Dynamical Networks: Analysis of P2P Networks

Surender Reddy Yerva[*],
Ecole Polytechnique Fédérale de Lausanne.

Abstract

Peer-To-Peer(P2P) networks are becoming ubiquitious.There are different kinds of P2P architectures and each of the architecture has its own merit and demerits. Peers interact and make decisions based on the local information. This local decisions have impact on the global behavior of the network. As a part of this project, first we model the dynamics involved in the network construction phase and then we consider different kinds of p2p architecutes and explore the networks that are formed at the end of network-construction phase. We are interested in the graph properties like: diameter of the network, node clustering and if small world and scale-free properties are exchibited. Studying this would help us comment on the behaviour of the network during its regular P2P operations(resource lookup).

Introduction

A peer to peer (P2P) network is an overlay network. Unlike traditional client-server model, a peer in P2P network plays the role of both client and server. It acts as a client when it consumes resources of other peers and as a server when its resources are used by the other peers.

For ease of resource-lookup all the resourses on the network are mapped onto resource-identifier space. This is usually done by using some hash-functions. There are different peer-to-peer network architectures.

- Centralized Structure:In this architecture, when a peer has some resource to share, it publishes this information on some central server. When it needs to consume a resource on the network, it searches the central server and once the destination-peer is identified, it contacts the destination peer for resource. Here the central-server acts as single source bottleneck. Napster[2] had this kind of architecture. This P2P architecture will not be studied in this project.

- Unstructured P2P: Here each peer is connected to some number of random peers. Here there is nothing as distributed index. If a peer has to search for some resource it just floods the network. Gnutella[8] and FreeNet[3] are two such kinds of networks. Here there can be variation where a peers likes to connect to well connected peers rather than some random peers.

---

[*]surenderreddy.yerva@epfl.ch

- Structured P2P: The resource identifier space in this architecture is split into segments and each peer will be responsible for one of the segments. To have reference to the complete index space each peer maintains links to some number of peers responsible for the other segments. The links are represented at a peer in the form of routing table. Chord[7], PGrid[1] etc are such kind of structures.

- Hybdrid P2P: This architecture is a combination of unstructured and structured architectures. Here few peers noted as Super-Peers form an unstructured network and the remaining peers form a structured network and will be attached to one of the super peer. Whenever a resource is searched for, it is initially looked for in the local structured network and if no resource is found the search request is forwarded to the local super-peer. The superpeers will inturn try to answer the query.

One of the important phase in the lifetime of a Peer-To-Peer network is the initial network construction phase, which is referred to as BOOTSTRAP phase. The involved peers meet and decide on the overlay connections based on local information. The bootstrap phase algorithms(which represent the dynamic process at a node) decide the connections to be maintained, which in turn decide the global structure of the network. At the end of bootstrap phase, each peer maintains references to some chosen number of peers in the form of local routing table. This routing table along with the search strategy will be used to answer any resource lookup queries. The latency for the query results and the network bandwidth consumed will depend on the kind of overlay structure that will be formed.

In this project, formally we model the dynamics involved in the bootstrap(network construction) phase of a strucutured peer to peer architecture. The formalism can be easily extended to other kinds of architectures. As a part of this project we are interested in different properties of the resulting final network. The final structure of the p2p network depends on the kinds of bootstrap phase algorithms. Given the time constraint we just describe the model representing the dynamics involved in the bootstrap phase and we simulate the expected graph structures for different p2p architectures and then infer properties on those graphs. Based on the properties we conclude about the complexity involved in the normal P2P operations.

Formal Description

In this section we will have the formal description of the bootstrap phase of the structured P2P network. Here the network graph is time varying during the bootstrap phase but once the network is constructed the graph is no more time varying.

In the structured p2p the resources are mapped to resource identifier space. This is accomplished by the use of hash functions. We assume that the resources are mapped on to a [0 1] scale. Initially each peer is responsible for the complete identifier space. As peers keep meeting during the bootstrap phase, the peers try to take responsibility of some segment of the resource identifier space. At the same time, some references are kept to the other resource identifier segments, so as to have access to complete identifier space.

Following are the different aspects of the model:

Graph

- V: Peers that are part of P2P Network.

- E: The edges are directed edges. There is an edge from Peer P1 to Peer P2 if there is an entry of P2 in the routing table of P1.

Dynamical System

At each vertex v, we have:

- State Variables:

  - $X^{(v)}$: The state space at a vertex is given by [a b) where $0 \leq a \leq b \leq 1$. Range of the resource identifier space for which the vertex is responsible.

  - $T^{(v)}$: The state space at a vertex represents the routing table. This table maps a resource-id to the responsible peer.

- $S^{(v,in)} = V \cup \{0\}$

  At a vertex "v" one of the peer "w $\in$ V" is chosen for bootstraping. "v" and "w" will now be involved in an exchange process. The choice of the peer with which bootstraping is done will be taken care by bootstrap algorithms at the node. This we dont discuss/model as it is out of scope.

- $S^{(v,out)} = V \cup \{0\}$

  Whenever the bootstraping is complete with a particular peer, the peer id will be sent as output. This information will be used by the bootstrap algorithm for bookkeeping. 0 means bootstrap process failed while "w $\in$ V" means that the current peer and "w-peer" were involved in successful bootstrap process.

At each edge e, we have:

- $S^{(e)} = (X \times T) \cup \{0\}$

  When two peers are involved in the bootstrap phase each of them exchange their current state(the identifier space for which it is responsible and its current routing table). While "0" means no information is exchanged on an edge e.

Dynamical System at vertex v:

- State Transition Functions:

  - $F^{(v)}$: This state transition function based on the identifier space of the current peer and the other peer involved in bootstraping, decides the new resource identifier segments to which the peers will be responsible.

  $$F^{(v)}(x, s^{(e_1)}, \ldots, s^{(e_n)}) = y$$

where x = [a b) with $0 \leq a \leq b \leq 1$ and only one of the edge is non-zero, which is, $s^{(e_i)} = $ [c d) with $0 \leq c \leq d \leq 1$. Finally y = [e f) with $0 \leq e \leq f \leq 1$: the new resource id range for which this peer will be responsible and is decided based on the bootstrapping algorithm used.

- $R^{(v)}$: This state transition function takes the local routing table and routing table of the remote bootstrap peer as inputs. Based on the algorithms involved a new routing table is formed.

$$R^{(v)}(\text{t},s^{(e_1)},\ldots,s^{(e_n)}) = \text{z}$$

where $\text{t} \in \text{T}$ is the local routing table and only one of the edge is non-zero, which is, $s^{(e_i)} \in \text{T}$ is the routing table of the remote peer. Finally $\text{z} \in \text{T}$ is the new local routing table and is decided based on the bootstrapping algorithm used.

- Output Function : $G^{(v)}$
  The output function based on the input $s^{(v,in)}$ would choose one of the edges $s^{(e_i)}$ and output the current state (x,t) on this edge. Also based on the outcome of bootstraping a value of remote-peer-id or $\{0\}$ is output on $s^{(v,out)}$.

$$G^{(v)}(\text{x,t},s^{(e_1)},\ldots,s^{(e_n)},s^{(v,in)}) = \left(s^{(e_{n+1})},\ldots,s^{(e_{n+k})},s^{(v,out)}\right)$$

where "x,t" is the state associated with given vertex representing the current resource-id range the vertex is responsible and the local routing table respectively.

During the bootstrap phase, based on the model described and on the involved bootstrap algorithms, a structured P2P network will be formed. The complete network is represented in the form of routing tables at each of the peers.

We assume the bootstrap phase is run and we have the P2P network structure available. In the next section we do graph analysis on different kinds of graphs that are typically formed during the different P2P network architecture bootstrap phase. Once the network is stabilized the latency involved in answering a query depends on the properties of the graph structure that is formed.

Object of Interest

In this project we would like to explore the network properties like

- Number of components, clustering coefficients and the degree distribution of the nodes.

- Diameter of the network and average distances (Average number of hops required for answering a query)

- Whether the network exhibits small-world and scale-free properties

for the various p2p scenarios

- Unstructured P2P Networks: Erdos-Renyi random graphs and Preference based random graphs

- Structured P2P Networks: Each peer maintains links to (a) peers which are exponential distances away (b) peers which are immediate neighbors and (c) some peers which are immediate neighbors and some random peers.

Simulations and Results

Matlab along with MatlabBGL[4] is used for the simulations.

Unstructured P2P

In this section we study two kinds of unstructured P2P networks. For unstructured P2P case the edges among the nodes are undirected.

Modified Erdös-Réyni Random Graphs

For the first scenario, each peer tries to maintain some fixed number of random connections. Here we use modified version of Erdos-Reyni random graph construction algorithms[6], and at the same time making sure that there is only one single component.

In this experiment we choose number of nodes N = 8000 and each node tries to maintain k=3 number of random connections.[1] Observations:

Figure 1: Modified Erdos-Renyi based random graph

- Number of components N_c=1.

- As the graph is undirected, the degree of node is same as indegree or outdegree of the node. Even though we have chosen a constant "k" to be number of edges per node, as there is randomness involved, we do not expect the degree distribution to be a constant. This distribution is shown in the figure 2 Also at the same time we do not see any power law distribution of degree, thus this graph structure does not exhibit scale-free property.

- $d_{max} = 11$; $d_{avg} = 6.54$; [2] Since these values are not of the order of log(N), the small world property is not exhibited by this structure.
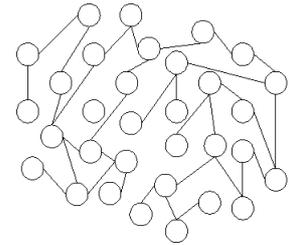
---

[1]Care is taken to ensure that there is only one single component so that every peer can reach every other peer

[2]Even though we have small values for these parameters, it will not be possible for a query to have these number of hops because it is difficult to choose a shortest path in a decentralized environment.

- Clustering coefficient: C = 0.07. Low value of clustering coefficient indicates it is less likely that the neighbors of a node to be neigbhors among themselves.
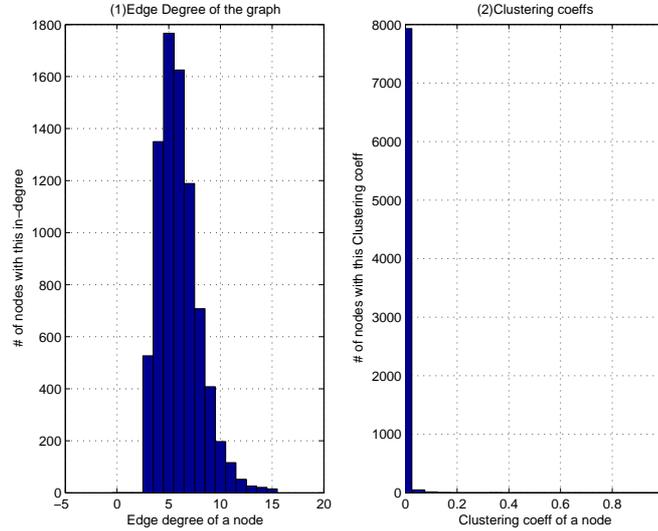

Figure 2: Unstructured P2P based on modified Erdos-Renyi random graphs

Preferential Attachment Random Graphs

In the second scenario each peer tries to maintain random connections to nodes based on the popularity. For this experiment we have the number of nodes N = 8000. Again each node tries to maintain k=3 number of random connections. To take preferential attachment into consideration, the simulation of the graph is done in two phases. In the first phase at any intermediate time a new node tries to connect to existing nodes based on degree distribution of the existing nodes. In the second phase the remaining k-1 connections are made at random. Observations:

Figure 3: Random graph considering preferential attachment

- Number of components N_c=1.

- As this is undirected graph the degree of a node is same as indegree or outdegree of the node. In the figure 4 we see the degree distribution.Unusually we find some nodes with very high degree(some nodes have degree of 250 or so). Here we see power law kind of degree distribution, thus this graph structure does exhibit scale-free property.
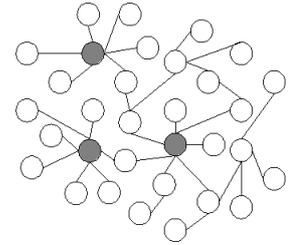
6

- $d_{max} = 6$; $d_{avg} = 4.32$; We see small values for this parameter inspite of huge N=8000. This indicates a compact graph where nodes are very close to each other and there are some nodes(hubs) with huge fanout. [3]

- Clustering coefficient: C = 0.08. Low value of clustering coefficient indicates it is less likely that the neighbors of a node to be neigbhors among themselves. This is because the way the graph is constructed.
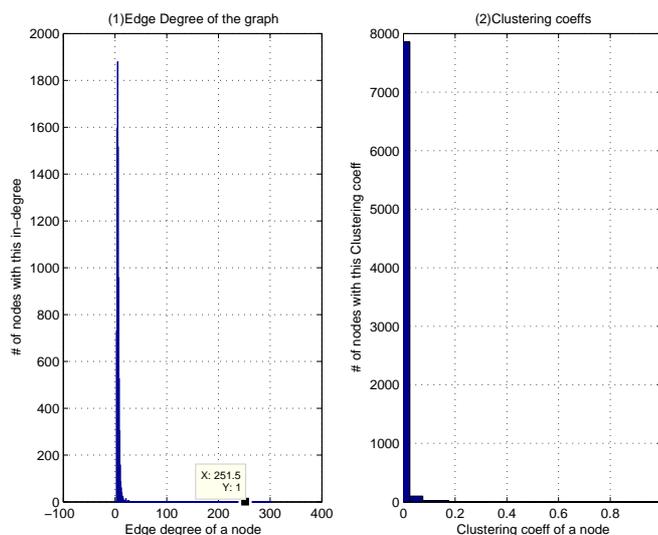

Figure 4: Unstructured P2P based on preferential attachment

---

[3]This kind of low values are desirable for the search operations in P2P. But this will not be feasible because majority of the queries would pass through these hubs and if the hubs are not computationally strong then they will not be able to withstand the load.

Structured P2P

For all the structured P2P scenarios, as assumed the resources are mapped to the resource-identifier space. The peers are numbered and arranged according to the resource-id segment the peer is responsible. Additionally each peer maintains references to the other peers.

References at exponential distances

In this scenario, a peer maintains references to the peers that are increasing exponential distances away. For example peer P_1 maintains references to the peers that are at distances of 1,2,4,8,16,...
i.e. to the peers P_2,P_3,P_5,P_5,P_{17},....
The experiment simulates N=8000 nodes, and each node tries to have logN = k = 13 number of references to the other peers. Observations:

- Number of components N_c=1.

- The outdegree of each of the node is a constant, because it is the parameter of the simulation.

- As there is no randomness involved, the indegree of each of the node is observed to be a constant. As we do not see any power law distribution of indegree and outdegree, this graph structure does not exhibit scale-free property.

Figure 5: Peer references at exponential distances

- $d_{max} = 12$; $d_{avg} = 6.432$; Since these values are of the order of log(N), the small world property is exhibited by this structure.

- Clustering coefficient: C = 0.125. As the references maintained by a node are at increasing exponential distances, it is less likely that the neighbors of this node to be neighbors among themselves.
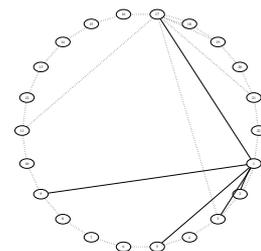
References to immediate neighbors

For this scenario, each peer maintains references to some fixed number of references to the immediate numbers. For example, if we assume a peer maintains k=6 references, then peer P_1 maintains references to the peers that are at distances of 1,2,3 on either direction i.e. to the peers P_{N-3},P_{N-2},P_{N-1},P_2,P_3 and P_4. For simulation, we use Newmann-Watts[5] method of random graph generation.
For this experiment we have number of nodes N=8000 and each node maintains k=14 references i.e. 7 in each direction. Observations:
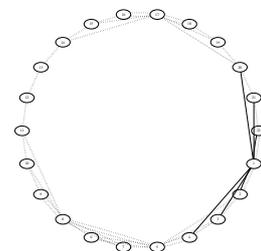

- Number of components N_c=1.

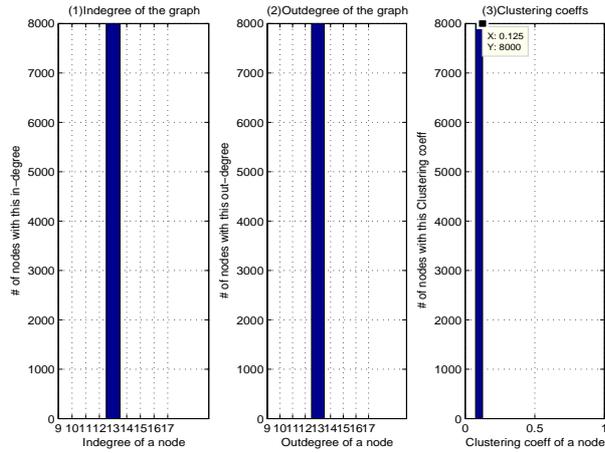Figure 7: Peer references immediate neighbors

Figure 6: Peer references at exponential distances

- The outdegree of each of the node is a constant, because it is the parameter of the simulation.

- As there is no randomness involved, the indegree of each of the node is observed to be a constant. As we do not see any power law distribution of indegree and outdegree, this graph structure does not exhibit scale-free property.

- $d_{max} = 572$; $d_{avg} = 286.12$; Since these values are of the order of log(N), the small world property is exhibited by this structure. Here we observe the values to be very large because to reach distant peers, the paths have to go through immediate neighbors and their immediate neighbors and so on.

- Clustering coefficient: C = 0.675 . Here the clustering coefficient is higher than the previous case. Here the neighbors of a node are more likely to be neighbors among themselves.

References to immediate neighbors and some random distance neighbors

This is an in between scenario of the above two scenarios. A peer maintains "k" fixed number of references to the other peers. Out of this "k" some fixed number "m" number of references are made to the immediate neighbors, while the remaining "k-m" references are maintained with some random peers.

For example when k=6 and m=4, the peer P_1 maintains references to immediate neighbor peers $P_{N-2}, P_{N-1}, P_2, P_3$ and two random peers say P_{11} and P_{17}.

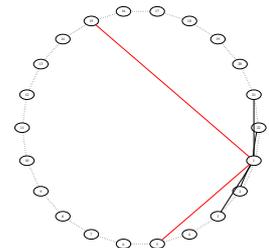Here N = 8000 nodes, while k = 14 of which 12 are immediate

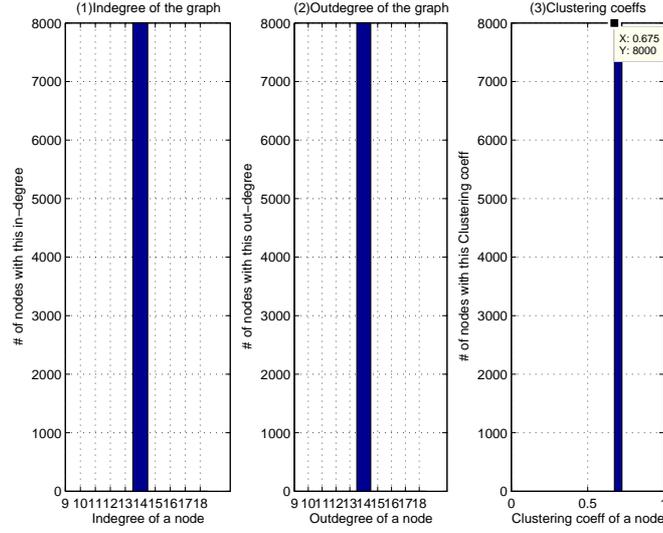Figure 9: Peer references some immediate neighbors and some random neighbors

9

Figure 8: Peer references immediate neighbors

neigbors (6 on either side) and m = 2 random references.
Observations:

- Number of components N_c=1.

- The outdegree of each of the node is a constant, because it is the parameter of the simulation.

- As there is some randomness involved, the indegree of each of the node is observed to be spread around a constant value. As we do not see any power law distribution of indegree and out-degree, this graph structure does not exhibit scale-free property.

- $d_{max} = 7$; $d_{avg} = 4.832$; Since these values are of the order of log(N), the small world property is exhibited by this structure.
  Here we observe the values significantly less than the previous "immediate neighbors" case because we are maintianing few references to distance peers at random. So when a peer has to reach a distant neighbor, using the distant references a peer can reach the neighborhood of distant peer. Once in the neighbor by making use of immediate neighbors a peer converges to the distant peer much faster.

- Clustering coefficient: C = 0.475. As this scenario is in between the above two scenarios, the clustering coefficient as expected is greater than the first case and less than the second scenario.
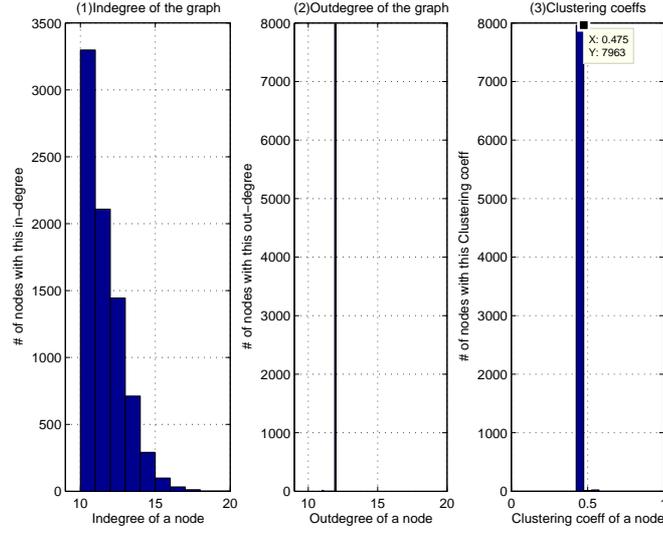
Figure 10: Peer references some immediate neighbors and some random neighbors

Conclusion

We have modelled the dynamics involved in the bootstrap phase of a structured p2p network. We studied the various graph properties on the different p2p network architectures. We summarise the main properties observed in the different p2p architectures in the following table.

Table 1: Graph properties of different P2P architectures

Structure	N	Nc	In	Out	Dmax	Davg	logN	SW	SF	C
UnStructured										
Erdos-Renyi(modified)	8000	1	13	13	11	6.58	13	F	F	0.07
Prefer	8000	1	13	13	6	4.32	13	F	T	0.08
Structured										
Exponential Distances	8000	1	13	13	12	6.43	13	T	F	0.125
Immediate Neighbors	8000	1	14	14	572	286	13	F	F	0.675
Hybrid	8000	1	13	13	7	4.83	13	T	F	0.475

Table 2: Legend

N	Number of nodes in the graph
Nc	Number of components/strong-components
In	Indegree of a node
Out	Outdegree of a node
Dmax	Diameter of the graph
Davg	Avegare distance between two nodes
SW	Small world property
SF	Scalefree network
C	Clustering coefficient

References

[1] Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. P-grid: a self-organizing structured p2p system. *SIGMOD Rec.*, 32(3):29–33, 2003.

[2] Bengt Carlsson and Rune Gustavsson. The rise and fall of napster - an evolutionary approach. In *AMT '01: Proceedings of the 6th International Computer Science Conference on Active Media Technology*, pages 347–354, London, UK, 2001. Springer-Verlag.

[3] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: a distributed anonymous information storage and retrieval system. In *International workshop on Designing privacy enhancing technologies*, pages 46–66, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[4] David Gleich. Matlabbgl. *http://www.stanford.edu/ dgleich/programs/matlab_bgl/.*

[5] M. Newman, D. Watts, and S. Strogatz. Random graph models of social networks. In *Proc. Natl. Acad. Sci., to appear.*

[6] Erdos P. and A. Renyi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, 2001.

[8] Yong Wang, Xiaochun Yun, and Yifei Li. Analyzing the characteristics of gnutella overlays. In *ITNG '07: Proceedings of the International Conference on Information Technology*, pages 1095–1100, Washington, DC, USA, 2007. IEEE Computer Society.