Designing a P2P Ad-hoc Secured Collaboration Tool

Tarun Abhichandani
Kristie Kosaka
Samir Chatterjee
School of Information Systems and Technology
Claremont Graduate University
Claremont, CA – 91711.
{tarun.abhichandani, kristie.kosaka, samir.chatterjee}@cgu.edu

1. Introduction

We are witnessing the continuing progress of client-server based voice, video and conferencing applications that help us collaborate with participants on the Internet. Following this continuum, several recent standards such as Session Initiation Protocol (SIP) [1] and H.323 [2] have enabled voice and video conferencing over Internet Protocol (IP). However, experience has revealed that these systems are expensive, require centralized services, and often require intensive administrative support to configure and maintain[1] [3]. Peer-to-Peer (P2P) technology [4], on the other hand, has the potential to replace expensive centralized entities and offer true distributed systems that can facilitate minimal infrastructure for communication capability. P2P systems are highly robust, scalable, and lack any centralized server that could be a bottleneck. A pure P2P system enables individual peers to communicate with no dependence on servers. All users are peers that communicate in a distributed fashion. Access to resources coupled with the ease of installation to get started has fueled the success of P2P. The authors purport that such decentralized or server-less environment can promote ad-hoc collaboration.

We define ad-hoc collaboration to imply environments that support "spontaneous" distributed collaboration, where two or more people from different locations decide to have an unplanned interaction with one another. Studies have assigned different terms to similar environments – spontaneous collaboration [5] and coalition [6], for instance. The definition used in this study has been influenced by Berket and Agarwal [7]. Ad-hoc collaboration includes all the features that other kinds of group-work support. The distinguishing factor, however, is the support of ad-hoc scenarios. Frequently on-line collaboration includes the sharing of applications, often concurrently, while content is being created or shared by multiple participants, e.g., Webex [8] or Microsoft Live Meeting [9]. Another kind of teamwork involves collaboration on files, typically with unstructured content and often on a discontinuous basis, e.g., Lotus' Domino Document Manager [10]. This category involves heavy use of file-sharing and e-mail. Multimedia (IP telephony and video-conferencing) for which expensive servers are often required is yet another type of collaboration. Due to

[1] Internet-Draft accessible at http://www.p2psip.org/drafts/draft-bryan-sipping-p2p-01.txt has clearly alluded to such issues.

DESRIST 2006. February 24-25, 2006, Claremont, CA.

increased mobility and emergent scenarios, the need for ad-hoc collaboration has been identified. One of the prime examples of an emergent scenario is "network-centric warfare", a highly mobile server-less battlefield environment.

The objective of this paper is to explain the design of a P2P infrastructure that enables ad-hoc collaboration. The infrastructure must support various communication and security capabilities so that a wide variety of applications can be supported with a high degree of assurance and trust. Based on the objective, the purpose of this paper is threefold: 1) to identify requirements for a real-time ad-hoc collaboration using P2P, 2) to explore new architectures that bind P2P with signaling standards such as SIP to design and develop a server-less multimedia collaboration system, and 3) to discuss the security challenges of such ad-hoc collaboration systems.

The paper is organized as follows. Section 2 provides a detailed background of P2P and SIP as converged technology and presents design requirements for P2P-based ad-hoc collaboration. The details of a P2P-based architecture, formulated by the authors, are described in Section 3. The dynamics of trust involved in ad-hoc collaboration are outlined in the fourth section. Section 5 describes the current status of the study. Section 6 concludes by identifying the future work planned for the study.

2. Background

2.1 P2P

P2P is a class of applications that take advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet [11]. The central idea in P2P systems is that every node in a network is a server and a client. Every node is able to exchange resources and collaborate with no dependence on a central entity. P2P systems are increasingly being considered for business applications as it enables valuable externalities, lower cost of ownership, anonymity and privacy [4]. P2P is being viewed as an infrastructure that could effectively serve as an operating system for the Internet, which is global-scale and heterogeneous in nature [12]. In delivering this vision, P2P is considered as an important development in distributed computing where the network is really the computer. In this kind of infrastructure, scalable IP routing can be employed to connect networked devices that are spread all over the Internet. This interconnection and communication is important so that different kinds of media can be exchanged. Principles of P2P with a mix of centralization and decentralization have been employed in e-mail, IP routing and Usenet applications [13]. There have been various initiatives that have used P2P infrastructure – voice and video conferencing[2] [14, 15] and content sharing [16]. Operationally, P2P needs to provide improved capabilities to a user, wherein a

[2] Skype, a commercial initiative, is a noteworthy contribution towards voice and video conferencing

user's machine is a server and a client. P2P infrastructure needs to be aware of asymmetrical capability, unreliable behavior of users and creation of working groups.

P2P topologies can be hierarchical, centralized, decentralized or hybrid, as illustrated in Figure 1. These topologies or their variants have been in existence for quite some time. A well-known hierarchical system is Domain Naming Service (DNS) [17], wherein authority flows from the root name servers to the server for the registered name and often down to third-level servers [18]. Centralized P2P is wherein communication is completely centralized with many clients connecting directly to a single server. Traditional database systems, web server systems and SETI@Home [19] are some of the examples of centralized P2P topologies. In decentralized topology, there is no central point of control. Peers communicate symmetrically and have equal roles. Examples of a pure P2P model include Freenet [20] and Gnutella [21], wherein resources are distributed among participating nodes. In a hybrid model, a central entity is contacted to obtain meta-information, such as the identity of the peer on which some information is stored. Actual resource sharing, however, is decentralized. Groove [14] is an example of a hybrid network. Just as hybrid model, there are other intermediate solutions that have superPeers, which have more information than other peers. A coarse taxonomy of P2P systems classify them into *distributed computing* (e.g., SETI@home [19]), *file sharing* (e.g., Gnutella [21], Freenet [20]), *collaboration* (e.g., Groove [14]) and *platforms* (e.g., JXTA [22]).

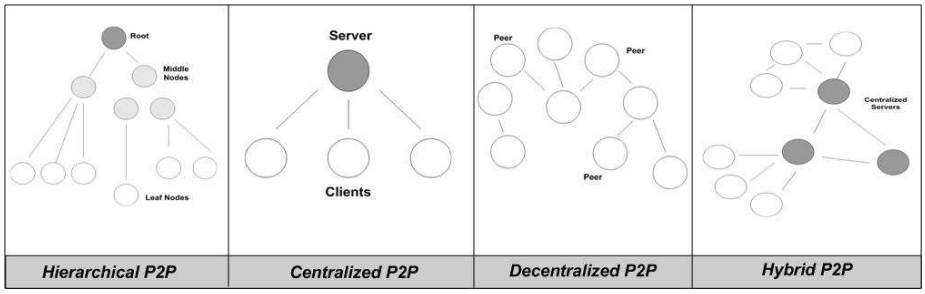

Figure 1: Peer-to-Peer Topologies

Overlay networks have been recommended for self-organizing networks [23], a critical feature of P2P infrastructure. Overlay networks have been identified to be effective design mechanisms for deploying heterogeneous, large-scale, P2P systems [24]. The purpose of P2P-based overlay networks is that they abstract the complicated connectivity of a P2P network to higher-level programmatical view of the peers that make up the network [18]. P2P overlays provide a good substrate for building large scale data sharing and content distribution applications [25]. Similar to general overlay networks, a wide array of research has been published with regards to facilities in P2P overlay. In the context of P2P systems, there are two types of overlays: unstructured and structured [25]. Unstructured overlays do not employ any standardized logic in their P2P tasks. Unstructured overlays organize nodes in a random graph and use flooding or

random walks on the graph to query content stored by overlay nodes [25]. Gnutella [21] and Freenet [20] are examples of such overlays.

Structured overlays, on the other hand, use certain methodology while searching and disseminating information regarding resources. Structured overlay networks such as CAN [26], Chord [27], Pastry [28], Tapestry [29] and Bamboo [30] create a virtual topology on top of the physical topology. They are formed to overcome the inefficiencies of the unstructured overlay [25]. Structured algorithms either guarantee logarithmic bounds with respect to the size of the peer community, or argue that logarithmic bounds can be achieved with high probability.

In structured overlays, there is a hash table that stores information in the form of *key* and a *value*. Key is a certain m-bit identifier, which varies based on the types of structured overlay. Value can be an address, a document or an arbitrary data item [27]. Information in hash tables is stored across the P2P network with the nodes that form the overall network. The responsibility of maintaining the information in hash tables is distributed among all the nodes; hence the term *Distributed Hash Tables* (DHT). In addition to DHT, there is certain routing intelligence that is implemented in P2P networks. Figure 2 illustrates one of these overlay algorithms – Chord.

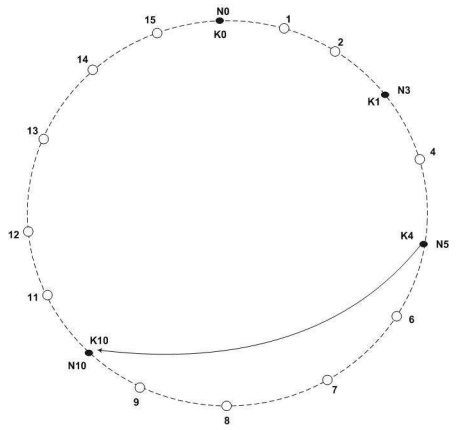

Figure 2: An Example of Structured Overlay - Chord

Chord provides a facility for consistent hashing, which has distinct advantages of balancing load and improving scalability. The specification of Chord protocol supports just one operation: given a key, it maps the keys onto a node [27]. The basic idea of this protocol is that of a clock. In a clock, we arrive at an hour using modulo function. At 1:00pm, we arrive at "1" using this function: 13 (12+1) modulo 12 which gives us a remainder of 1. In Chord, an m-bit identifier gives us this number (12 in a clock). If m-bit identifier is 3, modulo value will be 8 (2^3) and so on. Every node in a Chord network gets an identifier based on its IP – a node identifier. Every key, a document name, gets its identifier based on its name – a key identifier. Both the values are assigned identifiers using a base

hash function – Secured Hash Algorithm (SHA-1). These identifiers are arranged in a single dimensional circular space.

As enumerated in Figure 2, consider 4 nodes with identifiers – 0, 3, 5 and 10. These node identifiers are calculated based on the IP address of the nodes. If there are four key (k) identifiers that represent name of the files – 0, 1, 4 and 10. k identifiers are assigned to first node whose identifier is equal to k or follows k. Consequently, node 0 will have key 0, node 3 will have key 1, node 5 will have key 4 and node 10 will have key 10. If the information to be stored on the network is less and the number of nodes is few then every node needs to know the destination of its successor. However, this is not scalable in real-time networks. Therefore, nodes in Chord network need to store routing information in a table called a finger table [27]. The changes in the network caused due to the exiting and entering of nodes, known as the rate of churning is handled by stabilization methods employed in the protocol.

All the DHT algorithms, mentioned above, provide certain basic components – node and key identifier space, rules for associating keys to nodes, routing tables and rules for updating routing tables [31]. However, the difference is in the way they employ different routing algorithms [32]. There have been attempts to devise a common Application Programming Interface (API) for different DHT algorithms [32]. Table 1 illustrates differences in various DHT facilities. Chord, as explained above, arranges the network in circular space. CAN, on the other hand, arranges the network based on certain number of dimensions (d). As a result, the routing state, hops and the churn rate depend on the number of dimensions based on which the nodes are arranged. In case of Pastry and Tapestry, nodes in the network are arranged based on logarithmic base of the node identifier.

Table 1: Comparison of Various DHTs

Overlays	Comparison Criteria					
	Model	**Parameters**	**Hops to locate data**	**Routing State**	**Peer joins and leaves**	**Distinctive feature**
Chord	uni-dimensional, circular ID space	N – number of peers in the network	logN	logN	$(\log N)^2$	Replicate data on multiple consecutive peers, Application retries on failure
CAN	multidimensional ID space	N – number of peers in the network d – number of dimensions	$d \cdot N^{1/d}$	$2 \cdot d$	$2 \cdot d$	Multiple peers responsible for each data item, Application retries on failure
Pastry	Plaxton-style global mesh	N – number of peers in the network b – base of the chosen identifier	$\log_b N$	$\log_b N$	logN	Replicate data across multiple peers, Keep track of multiple paths to each peer
Tapestry				$b \cdot \log_b N + b$	logN	

2.2 Session Initiation Protocol (SIP)

Existing IP telephony and video-over-IP solutions use a client-server approach as in Internet Engineering Task Force (IETF) Session Initiation Protocol [1] and International Telecommunications Union (ITU-T) H.323 standard [2]. User Agents (UA) communicate with centralized servers (e.g. proxy or registrar) for every domain. UAs in the domain register their IP addresses with the server so that others can reach them. While scalability and reliability of such server-based systems are achieved, a majority of the cost is associated with maintenance, configuration and typically requires a dedicated system administrator for each domain. Setting up and quickly tearing down such networks for ad-hoc collaboration is simply not possible.

Several initiatives are underway to include P2P architectures using SIP as a signaling standard. The P2P-SIP working group (http://www.p2psip.org/index.php), under the auspices of IETF, has been formed with a view to utilize the distributed nature of P2P in a SIP network resulting in eliminating or reducing the need for centralized servers. There are a number of IETF drafts that identify various topical issues in implementing such applications. Furthermore, there are various proposals that have been published in some of the major conference proceedings [3, 33].

Our objective is to design and develop an ad-hoc collaboration tool that utilizes the capabilities of underlying DHTs to map users to location and takes full advantage of standards such as SIP for establishing multimedia sessions on the Internet.

2.3 Design Requirements

To support ad-hoc collaboration, there are specific design requirements that need to be considered. We present the requirements pertaining to such systems based on solutions that have been adopted in this study:

A. P2P-based requirements

1) *Self-organization*: Peers in an ad-hoc infrastructure form overall network. It is a common occurrence that peers in this network leave and join in a random fashion. Further, the resources such as files and meta-information might be reorganized to reflect the existing status of the network. P2P networks are designed to support such self-organization, features similar to the real-world phenomenon of social interactions. Ad-hoc networks are formed on a random basis like social networks with resources are located at distributed peers. Overlay networks have been proposed as a solution to support extendability of topologies in P2P networks [24].

2) *Efficient node or resource lookup*: Finding resources or nodes in ad-hoc infrastructures necessitate few hops so that the latency can be kept to a minimum. In a P2P system, resolution can be accomplished through unstructured or structured overlays. Structured overlays are an active research area within the P2P community and offer highly scalable and efficient low-latency search and retrieval of data over an overlay network. Extending these structured overlays, Proximity Neighbor Selection (PNS) [34] algorithm provides options for resolving resources among nodes that are in routing proximity. This is particularly desirable in the case where the usernames are of an email address format user@domain. Collaboration among users who are within a similar name space, and presumably communicate frequently, will be located in nearby machines.

3) *Completeness*: Name lookup or URI lookup query must always complete and return valid responses. For example, if a resource is present (the user is registered), we return the location of this user, and return a failure if the user is not present. Most of the structured overlay algorithms satisfy this requirement.

4) *Stabilization*: Ad-hoc infrastructure needs to support variable churn-rates that may result if an existing peer goes down or a new node joins the network. P2P routing algorithms are designed to stabilize the overall network against such random changes [24]. Further, such algorithms are also designed to redistribute resources that are placed all over the network. For example, if a peer node goes down, we may lose SIP registration information. P2P algorithms, in this case, will redistribute the information over the network. Stabilization is an important way of providing fault-tolerance towards the overall network [24].

5) *Routing*: Every peer in a P2P network stores routing information of nearby peers that can be reached. Information on every peer node collectively forms complete information of the overall network [18]. Different structured overlays prescribe different approaches for storing the routing information.

B. Infrastructure requirements

6) *Auto-configuration*: The peers in the ad-hoc network need to be capable of self and auto configuration. To enumerate select configuration, a peer should be capable of registering the node, user and joining the ad-hoc network. This should occur with minimal user involvement.

7) *Network Address Translation (NAT) / Firewall Traversal*: Peers should be capable of traversing NAT and Firewall entities with no administrative interference. A number of solutions are being tested in standards organizations – Simple Traversal of User Datagram Protocol Through Network Address Translators (STUN) [35], Traversal using Relay Network

Address Translators (TURN)[3] and Interactive Connectivity Establishment (ICE)[4]. These solutions are under consideration and are being widely tested. The main drawback, however, is that they are inherently client-server based solutions.

C. SIP-based requirements

8) *Basic and Advanced Session services*: In order to enable peers to communicate, ad-hoc networks need to provide for creating sessions and exchanging media over these sessions. Artifacts over the network need to utilize established standards such as SIP [1] to establish point-to-point and multi-point voice and video sessions. SIP provides for such capabilities. SIP is also designed to provide Instant Messaging (IM), Presence and file-transfer capabilities [36]. It is also designed to advanced session services such as caller ID and voice messaging.

D. Security requirements

9) *Authentication*: When a user calls using this new P2P-SIP system, one should be able to make sure that it is coming from a legitimate user and not some imposter who is masquerading as that user. Authentication can be easily achieved if public-key infrastructure is present by using message digests (MD5 or SHA-1). Since PKI is a server-based solution, it will be important to adapt PKI to operate in a totally distributed environment.

10) *Group key generation and secured distribution*: For P2P conferencing, an initiator must create a group key that would be used to encrypt all messages between collaborators. Further, once such a key is generated, it must be securely shared with other members. Several recent works on how to do that have been recently addressed [37, 38].

11) *Confidentiality through encryption*: All messages (voice, video and data) must be encrypted to provide privacy and confidentiality of communication. Using a P2P-PKI approach, this can be achieved.

12) *Trust Issues*: In a P2P environment, trust becomes of paramount importance since misbehaving peer nodes could wreak havoc. When centralized authority services are absent, each peer has to generate and maintain its own trust in a purely PGP fashion. We will share details on this aspect in later sections.

E. Media Requirements

[3] TURN is an Internet-Draft and is available at http://www.ietf.org/internet-drafts/draft-rosenberg-midcom-turn-08.txt
[4] ICE is an Internet-Draft and is available at http://www.ietf.org/internet-drafts/draft-ietf-mmusic-ice-05.txt

13) *Two-party media exchange or multi-party conferencing:* While voice does not pose a major hindrance among participants, video needs to be mixed and retransmitted among interested individuals. To perform such mixing, central entities such as conference servers are required. There are some initiatives that focus on negating the dependency on such servers [39-41]. However, concrete and replicable solutions have not been found by the authors.

3. Design and Architecture of a P2P-SIP Ad-Hoc System

The primary purpose of the study is to design and implement a peer (a server and a client) that is able to utilize underlying overlay algorithms for routing, resource and node resolutions and create ad-hoc sessions among participants. This section describes the solutions for various design requirements and Section 5 details the progress of implementing this ad-hoc peer.

Figure 3 illustrates the design requirements for ad-hoc collaboration and various solutions that can be utilized to offer a cohesive application for satisfying these disparate requirements. The overall objective of the study is to:

- Provide a P2P infrastructure capable of discovering nodes in a structured manner, detailed in Section 2.
- Provide features for session creation and termination between two or more parties. SIP is an appropriate technology for managing sessions between two or more ad-hoc participants.
- Consider various scenarios that exist in an ad-hoc environment.
- Provide assurance for secured transmission and trust-building between participants.

The above-mentioned objectives warrant certain specific qualities that need to be implemented:

- **Infrastructural Transparency** by providing for wireless or wireline support, traversal for NAT or Firewall and IPv4 or IPv6 support. IP Tunneling and Interactive Connectivity Establishment (ICE)[5] are two methods for traversing NAT or Firewall. This kind of transparency is important as the scenarios are ad-hoc in nature.
- **Infrastructural Administration** of overlay networks. This includes successful resolution of nodes, users and resources that are distributed in nature. Further, ensuring that the infrastructure is stable under high churn rates – joining and leaving of nodes. These facilities can be implemented by utilizing structured P2P algorithms.
- **Session management** by utilizing SIP-based session creation, transmitting media and files over these sessions and session termination. Providing support for creating sessions between two or more ad-hoc participants.

[5] Interactive Connectivity Establishment, an Internet-Draft is available at http://www.jdrosen.net/papers/draft-ietf-mmusic-ice-05.txt

- **Media support** by utilizing established standards such as Session Description Protocol (SDP) and Real-Time Protocol (RTP). Using various P2P and SIP-based initiatives for video, voice, text and file-transfer.
- **Trust and assurance** by providing trust scores of ad-hoc participants and secured key generation.

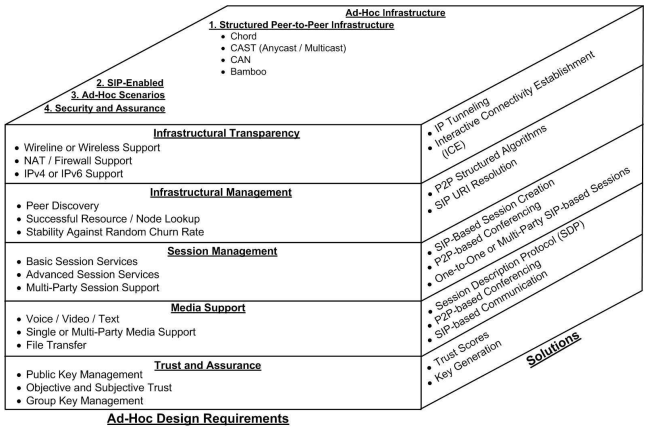

Figure 3: Overall Architecture

To implement the ad-hoc design requirements detailed above, a peer that supports various operations needs to be designed. The focus of implementation of this study is a peer (henceforth referred to as ad-hoc peer) as the infrastructure is P2P in nature. The fundamental assumption of this architecture is that the ad-hoc peer can be reached with an IP address (IPv4 or IPv6). The capabilities or solutions, illustrated in Figure 2, need to be implemented at different Internet Protocol (IP) layers – Application, Transport, Network, Data Link and Physical.

Over Application layer, SIP, P2P and Trust protocol needs to be implemented. SIP protocol provides session management features. Further, SIP provides certain addressing schemes such as SIP URI with which a user can be identified. Structured P2P protocols provide routing intelligence, node resolution and resource resolution over P2P infrastructure. Trust, described in the next section, will assure ad-hoc participants of their secured interactions with others. Over Transport layer, in addition to TCP and UDP, Real-Time Protocol (RTP) [ref required] needs to be implemented so that real-time media such as voice and video can be transmitted. The ad-hoc node needs to support IPv4 and IPv6 in the network layer. Further, as alluded to above, an ad-hoc peer needs to support wireline as well as wireless communication infrastructure in Physical and Data Link layers. Figure 4 illustrates the protocol stack for an ad-hoc peer.

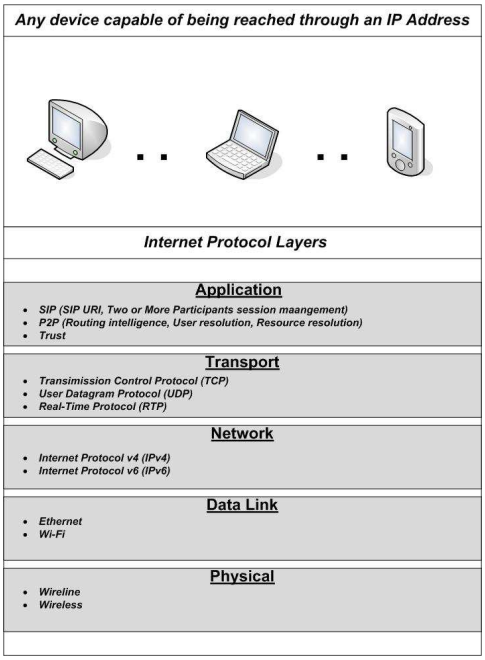

Figure 4: Protocol Stack for an Ad-Hoc Peer

4. Trust

Trust is the most important basis for security awareness between persons. Some common words associated with the notion of trust include: belief, confidence, faith, hope, expectation, dependence and reliance on the integrity, ability, or character of a person or thing to behave properly. Humans use trust every day to promote interaction and accept risk in situations where only partial information is available [42]. Without risk, there is no reason to trust. With operational trust, a peer must make a rational decision based on knowledge of possible outcomes for trusting and not trusting. One goal of a trust system is to enable one peer to determine the likelihood that another peer will behave properly [43].

Ad-hoc groups are self-provisioning; members can invite other collaborators to join their group, and assign them various privileges; they may be affiliated with many different organizations. A group is described as a number of peers that are governed under a set of rules describing minimal conditions of that group. Groups are presumed to be dynamic, meaning the membership is continually changing. Existing trust models rely on a centralized database, but P2P systems are decentralized; this makes managing trust an important problem to resolve since users must be their own trust managers [44]

Pretty Good Privacy (PGP) lets users increase trust in a transitive manner; and to select a key length appropriate for the situation. Members should be able to express trust in more dimensions, more richly, than the single dimension of trust with PGP [45]. In trust and reputation based models, each peer attaches a trust score to every other peer to avoid interacting with those having insufficient trust levels. [46]. True collaboration teams usually start with known people, with

unknown members added as needed. Ad-hoc interaction can only occur between "strangers" if a sufficient level of trust exists between them. Trust evolves over time as a result of experiences from interactions and local observations [47].

Some of the concepts proposed to dynamically maintain trust levels for P2P file sharing have potential applicability in the ad-hoc real-time collaborative setting [48]. An ad-hoc collaborative P2P-based trust system must be simple, process efficiently, be highly transparent, require minimal human interaction and accommodate both the "on the fly one time" collaboration as well as longer-term ad-hoc collaborative group relationships. Many trust attributes exist, however initially we incorporate just three dimensions: authentication confidence, availability and contribution. PGP's web of trust concepts will be applied. A key is deemed valid if it is either signed by the owner of the key ring or signed by enough key holders who are trusted by the key owner. Availability considers how recently the member has been on-line and the number of session time minutes. Quality, as measured in the number of kilobytes sent, and quality, a subjective peer measure of the relevance/ value of the information shared, are incorporated into the contribution value. Each peer will receive one trust score, based upon an algorithm that incorporates the various trust values, weighted to recognize that some attributes are more important. Table 2 proposes an initial set of attributes for calculating a trust score. Each trust attribute will be assigned a value between 0 to 10 points. The trust score algorithm will normalize these values.
$T = \sum (A + B + C + D + E + F + G)$ - this approach treats each attribute independently, summing the normalized values.
$T = \sum (A + B + C)\ W_1 + (D + E)\ W_2 + (F*G)\ W_3$ - this second approach groups attributes by dimension, then weights each dimension to derive a single trust score (T). The trust formula will incorporate logic to reflect that quantity alone is insufficient to improve one's trust score; contribution should represent the composite value of both quantity and quality. How to test for weaknesses during the iterative design process must still be defined.

Table 2: Trust Model Attributes

#	Dimension	Attribute	Rationale	Proposed Scoring
1	Authentication Confidence	Age of key (A)	The longer the length of time a user has been a member of the group, (as evidenced by the age of his public key) the higher the potential trust level.	The days will be broken into groupings and assigned point values. For example, 0 days (new key signed on current day) would be worth 0 points. A key that was 1 – 7 days might be worth 1 point. A maximum of 10 points, for any key with an age of 365 days or more.
2	Authentication Confidence	Introducer (B)	If a member was introduced by someone having a high trust level, then the new member will receive a higher level of trust (relates to positive association).	Self-signed certificates will receive no points for this attribute (no trust) One field of the public key will be used to carry the trust value associated with the introducer
3	Authentication Confidence	Number of members who have signed the	The more people, especially all those with higher levels of trust, who have signed that member's key, the greater the trust in that member's certificate, could be a	If no other group members have signed the key, then this attribute will have 0 points. A maximum of 10 points could be awarded, with 1 or 2 points being credited for each signer of the key, based on signer's trust

		certificates (C)	strong value in calculating the trust level	score
4	Availability	Last Date On (D)	The more recent the activity in that group, the higher the confidence / trust. There is less trust in those who are casual users in a collaborative setting. –relates to trust in the other party's "availability" in terms of likelihood to be on-line	Range from 0 points to 10 points Maximum points are assigned to those who have been on line most recently. 0 indicates has not been on the system within past 60 days Up to 10 points indicating was on the current date.
5	Availability	User Minutes (E)	Good behavior is equated with those members who are regularly active in the site, and stay on-line for more time. – relates to trust in the other party's "availability"	1 point will be awarded for every 60 minutes of session time.
6	Contribution	Quantity - Data "shared" (sent or transferred to others) (F)	Group collaboration flourishes in an environment where ideas and information is freely and regularly exchanged. One indication that a group member is contributing to the group is based upon the volume of data (e.g. KB) sent to others – what this doesn't address is the "quality" of the data. Perhaps a user should be able to update their own trust rating through local experiences.	Points will be credited for every 50 kilobytes of data that is sent by one peer member. Being decided - - whether to deduct points for non-contributing long-term members - whether/how to weight more heavily data sent more recently
7	Contribution	Quality (G)	Groups have a common goal or focus. The relevance and quality of one's contributions to the group is a purely subjective evaluation. -- A simple user interface is required to provide a peer member with the option to assign a quality rating to another peer.	Quality and quantity should be multiplied together in the formula, once weighted. This way, if users post lots of information of zero quality, it does not add to their trust rating in any way.

5. Current Status: Implementation and Evaluation

The design, described in previous sections, is being implemented by the authors. The overall approach is to formalize an intelligent ad-hoc peer. An intelligent ad-hoc peer includes P2P-based, SIP-based and Trust-related design requirements as detailed in Section 2. The ad-hoc peer will be capable of performing P2P-based structured overlay routing, resource and node resolutions, support SIP-based signaling protocol and include trust-related principles detailed in Section 4. In addition to including these features, the peer will have to be independent of various infrastructural settings such as wireline or wireless connection and NAT or Firewall interference. These combined features will enable participants in various ad-hoc scenarios to use this ad-hoc peer to exchange resources with other peers in either synchronous or asynchronous fashion and make decisions regarding the trustworthiness of other users or peers. Presently, SIP-based behavior has been implemented within this ad-hoc peer. The P2P-based feature is being included by utilizing existing open-source initiatives. Methodology for Automatically Creating, Evaluating, and Designing Overlay Networks (MACEDON) [49], an open-source project initiated by University of California,

San Diego is being utilized to implement structured overlay networks. This project provides facility of including various structured overlay methodologies that have been described in previous sections. Subsequently, trust–related principles will be included so that the ad-hoc peer is able to provide security and trust recommendations to the ad-hoc participants.

The design is being implemented in a way that accurate performance evaluation can be performed. The performance will be evaluated based on certain measurements. The primary measurements will be on the basis of scalability, reliability and latency. Scalability measurements will indicate whether the underlying P2P structured overlay supports random extensibility of the topology or node arrangements. Latency measurements will indicate whether nodes and resource resolutions are done in logarithmic bounds of overlay networks. Reliability will be evaluated to examine the degree of consistency provided by the network. To make this evaluation possible, the architecture is implemented over the NS-2 [50] simulation application. NS-2 is a discrete event simulator over which large-scale as well as small-scale simulations can be performed to effectively evaluate performance of architecture. NS-2 is targeted at networking research as it provides support for various networking protocols so that comparison between different approaches can be evaluated. Figure 5 illustrates the simulation testbed.

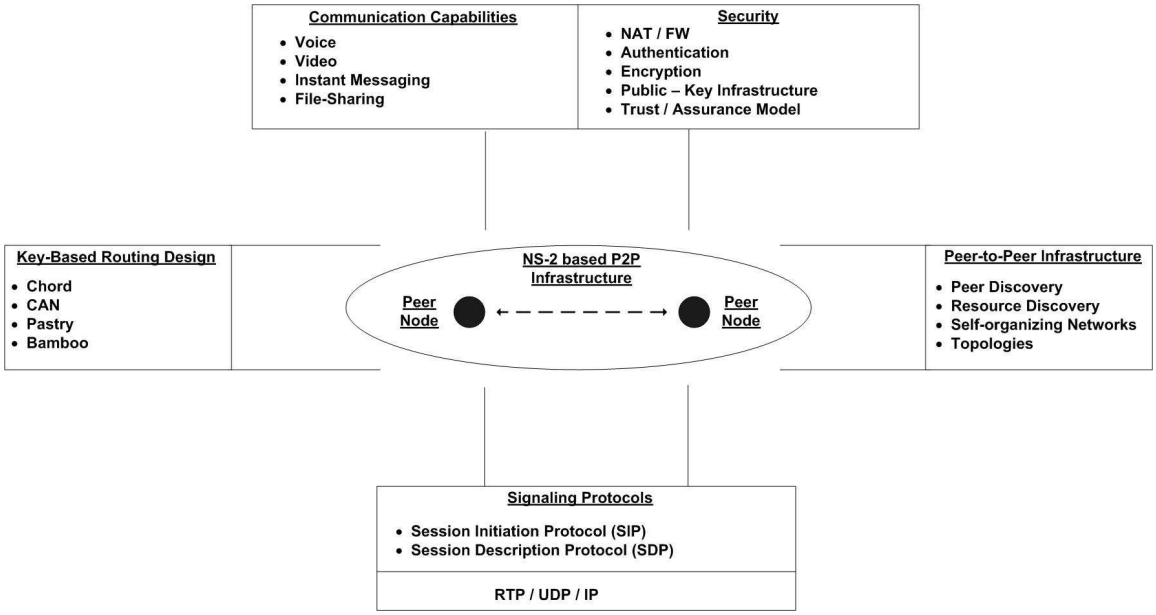

Figure 5 : Overview of Simulation Testbed

A general purpose simulation tool will be used to evaluate the proposed trust model. We want to assess whether the attributes being monitored do have the desired effect upon an ad-hoc peer's trust score. We hope to evaluate the time it takes to obtain trust values as well as scalability issues. Some simulation tests

will involve manipulating the variables, such as the weighting of the trust attributes and dimensions. Once we are satisfied that the model is functioning reasonably appropriately, we intend to simulate the effects associated with the dynamic membership of adhoc collaboration groups; this will be accomplished by adding and removing peers during the simulation process.

6. Future Work

The subject area of this study is of great interest to researchers in the field of IS. This study focuses on testing theories and developing systems and models for the field of P2P. Developing systems and models and evaluating existing theories have been identified as pre-requisites for utilizing design science research methodology [51, 52]. Influenced by the design science methodology, this study evaluates existing theories by implementing them in a specific design and attempts to formulate a novel architecture for enabling collaboration among ad-hoc, infrastructure-less participants.

The underlying design of this architecture has been finalized; it is being implemented over NS-2 and includes SIP-based session creation between participants. P2P-based behavior is being implemented over this architecture. Afterward, trust-related attributes will be included to form a secured and trust-assured collaboration. Some preliminary performance evaluation results should be available for presentation.

The long-term objective of this research is to recommend design principles. upon which ad-hoc peers can be formed. The outcome of this research will be specific sets of algorithms on which ad-hoc peers will be formed that would require no or minimum centralized infrastructure entities.

References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, N., and Schooler, E., "SIP: Session Initiation Protocol," Internet Engineering Task Force RFC 3261, June 2002.

[2] Toga, J. and Ott, J., "ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations," *Computer Networks and ISDN Systems*, vol. 31, no. 3, pp. 205-233, 1999.

[3] Singh, K. and Schulzrinne, H., "Peer-to-Peer Internet Telephony using SIP," appeared in the Proceedings of Network and Operating System Support for Digital Audio and Video, Skamania, WA, June 13-14, 2005.

[4] Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., and Xu, Z., "Peer-to-Peer Computing," Hewlett-Packard Laboratories, Palo Alto 2002.

[5] Hindus, D. and Schmandt, C., "Ubiquitous Audio: Capturing Spontaneous Collaboration," appeared in the Proceedings of ACM Conference on Computer Supported Cooperative Work, Toronto, Ontario, Canada, 1992.

[6] Zager, D., "Collaboration on the Fly," appeared in the Proceedings of Academic/Industry Working Conference on Research Challenges, Buffalo, NY, 2000.

[7] Berket, K. and Agarwal, D., "Enabling Secure Ad-Hoc Collaboration," appeared in the Proceedings of Workshp on Advanced Collaborative Environment, Seattle, WA, 2003.

[8] "Webex: Web Conferencing, Online Meetings and Video Conferencing," accessed at *http://www.webex.com/* on October 11, 2005

[9] "Microsoft Live Meeting (PlaceWare)," accessed at *http://esd.placeware.com/LM2005test/* on October 11, 2005

[10] IBM, "Lotus Domino Document Manager," accessed at *http://www.lotus.com/lotus/offering4.nsf/wdocs/domdochome* on October 11, 2005

[11] Shirkey, C., "Listening to Napster," in *Peer-to-Peer: Harnessing the Benefits of Disruptive Technology*, Oram, A., Ed. Sebastopol, CA, 2001, pp. 21-37.

[12] O' Reilly, T., "Remaking the Peer-to-Peer Meme," in *Harnessing the Benefits of Disruptive Technology*, Oram, A., Ed. Sebastopol, CA: O'Reilly and Associates, 2001.

[13] Usenet Group, "Usenet," accessed at *http://www.usenet2.org* on October 11, 2005

[14] Groove Networks, "Groove Virtual Office," accessed at *http://www.groove.net/home/index.cfm* on October 11, 2005

[15] Edwards, W. K., Newman, M. W., Sedivy, J. Z., Smith, T. F., and Balfanz, D., "Using Speakeasy for Ad Hoc Peer-to-Peer Collaboration," appeared in the Proceedings of Computer Supported Cooperative Work, New Orleans, LA, 2002.

[16] EarthLink, "EarthLink SIP Share: SIP-based P2P Content Sharing Prototype," accessed at http://www.research.earthlink.net/p2p/ on October 11, 2005
[17] Mockapetris, P. and Dunlap, K., "Development of the Domain Name System," *ACM SIGCOMM Computer Communications Review*, vol. 25, no. 1, pp. 112-122, 1995.
[18] Taylor, I., *From P2P to Web Services and Grids: Peers in a Client/Server World*: Springer-Verlag, 2005.
[19] University of Southern California, "SETI@Home," accessed at http://setiathome.ssl.berkeley.edu/ on October 13th, 2005
[20] Clarke, I., "The Free Network Project," accessed at http://freenet.sourceforge.net on October 13, 2005
[21] "Gnutella," accessed at http://www.gnutella.com/ on October 13, 2005
[22] JXTA, "Project JXTA," accessed at http://www.jxta.org on October 13, 2005
[23] Doval, D. and Maohony, D. O., "Overlay Networks: A Scalable Alternative for P2P," *IEEE Internet Computing*, vol. 7, no. 4, pp. 79-82, 2003.
[24] Darlagiannis, V., Mauthe, A., and Steinmetz, R., "Overlay Design Mechanisms for Heterogeneous, Large-Scale, Dynamic P2P Systems," *Journal of Network and Systems Management*, vol. 12, no. 3, pp. 371-395, 2004.
[25] Castro, M., Costa, M., and Rowstron, A., "Should we build Gnutella on a Structured Overlay?," *ACM SIGCOMM Computer Communications Review*, vol. 34, no. 1, pp. 131-136, 2004.
[26] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S., "A Scalable Content-Addressable Network," appeared in the Proceedings of Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, San Diego, CA, 2001.
[27] Stoica, I., Morris, R., Krager, D., Kaashoek, M., and Balakrihsnan, H., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17-32, 2001.
[28] Rowstron, A. and Druschel, P., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer system," appeared in the Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, 2001.
[29] Zhao, B., Huang, L., Stribling, J., Rhea, S., Joseph, A. D., and Kubiatowicz, J., "Tapestry: A Resilient Global-scale Overlay for Service Deployment," *IEEE Journal of Selected Areas in Communications*, vol. 22, no. 1, pp. 41-53, 2004.
[30] The Bamboo-DHT Group, "The Bamboo Distributed Hash Table: A Robust, Open-Source DHT," accessed at http://bamboo-dht.org/index.html on October 11, 2005
[31] Sit, E. and Morris, R., "Security Considerations for Peer-to-Peer Distributed Hash Tables," appeared in the Proceedings of First International Workshop on Peer-to-Peer Systems, Cambridge, MA, 2002.

[32] Ratnasamy, S., Stoica, I., and Shenker, S., "Routing Algorithms for DHTs: Some Open Questions," appeared in the Proceedings of First International Workshop on Peer-to-Peer Systems, Cambridge, MA, 2002.
[33] Bryan, D. A., Lowekemp, B. B., and Jennings, C., "SOSIMPLE: A Serverless, Standards-based P2P SIP Communication System," appeared in the Proceedings of International Workshop on Advanced Architectures and Algorithms For Internet Delivery and Applications, Orlando, FL, 2005.
[34] Castro, M., Druschel, P., Hu, C. Y., and Rowstron, A., "Proximity Neighbor Selection in Tree-based Structured Peer-to-Peer Overlays," Microsoft Technical Report MSR-TR-2003-52, 2003.
[35] Rosenberg, J., Weinberger, J., Huitema, C., and Mahy, R., "RFC 3489 - STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," Internet Engineering Task Force 2003.
[36] Earthlink, "EarthLink SIPShare: SIP-based P2P Content Sharing Prototype," accessed at *http://www.research.earthlink.net/p2p/* on October 13, 2005
[37] Ellison, C. and Dohrmann, S., "Public-Key Support for Group Collaboration," *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 547-565, 2003.
[38] Kim, Y., Perrig, A., and Tsudik, G., "Tree-Based Group Key Agreement," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 60-96, 2004.
[39] Chu, Y.-h., Rao, S. G., and Zhang, H., "A Case for End System Multicast," appeared in the Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems, Santa Clara, CA, 2000.
[40] Lienhart, R., Holliman, M., Chen, Y.-K., Kozintsev, I., and Yeung, M., "Improving Media Services on P2P Networks," *IEEE Internet Computing*, vol. 6, no. 1, pp. 73-77, 2003.
[41] Lennox, J. and Schulzrinne, H., "A Protocol for Reliable Decentralized Conferencing," appeared in the Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video, Monterey, CA, 2003.
[42] English, C., Nixon, P., Terzis, S., McGetrick, A., and Lowe, H., "Dynamic Trust Models for Ubiquitous Computing Environments," accessed at *http://www.teco.edu/~philip/ubicomp2002ws/organize/paddy.pdf* on March 15, 005, 2005
[43] Dewan, P., "Peer-to-Peer Reputations," appeared in the Proceedings of 18th Annual International Parallel and Distributed Processing Symposium (IPDPS'04), 2004.
[44] Aberer, K. and Despotovic, Z., "Managing Trust in a Peer-to-Peer Information System," appeared in the Proceedings of Conference on Information and Knowledge Management, Atlanta, GA, 2001.
[45] Stallings, W., *Network Security Essentials*, 2nd edition ed. Upper Saddle River, N.J: Prentice Hall, 2003.

[46] Gummadi, A. and Yoon, J. P., "Modeling Group Trust for Peer-to-Peer Access Control," appeared in the Proceedings of 15th International Conference on Database and Expert Systems Applications, Zaragoza, Spain, 2004.
[47] Mayer, R. C., Davis, J. H., and Schoorman, F. D., "An Integrative Model of Organizational Trust," *Academy of Management Review*, vol. 20, no. 3, pp. 709-734, 1995.
[48] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H., "The EigenTrust Algorithm for Reputation Management in P2P Networks," appeared in the Proceedings of 12th International Conference on World Wide Web, Budapest, Hungary, 2003.
[49] The MACEDON Project, "MACEDON," accessed at *http://macedon.ucsd.edu* on October 13th, 2005
[50] Information Sciences Institute, "The Network Simulator - ns2," accessed at *http://www.isi.edu/nsnam/ns/* on October 13th, 2005
[51] Hevner, A., March, S. T., Park, J., and Ram, S., "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, 2004.
[52] Rossi, M. and Sein, M., "Design Research Workshop: A Proactive Research Approach," appeared in the Proceedings of Information Systems Research Seminar in Scandanavia, Helsinki School of Economics, 2003.