SuperstringRep: Reputation-enhanced Service Discovery

Ryan Wishart, Ricky Robinson, Jadwiga Indulska
School of Information Technology and Electrical Engineering
University of Queensland
Brisbane, Australia
{wishart, ricky, jaga}@itee.uq.edu.au

Audun Jøsang
CRC for Enterprise Distributed Systems Technology (DSTC)
Brisbane, Australia
ajosang@dstc.edu.au

Abstract

Service discovery protocols are used in distributed systems to locate services for clients. The services that are located as well as the clients requesting service are commonly assumed to be trustworthy and reliable. This assumption can lead to security problems, particularly when clients and services are transient, as is the case with mobile networks. Authentication is commonly used in an attempt to circumvent this problem, but this only provides proof of identity, and does not vouchsafe behaviour. In this paper we present a new protocol, SuperstringRep, which combines service discovery with service scores to create a system-wide score for services which reflects the quality of service they offer. This integration of service discovery and reputation provides the service scores right when the client needs them: while selecting a service to interact with.

1 Introduction

Service discovery protocols are used to enable clients to find services matching their needs. The dynamicity of current distributed environments (services appearing and disappearing) introduces a problem: how can a client trust a service it has never seen before? The service could be certified by a trusted third party, though this is not a guarantee that the service will behave, or that the third party has not been co-opted. There is currently no reliable means for clients to evaluate the danger posed by services they have never interacted with. We present a solution derived from the field of P2P where researchers have proposed the use of entity reputation to influence interaction choices. Reputation systems keep track of what other nodes think of a particular node. Nodes wanting to contact the particular entity query the reputation system first. The reputation system provides a score derived from the opinions (which we henceforth refer to as testimonials) offered by other nodes which have interacted with the entity in the past. This information can then be used to determine whether or not to proceed with the interaction.

We present a protocol, SuperstringRep, which provides service discovery and reputation information to clients. The protocol is a combination of the Superstring service discovery protocol (Robinson and Indulska 2003) and a reputation system. The approach that we have taken has significant benefits. Firstly, the assumption that all services in the system are trustworthy can be discarded. In its stead is the notion of trust based on the service's actual performance. Secondly, the reputation information is provided to clients precisely when they require it: when selecting a service. This enables clients to make informed choices before entering into potentially damaging interactions. Lastly, integrating the reputation system into the service discovery protocol enables it to use the service discovery infrastructure for gathering and storing reputation information.

We also present a formula to calculate the service score for services and explain how we merged the reputation system with the Superstring service discovery protocol to create SuperstringRep.

The remainder of this paper is organised as follows. Section 2 presents related work. Section 3 covers the operation of SuperstringRep, while Section 4 presents a usage scenario demonstrating the functionality of SuperstringRep. In Section 5 we examine the operation of the reputation protocol, and determine the affect of different values for parameter terms in the reputation function. Section 6 discusses future work while Section 7 provides a conclusion and summary.

The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

2 Related Work

This section covers related work and provides background information.

2.1 Service Discovery and Superstring

Network resources and services can be located using a service discovery protocol. Several approaches to service discovery exist. Some approaches, such as that used by Jini (Jini 2001) and Service Location Protocol (SLP 1997), employ a centralised repository of service descriptions to which service advertisements and client queries can be sent. In other approaches, each device on the network can be queried about the services it offers. Bluetooth SDP (Bluetooth 2001) and Salutation (Salutation 1999) are examples of this model. Service discovery protocols such as Superstring (Robinson and Indulska 2003) and INS/Twine (Balazinska, Balakrishnan and Karger 2002) are intended to scale to the wide-area. These protocols are built on top of distributed hash table algorithms.

To demonstrate the viability of embedding a reputation system in service discovery we have chosen to use the Superstring protocol (Robinson and Indulska 2003) as it provides a powerful query language and preference mechanism.

The Superstring protocol uses a set of distributed query resolvers to answer service requests. The resolvers route amongst themselves using a distributed hash table routing structure. The distributed hash table structure is highly efficient, enabling messages to be routed within *log N* hops, where N is the number of nodes in the network. Messages are given an ID value based on their hash. This value is used in routing, with messages being sent to the resolver whose identifier is equal to or succeeds the message identifier. The messages move twice as close to the destination on each successive hop through the network.

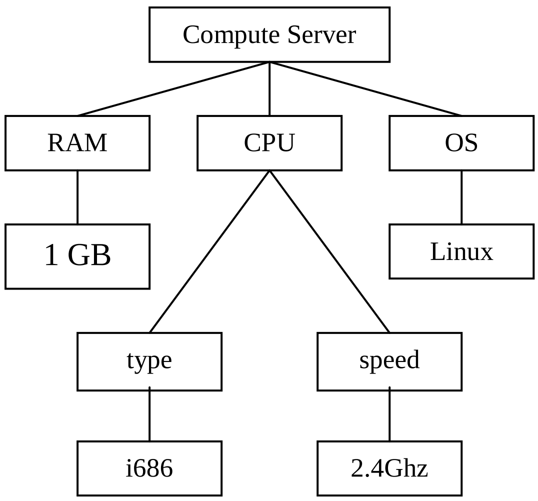

Figure 2-1 Example Superstring Service Advertisement

Services and clients communicate with Superstring resolvers using service advertisements and service queries, respectively. Advertisements and queries utilise a hierarchical format, with the root element describing the type of the service. An example service advertisement is given in Figure 2-1. The advertisement describes a compute server with 1GB of RAM and a 2.4Ghz processor running the Linux operating system.

The root element of the service description or query is hashed to produce a key for the distributed hash table. The key, along with the service description or query, is then routed to the appropriate Superstring resolver. Upon reaching this resolver, service descriptions are then distributed to a hierarchy of resolvers in such a manner that no resolver stores the entire description. Queries are then resolved by this hierarchy of resolvers. The hierarchy ensures that no resolver is overwhelmed by queries or advertisements. For any query, each resolver has a very small amount of work to do, meaning responses are timely and message queues remain short.

2.2 Reputation Systems

This section provides a further examination of reputation systems. As their basic functionality has been discussed in the previous section it will not be covered again. Instead, we continue the discussion of reputation systems by examining their architecture.

A reputation system can be classified as either centralised or distributed. In the centralised approach all ratings are gathered and maintained at a well-known, and fully trusted, repository. Entities wishing to obtain scores for potential partners can do so by contacting the repository directly. This centralised approach has the advantage of offloading all storage and calculation to the repository. This compares with the distributed approach employed in (Dingledine, Mathewson and Syverson 2003, Yu and Singh 2002), which requires all entities to remember their past interactions, and be able to provide testimonial ratings for any entity that they have encountered previously. In addition to the extra load placed on the client entities, the process of reporting the testimonial ratings introduces privacy issues: providing a report on an entity implies that you have interacted with it in the past. Through careful querying, any network entity can determine the interaction history of any other network entity. The central approach overcomes this by not releasing any testimonial ratings, only service scores.

In all reputation systems the calculation of service scores is handled by the reputation engine. The six types of reputation engine can be classified as:

- Simple summation models
- Discrete trust models
- Bayesian models
- Fuzzy models
- Belief models
- Flow models

The simplest are those using a summation engine. The engine takes the rating of an entity as the difference between all the positive ratings and all the negative ratings. This principle is used in eBay's reputation forum mechanism. A more advanced mechanism is used by Amazon.com, which takes the average of all the ratings. Weighted averaging can also be used, with weights determined by trustworthiness, age, sex, etc.

In contrast to the numerical summation models, discrete trust models, as used by (Abdul-Rahman and Hailes 2000), break trust into discrete categories such as very trustworthy, trustworthy, untrustworthy and very untrustworthy. As there is no mathematical basis to the engine, heuristic measures like look-up tables have to be used when updating reputation scores.

A more advanced engine is employed in Bayesian reputation models like (Josang and Ismail 2002), which use a statistical approach to calculate reputation. In this approach the reputation engine looks at the previous ratings of an entity and uses statistical methods to determine the most likely future rating. This is then given as the reputation score for the entity.

The belief theory approach to reputation relies on probability theory. Two examples using this approach are (Josang 1999), and (Yu and Singh 2002). The first represents belief in particular statements, such as service X is trustworthy, as opinions. Opinions are comprised of belief, disbelief and uncertainty regarding the statement. The level of uncertainty regarding the three values is also captured. Opinions can be mapped to Beta Probability Density Functions (PDF), meaning that the model is both belief and Bayesian based. The work by Yu and Singh takes a different approach, representing reputation as two possible outcomes: trustworthy and untrustworthy. The reputation of an entity is the difference between these two values.

Fuzzy logic reputation models make use of fuzzy logic membership functions to describe to what degree an agent can be said to be trustworthy or untrustworthy. An example of this technique is the REGRET reputation model (Sabater and Sierra 2002).

The final category, flow models, have a constant rating for the entire community. For one entity to increase in reputation, others must decrease. This paradigm is used by Google's PageRank algorithm to model the connectedness of web pages. The algorithm calculates a probability distribution over all web pages, assigning a rank (PageRank) to each page. The sum of the PageRanks is one. Pages with many incoming links have a high PageRank. The addition of a link from page A to page B increases the PageRank of B at the cost of slightly reducing the PageRanks of other pages.

In SuperstringRep we use the Bayesian reputation model as it is solidly based in statistical theory, yet is simple to implement. In the following section we examine the operation of SuperstringRep and explain its Bayesian reputation model.

3 SuperstringRep

The service discovery process introduces clients and services to other network entities they may never have encountered before. In the event that one of the parties proves to be malicious, the victim has no means of recourse, and no way to warn other entities. A reputation system for service discovery overcomes these problems by allowing entities to rate one another. The scores can be used by client entities to determine whether or not potential interaction partners are trustworthy. The need for such a system has been clearly established by prior work in (Yolum and Singh 2003, Yu and Singh 2001, Yu and Singh 2002, Yu and Singh 2003). In this paper we present SuperstringRep, a service discovery protocol with a built in reputation system. Integrating the reputation system with the service discovery protocol enables clients to obtain reputation information while selecting a service. The benefit of our approach is that it is not a total redesign of the service discovery mechanism. Although we demonstrate it in the context of SuperstringRep, running over Superstring, the reputation system is general enough to apply to other service discovery protocols currently in widespread use.

The basic operation of a reputation system was outlined in Section 2.2. The operational requirements of such a system can be summarised as:

- The ability to gather testimonial ratings from other entities.
- The ability to rate an interaction partner after the interaction has taken place.
- The ability to calculate service scores based on the combination of personal experience and testimonial ratings provided by other entities.
- The ability to distribute testimonial ratings for past interactions to interested parties.

The process of gathering information from and distributing information to clients are core functions of the service discovery process. In SuperstringRep we extended the existing mechanisms in the Superstring service discovery protocol to support the collection of ratings and the provision of service scores for the reputation system. The task of storing the testimonial ratings and generating a service score for a service is performed by the Superstring resolvers.

3.1 Obtaining reputation information

In our protocol, SuperstringRep, the service score for a service can be used as a search parameter on which the client can base her selection. When the parameter is not used to automatically tailor the selection, the service score of each matching service is returned, along with the service description, to the client.

3.2 Testimonial Rating Representation

The reputation system requires that the client entities provide an indication of their level of satisfaction with a service after having interacted with the service. This testimonial rating can be represented in a variety of different forms, depending on the reputation engine used. As discussed in Section 2, common approaches use:

- A single numerical value indicating satisfaction or dissatisfaction with the service. This is usually expressed over some range, e.g. (Salutation 1999).
- A discrete ranking system such as: Very Trustworthy, Trustworthy, Untrustworthy, Very Untrustworthy.
- A two part ranking capturing both the level of satisfaction and the level of dissatisfaction (r, s), as is used by the Bayesian approach (Josang and Ismail 2002).
- A ranking capturing the trust and distrust in an entity (t, d), as used by Yu & Singh (Yu and Singh 2002).
- A three part ranking like that used by the belief model developed in (Josang 1999). The three values capture belief, disbelief and uncertainty. This is represented as (b, d, u).

In the SuperstringRep protocol, clients submit a testimonial rating as a single numerical value in the range (Salutation 1999). This value is fed into a Bayesian probability function to calculate the service score based on testimonial ratings provided by clients. The Bayesian function requires two parameters as input, the client's level of satisfaction, and dissatisfaction with the service. The system splits the testimonial rating value into the two parts needed by the Bayesian formula using the equation described in Section 3.6.

3.3 Determining Client Satisfaction

After an interaction between a client and a service, the client entity determines a rating to be given as feedback for the service. This rating is provided to SuperstringRep as a testimonial rating representing the client's opinion of the service.

The rating of the service requires an objective evaluation be performed by the client. The establishment of criteria for this evaluation are considered to be outside the scope of this paper.

3.4 Reporting Interactions to the Service Discovery Mechanism

Once the client has calculated a testimonial rating, it contacts the nearest Superstring resolver. The structure of the testimonial rating is depicted in Figure 3-1. The testimonial rating from the client is structured as a hierarchy of elements for compatibility with the Superstring service query mechanism.

In SuperstringRep the client ID is taken as the IP address of the client.

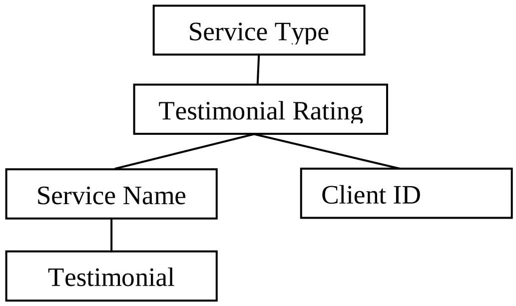

Figure 3-1 Structure of a SuperstringRep Testimonial Rating

The Service Name and Testimonial Value elements contain the name of the service the client interacted with and the client's testimonial rating of the service, respectively.

3.5 Storage of Testimonial Ratings

The SuperstringRep protocol stores client testimonial ratings on the distributed network of Superstring resolvers. Each of the testimonial ratings is associated with the identity of the client that provided them. In SuperstringRep the IP address of the client is used as an identifier. As Superstring and SuperstringRep operate over TCP, the IP address for the client can be reliably obtained from the TCP protocol.

Testimonial ratings, once gathered, are stored on a per service basis together in a table. An example representation is given in Figure 3-2.

Client IP	**Testimonial Rating**	**Timestamp**
102.1.2.3	0.75	14:20:02 8-8-2004
192.68.27.13	0.80	16:10:34 12-8-2004

Figure 3-2 Example Testimonial Ratings for a Service

The left hand column contains the IP address used by SuperstringRep to identify the client. Each client is only allowed one testimonial rating per service. If a client provides multiple ratings for a particular service, only the most recent will be taken.

The central column in the table is the testimonial rating provided by the client. The right hand column is a timestamp of when the testimonial rating was provided. The timestamp is used to determine how much a testimonial should affect the service score. This weighting is based on the age of the testimonial, where age is measured as the time difference between the most recently submitted testimonial and the testimonial in question. The aging process is further discussed in Section 3.8.

3.6 Calculation of Service Score

The score of a service is calculated by combining client testimonial ratings together. A Bayesian statistical approach, developed in (Josang and Ismail 2002), is then applied to this aggregate value to obtain the score of the service. The testimonial rating values, referred to as *v*, that clients provide are captured as single numerical values in the range [0,1]. To comply with the Bayesian formula, the single value is broken into values for satisfaction, *r*, and dissatisfaction, *s*, by the following equations:

$$r = v, \quad s = 1 - v \tag{1}$$

The Bayesian approach used in SuperstringRep calculates the score of a service as the expected value of the probability density function (PDF) for that service's cumulative *r* and *s* values. The beta PDF is defined for the probability variable *p* indexed by the values *α* and *β* as:

$$beta(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$
$$0 \le p \le 1 \; and \; \alpha, \beta > 0$$
$$where \; p \ne 0 \; if \; \alpha < 1 \; and \; p \ne 1 \; if \; B < 1$$

The density function gives the probability of p having a particular value. When used to calculate the score, we use α as the cumulative satisfaction with the service, and β as the cumulative dissatisfaction with the service. The relationship between (α, β) and (r, s) is given by the following equations:

$$\alpha = r + r_{\text{base}}, \quad \beta = s + s_{\text{base}} \tag{2}$$

where r_{base} and s_{base} are the Bayesian base rate parameters satisfying $r_{\text{base}} + s_{\text{base}} = 2$. For simplicity we will set the base rate so that $r_{\text{base}} = 1$ and $s_{\text{base}} = 1$ in this study, but the base rate can be used to reflect the base rate reputation in the community as a whole.

In Eq.(2), the values of *r* and *s* represent the accumulated testimonial ratings for the service, and the PDF then represents the service score for the service. In order to have a simpler representation of the service score, the probability expectation value can be used (Josang and Ismail 2002). The expected value of *p* is defined as:

$$E(beta(p \mid \alpha, \beta) = \frac{\alpha}{\alpha+\beta} \tag{3}$$

This function gives the most likely future service score given the history of the service. The value is in the range [0,1], providing an easy to understand service score for services, with 0 representing the worst possible score and 1 representing the best possible score.

3.7 Combining Testimonial Ratings

The testimonial ratings referring to a particular service, which we will call Z, can be combined together to form an aggregate testimonial rating for Z. This aggregate testimonial rating is used to calculate the service score for Z.

The testimonial rating regarding Z from a particular client, X, can be expressed as a vector $P_Z^X = [r, s]$. The super script refers to the origin of the testimonial rating, and the subscript to the service being rated. Vector addition can be used to aggregate the vectors for each client that provided a testimonial rating for Z to obtain $P(z)$, the aggregate testimonial rating for Z. As an example, if service Z had two clients, X and Y, provide testimonial ratings, then the value for $P(z)$ would be $P(z) = P_Z^X + P_Z^Y = [r_X + r_Y, s_X + s_Y]$. The more general form of this is

$$P(z) = \sum_{x \varepsilon C} P_Z^X \tag{4}$$

Eq. (4) can be read as when a client X is an element of the set of clients that have provided a testimonial rating regarding service Z, which we call C, the aggregate rating for Z is the sum of all the testimonial ratings provided by the elements of C.

As stated in (Whitby, Josang and Indulska 2004), once this aggregate testimonial rating has been calculated for service Z, the reputation beta PDF can be obtained, and the service score *R*(*Z*) computed.

$$beta(P(Z)) = beta(p \mid r_Z + r_{\text{base}}, s_Z + s_{\text{base}}) \tag{5}$$
$$R(Z) = E\big(beta(P(Z))\big) \tag{6}$$

By substituting the beta PDF terms from (5) into the equation given in (3), R(Z) can be calculated in terms of *r* and *s*.

$$R(Z) = \frac{r_Z + r_{\text{base}}}{r_Z + s_Z + r_{\text{base}} + s_{\text{base}}}$$

where $r_{\text{base}} = 1$ and $s_{\text{base}} = 1$. This equation is used by the Superstring resolvers in SuperstringRep to calculate the service score of a service once they have determined the aggregate testimonial rating for that service.

3.8 Aging of Testimonial Ratings

Old testimonial ratings should not have undue influence on the score of a service, as they represent out of date information. This can be prevented by introducing a forgetting factor, λ, where $0 < \lambda < 1$. This forgetting factor is applied to each of the testimonial ratings regarding a service (Josang and Ismail 2002, Whitby, Josang and Indulska 2004). In SuperstringRep, the aging factor is applied as follows

$$P_{Z,T}^X = \lambda^{Tr-Tc} P_Z^X$$

where the terms *Tr* and *Tc* represent the time of the most recent rating of the service Z, and the time the current testimonial rating, P_Z^X , was submitted, respectively.

4 SuperstringRep Usage Scenario

In this section we demonstrate the operation of our reputation enhanced service discovery protocol, SuperstringRep, using a fictional scenario.

An Online Data Storage facility, ODS, has just started. To attract customers, it advertises its services on the local service discovery network, which makes use of the SuperstringRep protocol. ODS registers itself with SuperstringRep by contacting its nearest Superstring resolver. ODS then submits a Superstring service advertisement providing its details to the resolver. The specifics of this procedure are considered outside the scope of this paper, but have been addressed in previous published work by the authors (Robinson and Indulska 2003). Some time following the registration of ODS, a client entity, Alice, develops the need for an online data storage facility. Having no knowledge of which services are available, she proceeds to contact her nearest SuperstringRep resolver.

Alice determines her needs and issues a Superstring service query specifying the type of service she requires. The resolver accepts the query, and then proceeds to resolve the service query using the procedure explained in Section 2.1. The protocol determines that there is only one service matching Alice's query. The service score for the service is calculated using the equation developed in Section 3.6. The testimonial rating entries corresponding to ODS at the time of Alice's request are given in Figure 4-1.

Client IP	Testimonial Rating	Timestamp
102.1.2.3	0.80	04:30:00 23/8/2004
192.22.21.8	0.90	05:30:00 23/8/2004
103.2.4.9	0.825	22/8/2004

Figure 4-1 ODS Testimonial Rating Table

The calculation of the *r* and *s* values is performed using Eq.(1).

Client IP	Testimonial Rating	r-value	s-value
102.1.2.3	0.80	0.80	0.2
192.22.21.8	0.90	0.90	0.1
103.2.4.9	0.825	0.825	0.175

Figure 4-2 R and S values for ODS score calculation

The values are then aged with reference to the most recent testimonial rating. We take the time difference to be expressed in days, and the value of λ to be 0.9. The affect of varying the value of λ is examined in the following section.

Client IP	Timestamp	Tr-Tc	λ r	λ s
102.1.2.3	23/8/2004	0	0.80	0.20
192.22.21.8	22/8/2004	1	0.81	0.09
103.2.4.9	19/8/2004	4	0.541	0.115
SUMMATION OF VALUES			2.151	0.405

Figure 4-3 Table of values calculated for P(ODS)

The service score of the service ODS is then the expected value of the sum, calculated using Eq. (6).

$$R(ODS) = E\big(beta(P(ODS))\big) = 0.691$$

After examining the service description and the score of 0.691, Alice decides to use the service provided by ODS. The service does not meet her expectations and she gives ODS an unsatisfactory testimonial rating of 0.3. The testimonial rating is provided to the closest Superstring resolver. Once obtained, Alice's testimonial rating is added to the table maintained for ODS.

Client IP	Testimonial Rating	Timestamp
102.1.2.3	0.80	04:30:00 23/8/2004
192.22.21.8	0.90	05:30:00 23/8/2004
103.2.4.9	0.825	22/8/2004
192.29.2.58	0.65	25/8/2004

Figure 4-4 Updated ODS Testimonial Rating Table

After considering the poor service she received, Alice decides to change her rating of ODS. She submits another testimonial rating to the Superstring resolver with a new rating of 0.1.

Client IP	Testimonial Rating	Timestamp
102.1.2.3	0.80	04:30:00 23/8/2004
192.22.21.8	0.90	05:30:00 23/8/2004
103.2.4.9	0.825	22/8/2004
192.29.2.58	0.1	26/8/2004

Figure 4-5 Updated Testimonial Rating Table for ODS

The updated testimonial rating overrides the previous entry made by Alice. The new service score for ODS, R'(ODS), based on the testimonial rating that Alice

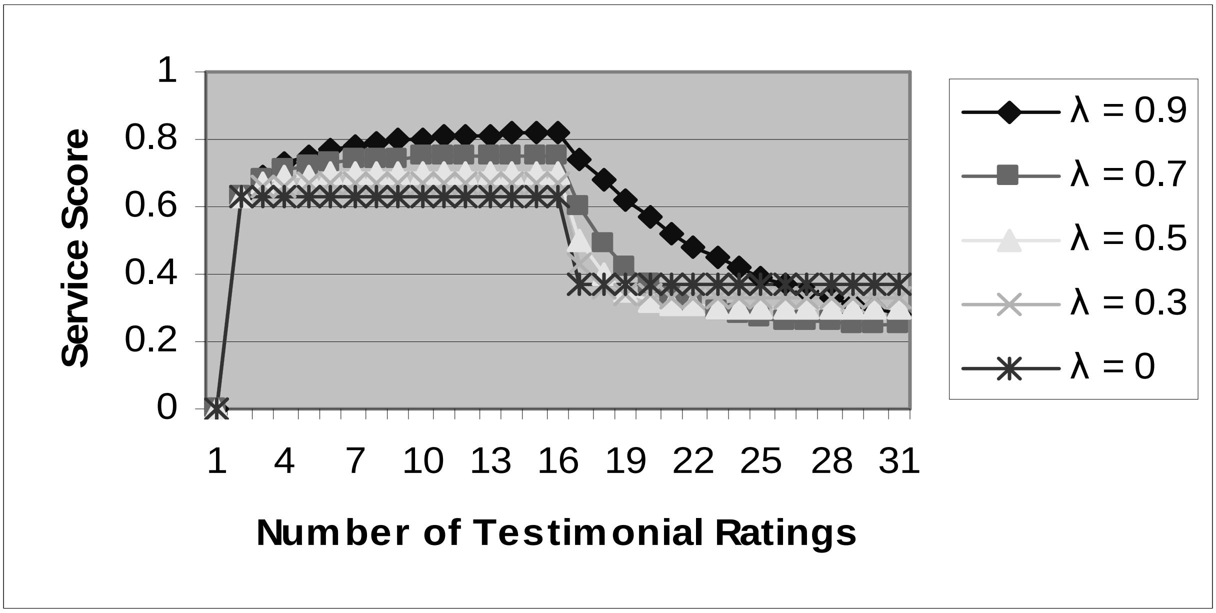

Figure 5-1. Plot of Service Score vs Number of client Testimonial Ratings submitted

provided, would be calculated according to the values provided in Figure 4-6.

Client IP	Timestamp	Tr-Tc	λ r	λ s
102.1.2.3	23/8/2004	3	0.583	0.162
192.22.21.8	22/8/2004	4	0.59	0.065
103.2.4.9	19/8/2004	7	0.394	0.083
192.29.2.58	26/8/2004	0	0.55	0.45
SUMMATION OF VALUES			2.117	0.76

Figure 4-6 Final step of calculations for ODS's Service Score

Giving a value for R'(ODS) of $R'(ODS) = 0.639$

As can be seen in the next section, altering the value of the aging factor, λ, leads to more drastic changes in the score of the service.

In the following section we examine the operation of the reputation system used in SuperstringRep and discuss the relevance of values for the particular parameters in the formula.

5 Evaluation of SuperstringRep Reputation Protocol

In this section we evaluate the operation of the reputation system used in SuperstringRep. This was done to determine its suitability for reputation services.

The aging factor used by the protocol to discount older values and lend greater weighting to new testimonial ratings was examined. This was achieved by varying the value of the aging factor λ. The results of the evaluation are shown in Figure 5-1. The service score of the service is plotted as a function of the number of testimonial ratings provided by client entities. The test was conducted using thirty (30) testimonial ratings. The testimonial ratings provided by clients were all of equal value and initially positive. The testimonial ratings all changed to negative after the 15th testimonial rating was submitted. This is reflected in the graph with the increase in service score until the 15th testimonial rating, at which point the service score drops as negative testimonial ratings accumulate.

When the aging factor is set to 0, the service score of the service is taken as the most recent testimonial rating. Conversely, when the aging factor is set to 1, all testimonial ratings have equal weight. Both of these extreme cases are undesirable. Remembering none of the previous testimonial ratings means a service score can fluctuate wildly. Weighting all testimonial ratings equally makes the service score slow to respond to changes.

A high value for λ, i.e. close to 1, extends the period required for the score to become stable. This phenomenon can be observed in Figure 5-1 when the number of testimonial ratings becomes greater than 16. The service score changes slowly, particularly when a rapid change in the quality of the service occurs. This is undesirable for a reputation system, where the reputations are meant to provide an accurate reflection of the current quality of service offered.

As can be seen in Figure 5-1, smaller values of the aging factor, such as 0.5 and 0.3, cause the system to respond more quickly to abrupt changes. As discussed in (Josang and Ismail 2002), service scores making use of the smaller λ values become stable more quickly. The value of the stable region is also less extreme.

These findings suggest that the SuperstringRep protocol should make use of small λ values in order to be more responsive to changes in service activity. The exact value will depend on the demands of the environment in which the protocol operates.

6 Future Work

There are several problems that exist with the current solution which we will address in future work. One of these is the lack of incentive for clients to provide testimonial ratings for services once they have interacted with them. In addition to this, clients that do provide testimonials may provide overly positive ratings (Resnick and Zeckhauser 2002). This is a problem for reputation systems. Whether done maliciously or out of kindness, extreme ratings constitute an attack against the reputation system and result in distorted service scores. Our current implementation of SuperstringRep does not attempt to remove these incorrect ratings. This problem could be addressed by using a filtering mechanism, as proposed in (Whitby, Josang and Indulska 2004). The filter uses a statistical approach based on the assumption that honest ratings follow a statistical trend. Ratings that stray from this trend are removed.

Services that provide inconsistent levels of service to different clients are difficult to detect in reputation systems. These inconsistencies can manifest as bullying of particular clients or services providing poor service once the service score has reached a suitably high level. When the score starts to drop, the service returns to good behaviour. The cycling between good and bad behaviour can be detected, but if the cycles are relatively short, it could be mistaken for negative client feedback as opposed to an actual change in service behaviour.

Another concern is ballot box stuffing, an attack whereby clients provide more than the legitimate number of testimonials. In SuperstringRep clients are identified based on their IP address. Each IP address is only allowed one testimonial per service. If multiple testimonials are submitted, only the most recent is retained. This scheme prevents ballot box stuffing when client IP addresses are relatively static, but fails if IP addresses change frequently.

The several problems outlined in this section are all areas of current research in the field of reputation management. Our SuperstringRep protocol addresses some of these problems within a limited context. In future work we intend to incorporate the filtering mechanism of (Whitby, Josang and Indulska 2004) to increase the reliability of our hybrid reputation and service discovery protocol and make it more robust against misbehaving clients and services.

7 Conclusion

This paper presented SuperstringRep, a protocol combining the Superstring service discovery protocol with a reputation system. The protocol operates using an elegant architecture which provides client anonymity without the bottleneck of a centralised system. Unlike a fully distributed reputation system, clients do not store their testimonials. Instead the testimonial ratings are handed over to the distributed Superstring resolver network. The resolvers maintain the testimonial ratings and use them to calculate the score for services. The scores for services are provided to clients as an augmentation to the service discovery process.

This method we use of including service scores with service information is efficient, as the information is provided to clients when they need it, without having to ask a third-party about a service's score.

The service scores are calculated using a Bayesian reputation engine. The method used has a sound basis in statistical theory, ensuring service scores are meaningful. The reputation engine also takes into account the general level of trustworthiness in the community of clients and services. This value is used to tune the scores it generates for services.

Lastly, the reputation system for SuperstringRep has only been demonstrated running with Superstring. In fact it is general enough to be applied to any service discovery protocol capable of supplying the necessary infrastructure. All that is required is a means of gathering, maintaining and distributing client testimonial ratings and service scores. These basic operations are already fulfilled by common service discovery infrastructure like (SLP 1997, Jini 2001). As such, extending them to support our reputation protocol should not represent a significant challenge.

8 Bibliography

Abdul-Rahman, A. and Hailes, S. (2000): Supporting Trust in Virtual Communities. *Hawaii International Conference on System Sciences*, Maui, Hawaii.

Balazinska, M., Balakrishnan, H. and Karger, D. (2002): INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. *Pervasive 2002*, 195-210, LNCS.

Bluetooth: Bluetooth Specification version 1.1, Bluetooth SIG, https://www.bluetooth.org/spec/, Accessed 2004 30th Aug.

Dingledine, R., Mathewson, N. and Syverson, P. (2003): Reputation in P2P Anonymity Systems. *Workshop on Economics of Peer-to-Peer Systems*, Berkley CA, USA.

Jini: Jini Technology Core Platform Specification v1.2, Sun Microsystems, http://wwws.sun.com/software/jini/specs/, Accessed 1st Aug 2004.

Josang, A. (1999): Trust-Based Decision Making for Electronic Transactions. *NORDSEC'99*, Stockholm, Sweden.

Josang, A. and Ismail, R. (2002): The Beta Reputation System. *Bled Electronic Commerce Conference*, Bled, Slovenia.

Resnick, P. and Zeckhauser, R. (2002): Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. *Advances in Applied Microeconomics* Vol 11.

Robinson, R. and Indulska, J. (2003): Superstring: A Scalable Service Discovery Protocol for the

Wide-Area Pervasive Environment. *ICON'03*, Sydney, Australia, 699 - 704.

Sabater, J. and Sierra, C. (2002): Social ReGreT, a reputation model based on social relations. *SIGecom Exchanges* 3(1): 44 - 56.

Salutation: Salutation Architecture Specification (Part 1) v2.0c, The Salutation Consortium, http://www.salutation.org/, Accessed 30 Aug 2004.

SLP: Service Location Protocol, IETF, http://www.ietf.org/rfc/rfc2165.txt, Accessed 30 April 2004.

Whitby, A., Josang, A. and Indulska, J. (2004): Filtering Out Unfair Ratings in Bayesian Reputation Systems. *AAMAS 2004*, New York, USA.

Yolum, P. and Singh, M.P. (2003): Self-Organizing Referral Networks: A Process View of Trust and Authority. *ESOA*, Melbourne, Australia.

Yu, B. and Singh, M.P. (2001): Towards a Probabilistic Model of Distributed Reputation Management. *Proceedings of Fourth International Workshop on Deception, Fraud and Trust in Agent Societies*, 125 -137.

Yu, B. and Singh, M.P. (2002): An Evidential Model of Distributed Reputation Managment. *AAMAS'02*, Bologna, Italy, 294 - 301.

Yu, B. and Singh, M.P. (2003): Detecting Deception in Reputation Management. *AAMAS'03*, 73-80.