Publish-Subscribe Systems via Gossip: a Study based on Complex Networks

Stefano Ferretti
Department of Computer Science, University of Bologna
Bologna, Italy
sferrett@cs.unibo.it

ABSTRACT

This paper analyzes the adoption of unstructured P2P overlay networks to build publish-subscribe systems. We consider a very simple distributed communication protocol, based on gossip and on the local knowledge each node has about subscriptions made by its neighbours. A mathematical analysis is provided to estimate the number of nodes receiving the event. These outcomes are compared to those obtained via simulation. Results show even when the amount of subscribers represents a very small (yet non-negligible) portion of network nodes, by tuning the gossip probability the event can percolate through the overlay.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques

General Terms

Algorithms, Performance, Theory

1. INTRODUCTION

Publish-subscribe is a distributed paradigm that gained a lot of attention in the last years. Today, it is widely used in several large-scale distributed applications, such as checking stock exchange quotations, information dissemination, targeted advertising, multiplayer online games, decentralized business process execution, workflow management, and discovery [2]. The interesting feature of a publish-subscribe system is that it allows nodes to communicate asynchronously in a loosely and decoupled manner. In a publish-subscribe system, there are nodes which are interested in receiving some type of contents; they are referred as *subscribers*. *Publishers* are those actors who produce information. Loose-coupling is achieved since producers do not have information on the identity and number of subscribers, as well as consumers subscribe to specific information without knowing the identity and number of possible publishers.

SIMPLEX '12, April 17 2012, Lyon, France.

Publish-subscribe systems can be implemented by resorting either to centralized or distributed solutions. The typical lack of scalability and fault-tolerance of centralized solutions led to the study of distributed solutions. Among the plethora of different solution, an interesting approach is based on unstructured Peer-to-Peer (P2P) overlay networks [8]. In an unstructured P2P overlay, links among nodes are established arbitrarily. They are particularly simple to build and manage, with little maintenance costs, yet at the price of a non-optimal organization of the overlay. Peers locally manage their connections to build some general desired topology and links do not depend on the contents being disseminated [4]. Publish-subscribe systems can be built on top of unstructured networks by adopting either a gossip-based communication protocol, or some more sophisticated algorithm to route messages in the overlay. Contents can be replicated or not, as well as queries. In any case, we might sum up that these systems can be effectively employed when: i) the number of nodes is very high and dynamic, with high churn rates; ii) there is a high number of publications to handle; iii) there is a high number of subscribers to a given type of contents and hence usually an event must be propagated to reach a non-negligible portion of nodes in the overlay.

In this work, we study if a general P2P publish-subscribe system can be implemented on top of unstructured overlay networks. In particular, to distribute events through the unstructured overlay, we consider a simple dissemination protocol which is based only on local knowledge among neighbour peers and gossip. We analyze such protocol through an analytical model which estimates the amount of subscribers that may receive a given event. The approach is quite general; the network topology can be set by defining the node degree distribution probability. Depending on the network topology, the proportion of subscribers in the overlay, and the gossip probability threshold, it is possible to understand if the event reaches only a limited amount of nodes, or if it is spread through the whole network, i.e. it might reach an infinite amount of nodes. Of course, this happens only when the network topology has a giant component.

Numerical outcomes are compared with those obtained via simulation. Outcomes confirm that a node subscribing to a given type of contents will receive an event matching its subscriptions with high probability. Of course, we are not suggesting here to replace completely structured schemes, usually employed to build publish-subscribe services, with unstructured overlays using gossip. Rather, our claim is that this solution represents an interesting alternative when dealing with large scale and highly dynamic systems. In this case, in fact, the costs for managing and maintaining a structured (or centralized) distributed system is quite high.

The remainder of this paper is organized as follows. Section 2 presents the system model. Section 3 states the local protocol executed at each node. Section 4 presents the mathematical model. Section 5 outlines results coming from a numerical analysis and simulation. Finally, Section 6 provides some concluding remarks.

2. SYSTEM MODEL

We consider the set of nodes organized as a P2P overlay network. Each node **n** is connected to a given subset of nodes, whose number is specified using whatever probability distribution. The overlay does not depend on the subscriptions made by peers in the P2P publish-subscribe system, i.e. the overlay is unstructured. We denote with p_i the probability that a peer **n** has i neighbours (its *degree*). We assume that the overlay has a high number of nodes. This assumption comes from the fact that the solution we are studying is thought for very large and highly dynamical systems. The high number of nodes, together with the random nature of contacts among peers in the overlay, augments the probability of having a low clustering in the network [7].

Events produced by publishers are included within messages spread through the overlay. Direct communication may occur only between neighbour nodes. Hence, to disseminate information through the overlay, peers must act as relays and forward messages to their neighbours.

Peers in the overlay may act as *subscribers* or *publishers*. Subscribers register their interest in an event, or a pattern of events. Then, they must be notified asynchronously when events are generated by publishers [5]. Such events may represent any kind of information which is usually filtered based on some event classification scheme. We are not going to describe in detail the plethora of existing methods to categorize events, since the particular approach is independent from the dissemination strategy. It is sufficient to assume that each event has some metadata associated to it, and that a subscription specifies a set of metadata the node is interested in. Peers in the overlay may be subscribers and publishers at the same time, even for multiple patterns of events. If a peer in the overlay is not a subscriber nor a publisher for a given kind of content, it will act as a relay.

Each peer **n** stores in its cache all the subscriptions of its neighbours. Once **n** receives a message containing a given event **e**, it is able to understand which neighbours are interested in receiving **e**. Nodes maintain in their caches information on messages which have been already handled, so as to avoid redundancy in the communication.

3. THE PROTOCOL

In the considered system, there are two main activities accomplished by peers. The first one is the subscription of a peer to a given event type. The other activity is concerned with the publication and dissemination of novel events.

The subscription protocol is very simple. When a peer **n** makes a novel subscription, it informs its neighbours. In turn, each node **m** receiving a message containing a novel subscription from a neighbour **n**, adds a related entry in its neighbour table. This way, each time **m** receives an event **e** matching this subscription, **m** sends **e** to **n**. When a node is no more interested in a subscription, it informs its neighbours that will remove the related entry.

Algorithm 1 Dissemination protocol executed at node **n**

```
Require: Event e generated at n ∨ e received from a peer m
 1: e ← REMOVEFROMBUFFER()
 2: if e already handled ∨ TTL(e) = 0 then
 3:    Return
 4: end if
 5: DECREASETTL(e)
 6: N ← n's neighbours \ m  {m = NULL if e originated at n}
 7: I ← {i|i ∈ N ∧ i's subscriptions match e}
 8: for all i ∈ I do          {send e to all neighbour subscribers}
 9:    SEND(e, i)
10: end for
11: for all i ∈ N \ I do    {gossip to the remaining neighbours}
12:    if RANDOM() < γ then
13:       SEND(e, i)
14:    end if
15: end for
```

The dissemination protocol is a push scheme: nodes which have novel information to disseminate forward messages to other peers. Algorithm 1 shows the pseudo-code of the algorithm. Once a given node **n** generates a novel event **e**, or upon reception of **e** from a neighbour **m**, **n** checks if it has already handled **e** in the past; in such a case, **n** drops **e** (lines 2–4). This reduces the amount of messages in the network. The event is dropped also if the Time-To-Live (TTL) associated to the event has reached a 0 value.

If **e** it is not dropped, **n** forwards it to the subset of neighbours whose subscriptions match the topics associated to **e**, with exception of **m** (lines 6–10). Then, **n** considers the remaining set of its neighbours, i.e. those nodes that are not interested in receiving **e**. For each node in this subset, **n** gossips **e** with a probability $\gamma \leq 1$ (lines 11–15).

An important aspect is concerned with the TTL value, employed to avoid that messages are forwarded forever in the net. In particular, such TTL must be sufficiently large to guarantee that the message can be spread through the whole network. An estimation of such diameter. can be obtained starting from the degree probability distribution, and in most kinds of nets it is usually a low number. Hence, based on this common assumption, we will not consider such TTL value in the model presented in the next section.

4. NETWORK COVERAGE

In this section, we analyze the performance of the decentralized P2P protocol presented in the previous section. We specifically focus on the coverage of the overlay, i.e. the average amount of subscribers $\langle s \rangle$ that receive a given event **e**. We denote with σ the probability that a node has made a subscription matching **e**, i.e. σ represents the portion of nodes in the overlay interested in receiving **e**. We model each single event dissemination as a standalone activity. In other words, the model treats the distribution of generated events as independent tasks. This is a correct assumption if peers have a buffer whose size is sufficiently large to handle simultaneous events passing through it. Conversely, the model should be extended to consider possible buffer overflows.

We consider networks with a large number of nodes. Following the approach presented in [7], we assume that links among nodes are randomly generated, based on a given node degree distribution. This does not represent a problem, since the overlays we are considering here are synthetic communication networks, which can be built using whatever algorithm chosen during the network design phase. A consequence of the random nature of the attachment process is that, regardless of the node degree distribution, the probability that one of the second neighbours (i.e. nodes at two hops from the considered node) is also a first neighbour of the same node, goes as N^{-1}, being N the number of nodes in the overlay. Hence, this situation can be ignored since the number of nodes is high.

4.1 Degree and Excess Degree Distributions

We denote with p_i the probability that a peer $\mathbf{n}$ has degree equal to i. Starting from $\mathbf{n}$, another measure of interest is the number of connections (links) that a node $\mathbf{m}$, which is a neighbour of $\mathbf{n}$, may provide, other than the one that connects $\mathbf{m}$ with $\mathbf{n}$. In particular, the probability that, following a link in the overlay, we arrive to a peer $\mathbf{m}$ that has other i links (hence its total degree is $i+1$) is $q_i = \frac{(i+1)p_{i+1}}{\sum_j jp_j}$. The probability q_i is often referred as the *excess degree distribution* [7]. Probabilities p_i and q_i represent two similar concepts i.e. the number of contacts of a considered peer (its degree), and the number of contacts obtained following a link (its excess degree), respectively. In the following, we introduce measures obtained by considering the degree p_i of a node, and considering the excess degree q_i of a link. In this last case, with a slight abuse of notation we denote all the probabilities/functions related to the excess degree with the same letter used for the degree, with an arrow on top of it, just to recall that the quantity refers to a link.

4.2 Probability of Dissemination

Given a peer $\mathbf{n}$ in charge of relaying an event $\mathbf{e}$, the probability that $\mathbf{n}$ forwards $\mathbf{e}$ to i of its neighbours is

$$f_i = [\sigma + (1-\sigma)\gamma]^i \sum_{j\geq i} p_j \binom{j}{i} [(1-\sigma)(1-\gamma)]^{j-i}, \quad (1)$$

which is obtained by considering all the possible cases of $\mathbf{n}$, having a degree higher than i, which forwards $\mathbf{e}$ to i neighbours either because they are subscribers to events matching $\mathbf{e}$ (with probability σ), either because they are not subscribers but $\mathbf{n}$ decides to gossip $\mathbf{e}$ (with probability $(1-\sigma)\gamma$). Moreover, $\mathbf{n}$ does not gossip $\mathbf{e}$ to its remaining $j-i$ neighbours, which not subscribed to topics matching $\mathbf{e}$ (with probability $(1-\sigma)(1-\gamma)$). In the rest of the discussion, for the sake of a more readable presentation, we denote $\Gamma = \sigma + (1-\sigma)\gamma$ and $1-\Gamma = (1-\sigma)(1-\gamma)$.

A similar reasoning can be made to measure the probability that, following a link we arrive to a node that forwards $\mathbf{e}$ to i other nodes. This probability is readily obtained by substituting, in (1) above, p_j with q_j, i.e.

$$\overrightarrow{f}_i = \Gamma^i \sum_{j\geq i} q_j \binom{j}{i} (1-\Gamma)^{j-i}. \quad (2)$$

To proceed with the reasoning, we need to introduce the generating functions for f_i, $\overrightarrow{f}_i$, as well as for p_i, q_i, i.e.

$$G(x) = \sum_i p_i x^i, \qquad \overrightarrow{G}(x) = \sum_i q_i x^i, \quad (3)$$

$$F(x) = \sum_i f_i x^i, \qquad \overrightarrow{F}(x) = \sum_i \overrightarrow{f_i} x^i. \quad (4)$$

In fact, if we consider the F generating function,

$$\begin{aligned} F(x) &= \sum_i f_i x^i = \sum_i \Gamma^i x^i \sum_{j\geq i} p_j \binom{j}{i} (1-\Gamma)^{j-i} \\ &= \sum_j p_j \sum_{i=0}^{j} \binom{j}{i} \Gamma^i x^i (1-\Gamma)^{j-i} \\ &= \sum_j p_j (\Gamma x + 1 - \Gamma)^j = G(\Gamma x + 1 - \Gamma) \quad (5) \end{aligned}$$

One might notice that all the coefficients of the introduced generating functions are probabilities. In fact, $G(1) = \sum_i p_i = 1$, as well as $F(1) = \sum_i f_i = 1$, and so on. Now, it is also possible to evaluate the average of the values f_i, by calculating the derivative of F measured at $x = 1$, since $F'(1) = \sum_i i f_i$. We have

$$F'(x)\Big|_{x=1} = \frac{dG}{dx}(\Gamma x + 1 - \Gamma)\Big|_{x=1} = \Gamma G'(1) = \Gamma\langle p\rangle, \quad (6)$$

where $\langle p\rangle$ is the mean node degree probability.

From a similar reasoning,

$$\overrightarrow{F}'(x)\Big|_{x=1} = \Gamma \overrightarrow{G}'(1) = \Gamma\langle q\rangle, \quad (7)$$

where $\langle q\rangle$ is the mean value of the excess degree, that is [7][1]

$$\begin{aligned} \langle q\rangle &= \sum_i i q_i = \frac{\sum_i i(i+1)p_{i+1}}{\sum_j jp_j} = \frac{\sum_i (i-1)ip_i}{\sum_j jp_j} \\ &= \frac{\langle p^2\rangle - \langle p\rangle}{\langle p\rangle}. \quad (8) \end{aligned}$$

4.3 Number of Receivers and Subscribers

We can now consider the whole number of nodes reached by a message starting from a given node, regardless of the number of hops. Let denote with r_i the probability that i peers receive an event, starting from a given node. Similarly, denote with $\overrightarrow{r}_i$ the probability that i peers are reached by the event dissemination, starting from a link. In general, $\overrightarrow{r}_i$ can be defined using the following recurrence,

$$\begin{aligned} &\overrightarrow{r}_0 = 0, \\ &\overrightarrow{r}_{i+1} = \sum_{j\geq 0} \overrightarrow{f}_j \sum_{a_1+a_2+\ldots+a_j=i} \overrightarrow{r}_{a_1} \overrightarrow{r}_{a_2} \ldots \overrightarrow{r}_{a_j}. \quad (9) \end{aligned}$$

Equation (9) can be explained as follows. It measures the probability that following a link we disseminate the event to $i+1$ peers. (The case $\overrightarrow{r}_0$ is impossible, since at the end of a link there must be a node.) In general, one peer is the one reached at the end of the link itself. Then, we consider the probability that the peer has other j links (varying the value of j). Each link k allows to disseminate the event to a_k peers, and the sum of all these reached peers equals to i.

Similarly, we can calculate r_k as follows

$$\begin{aligned} &r_0 = 0, \\ &r_{i+1} = \sum_{j\geq 0} f_j \sum_{a_1+a_2+\ldots+a_j=i} \overrightarrow{r}_{a_1} \overrightarrow{r}_{a_2} \ldots \overrightarrow{r}_{a_j}. \quad (10) \end{aligned}$$

In this case, we start from the peer itself, considering it has a degree equal to j; and as before, from its j links we can reach i other peers, globally.

[1] $\langle p^2\rangle$ refers to the second moment of p_i coefficients.

The use of generating functions may be of help to handle these two equations. In fact, if we consider the generating functions for r_i and $\overrightarrow{r}_i$,

$$R(x) = \sum_i r_i x^i, \qquad \overrightarrow{R}(x) = \sum_i \overrightarrow{r}_i x^i \tag{11}$$

then, after some manipulation typical for generating functions (e.g. [7]) we arrive to the following result

$$\overrightarrow{R}(x) = x \sum_{j \geq 0} \overrightarrow{f}_j [\overrightarrow{R}(x)]^j = x \overrightarrow{F}(\overrightarrow{R}(x)) \tag{12}$$

and, similarly,

$$R(x) = x \sum_{j \geq 0} f_j [\overrightarrow{R}(x)]^j = x F(\overrightarrow{R}(x)). \tag{13}$$

From the generating functions, we might recover the elements r_i, $\overrightarrow{r}_i$ composing them. Unfortunately, equations (12), (13) may be difficult to solve, depending on the degree probability distribution p_i which controls the whole introduced measures [7].

But actually, we are not interested that much in the single values of r_i, $\overrightarrow{r}_i$. In fact, it is easier and more useful to measure the average number $\langle r \rangle$ of peers that receive a given event through the dissemination protocol. To this aim, we can employ the typical formula for generating functions $\langle r \rangle = R'(1)$. In fact, taking the first equation of (11), differentiating and evaluating the result for $x = 1$, and since $r_0 = 0$, we have $R'(x)\Big|_{x=1} = \sum_i i r_i$, which is the mean value related to the distribution of r_i coefficients. We already observed that the coefficients of the introduced generating functions are probabilities, and thus $F(1) = \sum_i f_i = 1$, and similarly $\overrightarrow{F}(1) = 1$, $R(1) = 1$, $\overrightarrow{R}(1) = 1$. Hence, taking (13) and differentiating

$$\begin{aligned} \langle r \rangle &= R'(1) = \left[F(\overrightarrow{R}(x)) + xF'(\overrightarrow{R}(x))\overrightarrow{R}'(x)\right]_{x=1} \\ &= 1 + F'(1)\overrightarrow{R}'(1). \end{aligned} \tag{14}$$

Similarly, from (12),

$$\begin{aligned} \overrightarrow{R}'(1) &= \left[\overrightarrow{F}(\overrightarrow{R}(x)) + x\overrightarrow{F}'(\overrightarrow{R}(x))\overrightarrow{R}'(x)\right]_{x=1} \\ &= 1 + \overrightarrow{F}'(1)\overrightarrow{R}'(1). \end{aligned} \tag{15}$$

Thus, $\overrightarrow{R}'(1) = \frac{1}{1-\overrightarrow{F}'(1)}$. This last equation allows to find the final formula for $\langle r \rangle$,

$$\langle r \rangle = 1 + \frac{F'(1)}{1 - \overrightarrow{F}'(1)} = 1 + \frac{\Gamma \langle p \rangle^2}{(1+\Gamma)\langle p \rangle - \Gamma \langle p^2 \rangle}. \tag{16}$$

Now, $\langle r \rangle$ is the number of peers that receive the event, regardless if they are subscribers or simply relay nodes. To obtain the average number of subscribers $\langle s \rangle$ that receive the event, it suffices to multiply $\langle r \rangle$ by the probability that a peer is a subscriber σ, hence obtaining $\langle s \rangle = \sigma \langle r \rangle$.

4.4 Percolation Probability

As it is quite typical in complex network theory, it is actually easier to examine infinite networks rather than just large ones. The analysis of infinite networks, under conditions similar to those of large scale networks, allows to understand important peculiarities of the real networks and on protocols executed by their nodes. For instance, it is possible to understand if a message can percolate through the network. This assumption is perfectly reasonable in our scenario, since we consider very large dynamical systems (with a number of nodes that tends to infinity) where peers know only their neighbours and manage contents based on local knowledge about nodes' subscriptions.

Equation (16) has a divergence when $(1 + \Gamma)\langle p \rangle = \Gamma \langle p^2 \rangle$, which signifies that the event reaches an infinite number of nodes, i.e. the event percolates through the network. By looking at the parameters, this situation depends, first, on the nodes' connectivity, i.e. the node degree probability distribution p_i. In fact, the degree probability distribution determines if the overlay has a giant component (i.e. the largest subset of connected nodes which scales with the network size, and thus has a number of nodes whose limit tends to ∞), rather than being partitioned into a set of components of limited size [7]. The event can be spread to a large (infinite) number of nodes only when there is such a giant component; otherwise, the event can be sent to a limited number of nodes only. Studies exist that allow to understand how to build networks with a giant component [6, 7].

Second, the value of σ has influence on both the number of subscribers to be reached and on the dissemination of events. In fact, the higher σ the higher the probability that a node has some of its neighbours which are subscribers to a given type of events; these nodes will be receivers of the event and subsequently they will act as relays for such event.

Third and final, the gossip probability γ determines if the message event is spread through the network even when the amount of subscribers for a given event type is small, i.e. when σ has a very low value. Of course, setting $\gamma = 1$ allows to flood the event to the whole component (from which the event has been originated). This is a fair choice when the network has a tree-like structure, or when the network has a very low clustering. Conversely, a low value for γ should be employed when there are loops in the overlay.

A completely different scenario is concerned with the situation when the network is formed by limited clusters only (there is no giant component). In such a case, in fact, the number of reached nodes does not grow proportionally with the network size, and a finite number of subscribers might receive a published event.

5. EXPERIMENTAL RESULTS

This section presents an assessment performed by considering the analytical model and results obtained through a simulation of the distributed protocol. The two approaches provide similar outcomes. In particular, when the theoretical model estimates that an infinite amount of nodes is reached through the dissemination, simulations show that a significant portion of the simulated network receives the events, as expected.

The focus here is on network coverage. Another important metric to consider is the number of sent messages. In this sense, the protocol ensures that peers disseminate a given event at most once. Moreover, the tree-like structure of the overlay limits that multiple copies of the same event are received by a peer.

5.1 Theoretical Model

We employed the framework presented in Section 4 to assess the performance of the protocol, based on the overlay topology, i.e. node degree distribution, the subscription probability σ and the gossip probability γ. Figure 1 shows

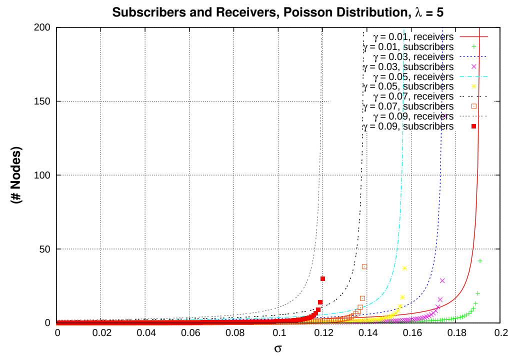

Figure 1: Receivers and subscribers: topology based on a Poisson degree distribution with mean $\lambda = 5$.

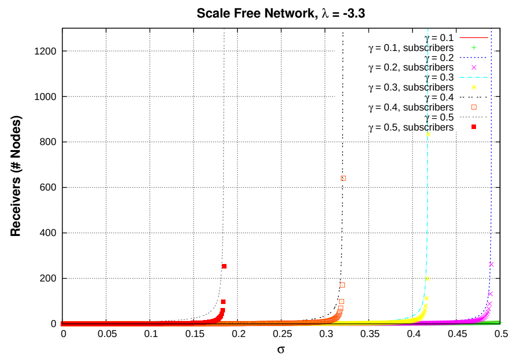

Figure 2: Receivers and subscribers: scale-free topology with exponent $\lambda = -3.3$.

the number of nodes receiving an event, spread through the network, when the unstructured overlay has a topology based on a Poisson node degree distribution with mean value $\lambda = 5$ (we tested the framework with other λ values, obtaining similar results). Lines in the chart correspond to the whole number of receivers (i.e. relay nodes and subscribers), while points correspond to the number of subscribers. Results are obtained varying the value of σ (on the x-axis), i.e. the portion of subscribers present in the overlay.

The figure shows that, for each specific γ value, there is a phase transition, i.e. as σ is varied there is an abrupt increment on the number of receivers (and subscribers), passing from a limited value to ∞, i.e. the event percolates through the network. This phase transition depends on the parameters used to set the distributed system. In fact, the value of σ not only represents the subscription probability, but it influences also the event dissemination in the overlay (a node forwards with probability 1 the event to each of its neighbours that subscribed to that event). Finally, the value of γ does not change the trend of the curves; basically, the higher γ the smaller the value of σ to have a transition.

Similar considerations can be made for Figure 2, where the estimated amount of receivers and subscribers is reported for a scale-free network with a degree distribution $\sim p^{\lambda}$, with $\lambda = -3.3$. The chart shows that for each curve there is a phase transition, where the number of receiving nodes passes from a limited (low) value to an infinite number.

5.2 Simulation

In order to assess the theoretical model proposed in the paper, we have built a discrete-event simulator mimicking

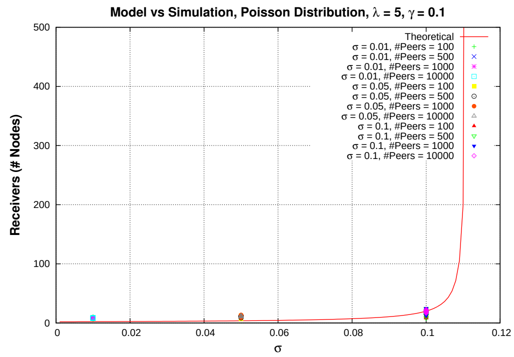

Figure 3: Model vs Simulation: topology based on a Poisson degree distribution with mean $\lambda = 5$.

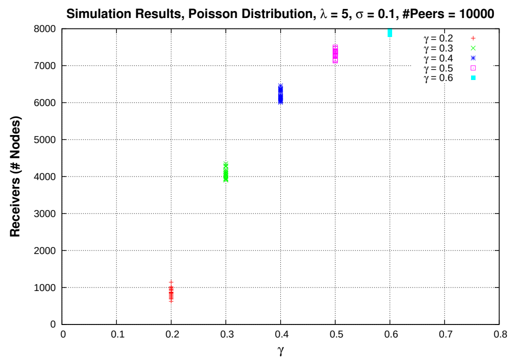

Figure 4: Model vs Simulation: Poisson degree distribution, varying γ above the phase transition. Number of receiving nodes obtained through simulation (the model returns an infinite sub-graph).

the presented protocol. The simulator was written in C code, by employing the GNU Scientific Library. It allows to test the behavior of a given amount of nodes executing a publish-subscribe distributed system employing the explained protocol. In particular, the simulator generates a random network based on the chosen degree distribution. Once having assigned a specific target degree to each node, a random mapping is made so that links are created until each node has reached its own target degree. During the initialization phase, for each node a random choice was made, in order to set that node as a subscriber of the event type or not, based on the probability σ.

We varied the network topology, the number of nodes and statistical parameters characterizing the network degree distribution. For each network setting, we repeated the simulation using a corpus of 20 different randomly generated networks. For each network, we analyzed the dissemination of 400 events published by random nodes. In the results that follow, for each generated network we show the average number of receiving nodes, i.e. subscribers and relays; this number allows to understand if the distributed protocol is able to disseminate the event through the unstructured network, using the presented protocol.

5.2.1 Poisson Degree Distribution

Here, we show results for networks generated through a Poisson degree distribution. Figure 3 shows results obtained from simulation and the theoretical model. We simulated different corpuses of networks, varying the number of nodes

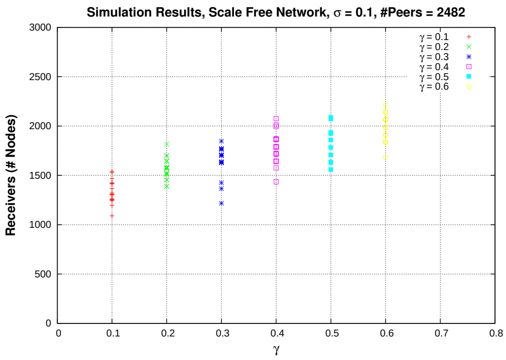

Figure 5: Model vs Simulation: scale free network, $a = 6, b = 1$, varying γ above the phase transition. Number of receiving nodes obtained through simulation (the model returns an infinite sub-graph).

and the value of the gossip probability γ. Each point in the chart corresponds to the average number of receivers for a simulated network. The line corresponds to the theoretical value measured using equation (16). All results from the simulations lie near the theoretical value, regardless on the considered number of simulated network nodes. Hence, the model captures the behavior of the distributed protocol.

Figure 4 shows results obtained in our simulations when $\sigma = 0.1$ while varying γ, above the phase transition. According to the model, the system is above the phase transition. Hence, assuming an infinite number of nodes in the network, an infinite number of receivers is reached. As concerns simulations, instead, we expect that a non-negligible portion of nodes is reached during the dissemination of an event. Of course, since the dissemination is based on rather low values of γ, σ probabilities, and since the network clustering of these considered networks is quite low (we employ a random attachment process to build links in the network [7]), it is unlikely that all network nodes receive the disseminated event. In fact, because of the tree-like structure of the net, every time we decide not to exploit a link, we might cut away some branch (and consequently some sub-graph) of the overlay. Results confirm our outlook. A non-negligible portion of nodes is reached in each configuration. Yet, the whole overlay is not covered completely. The amount of the reached nodes increases with the varied parameter γ. Of course, the entire net (or at least, the component to which the node belongs) can be reached by flooding the event.

Similar results were obtained for different networks built varying the statistical parameters of the random graph and the values of γ, σ. In substance, all this means that the protocol is able to spread a given event in the network in random graphs with Poisson degree distributions.

5.2.2 *Scale-Free Networks*

Scale free networks gained a lot of interest in recent years. These networks are characterized by a degree distribution following a power law. The presence of hubs, i.e. nodes with degrees higher than the average, has an important impact on the connectivity of the net. The interest on scale-free networks in this work relates to the fact that several P2P systems are indeed scale-free networks [3, 7].

To build scale-free networks, our simulator implements the construction method proposed in [1]. It builds a network of fixed size, characterized by two parameters a, b. More specifically, the number of nodes y which have a degree x satisfies $\log y = a - b \log x$, i.e. $y = \lfloor \frac{e^a}{x^b} \rfloor$. Thus, the total number of nodes is $N = \sum_{x=1}^{\lfloor e^{\frac{a}{b}} \rfloor} \frac{e^a}{x^b}$, being $\lfloor e^{\frac{a}{b}} \rfloor$ the maximum possible degree of the network, since it must be that $0 \leq \log y = a - b \log x$. Once the number of nodes and their degrees have been determined, edges are randomly created among nodes until nodes reach their desired degrees.

As made above for random graphs, Figure 5 shows results obtained in our simulations when we employ a scale-free network topology, with $\sigma = 0.1$ while varying γ, above the phase transition (similar tests were performed varying σ and keeping γ fixed, obtaining similar results). Again, based on the model an infinite number of receivers is reached (assuming a network of infinite size). From the simulations, a non-negligible portion of nodes is reached during the dissemination of events, that increases together with the γ parameter. Indeed, it is interesting to observe that when $\gamma = 0.6$, $\sigma = 0.1$ almost all network peers receive the event during the dissemination, and thus, almost all subscribers receive the published events. In the scenarios reported in the pictures, in fact, we employed scale-free networks generated through the choice of $a = 6$, $b = 1$, resulting in networks composed of 2482 nodes. In this case, simulation results provide average results above 2200 nodes. Again, this result is in accordance with the outcomes from the model, stating that an infinite number of nodes is reached with these settings. We performed simulations with different networks of different sizes, obtaining similar results (not reported here).

6. CONCLUSIONS

This paper analyzed the performance of an unstructured P2P overlay network that exploits a very simple dissemination strategy to build P2P publish-subscribe systems. Results show that by tuning the gossip probability it is possible to spread contents through the overlay, without the need to resorting to sophisticated dissemination strategies built on top of costly structured distributed systems. This is true when networks are large in size and the number of subscribers is not negligible.

References

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Math*, 10:53–66, 2000.

[2] A. K. Y. Cheung and H.-A. Jacobsen. Load balancing content-based publish/subscribe systems. *ACM Trans. Comput. Syst.*, 28:9:1–9:55, December 2010.

[3] G. D'Angelo and S. Ferretti. Simulation of scale-free networks. In *Simutools '09: Proc. of the 2nd Int. Conf. on Simulation Tools and Techniques*, pages 1–10, Belgium, 2009. ICST.

[4] J. Eberspächer and R. Schollmeier. First and second generation of peer-to-peer systems. In *Peer-to-Peer Systems and Applications*, volume 3485 of *Lecture Notes in Computer Science*, pages 35–56. Springer, 2005.

[5] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35:114–131, June 2003.

[6] A. D. Flaxman, D. Gamarnik, and G. B. Sorkin. Embracing the giant component. *Random Struct. Algorithms*, 27:277–289, October 2005.

[7] M. E. J. Newman. *Random graphs as models of networks*, pages 35–68. Wiley, 2005.

[8] W. W. Terpstra, J. Kangasharju, C. Leng, and A. P. Buchmann. Bubblestorm: resilient, probabilistic, and exhaustive peer-to-peer search. *SIGCOMM Comput. Commun. Rev.*, 37:49–60, August 2007.