

# Improving ICE Service Selection in a P2P System using the Gradient Topology

Jim Dowling<sup>1</sup>

Jan Sacha<sup>2</sup>

Seif Haridi<sup>1</sup>

<sup>1</sup>Swedish Institute of Computer Science, Kista, Sweden

<sup>2</sup>Distributed Systems Group, Trinity College Dublin

{jdownling,seif}@sics.se, jsacha@cs.tcd.ie

## Abstract

*Internet Connectivity Establishment (ICE) is becoming increasingly important for P2P systems on the open Internet, as it enables NAT-bound peers to provide accessible services. A problem for P2P systems that provide ICE services is how peers discover good quality ICE servers for NAT traversal, that is, the TURN and STUN servers that provide relaying and hole-punching services, respectively. Skype provides a P2P-based solution to this problem, where super-peers provide ICE services. However, experimental analysis of Skype indicates that peers perform a random walk of super-peers to find one with an acceptable round-trip latency. In this paper, we discuss a self-organizing approach to discovering good quality ICE servers in a P2P system based the walk Topology. The walk Topology uses information about each peer's ability to provide ICE services (open IP address, available bandwidth and expected session times) to construct a topology where the "better" peers for providing ICE services cluster in the center of the topology; this adaptation of the super-peer search space reduces the problem of finding a good quality ICE server from a random walk to a gradient ascent search.*

## 1. Introduction (ICE and Relay Peer Selection)

Recently, large-scale distributed and peer-to-peer (P2P) systems, such as Gogletalk and Skype, respectively, have enabled nodes behind Network Address Translation (NAT) gateways and firewalls, that we call NAT'd nodes, to provide services, such as Voice over IP (VoIP) telephony and file transfer. In these systems, peers cannot establish direct IP connections with NAT'd nodes without the use of a 3rd party server to either help the connecting node hole-punch through any intermediary NATs or to help establish an indirect connection via a relay (or rendezvous) node, which relays traffic to the NAT'd node. Two commonly used protocols for NAT traversal are Simple Traversal of UDP over NAT (STUN) [13] for hole-punching to set up a

direct UDP connection through a NAT, and an extension to STUN called Traversal using Relay NAT (TURN) [12], that provides relaying services to nodes behind NATs by routing UDP and TCP traffic to and from them. Peers in both Skype and Gogletalk attempt to use STUN to establish connections, and if this fails they fall-back to using a relay (TURN) node. Guha estimates that 9.6% of Skype's traffic travels over relay nodes [4], which corresponds closely to Google's estimate that 8% of the GoogleTalk traffic is routed using relay nodes [3].

Internet Connectivity Establishment (ICE) is a draft IETF specification [2] that defines how to establish sessions (primarily aimed at Session Initiation Protocol sessions) that can traverse NATs and firewalls. ICE specifies how both STUN and TURN protocols are used to enable nodes provide each other a set of candidate transport addresses for communication. The problem of how nodes find a host providing these ICE services (i.e., the STUN and TURN servers) is outside the scope of the specification. However, two different approaches have become popular: deploy ICE services on well-known names in a centralized server farm, as in relay.l.google.com and stun.l.google.com for Google Talk [3], or provide a decentralized solution, such as Skype, where peers with an open IP address and sufficient available bandwidth are upgraded to super-peers that provide STUN and TURN-like services to ordinary peers [10, 4, 1].

In Skype super-peers are connected over their own P2P topology, and ordinary peers connect to super-peers to both discover Skype users and use ICE services to establish VoIP sessions with NAT'd peers [5, 1, 10]. NAT traversal covers four cases in Skype: Initiator NAT'd, Recipient NAT'd, Both NAT'd (behind well-behaved NATs) and Both NAT'd (behind broken NATs) [10]. Only in the final case, do peers use relay nodes for VoIP traffic. Ren et al. observed that peers select several super-peers to act as a relay nodes, apparently at random, and measure the round-trip time (RTT) to the relay nodes [10]. If the RTT is below an acceptable level, the VoIP session stabilizes routing over those relay nodes, otherwise super-peers continue to be selected and probed until a few of them are discovered with a low-

enough RTT. They concluded from their experiments that peer selection of relay nodes is both slow, with a worst case in their experiments of 329 seconds taken for traffic to stabilize over selected relay nodes, and suboptimal in use of system resources, with most sessions probing over 20 super-peers before stabilizing on relay nodes [10]. We suggest that this poor performance is due to Skype performing a random walk of the super-peer topology, where no use is made of domain knowledge to improve search efficiency, see Figure 1 (a).

In the rest of this paper, we show how the the Gradient Topology [15, 14] enables the construction of an efficient, self-organizing system for providing ICE services, where ICE services are deployed on high-quality peers and efficiently discovered by clients using gradient ascent search. We show that the cost of finding high quality super-peers on the Gradient Topology using a general search algorithm, random walk, is substantially higher than gradient ascent search.

## 2. ICE Services on the Gradient Topology

The Gradient Topology is a self-organizing P2P topology where peers use a local utility level to adapt their connections to other peers, such that peers with highest utility levels are clustered in the center of the topology while peers with decreasing utility are found at increasing distance (or “orbits”) from the center [15, 14]. Gradient search to and from the center of the topology follows the peers’ utility gradient. As an application of the Gradient Topology, we are developing a P2P middleware that provides ICE services. In this system, there are two types of peers: NAT’d peers that cannot provide ICE services, and super-peers that have an open IP address, and, hence, can potentially provide ICE services. Only a sub-set of the super-peers are suitable for providing relay services: those peers with high available bandwidth and session times that are long relative to application usage scenarios. For different applications that use ICE services, such as VoIP systems, web servers or video streaming, typical application usage may produce different requirements on how long a super-peer’s session time should be to provide a reliable relaying service. In the Gradient Topology we can design super-peers, so that only those super-peers with high enough expected session times are upgraded to provide relay services, see Figure 1 (b). For context, in Skype, the median super-peer session time has been measured at 5.5 hours - easily long enough to provide relay services for VoIP conversations that have a median length measured at 2 minutes 50 seconds [4].

In our system, peers connect to the Gradient Topology and determine whether they have an open IP address or are NAT’d using a pool of well-known bootstrap servers. Peers with open IP addresses become super-peers and are

connected using the Gradient Topology, while the NAT’d peers are edge nodes that can have redundant connections to super-peers. In effect, the NAT’d peers are clients of the Gradient Topology, while the super-peers are servers that provide ICE services or routing to ICE services.

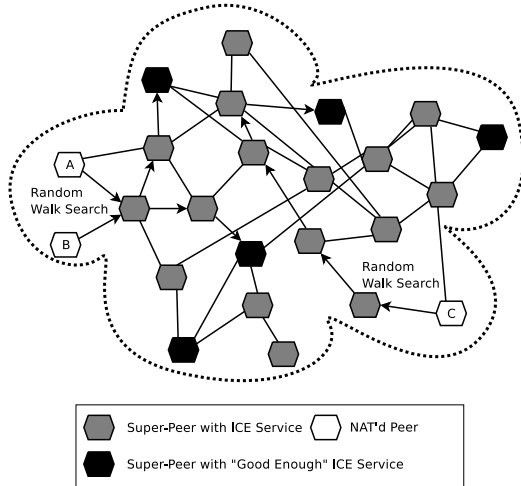
In order to construct the Gradient Topology, super-peers have a utility function that captures good properties for ICE services, and relay services in particular. The utility function can be customized to consider different host and application attributes, but our current utility function is based on available bandwidth at peers and their expected session time:

$$Utility(n_i) = (AvailableBandwidth_i * E(SessionTime_i))$$

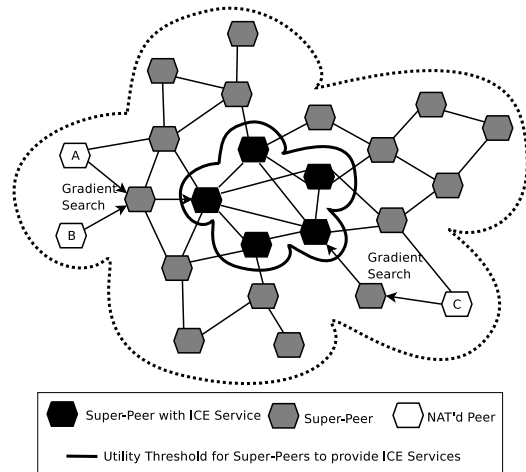
where  $AvailableBandwidth_i$  has a value greater than zero if it exceeds a minimum threshold value and the expected session time,  $E(SessionTime_i)$ , is calculated using a probabilistic model based on observations of the peer’s recent session times. Peers use the output of the utility function to adapt their connections to preferentially connect to other peers with similar utility levels. This causes peers with higher utility to migrate towards the center of the topology and peers with lower utility to migrate to the edge of the topology. Peers also maintain connections to random peers, which provide negative feedback on the excessive clustering induced by peers connecting to other peers with similar utility values. For more details on the algorithm, see [15, 14].

Search on the Gradient Topology can be done efficiently using gradient ascent search [14], see Figure 1 (b), where the search query includes a utility threshold that the super-peer providing the ICE service must exceed for the search to succeed. NAT’d peers can determine an appropriate utility threshold by estimating the distribution of utility levels of super-peers using an aggregation algorithm, such as the one described in [15]. Similarly, super-peers can use the estimated distribution of utility levels to determine if they should provide an ICE service or not. These utility thresholds can be customized for different applications.

Figure 2 shows the results of experiments performed on simulations of the Gradient Topology comparing random walk with gradient ascent. In the first experiment, the simulated P2P network is under constant churn, and new peers are assigned a session-time according to the Pareto distribution with an exponent of 1.5; this roughly corresponds to the power-law distribution of super-peer session times observed by Guha et al. for Skype [4]. Peers search for any super-peer with a utility above a given threshold, i.e., the super-peers that provide the ICE services. As can be seen in Figure 2 (a), the gradient ascent algorithm requires significantly fewer hops to find a high utility super-peer than random walk. This is because gradient ascent can exploit knowledge of peers’ utilities captured in the gradient topology, whereas the random walk algorithm is blind to peer



(a) Skype searches for better quality relay nodes using a random walk of its Super-Peer Topology



(b) Gradient Descent Searching for ICE Services on the Gradient Topology

**Figure 1. Searching on P2P Topologies for ICE Services**

utilities. In the second experiment, the performance of random walk and gradient ascent are evaluated when the system is under churn. The network contains 10,000 peers, and even under extremely heavy churn (with 10% of the peers replaced at every discrete time step), gradient ascent search continues to have a low failure rate for searches, see Figure 2 (b).

**Status of our P2P ICE Middleware** Currently, work is in progress on an implementation of the Gradient Topology, and we are leveraging implementations of STUN and relay servers in the Berkeley-licensed libjingle library [16, 3] to provide ICE services.

### 3. Related Work

The ASAP protocol addresses the problem of improving the selection of relay nodes by making the protocol aware of the nodes' Internet Autonomous System (AS) [10]. For NAT traversal, there have been recent attempts to extend ICE to cover TCP-based media, including the ability to offer a mix of TCP and UDP-based candidates [11]. There is also ongoing research on hole-punching for TCP, with systems such as STUNT, NATBlaster and P2PNAT demonstrating some success [5].

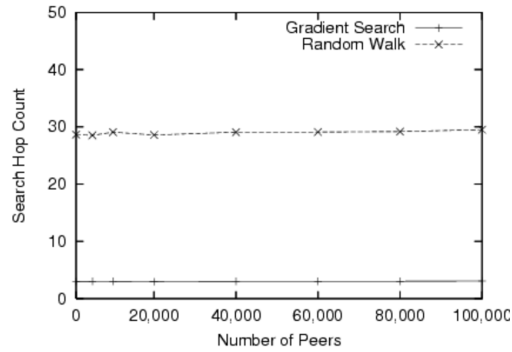
SG2 addresses the problem of developing a super-peer topology where clients connect to a single super-peer with a low round-trip time (RTT) [7]. In SG2, peers are connected in a random topology using Newscast, while super-peers connect to one another if their latency is below a threshold level. Jelasy and Kermarrec have simulated a gossiping

model, also based on Newscast, that partitions peers into "slices", where the protocol takes into account specific attributes of the nodes [6]. This system addresses the problem of the partitioning a P2P topology, but the authors have not yet proposed search algorithms for their topology.

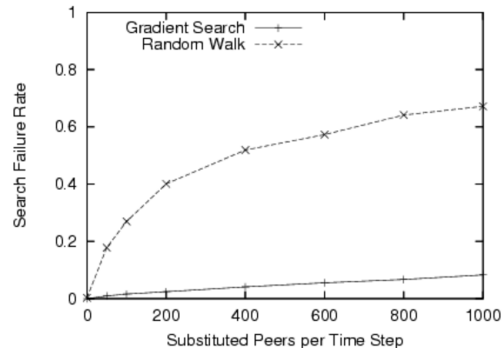
Semantic overlay networks (SONs) address a similar problem to the gradient topology, namely, how to structure a P2P topology to improve the search performance for resources in a system. They aim to exploit semantic structure in resources or in successive search queries, generally by constructing a topology that organizes resources in accordance with the semantic space of resources, such as documents or multimedia. Often resources are mapped on to a semantic space, for example, mapping feature vectors to a euclidean space, and the distance between resources represents their semantic closeness. This is similar to our notion of mapping node utility to the gradient topology, except that utility is defined on a per node, rather than per resource, basis. Examples of SONs include [17], a small world semantic overlay topology [8], and a semantic overlay clustering for super-peer networks [9], where peer-to-super-peer connections are adapted to improve search performance.

### 4. Conclusions

In this paper, we showed how a P2P topology can be adapted to build more efficient search algorithms to find ICE services deployed on super-peers. We described how the Gradient Topology can be used to build a P2P middleware that provides ICE services for NAT-traversal, and how it can be constructed so that the ICE services are deployed on peers with higher utility. We showed how gradient as-



(a) Average Search Hop Count



(b) Average Search Failure Rate.

**Figure 2. Comparison of Random Walk and Gradient Ascent Search on the Gradient Topology when searching for peers above the utility threshold.**

cent search algorithm is significantly more efficient than random walk when searching for ICE services on the Gradient Topology, and how gradient ascent search functions continues to work under high churn levels where random walk breaks down. This approach should not only improve the time required to find a good quality of ICE server, but also help optimize the global utilization of resources in the P2P system. Future work will involve a real-world implementation of the middleware and evaluation using applications built on the ICE services.

**Acknowledgements** This research was supported by a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme. The authors thank Ray Cunningham for comments on an earlier version of the paper.

## References

- [1] S. Baset and H. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In *IEEE Infocom*, 2006.
- [2] Cisco. Interactive connectivity establishment (ice). In *IETF Draft*, Jan 2007.
- [3] Google. Google talk, Accessed, Feb 22, 2007. <http://code.google.com/apis/talk/libjingle>.
- [4] S. Guha, N. Daswani, and R. Jain. An Experimental Study of the Skype Peer-to-Peer VoIP System. In *Proceedings of The 5th International Workshop on Peer-to-Peer Systems*, pages 1–6, 2006.
- [5] S. Guha and P. Francis. Characterization and measurement of tcp traversal through nats and firewalls. In *Internet Measurement Conference*, pages 199–211, 2005.
- [6] M. Jelasity and A.-M. Kermarrec. Ordered slicing of very large-scale overlay networks. In *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, pages 117–124. IEEE Computer Society, 2006.
- [7] G. Jesi, A. Montresor, and O. Babaoglu. Proximity-aware superpeer overlay topologies. In *Proceedings of the 2nd IEEE International Workshop on SelfMan*, volume 3996 of *LNCS*, pages 43–57. Springer-Verlag, june 2006.
- [8] M. Li, W.-C. Lee, and A. Sivasubramaniam. Semantic small world: An overlay network for peer-to-peer search. In *ICNP '04: Proceedings of the Network Protocols, 12th IEEE International Conference on (ICNP'04)*, pages 228–238, Washington, DC, USA, 2004. IEEE Computer Society.
- [9] A. Loser, F. Naumann, W. Siberski, W. Nejd, and U. Thaden. Semantic overlay clusters within super-peer networks. In *DBISP2P*, pages 33–47, 2003.
- [10] S. Ren, L. Guo, and X. Zhang. Asap: an as-aware peer-relay protocol for high quality voip. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, pages 70–79, 2006.
- [11] J. Rosenburg. Tep candidates with interactive connectivity establishment. In *IETF Draft*, October 2006.
- [12] J. Rosenburg and a. C. H. R. Mahy. Turn - traversal using relay nat. In *IETF Draft*, October 2006.
- [13] J. Rosenburg, J. Weinberger, C. Huitema, and R. Mahy. Stun - simple traversal of user datagram protocol through network address translators. In *RFC 3489*, March 2003.
- [14] J. Sacha, J. Dowling, R. Cunningham, and R. Meier. Discovery of stable peers in a self-organising peer-to-peer gradient topology. In *Proceedings of DAIS*, volume 4025 of *LNCS*, pages 70–83. Springer-Verlag, 2006.
- [15] J. Sacha, J. Dowling, R. Cunningham, and R. Meier. Using aggregation for adaptive super-peer discovery on the gradient topology. In *Proceedings of the 2nd IEEE International Workshop on SelfMan*, volume 3996 of *LNCS*, pages 77–90. Springer-Verlag, 2006.
- [16] P. Saint-Andre, J. Beda, S. Ludwig, J. Hildebrand, and S. Egan. Jingle ice transport. In *XMPP Draft*. XMPP Standards Foundation, Dec 2006.
- [17] S. Voulgaris, A.-M. Kermarrec, and L. Massoulie. Exploiting semantic proximity in peer-to-peer content searching. In *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 238–243. IEEE Computer Society, 2004.