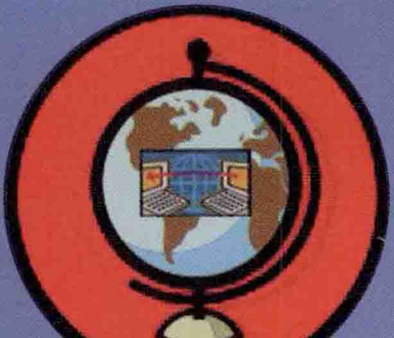
Hot Topics

LNCS 3640

Miguel Castro
Robbert van Renesse (Eds.)

Peer-to-Peer Systems IV

4th International Workshop, IPTPS 2005
Ithaca, NY, USA, February 2005
Revised Selected Papers



Springer

Miguel Castro Robbert van Renesse (Eds.)

Peer-to-Peer Systems IV

4th International Workshop, IPTPS 2005
Ithaca, NY, USA, February 24-25, 2005
Revised Selected Papers

Springer

Volume Editors

Miguel Castro
Microsoft Research
7 JJ Thomson Avenue, Cambridge CB3 0FB, UK
E-mail: mcastro@microsoft.com

Robbert van Renesse
Cornell University, Department of Computer Science
Ithaca, NY 14853, USA
E-mail: rvr@cs.cornell.edu

Library of Congress Control Number: 2005936069

CR Subject Classification (1998): C.2.4, C.2, H.3, H.4, D.4, F.2.2, E.1, D.2

ISSN 0302-9743
ISBN-10 3-540-29068-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29068-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11558989 06/3142 5 4 3 2 1 0

Preface

The 4th International Workshop on Peer-to-Peer Systems was held at Cornell on February 24th and 25th 2005. The IPTPS workshop continued to bring together researchers and practitioners from a variety of disciplines, including networking, theory, databases, security, and scientific computing. They described experimental findings, discussed challenges, and presented novel techniques.

We received 123 submissions. Submissions were limited to 6 pages, one page more than in previous years. The submissions were reviewed by a Program Committee consisting of 18 international experts from academia and industry. After a bidding process, each committee member was assigned 20 papers to review, generating 3 reviews for each paper. Controversial papers were assigned additional reviewers. The papers were then ranked based on originality, technical merit, and topical relevance, as well as the likelihood that the ideas expressed would lead to insightful technical discussions at the workshop. The program chairs suggested a program which was extensively discussed and revised by the entire committee to produce the final program.

We accepted 24 papers, which were organized into 8 sessions: Security and Incentives, Search, Multicast, Overlay Algorithms, Empirical Studies, and Network Locality, and two sessions on miscellaneous topics. Authors revised their submissions for a preproceedings distributed at the workshop. After the workshop, the authors revised their papers once more for the proceedings before you.

In order to focus discussions, attendance was restricted to Program Committee and to Steering Committee members and to at most two authors per paper. This resulted in 55 attendees from 9 countries at the workshop. Each session included 3 talks (20 minutes presentation and 5 minutes for questions), and a discussion panel (15 minutes). This format stimulated lively interaction between the participants of the workshop, and resulted in interesting and insightful discussions. The workshop was webcast, and followed by approximately 50 additional attendees from 10 countries.

The organization of the workshop involved many people. We thank the Program Committee for their hard work and for selecting an excellent program. Bill Hogan did an outstanding job with all of the local arrangements and maintaining the web server. Wenjie Wang and Sugih Jamin provided the live webcast. Twelve student scribes kept notes for the workshop report included in this proceedings. The Steering Committee provided guidance behind the scenes. Microsoft provided generous support. But, most of all, we wish to thank all participants of IPTPS 2005 for making this workshop a success.

July 2005 Miguel Castro and Robbert van Renesse

Organization

Workshop Co-chairs

Miguel Castro	Microsoft Research
Robbert van Renesse	Cornell University

IPTPS Steering Committee

Peter Druschel	Rice University
Frans Kaashoek	MIT
Antony Rowstron	Microsoft Research
Scott Shenker	UC Berkeley
Ion Stoica	UC Berkeley

Program Committee

Karl Aberer	EPFL
Mary Baker	HP Labs
Hari Balakrishnan	MIT
Bobby Bhattacharjee	Maryland
Miguel Castro	Microsoft Research
Peter Druschel	Rice
Hector Garcia-Molina	Stanford
Anne-Marie Kermarrec	INRIA
Barbara Liskov	MIT
Dahlia Malkhi	HUJI, Microsoft Research
Timothy Roscoe	Intel Research
Emin Gun Sirer	Cornell
Alex Snoeren	UC San Diego
Ion Stoica	UC Berkeley
Robbert van Renesse	Cornell
Maarten van Steen	VU Amsterdam
Helen Wang	Microsoft Research
Ben Zhao	UC Santa Barbara

Administrative Assistant

Bill Hogan	Cornell

Sponsoring Institution

Microsoft Corporation

Table of Contents

Security and Incentives

Search

Miscellaneous

Multicast

Overlay Algorithms

Empirical Studies

Miscellaneous

Exploiting Network Locality

Workshop Report

Mahesh Balakrishnan[1], Maya Haridasan[1], Prakash Linga[1], Hongzhou Liu[1], Venu Ramasubramanian[1], Sean Rhea[2], Manpreet Singh[1], Vidhyashankar Venkatraman[1], Vivek Vishnumurthy[1], Kevin Walsh[1], Bernard Wong[1], and Ming Zhong[3]

[1] Cornell University
[2] University of California Berkeley
[3] University of Rochester

Session 1: Security and Incentives

A Self-Repairing Peer-to-Peer System Resistant to Dynamic Adversarial Churn. Presented by Stefan Schmid.

Q: Is d (the dimensionality of the hypercube) constant? Does it scale? **A:** Yes, it scales. d is logarithmic in the number of peers in the system.

Q: Presumably in some cases you need to copy data between the core and periphery of nodes. Do you have any analysis? **A:** When a new peer joins the core, it needs to copy data, but when a peer becomes a core peer, it never becomes peripheral again unless there is a dimension change. I don't have an analysis of how often this happens.

Q: What happens to locality properties in the system? **A:** Locality is not a prime issue in our system. However, there is often a peer in the neighboring node that is geographically close. So this could be one heuristic to take locality into account.

Q: All routes are going to pass through core nodes. Could that be a problem? **A:** Yes, that is true. To address that, one optimization would be to distribute the cores' load into the peripheral peers.

A First Look at Peer-to-Peer Worms: Threats and Defenses. Presented by Lidong Zhou.

Q: Have you looked into imposing some structure on guardian placement? **A:** We haven't looked at it. Under our current model, having guardian nodes at strategic locations will improve the containment results. However, a worm might be able to infer those locations and change its path of infection. It is not clear in the end whether strategic placement helps.

Q: You consider that it takes zero time to generate an alert. How realistic is that assumption? **A:** We have some data based on our more recent work on alert generation mechanisms. It certainly varies with worms, but is under 1 second for some existing worms we have looked at.

Q: It seems like you are picking regular peers to become guardian peers. How do you know you can trust them? **A:** The alerts are verifiable and must be verified before they are accepted. Otherwise, they can be used to launch attacks on the receiving nodes. Fake alerts generated by malicious nodes will be dropped. This helps reduce the level of trust on guardian nodes.

Q: Patches to remove vulnerabilities may require human intervention and considerable time, even though detecting a signature for particular worms may be easier. Have

M. Castro and R. van Renesse (Eds.): IPTPS 2005, LNCS 3640, pp. 1–12, 2005.

you thought about the impact of this? **A:** Human intervention is not suited for containing p2p worms, which could spread very quickly. We are working on automating the process of generating and applying patches.

Q: How effective can guardians be? We could have a hybrid propagation mode where worms propagate also through links other than the peer-to-peer links. **A:** Hybrid worms require a combination of defense mechanisms. We haven't yet studied how those hybrid worms propagate and how effective our mechanism would be for those attacks.

Q: What if the guardian nodes are connected? **A:** We actually have a graph that shows what happens if we connect all the guardian nodes. It shows a visible but limited effect.

Kill the Messenger: A Taxonomy of Rational Attacks. Presented by Seth Nielson.

Q: Sometimes systems relax strict behavioral requirements from the nodes in order to handle exigencies. For example, tit-for-tat is relaxed when nodes join a p2p network after a failure. How would you classify attacks that exploit this? **A:** We classified those attacks under the excuses category but they certainly would also be policy attacks.

Q: Attacks are possible in systems, for example BitTorrent, where mechanisms (incentives) are good but the implementation is wrong. How do you classify these attacks? **A:** There may be attacks due to implementation faults, for example, a manufactured evidence attack can happen in BitTorrent. But, despite that, incentives in BitTorrent work well.

Q: BitTorrent is successful because payoff is low. If payoff is high, can the behavior change? **A:** Yes.

Q: Is tit-for-tat in BitTorrent beneficial? **A:** There would be more rational manipulation without tit-for-tat.

Q: Is BitTorrent the first system to implement rational incentives? **A:** Other systems have tried to apply incentive techniques as well, for example, Kazaa, but they were easily subverted. It is hard to build incentive mechanisms in a distributed manner. BitTorrent does pretty well.

Panel Discussion

Q: (For Lidong) You did not mention existing worms in Gnutella. **A:** I would not call them worms in the same sense, but rather viruses. They require users' actions to propagate and therefore their propagation is slow and can be relatively easily stopped.

Q: (For Seth) How does rational behavior work at the overlay routing layer? **A:** It would depend on the user's view of cost/benefit. We would need a cooperation model to tackle it.

Q: (For Lidong) Can worm propagation be mitigated by having different systems connected to each other as neighbors, for example, a Windows system with a Linux system? **A:** Diversity at one hop may yield better containment results under our current worm model. But, p2p worms can find two-hop neighbors and infect those directly. So diversity just among neighbors does not solve the problem completely.

Q: (For Seth) Do attacks go away if we can reason about client protocols? For example, we could get a certificate that all Gnutella nodes use the same Gnutella library. **A:** It could make it worse. If all the nodes are running the same software, then they are all vulnerable to the same attacks. However, it will eliminate rational attacks.

Session 2: Search

Brushwood: Distributed Trees in Peer-to-Peer Systems. Presented by Chi Zhang.

Q: One of your slides shows a distributed KD-tree built with Brushwood which allows nearest neighbor selection in an n-dimensional virtual coordinate space. Can you use this nearest neighbor selection scheme in the tree construction to make it proximity-aware, to allow for low latency/stretch routing? **A:** No, proximity neighbor selection is handled by the underlying skip graph. The skip graph performs proximity neighbor selection by selecting close neighbors for the large hops.

Q: What is the latency stretch of Brushwood? **A:** Similar to the stretch of Pastry.

Q: How does Brushwood perform node deletion? **A:** When a node leaves, its tree fragments are merged with a neighbor node.

Arpeggio: Metadata Searching and Content Sharing with Chord. Presented by Dan Ports.

Q: Do you know the average number of keywords in Gnutella queries? **A:** I don't have the exact number for you, but I can imagine it's something similar to web queries, where the average number is about 2.5.

Q: Instead of introducing Index Gateways to decide if metadata is already in the network, can you just query for the metadata to decide if it already exists? **A:** The Index Gateway has other advantages related to some things I didn't present in this talk. For example, since the availability of a file changes constantly, we expire the metadata on a regular basis. The metadata needs to be refreshed periodically to prevent expiration. The gateway knows when metadata will expire and is responsible for renewing it on all the index nodes.

Q: Since you have the sub-rings, can you just assign a specific node, like the last node, of the sub-ring to do that? **A:** That's one possibility, but our design is intended to keep the indexing system and the content distribution system independent.

Q: Suppose a node shares a lot of files. Does that mean it needs to join lots of sub-rings? What's the cost of that? **A:** The cost of joining a sub-ring is $O(\log N)$ communications, while the storage cost is constant.

Q: By "constant," do you mean per sub-ring? **A:** Per node in the sub-ring. **Q:** Suppose you are a member of K sub-rings. Does that mean you have storage cost proportional to K? **A:** Yes, but the cost is only a few bytes per sub-ring, as compared to many megabytes for the actual file data. It's negligible.

Q: I think Overnet is a deployed file sharing system that allows for keyword search using heuristic methods. **A:** I believe Overnet uses an inverted index table with some index side filtering. We are adding keyword sets that improve the distribution of load across nodes in the network.

Q: The idea of keyword-set indexing is trading storage for lookup time, but the storage overhead seems too high to me. Suppose there are 10 keywords per metadata and a query contains 3 keywords on average. The overhead will be a ratio of 1000. Isn't this too high? **A:** We are looking at files with a small amount of metadata. The FreeDB analysis shows that constructing the index requires an increase of only a factor of ten.
Q: Suppose you have only 10 keywords in a metadata block, why don't you just use a 10-dimension DHT? In this way, you can find the metadata simply by DHT

lookup? **A:** We don't believe this would work. It's not clear how to perform a multi-keyword search with this scheme.

Q: For FreeDB, how much storage space do you need? **A:** The total requires about one and a half billion index entries, but each index entry is very small, since it only needs to store the file metadata. For comparison, the total amount of audio data indexed by FreeDB is hundreds of terabytes.

Q: How would you do sub-string search? **A:** We don't provide support for sub-string search.

Q: It seems to me you are underestimating the size of the query because a query for Gnutella, unlike the web case, usually contains more than 3 keywords. **A:** Even if this is true, it isn't a problem if queries are larger. Suppose you have a query of size 6 and the maximum keyword-subset has a size of 3. You can still select a random three-keyword subset, and send the query to the index responsible. Because we're using index-side filtering, this still requires only transmitting relevant results.

Q: Have you thought about locality when choosing a node that's storing a file? **A:** The sub-ring lookups will give you a list of nodes that are sharing a file. You can choose one that's nearby. **Q:** By pinging each of them? **A:** Yes.

Overcite: A Cooperative Digital Research Library. Presented by Jeremy Stribling.

Q: Do you have the numbers for the search and storage costs of Google scholar? Do you think services supported by Google will obviate the need for Citeseer? **A:** We do not have any numbers on that but I don't think it will obviate the need for Citeseer. Google's services are free but there cannot be community control over them. For example, you will not be able to support new features that the community wants.

Q: How are you replicating the documents? **A:** Using the DHT. **Q:** What is the replication factor? **A:** Two. **Q:** How do you know that is enough? **A:** We feel this is good enough since nodes donated by universities can be assumed to be relatively stable. Even if it is not sufficient, we can alter the replication factor.

Q: You should be looking for volunteers. How can universities contribute resources to this system? **A:** The system is not completed. Once it is, we will make a formal announcement.

Q: You were talking about using one-hop DHTs. Instead, it makes a lot of sense if we can use existing, deployed DHTs such as OpenDHT or eDonkey. Have you ever thought of it? **A:** I don't think OpenDHT can support 760GB of data. **C:** This seems a perfect application for OpenDHT.

Q: There are certain things offered by centralized solutions that are very difficult to apply in the distributed case, such as spam filtering, correcting bad entries, and other administrative support. How do you plan to address them? **A:** First of all, Citeseer doesn't provide support for these features currently. But that is an interesting question and we have to think about it.

Q: Have you thought about security implications of distributing this system over the wide area. Potentially there are incentives for people to manipulate indices, the ranking and so on. **A:** Good point. We haven't thought about it yet.

Q: Aren't there copyright issues if you are going to replicate these documents? **A:** Legal issues may crop up. At present, Citeseer does not have any special agreements with the authors and it works well without them.

Q: Do you think a decentralized system is necessary at all? **A:** Let's see after we build it. I think it makes sense to try. **C:** I think Google scholar will need a lot of machines and bandwidth to make their system scalable. The p2p solution offers a simpler and cheaper way to do this. There is a technical and social side to the management issue and your system solves the technical side to these management issues.

Panel Discussion

Q: (For Jeremy) Ignoring non-computer science issues like copyrights, how would you compare using DHTs against regular mirroring or CDNs like Akamai? Even if you use DHTs, certain things have to be done in a centralized way as others have pointed out. Even if you want to extend features, a centralized authority (a committee of some form) may have to agree as to whether to do it or not. So issues against Citeseer may turn against Overcite as well. But leaving that aside, I am trying to think of the advantages that a pure distributed system would give when compared to the other alternatives. **A:** We do not know the exact answer, since we have not deployed our system yet. **C:** Leveraging Google-like solutions may require high bandwidth. Costs will definitely decrease if p2p is used. **C:** But why can't you mirror then? I can't seem to find a convincing answer as to why a p2p solution should be used. **C:** Never mind using p2p or mirroring. Finding an alternative to Citeseer is important and this work is a significant step in that direction.

Session 3: Miscellaneous 1

Peering Peer-to-Peer Providers. Presented by Michael Walfish.

Q: The benefit of DHTs is that you can run one per application—they could be decoupled. **A:** I'm not advocating coupling.

Q: Do you really expect people to run DSPs, and if so, what would their motivation be? **A:** That's what happened when NSF-Net turned into a bunch of Internet service providers, so there is some precedent.

NetProfiler: Profiling Wide-Area Networks using Peer Cooperation Presented by Venkata Padmanabhan

Q: Blame assignment may cause pushback; Can you certify diagnoses? **A:** It's hard for one client to prove a problem. It's easier to prove you *can* get to a website than that you *can't*. Our tool gives a better diagnosis for rational ISPs, which want to fix the problems identified in order to retain users.

Q: If you want to deploy the system on a large scale, and you are interested in diagnosing long-term failures, how do you handle churn in P2P systems? **A:** This is an interesting issue that we haven't yet looked into. But we already replicate data in P2P systems; we could replicate failure information. Also failure information may not need to be as fault-tolerant.

A Statistical Theory of Chord under Churn. Presented by Supriya Krishnamurthy.

Q: Is it easy to extend this analysis to consider latency distributions? **A:** Yes, I think so.

Panel Discussion

Q: (For Venkata) Can NetProfiler be used to detect DDoS attacks? **A:** We can use it to detect new virus outbreaks, and the initial steps (causality) involved in a DDoS attack.

Q: (For Michael) About the forest of DSPs model: does put/get scale like BGP, which uses aggregation? **A:** We're only talking about small multiplicative factors; people using the service can absorb that cost.

Q: (For Michael) I'm not aware of any peering storage services at present. **A:** What kind of peering services do you have in mind? **C:** The non-existence of peered storage services indicates a problem for you. **C:** Inktomi web cache peering is a good example.

Q: (For Michael) Can you think of particular examples that might benefit from DSPs? **A:** HIP. Keys identify hosts, values identify IP addresses. For music sharing, the "bring your own infrastructure" model works fine.

Session 4: Multicast

The Impact of Heterogeneous Bandwidth Constraints on DHT-Based Multicast Protocols. Presented by Sanjay Rao.

Q: I don't quite see what you mean about the multi-tree approach not being useful. **A:** If you have constant degree, the multi-tree approach is useful. But with varying degrees, you want the higher degree nodes higher in the tree in order to minimize depth, and multi-trees do not accomplish this.

Q: It seems that the issues you raise are not fundamental. I can think of some simple fixes that might work. For example, it is possible to remove the in-degree skew in Pastry. It would also help to make the degree proportional to the capacity of the node. You can do this by changing the way proximity neighbor selection works to bias it towards nodes with higher degree. **A:** Maybe you are right but I am pointing out these issues that have not been solved yet.

Q: Does SplitStream address these issues? **A:** (Miguel) I think it does! (Sanjay) I think it does not!

Chainsaw: Eliminating Trees from Overlay Multicast. Presented by Vinay Pai.

Q: What is the difference between BitTorrent and your multicast streams, really? **A:** BitTorrent distributes files, not live content. **Q:** But they both stream. Is it just your sliding window? **A:** Yes, it may be just the sliding window.

Q: You mention that your SplitStream implementation has problems with recovery. Did you try the version from Rice, which has more functionality? **A:** No, we didn't, because we thought that using the Macedon version would be fairer, since the others were implemented in Macedon. But Macedon's implementation of SplitStream is not complete.

Q: Did SplitStream use erasure coding to recover from errors in your experiments? **A:** No. I don't think that is implemented in Macedon.

Q: You mention for the DVD example that you use 8KB packets. How do you deal with the 1500 byte packet limit? **A:** Our implementation is over TCP. **Q:** Do you have a breakdown of costs and overheads. Say, the costs relative to the bandwidth of the stream, versus costs that are constant regardless of the size of the data, etc.? **A:** No, I don't have those numbers. We haven't looked at it.

FeedTree: Sharing Web Micronews with Peer-to-Peer Event Notification. Presented by Dan Sandler.

Q: It seems that this might already be solved. Can't Akamai and others do this? There is a lot of work on cooperative caching. **A:** This is very different because the data set is very volatile. I am a big fan of Akamai and CDNs but it seems that multicast is just a very natural fit for micronews.

Panel Discussion

Q: Why do multicast and not just file distribution with buffering? File distribution gives you time shifting like TiVo and people love it. **A:** There are lectures and video conferencing where you need low delay for real-time interaction.

Q: (For Vinay) Is the delay such that you can't get it with file distribution with buffering? With multicast, if you don't get the bits in time, they just go away forever, but with file distribution with buffering, you can still get all the bits. **A:** No, the bits don't have to go away permanently. As long as seeds still have the bits, they can be recovered. Also, you might have the availability window in Chainsaw spanning GBytes to allow time shifting. **C:** Trees might have better delay than Chainsaw. **C:** But Chainsaw might be more resistant to failures.

Session 5: Overlay Algorithms

Hybrid Overlay Structure Based on Random Walk. Presented by Xiong Yongqiang.

Q: You compute coordinates for nodes in an n-dimensional space using GNP and then map them to a one-dimensional space to enable DHT lookups. Two nodes that are close in the n-dimensional space map to close location numbers in the one-dimensional space but what about the converse property? **A:** It is possible that two nodes close together in the one-dimensional space are actually far apart in the n-dimensional space. So when a node joins, it gets the list of clusters that have location numbers that are close to its own. Then the new node needs to measure the latencies between the different clusters and decide which cluster to join.

Quickly routing searches without having to move content. Presented by Brian Cooper.

Q: With the two optimizations you proposed, every step of the route becomes deterministic. With high network churn, many walks are likely to fail because the target neighbor is down. **A:** If you worry about churn, you can combine these techniques with random walks. At some steps, you choose the next hop by document count and at others, you choose it randomly. An alternative is to perform two walks simultaneously: one based on document counts and a pure random walk.

Q: Did you try random walks biased by node degree in the power law network? **A:** Yes, we did try that. We looked at previous work that performs random walks biased by node degree. This works with the assumption that the node knows the data stored by its neighbors, which requires content movement. If we perform random walks biased by degree without content information, the result is roughly the same as a pure random walk. That's the reason we didn't bias by degree but by document count.

Q: When you use a square root topology, what does the CDF of node degrees look like compared to power law networks? **A:** The peers with the largest degrees have significantly lower degree than in power law networks. The skew is significantly smaller than in power law networks.

Practical Locality-Awareness for Large Scale Information Sharing. Presented by Dahlia Malkhi.

Q: I am worried about coloring and vicinity balls. You assume that a node's vicinity ball has at least a node from each color but that's not guaranteed. **A:** When we get to probability 2^{-80}, that's good enough. If you are still concerned that with churn we cannot find a node of the desired color, that's not a problem. We just go to another neighbor. **Q:** Can you take those extra local hops and still be within the routing stretch of 2? **A:** Our experimental cumulative stretch graph shows stretch above 2 for about 5% of the nodes. That's the difference between the system and the formal algorithm.

Q: Have you done a worst case analysis of node degree? **A:** Yes, it's $\sqrt{n} \cdot \log n$.

Q: What happens when the number of nodes grows or shrinks? **A:** Obviously we need to estimate $\sqrt{n}$, but it does not have to be accurate. This is pretty easy: as the network grows all the red nodes will realize the network is growing and split into two different shades of red. When the network shrinks that's even easier: for example, the red and blue nodes decide to have the same color.

Panel Discussion

Q: We regularly see new routing protocols being proposed. Have we reached a point where instead of proposing new protocols, we should be evaluating how existing protocols serve specific applications? **A:** You are right. I agree that we should tweak the knobs inside each algorithm, instead of trying to engineer a new algorithm.

Q: It's interesting to note that Brian and Dahlia's papers have taken very different approaches: Dahlia's paper talks of 2-hop routes by keeping a lot of state, whereas Brian's paper throws away state but has routes with average hop-lengths of about 8000. **A:** (Brian Cooper) Yes, there are tradeoffs involved but it is difficult to maintain invariants when there is churn if the degree is large. **C:** (Dahlia Malkhi) But without large enough degree, the network will be partitioned under churn.

Session 6: Empirical Studies

An Empirical Study of Free-Riding Behavior in the Maze P2P File Sharing System. Presented by Zheng Zhang.

Q: Is free-riding a real problem? Do free-riders consume that many resources that we need to figure out a way to make them stop? **A:** Yes, free-riding is a problem. The system performance can be improved by getting a better utilization of free-rider's resources. **C:** A brand new user starts with more points than a user that consumes some files. You could consider a version of the system where the user starts at the bottom of the hierarchy, with no points to use, as a way to reduce whitewashing.

Q: Should you set a bottom limit on the point system? **A:** We haven't thought about it. It's a good suggestion. **C:** Once you get past a certain point, it doesn't matter how low your points get. There's only an upper bound on bandwidth. **Q:** But it affects your position in the queue, which is the log of points. **A:** Yes, that's true.

Q: I'd like to ask about attacks where people cheat to get more points. For the fake file attack or colluding attack, maybe some reputation mechanism can be adopted, for example, Pagerank or EigenTrust. **A:** Absolutely. We have an algorithm to detect that. There's another related problem that happens in Maze. Sometimes a student transfers a file from his office to his dorm, and ends up getting points for that transfer.

Q: Does altruism really exist in Maze? From the graphs, it seems like there are a couple of users that upload a few terabytes? Do they offset the free-riders? **A:** Yes. Our top 10 list motivates people to upload to gain recognition.

Q: What are the typical types of data you found on Maze? **A:** Interesting images, Hollywood DVDs, software.

Clustering in P2P exchanges and consequences on performances. Not presented.

The BitTorrent P2P File Sharing System: Measurement and Analysis. Presented by Johan Pouwelse and Pawel Garbacki.

Q: When the tracker crashes everything stops. Why doesn't the RIAA DoS trackers? **A:** It's not true that everything stops when the tracker fails. What happens is that no new peers can join the system, but the joined peers can continue. The peer selection mechanism is done locally. When the tracker fails, it's not possible to detect the newcomers, but a peer can continue downloading. One of the things we're trying to do now is to distribute the tracker's functionality, for example, having the peers gossip the IP addresses of peers in the system. **Q:** And who runs the trackers now? **A:** Web sites like Supernova run big trackers.

Q: How do you measure the peer uptime? **A:** We just contact the peers and ask them for some random pieces to check whether they are up or not. We do this once every few minutes.

Q: Is there any data downloaded that is not copyrighted material? Have you collected any statistics to check whether there is different behavior of users for copyrighted and not copyrighted material? **A:** It depends on the tracker list. There are basically three types of users. The first group consists of the Linux Open Source population, which shows altruistic behavior and has high performance. The second group consists of regular users, sharing illegal content, and which have no sharing ratio requirements. And the last consists of a membership-based community, that enforces a strict sharing ratio to further stimulate the performance of the network.

Q: How well does tit-for-tat work in BitTorrent? Do we need additional mechanisms? **A:** Tit-for-tat outperforms the queuing mechanism in EDonkey or the aggressive seed-jumping policy used by Pirate Bay.

Panel Discussion

C: You have been talking about altruism in file sharing systems. But it is not really altruism, it is just contributing your idle CPU time and unused bandwidth capacity for stealing copyrighted material.