Workshop Report for the 3rd International Workshop on Peer-to-Peer Systems (IPTPS 2004)

February 26–27, 2004

Sea Lodge Hotel, La Jolla, CA, USA

Sriram Ramabhadran, Sumeet Singh, and Kiran Tati

Department of Computer Science and Engineering
University of California, San Diego
{nramabha,susingh,ktati}@cs.ucsd.edu

1 Miscellaneous

Shiding Lin, Qiao Lian, Ming Chen, and Zheng Zhang, **A Practical Distributed Mutual Exclusion Protocol in Dynamic Peer-to-Peer Systems.** Presented by Zheng Zhang.

C: Some of the ideas are similar to previous work on Byzantine quorum systems.

Q: How is a consistent ordering on client requests obtained? **A:** A logical clock is used to order client requests. Looser semantics result in a quasi-FIFO ordering of requests rather than strictly FIFO.

Q: Mapping between resource and its replicas is non-atomic due to the presence of churn. How does this affect the protocol? **A:** The protocol relies on the DHT to eventually stabilize and present a consistent replica set. Large churn rates could result in no overlap between the previous set of replicas and the new set of replicas, and therefore the protocol performance may degrade (with possible correctness issues).

Q: What kinds of assumptions are made about the DHT and synchrony in the formal proof of correctness? **A:** One assumption is that two nodes do not simultaneous claim ownership of a replica (could arise due to churn).

Nicolas Christin, John Chuang, **On the cost of participating in a peer-to-peer network.** Presented by Nicolas Christin.

Q: Does your analysis extend if you consider maximum load rather than average loads? In particular, does the star topology result in the social optimum? **A:** We need to extend the analysis to other utility functions, such as non-linear functions of load and delay. More general utility functions model cases where maximum rather than average is important.

C: Randomness in choosing node identifiers is a source of asymmetry because a real system will not be perfectly balanced in terms of load, messages routed, etc. It would be interesting to extend the analysis to account for this.

Q: How will hierarchical constructions where small clusters of nodes are tied together using a DHT help, given that most requests will still have to be routed through the DHT? **A:** It depends on how the clusters are constructed, along with the choice of routing algorithm. A desirable construction is where most of the requests are satisfied locally (through a caching mechanism), while only cache misses go through the DHT.

Mema Roussopoulos, Mary Baker, David Rosenthal, TJ Giuli, Petros Maniatis, Jeff Mogul, **2 P2P or Not 2 P2P?** Presented by Mema Roussopoulos.

C: Lack of trust is mentioned as a danger sign but it is also possible that lack of trust among participating entities motivates a distributed design. This is because some applications are too critical to trust any one centralized entity to implement. DNS is one such example.

Q: Peer-to-peer does not necessarily imply nodes are owned by different entities; it may be regarded as a paradigm for designing large-scale distributed systems. Therefore, it is not clear that somebody with a large budget may not want a peer-to-peer solution. **A:** In our model, peer-to-peer is considered to be a system with autonomous nodes rather than a distributed system with a common administration.

C: Mutual trust, accountability and manageability are all important considerations. Therefore it is not important what is the order of things appearing in the decision; they are all things to be considered when evaluating a peer-to-peer solution.

C: A distinction should be made between being able to design a peer-to-peer solution and wanting to use a peer-to-peer solution. In other words, technical issues should be separated from other issues.

Q: Areas where the decision tree indicates trouble spots are all active areas of research; therefore it is conceivable that a peer-to-peer solution will become desirable as the field matures. Is your decision tree predicting future trends with respect to the success of peer-to-peer in these areas? **A:** These areas are potential trouble spots where peer-to-peer solutions must be engineered carefully for them to be successful.

2 Networking

Hung-Yun Hsieh, Raghupathy Sivakumar, **On Transport Layer Support for Peer-to-Peer Networks.** Presented by Hung-Yun Hsieh.

Q: What is the relationship between your work and rate-less codes (IPTPS 2003)? **A:** Rate-less codes are application layer approaches used for file downloads from multiple sources. While the use of rate-less codes does not require the support from the transport layer, it incurs extra processing (for encoding and decoding) and communication (for transmitting redundant data) overheads. In addition, its support for multipoint-to-point communication is application-dependent (e.g., for non-real-time file downloads).

Q: If content is to be downloaded from multiple sources, there must be some way of verifying that the content is the same at all the sources. Shouldn't this be implemented at the application layer rather than at the transport layer? **A:** Yes, we assume that the peer-to-peer lookup service is responsible for providing the list of supplying peers with replicated content to the requesting peer. The validity of the data/content being transferred can be considered higher layer semantics (e.g., presentation or application layer). Note that even in the case of TCP, it does not verify the content being transferred–its functionality is merely to reliably transfer the data given by the application.

Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou, **Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming.** Presented by Venkat Padmanabhan.

Q: How sensitive is the performance of the system to the accuracy of the dropping decision? **A:** Identifying the source of congestion is important. For example, if congestion is occurring at a parent, it is best to switch to a better parent. If congestion is happening at a peer, the choice of which parent to drop is not absolutely critical. Again, there are considerations of base layer versus enhancement layer, where the base layer is much more important than the enhancement layer.

Ben Y. Zhao, Ling Huang, Anthony D. Joseph, John D. Kubiatowicz, **Rapid Mobility via Type Indirection.** Presented by Ben Zhao.

Q: Where are the Tapestry nodes (wired/wireless)? **A:** Tapestry is used to implement the overlay mobility infrastructure; the mobile end-hosts use a proxy mechanism.

Q: Is the ability to locate a PDA/laptop relevant, considering they are not likely to be content providers? **A:** We are providing the functionality (indirection service), there may be potential applications in the future.

Q: How is group formation done? **A:** Try to predict common mobility patterns, e.g., car-pooling.

Lidong Zhou, Robbert van Renesse, **P6P: A Peer-to-Peer Approach to Internet Infrastructure.** Presented by Lidong Zhou.

Q: Sites may have multiple IPv6 prefixes. How does this affect your architecture? **A:** P6P can easily support sites with multiple IPv6 prefixes. Such a site needs to install one routing table entry for each IPv6 prefix it owns.

Q: How inherent is IPv6 to your architecture; in other words, what features of IPv6 are used by P6P? **A:** P6P is a proposal to separate the network abstraction presented to end sites (hosts) from the core Internet infrastructure. We believe IPv6 is a reasonable abstraction for end applications. In theory, we could have picked any other appropriate abstractions. In practice, because OS support for IPv6 exists on end hosts, adopting IPv6 facilitates the deployment of P6P and is therefore essential for the practicality of P6P.

Q: How is P6P different from I3? **A:** Like I3, P6P also embrace indirection and the separation of identifiers from location. While I3 aims at presenting a general indirection infrastructure, P6P focuses on a concrete and practical proposal that is more likely to be adopted. In particular, unlike I3, which offers an entirely new API to end applications, P6P advocates IPv6, which is already supported in major operating systems.

Q: Can you quantify the routing overhead? **A:** P6P is not yet deployed, so quantifying overhead is difficult. The routing cost consists mainly of routing lookups and packet tunneling. The latter is relatively well understood because it has been widely used.

3 Routing

Jinyang Li, Jeremy Stribling, Thomer M. Gil, Robert Morris, Frans Kaashoek, **Comparing the performance of distributed hash tables under churn.** Presented by Jinyang Li.

Q: Do you take failed operations into consideration? **A:** Yes. All protocols retry failed/incorrect lookups for up to 4 seconds. So a low success rate (e.g., $<$ 90%) would translate into high lookup latency ($>$ 400 ms).

Q: Is the spread of points on your convex hull graph of any importance? **A:** The spread of points depends on the range of values we choose to vary for each parameter and hence is somewhat arbitrary and does not mean anything much. We also acknowledge that the convex hull found is conservative; it is possible that there are "better" parameter value combinations that we have not explored.

Q: How about protocols that adaptively tune the parameters? **A:** We presented convex hulls for different DHTs. However, that does not mean a given DHT can achieve its convex hull performance, as it needs to optimally pick its parameters. A protocol that adaptively tunes its parameters might operate closer to its ideal convex hull.

Q: In a real system with 100's of parameters it is difficult to isolate the effect of any one parameter using the approach used in this paper; for example, Oracle vs. DB2. Is this a limitation with your approach? **A:** We are lucky that the DHTs under study only have a couple of parameters. Even so, we find many parameters are not really meaningful to the DHT cost-performance tradeoff. The current method of varying all parameters simultaneously is not scalable to hundreds of parameters. However, for a real system with hundreds of parameters, it is likely most of them are not critical. It is conceivable to devise some optimizations to quickly identify these non-critical parameters and throw them out.

Q: What is the effect of congestion and queuing on your simulation model? **A:** We do not simulate queuing delays. We believe the load of DHT lookups and maintenance on the network is relatively small compared with application level traffic and will not be a source of congestion. As can be seen from our graphs, the bandwidth usage of DHT level traffic is very small.

Q: What is the impact of data transfer? **A:** Data transfer is not the focus of the work. One might argue that data transfer traffic might dwarf the mainte-

nance/lookup traffic of DHT operations. If so, this would imply we might want to operate DHTs with generous bandwidth costs constraints. And this paper says that all DHTs under study can be tuned to perform similarly to each other when they generate larger amounts of traffic.

Q: How does this DHT perform under different workloads? **A:** We are currently working on extending the model to include different workloads. The analysis in the paper focuses on one type of workload where the lookups are few. One can imagine things becoming different when lookups are abundant; less lookup hops lead to decreased lookup traffic, learning from lookups become more effective, etc.

Sergio Marti, Prasanna Ganesan, Hector Garcia-Molina, **DHT Routing Using Social Links.** Presented by Sergio Marti.

Q: Social networks are not uniformly distributed. This leads to different properties of the DHT, and in particular may lead to overloading of some well-connected nodes. How do you deal with this? **A:** Load-balancing is a problem, but simple strategies can be used to address this. For example, removing the friend links of the most well-connected nodes results in an approximately even distribution of load.

Q: Does trusting a friend translate to trusting a friend's machine? **A:** That is one of purposes of probability rating; trust in a friend is essentially the same as trust in the ability of the friend to manage his machine.

Q: What is the basis of modeling trust as a probability, i.e., what does it mean if you say that you trust someone with probability 0.9? How is the recursive trust effect modeled correctly, if I trust you and you trust somebody, how much do I trust your friend? **A:** One of the things we experimented with is using a step function instead of a multiplicative model for trust. Modeling trust as a probability is not perfect but does capture a lot of intuition.

Q: DHTs are supposed to be distributed, but social links can only be obtained by a centralized database. **A:** The idea is to use existing infrastructure such as AOL and Yahoo Messenger.

Rodrigo Rodrigues, Charles Blake, **When Multi-Hop Peer-to-Peer Routing Matters.** Presented by Rodrigo Rodrigues.

Q: For database sizes of 1TB, is there really a need for P2P? **A:** You do not want to mix technical issues with motivation issues for P2P.

Q: How many stale routing entries are there? And what is the cost incurred in querying stale data? **A:** We target a large fraction of stale entries (1/3 in our concrete example). The per-object availability can remain as high as desired by setting redundancy levels appropriately. Fetch latency can remain low provided we issue several fetch requests in parallel. The main problem would be increased timeouts on synchronous store operations, but we expect stores to be less frequent than fetches, and we think that waiting for all replicas to reply to a store is a bad policy.

C: Although the paper used HDD growth rates as an indicator of future trends, it is not necessarily the case that this is the same as growth in the amount of content available for sharing via P2P.

4 Load Balancing and Searching

Subhash Suri, Csaba Toth, Yunhong Zhou, **Uncoordinated Load Balancing and Congestion Games in P2P Systems.** Presented by Yunhong Zhou.

Q: Why use a bipartite graph to model the network when a node in a P2P system can be both a client and a server? **A:** It does not really matter; if a node is both a client and a server, it can be modeled as two nodes. Therefore a bipartite graph can be derived from the more general graph of the peers.

David R. Karger, Matthias Ruhl, **Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems.** Presented by Matthias Ruhl.

Q: In a real system, some objects are more popular or bigger in size than others. Therefore uniformity of key-space may not be the critical issue. **A:** The second protocol can handle arbitrary distributions of load and popularity. However, it has the disadvantage that nodes can move to arbitrary locations in the key-space, thereby potentially creating a security problem.

Q: Instead of using log(n) virtual nodes, can we distribute among the log(n) finger nodes to get the same effect? **A:** It is not clear that this would work.

Q: How does the routing work when nodes can move to arbitrary places in the key-space? **A:** See the details in the paper.

Boon Thau Loo, Ryan Huebsch, Ion Stoica, Joseph Hellerstein, **The Case for a Hybrid P2P Search Infrastructure.** Presented by Boon Thau Loo.

Q: How do you handle the fact that some inverted lists may be too big (popular keywords)? **A:** Stop-words need to be filtered. While this step alone is not sufficient to make keyword search feasible on a DHT, it is necessary.

Q: Apart from query, have you considered the effect of downloads? **A:** This is part of future work.

Q: What is the connection between rare items and rare query terms? **A:** On the average, rare items are likely to have rare keywords.

C: What is classified as a rare item is an artifact of your measurement methodology. Just because your search algorithm is not able to find it does not necessarily mean that the object is rare.

Q: Have you considered other hybrid architectures (for example a hierarchy of DHTs)? This may make sense to limit popular search items to a smaller number of nodes while using the global network to search for more rate items.

A: This is an interesting idea except that building DHTs over smaller sets of nodes does not exploit the scalability of DHTs.

Shuming Shi, Guangwen Yang, Dingxing Wang, Jin Yu, Shaogang Qu, Ming Chen, **Making Peer-to-Peer Keyword Searching Feasible Using Multi-level Partitioning.** Presented by Zheng Zhang.

(No questions, authors unable to attend.)

5 Miscellaneous

Alan Mislove, Peter Druschel, **Providing Administrative Control and Autonomy in Peer-to-Peer Overlays.** Presented by Alan Mislove.

(No questions recorded.)

Robbert van Renesse, Adrian Bozdog, **Willow: DHT, Aggregation, and Publish/Subscribe in One Protocol.** Presented by Robbert van Renesse.

Q: Almost any DHT can provide multicast, why is Willow special? **A:** In our case multicast is used internally to keep the tree together. Willow also supports aggregation and pub/sub routing.

Helen J. Wang, Yih-Chun Hu, Chun Yuan, Zheng Zhang, Yi-Min Wang, **Friends Troubleshooting Network: Towards Privacy-Preserving, Automatic Troubleshooting.** Presented by Helen Wang.

Q: How representative were the 20 troubleshooting cases you looked at? How likely is it that everybody will have the same value for the registry? **A:** We do not know how representative they are, the 20 cases are real-world troubleshooting cases some of which are gathered from our co-workers' troubleshooting experiences, and some of which are obtained from MS customer tech support. For the second question, the PeerPressure troubleshooting algorithm essentially measures the uniqueness of a suspect entry value among the sample set entry values and ranks the uniqueness metric among the suspects. This is effective for most cases. However, for a highly customized computer, its unique configurations would stand out as mis-configurations – and this is one source of false positives of the PeerPressure algorithm.

C: There is a lot of work in statistics that is potentially relevant, especially anonymous census. **A:** However, the existing techniques cannot address the anonymous accumulation of all the parameters that are needed by Friends Troubleshooting Network.

Q: How do you handle multiple entries? **A:** We currently do not handle multiple entries but this is a future avenue of research.

Q: PeerPressure ranks the suspects on their probabilities of being sick. Then what is next? **A:** The key benefit of PeerPressure is that it significantly narrows down the root cause candidate set. The next step is to trial-and-error with each entry from the lowest rank on and to see whether correcting that entry cures

the problem. It is true that this could be tricky since there can be rippling configuration changes from a single configuration change.

Q: The configurations may be site-specific, which you do not know ahead of time. How do you handle that? **A:** In this case, search could be scoped within a domain first. Basically such heuristics could be incorporated into search.

Brad Karp, Sylvia Ratnasamy, Sean Rhea, Scott Shenker, **Spurring Adoption of DHTs with OpenHash, a Public DHT Service.** Presented by Brad Karp.

Q: OpenHash supports both Put/Get and ReDir, which do you expect people to use more? **A:** We don't know yet. We expect that both will be heavily used.

Q: Why don't you want desktops to run this infrastructure? **A:** We are hoping to start by limiting the nodes that participate. As the project advances we may allow additional nodes to participate.

Q: Have you looked at the data management issues (for example infrastructure nodes may fail)? **A:** We have not looked at it yet.

Q: Since it is a public infrastructure don't you have to worry about security issues? **A:** These are open issues.

6 Applications

Emil Sit, Frank Dabek, James Robertson, **UsenetDHT: A Low Overhead Usenet Server.** Presented by Emil Sit.

Q: Does your scheme increase the potential for censorship? **A:** The data is fairly well distributed, an attacked would need to control a very large set of nodes.

Q: Existing system exploits geographic locality. Can you do that? **A:** Certainly you can form sub-groups to exploit locality.

Q: How does your approach compare to the alternate suggestion for USENET to use content on demand? **A:** Implementing content on demand in current USENET is technically very difficult, there any many issues to be dealt with.

F. Le Fessant, S. Handurukande, A.-M. Kermarrec, L. Massoulié,, **Clustering in Peer-to-Peer File Sharing Workloads.** Presented by Fabrice Le Fessant.

Q: Have you used your observations to predict how much improvement you can obtain as a result of the locality properties of P2P? **A:** We don't know yet as we have not done the analysis.

Virginia Lo, Daniel Zappala, Dayi Zhou, Yuhong Liu, Shanyu Zhao, **Cluster Computing on the Fly: P2P Scheduling of Idle Cycles in the Internet.** Presented by Virginia Lo.

Q: How do you make the quizzes indistinguishable from actual results? **A:** Sometimes the queries are the real thing.

Q: How do you see these reputation systems play out? **A:** It's a hard problem and we are not exactly interested in a totally secure reputation system.

7 Security

Baruch Awerbuch, Christian Scheideler,, **Robust Distributed Name Service.** Presented by Christian Scheideler.

Q: Does the assumption about join and leave rate apply to bad guys? **A:** You can come up with cryptographic puzzles, for example.

Q: What if the bad guys all behave very well for a while? **A:** The system is acting on a proactive fashion even when the peers are behaving well; the reliability argument is based on statistics.

Q: How much accuracy do you need in estimating N? Could N just be larger than the number of nodes in the system or do you have to get N right? **A:** You have to get N right to within a constant factor.

William K. Josephson, Emin Gün Sirer, Fred B. Schneider,, **Peer-to-Peer Authentication With a Distributed Single Sign-On Service.** Presented by Gün Sirer.

Q: How do you manage names in CorSSO? **A:** Every authentication server has its own namespace. The names from different authentication servers are combined to create a global name for each principal. These global names are associated with a shorter name at each application server. This approach allows CorSSO to separate the management of the name spaces from each other, yet link them to identify principals uniquely. It also allows legacy services to transition to use CorSSO without having to change their existing user names.

Q: Do you need a new public-private key pair to be generated for each service? **A:** We need a key pair for each sub-policy.

Q: Can the identifiers change over time? **A:** The identifier can change over time.

Q: How do you do replication of client certificates? **A:** Certificates are timed.

Antonio Nicolosi, David Mazières, **Secure Acknowledgment of Multicast Messages in Open Peer-to-Peer Networks.** Presented by Antonio Nicolosi.

(No questions recorded.)

8 Routing

Moni Naor, Udi Wieder, **Know thy Neighbor's Neighbor: Better Routing for Skip-Graphs and Small Worlds.** Presented by Udi Wieder.

Q: Why don't we get 50**A:** A non-hop is counted as two hops

Jingfeng Hu, Ming Li, Ning Ning, Weimin Zheng, **SmartBoa: Constructing P2P Overlay Network in the Heterogeneous Internet Using Irregular Routing Tables.** Presented by Ming Li.

(No questions.)

David R. Karger, Matthias Ruhl, **Diminished Chord: A Protocol for Heterogeneous Subgroup Formation in Peer-to-Peer Networks.** Presented by David Karger.

Q: Would it be better to have 2 small DHTs as opposed to 1 DHT? **A:** I don't think so, even if the node sets are disjoint, you can get more robustness against failures by attaching the two groups together. When you start building cache tables the lookup hops don't matter.

Q: Are we moving towards a service based model? **A:** OpenHash does have this idea of separating out this layer of service, you can open the OpenDHT layer to trusted nodes, but then who decides who are trusted? On the other hand, you can limit to high bandwidth nodes.

C: One situation where smaller DHTs may be better is when nodes are geographically close together.

Acknowledgements

We thank the authors for their valuable comments and feedback.