dominate the interest traffic ($P_r = 1.0$). We observe that `DPF` shows the worst starvation performance. The bad performances of `DPF` is due to the fact that refreshed interests benefit only from pending publications. `PPF` performs better than `DPF` because it favors pending publications, by making them more persistent than dispatched publications. In fact, this behavior increases the probability of interests to be satisfied locally. The `MRU`, `MF` and `MFU` satisfies almost all the interests, probably because of the efficiency of their dropping policies (least recently, less fresh and less frequently used). Moreover, when the cache size of the MRs becomes large enough, the starvation declines to zero meaning that the cache capability of the network is enough to serve any kind of interest. Regarding interest and message traffic, the percentage of forwarded interests and publication messages is still very low (less than $2 - 3\%$) with the exception of the `DPF` strategy where in small cache values, almost 20% of the interests and messages are forwarded in the mediation network. This arises from the fact that the rarity of pending publications locally forces MRs to forward unsatisfied interests to the rest of the network.

C. Impact of the refresh probability

We vary the refresh probability between 0.2 and 1 and observe the impact on the starvation of the interests when each MR's cache has 100 slots. In Fig. 3 also, almost all the proposed caching strategies are able to satisfy the clients' interests, leaving only a small portion of unsatisfied interests when the refresh probability is large enough (only 2% of the interests are not satisfied when $P_r = 1.0$). Even

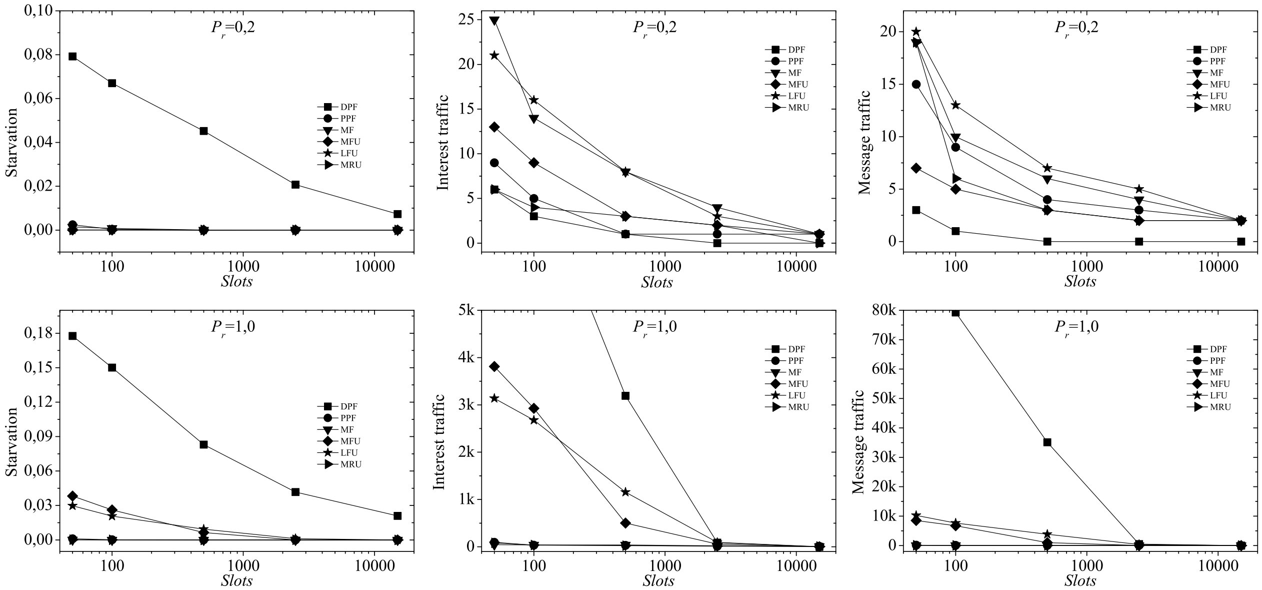

Fig. 2. Starvation, interest and publication/message traffic vs cache size (slots per MR), for two refresh probabilities ($P_r = 0.2$ and $P_r = 1$ respectively). In the legends are the corresponding selection policies.

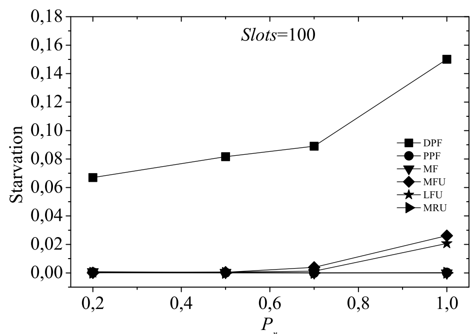

Fig. 3. Starvation vs refresh probability, when each MR has a cache of 100 slots.

when using the `DPF` caching strategy, which favors the least pending publications, only $7-15\%$ of the total interests are not satisfied.

V. Conclusion and Future Work

In this paper, we described a service model which captures the heterogeneity of information consumers' requirements and seamlessly supports content-based information retrieval and dissemination. Also, we investigated and evaluated different caching strategies in order to regulate the trade-off between caching efficiency and the offered QoS. The results show that using the appropriate caching strategy, together with the proposed service model, it is possible to scale decentralized content-based publish/subscribe networks for Internet-scale content distribution. This work can be extended in many ways from exploring new caching strategies to enabling the mobility of consumers and mediation routers.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement N° 224263 - OneLab2 and V. Sourlas work was supported by the Greek Ministry of National Education and Religious Affairs (E.S.P.A.- HRAKLEITOS II"). We would like to thank M. Jong Gun Lee and M. Paul V. Mockapetris for their invaluable comments and feedback.

References

[1] M. Diallo and S. Fdida , "IOA-CBR: information overload-aware content-based routing," Proc. 4th ACM Inter. Conference on Distributed Event-Based Systems, Cambridge, UK, 2010.

[2] M. Diallo and S. Fdida , "Avalanche: towards a scalable content-based Pub/Sub network service," Proc. of the 3rd ACM Inter. Conference on Distributed Event-Based Systems, Nashville, TN, 2009.

[3] S. Fdida and M. Diallo, "The network is a database," Proc. 4th Asian Conference on Internet Engineering, Bangkok, Thailand, pp. 18 - 20, Nov. 2008.

[4] V. Sourlas et al., "Caching in content-based publish/subscribe systems," Proc. 28th IEEE Conference on Global Telecommunications, Hawaii, USA, Nov. 2009.

[5] V. Sourlas et al., "Mobility Support Through Caching in Content-Based Publish/Subscribe Networks," 10th IEEE/ACM Inter. Conference on Cluster, Cloud and Grid Computing, pp. 715-720, 2010.

[6] A. Carzaniga et al., "Design and evaluation of a wide-area event notification service," ACM Trans. On Comp. Syst., vol. 19, pp. 332-383, 2001.

[7] A. Carzaniga et al. "Achieving Expressiveness and Scalability in an Internet-Scale Event Notification Service," Principles Of Distributed Computing 2000.

[8] M. Chen et al., "Content distribution for publish/subscribe services," Middleware '03: Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware, 998–998, 2003

[9] M. Cilia et al., "Looking into the past: enhancing mobile publish/subscribe middleware," Proc. 2nd Inter. Workshop on Distributed Event-Based Systems, pp. 1-8, San Diego, California, 2003.

[10] T. W. Yan et al., "The SIFT information dissemination system," ACM Trans. Database Syst. 24, vol. 4,pp. 529-565, Dec. 1999.

[11] D. Chappell, "Enterprise Service Bus," O'Reilly Media, Inc., 2004.

[12] T. Leighton, "Improving performance on the Internet," Commun. of the ACM, 2009.

[13] http://peersim.sourceforge.net/.