Design of a Scalable Clearing House Architecture

Lakshminarayanan Subramanian, Chen-Nee Chuah, Ramakrishna Gummadi
{lakme, chuah, ramki}@cs.berkeley.edu
Department of Electrical Engineering and Computer Sciences, U. C. Berkeley

Abstract—
A Clearing House (CH) coordinates the interactions between the various Internet Service Providers (ISPs) in the network. The structure of the CH is designed for efficient provisioning of resource reservations and secure billing services. In addition, the CH must scale over both distance and number of ISPs to support the operation of ICEBERG in the wide area. Three key ideas employed in this design to make the CH scalable to a large user base are *hierarchy*, *aggregation*, and *indirection*. We assume the network is composed of various basic domains (based on either administrative or geographic boundaries) which are aggregated to form *logical domains*. This introduces a hierarchical tree of the logical domains and a distributed CH architecture is associated with each logical domain in the tree. Each of these distributed architectures maintains only the aggregated state of their domains. Call setup time is reduced by employing powerful statistical techniques like Kalman filters to estimate the call traffic across various links and perform advance reservations. The report also discusses the scalable security, billing and privacy issues in the Clearing House.

I. Design Requirements

Before presenting our design for the Clearing House (CH), it would be necessary to understand the basic requirements and the goals that need to be satisfied by the CH architecture. One of the basic requirements of the CH is to extend rather than modify the existing architecture of the ISPs to minimize the development cost. The CH enhances the services and performance of the system by adding some functionality to the ISPs.

The basic goals of our Clearing House architecture are to:

- provide a near-optimal match of user-specified quality-of-service guarantees for call requests with the available resources of the ISPs.
- build a scalable infrastructure for the Clearing House. Scalability is measured along the following lines:
  - increase the throughput measured as the number of calls served proportionally with an increase in the processing power of the CH nodes *incremental scalability*.
  - reduce the state maintained in each node of the CH and the ISP routers.
  - reduce the call setup time with increase in processing power.
  - support scalable billing services.
- support fault tolerance and recovery.
- optimize network utilization under heavy traffic loads.
- separate call-setup and resource reservation in order to reduce the resource reservation time during call setup.
- support multicast operations and build a near optimal multicast tree.
- support mobility and dynamic path changes.
- provide a secure real-time billing architecture.
- ensure privacy of both ISPs and users.

In designing the Clearing House architecture, we make the following assumptions about the ISPs:

1. All edge routers in an ISP have the capability of collecting traffic statistics and estimating bandwidth requirements.
2. Control paths and data paths are separated.
3. The information posted by an ISP to the CH is trustable.
4. Routers have specific mechanisms for measuring average queuing delay and delay variance.
5. Every ISP has an inbuilt monitoring capability.

II. Clearing House Infrastructure

In this section, we provide a complete description of the Clearing House and its various functionalities. Our architecture contains two types of Clearing Houses, namely the *Internal Clearing House (ICH)* and the *External Clearing House (ECH)*. The ECH or third-party CH acts as the coordinator among the various ISPs in the network and the ICH is an agent present in every ISP, and is responsible for routing intra-ISP calls and monitoring the incoming and outgoing traffic from other ISPs into its own network. The ECH on the other hand, handles routing of all inter-ISP calls.

Every ECH performs the following operations:

1. Perform hierarchical routing of call traffic between domains.
2. Aggregate call requests between domains and service aggregate requests rather than individual calls.
3. Compute a near-optimal path for routing calls in the network.
4. Perform resource reservation along a path from one service provider to another service provider.
5. Receive information about reservation enhancements and online reservation costs from ISPs along various paths.
6. Inform the user about the billing prices of the various service providers in a certain domain.
7. Store the billing prices, quality and latency provided by various service providers and constantly update the information based on the available bandwidth and latency of the network in its domain.
8. Receive multicast requests and compute a multicast tree in coordination with other ECHs.
9. Monitor and verify the QoS guarantees of service providers for the calls routed through different ISPs.

The ICH, on the other hand, performs a different set of operations:

1. Route intra-ISP calls (perform hierarchical routing to achieve scalability).
2. Monitor incoming and out-going traffic, estimate the inter-domain external traffic (traffic from other ISPs), and perform reservations in its network for supporting this traffic.
3. Update the ECH about the reservation and the guaranteed traffic that the ISP can support.
4. Update routing tables in edge routers for supporting different calls.
5. Monitor mobility of users and performing dynamic call routing.

A. Hierarchy and Aggregation

The first step involved in setting up this architecture, is to divide the network into non-intersecting *basic domains*. A basic domain refers to a physical region which can be a collection of various area codes. Adjacent basic domains are clustered to form larger *logical domains* which may refer to geographic boundaries like states or small countries. The domain structure helps in building a *hierarchical tree* in the network. A distributed CH architecture is associated with every domain represented in the tree. The number of nodes in the distributed system will vary as a function of the load on the system and the level of the node in the hierarchy.

An ISP can span across many domains. A CH in the hierarchy aggregates all inter-domain call requests to a particular domain and sends this aggregated request to the parent CH. In other words, all call requests between two domains would be aggregated as a single request at a parent CH. Therefore, a CH of a logical domain that is a collection of K logical sub-domains would contain $O(K^2)$ call requests. Typical values of K are around $10 - 50$. Only the Clearing Houses at the local operators maintain per-flow state information.

B. Clearing House State

A distributed third-party CH is associated with every node in the *hierarchy*. A CH of a basic domain is called a *Local Clearing House (LCH)* and all other Clearing Houses up in the hierarchy are called *Global Clearing Houses (GCH)*. All service providers present in a domain advertise their prices to the *Local Clearing House*. The service providers would offers various prices based on the domain of the final destination. (e.g., call Canada 7/9cents/min). The service providers also advertise the paths between service providers to the *LCH*.

The LCH keeps track of the set of all paths between users and service providers in a domain and the paths between service providers in its basic domain. It also keeps track of the list of prices of calls within its basic domain and sends the list of advertised prices for a domain outside the basic domain of the CH to the next CH in the hierarchy. The *LCH* sends all paths that pass from/to a point in its basic domain to a domain outside to the next level in *H-Tree*. A global CH keeps track of the paths that run between two sub-domains and its corresponding latency, average queuing delay, variance of queuing delay and reservation status. It also keeps track of the advertised prices of the various service providers for calls within its domain.

C. Hierarchical Routing

The shortest path routing algorithm is not scalable because of the following reasons:

1. Every router needs to maintain $O(n)$ routing table information if one needs to perform shortest path routing where n is the number of nodes in the network.
2. Computing the shortest path between two points would at least take $O(e)$ time where e is the number of edges in the graph.
3. Computing shortest path for all call requests on the fly would be a very cumbersome and complicated job, since the amount of state information that needs to be maintained is very high.

Considering the scalability requirements of ICEBERG, we would employ a different kind of call routing mechanism called Hierarchical routing. This routing mechanism can be easily supported using the domain model for the Clearing House. The hierarchical routing does not always route through the shortest path but in most cases, the cost of the path is comparable to the optimal cost. Given a sender and a receiver, we compute the shortest path from the sender to the receiver in the following manner:

1. In the first step, we go up in the hierarchy from the sender's LCH to find the GCH whose domain contains both the sender and the receiver. Call this GCH as A. Let $s(A)$ and $r(A)$ denote the children of A which contain the sender and receiver.
2. A finds the best path between the sub-domains $s(A)$, $r(A)$. Let it pass from x in $s(A)$ to y in $r(A)$.
3. Find the best path from sender to x and y to receiver recursively.

In this routing algorithm, once the GCH, A has been located, the route from the sender to the receiver is found in a distributed fashion by the various CHs.

D. Caching

Caching can help in improving the performance of the ICEBERG system tremendously. An LCH or GCH can cache the paths that were used for previous call requests. The total number of inter-domain paths in a particular CH node is $O(K^2)$ if K is the number of sub-domains of a particular logical domain. For typical values of K which range from $10 - 50$, the cache can store all inter-domain paths. A local cache in a LCH can also store the price listings of various service providers to different destinations.

E. RxW Scheduling

RxW scheduling is a very good algorithm to be employed in the Clearing House scheduler. Given two sub-domains i and j of the logical domain of a CH, let R_{ij} and W_{ij} represent the total number of call requests and maximum waiting of the call requests between i and j. RxW scheduling algorithm schedules the aggregate call request with the maximum value of $R_{ij} \times W_{ij}$. Hence we maximize the throughput(number of call requests) serviced without unduly affecting the response time for call requests.

F. QoS guarantees

Every edge router in the network measures the average queuing delay and the queuing delay variance at its router and informs the CH about these values. When the CH computes the shortest path between two domains, the CH includes these delay metrics in the computation and outputs the path with minimum delay. Delay jitter calculations along a path can also be performed in a similar fashion.

G. Strengths of our Approach

The basic strengths of our approach are:

- The state that needs to be maintained by the entire CH architecture is shared between various the various CH nodes. Every CH node maintains only the state for its domain.

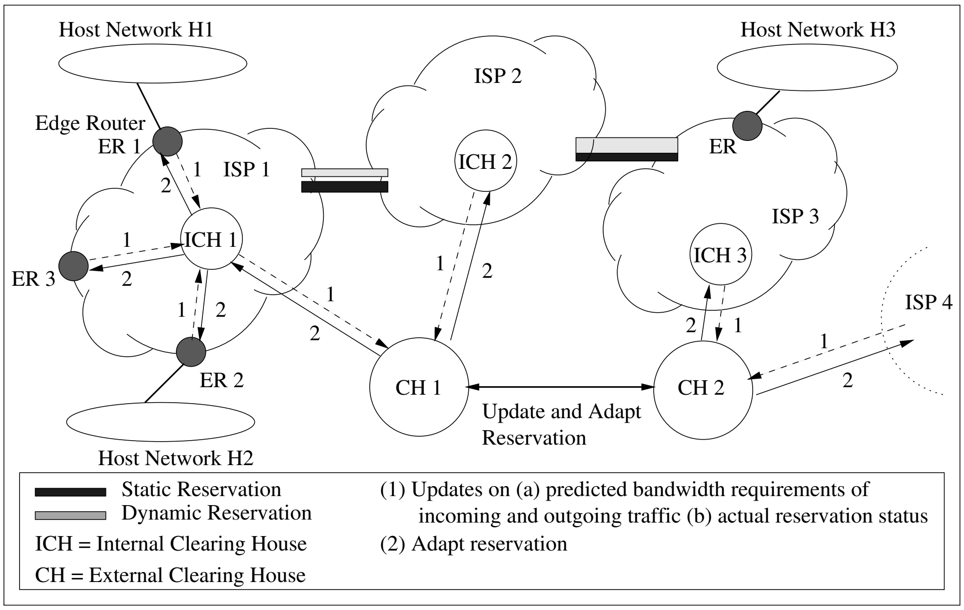

Fig. 1. A High-level view of the CH architecture during call setup.

- This architecture is optimized for making decisions based on locality. Call setup within a particular domain is made very fast.
- size of the queue of call requests in every CH is bounded due to call aggregation.
- Caching of inter-domain paths can enhance the performance of the system considerably.
- billing of calls can be localized and one can provide real-time billing in this architecture.
- process of path discovery is load balanced between various CH nodes.
- core routers in ISPs do not maintain much information.
- RxW scheduling not only increases the throughput of the system by servicing a lot of call requests in one shot but also services call requests with a high waiting time.

III. Resource Reservations and Traffic Monitoring

A. Overview

This section describes the resource reservation and traffic monitoring mechanisms involved in the Clearing House (CH) structure, which are critical for providing quality-of-service (QoS) in wide-area networks.

Currently, ISPs negotiate the volume and the price of high-priority traffic to be exchanged between different domains through service level agreements (SLAs). Based on SLAs, ISPs engineer their network so that they can support the high-priority traffic without congestion. However, the fluctuation of local traffic volumes produced by end-users has to be reflected in the SLAs between core networks. Many questions arise in this scenario. How and when is the SLA between two ISP established? What happens when the traffic travels through many ISPs? Our CH architecture approaches these questions based on two basic concepts: aggregation and logical hierarchy.

Figure 1 shows a typical wide-area scenario that spans multiple ISPs. We assume each edge router(ER) has (a) a monitoring system that measures the incoming and outgoing traffic, and (b) a Kalman filter that predicts the bandwidth requirement over time. Each edge router notifies the local CH of the dynamic changes in bandwidth requirements, and reservation status. The Internal Clearing House (ICH) maintains the resource reservations between different edge routers within the same ISP (intra-ISP). The ICHs in the same logical domain (ICH domain) send regular updates to the External Clearing House (ECH), which keeps track of the "reservation status" between different ISPs, i.e. whether resource reservation exists in a particular path and if so, how much bandwidth is reserved. Based on this information, the ECH facilitates the SLA negotiations between different ISPs (inter-ISP).

Admission Control: Whenever a sender wants to make a call to a receiver, there should be sufficient resources along the particular path from the sender to the receiver. Since on-line resource reservation is very costly, the goal of our design is to minimize the amount of per-link reservation that needs to be made for a particular call. Based on the reservation status of each ISP, a particular path is chosen such that the number of new per-link resource reservations is minimized.If the ICH fails to locate any links with sufficient resources reserved to complete a chosen path, the ER will either block the new call, or renegotiate with the user for the amount of required resources to carry the call. The decision involves some trade-offs in the user perceived quality of the call, the cost for increasingly scarce resources, and the additional latency to complete call setup.

B. Resource Reservations and Bandwidth Predictor

One very important feature of our design is that resource reservation is not performed on a per-call basis, but aggregated over various connections that use a particular link. Resource reservations are set up from ISP to ISP, instead of end-to-end. There are two types of reservations: static and dynamic. A static reservation of a particular link is one that is reserved in advance based on average aggregate traffic. Such kind of reservations can exist between ISPs or within a single ISP (between ERs). A dynamic reservation is one that is set up inter- or intra- ISPs based on credit. An ISP or ER pays another service provider a certain amount of credit to reserve resources for a certain amount of time. The credit for a particular reservation will decide the duration of a reservation, and one can enhance the reservation by increasing the credit of the existing reservation, instead of requesting a separate reservation.

B.1 Static Reservations

We assume that the ER has a monitoring system that collects statistics of the incoming and outgoing priority traffic for different time of the day, and different day of a week. The average traffic volume is computed for over a specific time window(e.g. an hour), and used as a predictor for bandwidth usage along a specific incoming and outgoing link of an ER. This information is stored in the ICH, and updated when necessary.

ICH uses the stored information about average traffic volume to set up static reservation between different ERs. Similarly, ECH collects statistics about the traffic flow between different ISPs, and negotiate static reservations between them.

B.2 Dynamic Reservations

One option to cope with the changing user requirements is to signal each change in flow activities at ER through the ICH to the core networks. However, this requires ICH and core networks to keep per-flow state information, and would lead to the same scaling problem that Integrated Services architecture faces. In the CH design, the reservation requests between ER and ISPs reflect aggregated changes, and only propagate to the nearest ICH or CH. The regular updates of reservation status are decoupled from the actual reservation requests.

Each ER uses statistical techniques to predict the amount of flow that will use a intra-ISP link based on average volume (offline measurement) and dynamic measurement of traffic arrivals. ER sends regular updates to ICH. When this bandwidth predictor exceeds the existing reservation, ICH will issue a reservation request along that particular intra-ISP link. The destination ER will process the reservation request and increase the amount of resources reserved along the link. However this changes in resource reservation is not propagated back to ICH. This information will be passed along to ICH during the regular update of reservation status from this ER.

Similarly, each ECH issues resource reservation requests when the inter-ISP bandwidth predictor exceeds the existing reservation. The reservation requests propagate along multiple ISPs until it reaches the ISP that serves the destination host network. Again, the change in reservation status between different ISPs is not propagated back to the CH immediately or to previous ISP. The CH will be informed of the changes in reservation requests only through regular updates from ICHs.

C. Discussions

The resource reservation described above meet the following design goals:

- **Scalability:** Resource reservation requests are made for aggregate connections instead of individual call requests. Neither the CH nor the ISPs need to maintain per connection state. This allows the CH to scale nicely.
- **Heterogeneity:** The CH design provides resource reservation and billing mechanisms that work across heterogeneous access networks and services. Since reservation is done using time-based credits, service han-doffs can be performed without having to tear down resource reservations on every link.
- **Minimize Signaling Overhead and Setup Time:** The flexibility in allowing existing resource reservations to be enhanced greatly reduces the need to set up new reservations for every single link whenever a new call is made. Since reservations may already exist on some links in the sender-to-receiver path, both resource reservation and call setup time can be reduced.
- **Inter-operability in the Wide Area:** The CH's design can easily be extended to interoperate with future Internet architecture that provide different service levels in various access networks (e.g., Differentiated Service or Integrated Service with RSVP).

IV. Security, Authentication and Privacy Issues in Clearing House

In this section, we discuss the various security and privacy issues related to the use of the Clearing House. We first list the goals of the CH with respect to security and privacy. These are as follows:

1. *Secure, Scalable, and Correct billing.* The Clearing House must support secure, scalable, and correct billing. By this, we mean that the CH should support mechanisms that will facilitate the user being charged for the exact (or reasonably approximate) length of time the call lasts. It must also enable recording this billing information securely–does the bill correspond to the charges invoked only by the participating ISPs?, does the call correspond to the resource reservation made beforehand by the call agent for this call (i.e., is it an authorized call?), for the purpose of billing the user, or if the user avails himself of special offers made available by the participating ISPs, has the CH properly authenticated the user, as well as itself to the user?, etc., without the CH itself getting involved during the entire length of the call directly. The last requirement, if not observed, does not lead to a scalable solution. Thus, the CH is charged with providing **scalable**, **correct**, **authenticated**, and **authorized** billing.
2. *Trust management.* We observe that the Clearing House is also a participant in monitoring the network, parts of which may not be directly be under the domain of the maintainers of the CH house. This raises an important issue: how can one specify policies for controlled exposure of information so that the network can be monitored, if one requires information about the internal state of the network beyond that available from end-to-end measurements (for example, questions such as what is the packet drop probability at a particular router cannot be easily answered without help from the underlying network). The management of the security policies applicable to a domain can be effectively carried out using a trust management system. The CH can incorporate such a system implementing the rules governing the use of this information about a domain, as well as its violations.

3. *Privacy management.* Note that it is relatively easy to provide privacy in the current Internet (at least theoretically) than in a network such as ICEBERG, simply because there is billing involved in the use of the latter, perhaps on a per-call basis, which makes matters complicated. Ideally, we would like to provide a service where the legitimate user must not need to reveal his identity to *every* provider in the call path simply for the sake of billing: only the user's home provider would need to record the details of a call, such as the date and place of origin. Since the CH is the intermediary in call setup, maintenance and billing, it has a significant role to play in the overall privacy maintenance.

Having discussed the various tasks of the Clearing House, we discuss below each of the tasks in turn in greater detail, including various implementation choices that can be made, and their advantages and disadvantages.

1. *Secure and Scalable Billing.* The Clearing House must be scalable to millions of calls a day. This requires that it should not involve itself in handling the data path during the duration of the call, except, perhaps, by way of monitoring the call for gathering information about the state of the network. The CH is only involved in setting up and maintaining the control path of a call. To provide scalability of this nature, we introduce a level of indirection in managing the call; the CH generates all the appropriate authorization and billing tickets, which, in turn, are given as needed to the ISPs in the call path. Thus, during call setup, the CH performs authentication of the user (both for the purpose of billing, and for the purpose of qualifying the user for possible benefits, for example, a certain ISP may want to provide benefits only to a special class of customers), and gives him two kinds of tickets to be presented as needed to the network: *authorization* and *billing* tickets. These tickets have a certain granularity, say, of one minute of valid usage time. After the user obtains these tickets in some number (initially, the user can indicate the approximate duration of the call, or a safe default can be chosen), he presents these tickets, as needed, to each of the ISPs in the call path, in proper amounts, dynamically: at the end of every minute, say, we can transmit the tickets on the control path. The authorization tickets enable the ISPs to ascertain that the call if valid, and the resources reserved for this call must, therefore, be provided for the call.

   The CH authenticates itself to the user through X.509-like certificates carrying the identitity of the clearing house and signed by a certificate authority trusted mutually by the CH and the user.

   The billing tickets are necessary so that off-line settlements between the Clearing House and the ISPs can be done, and the user can get a single bill at the end of the month depending on the number of tickets used. In this last respect, the CH acts much like a credit card company, albeit being more secure. The method of indirection proposed for receiving service is similar to that used in Kerberos, which is a scalable system for authentication, and which has ticket granting servers resembling the CH. We need the billing tickets because, unlike a traditional phone company that is huge and widely trusted (the main reason being that the barrier of entry is so high, so only well-established companies can offer services in the first place), we envisage ICEBERG services being provided by a large number of independent ISPs and iSPs (ICEBERG Service Providers), who may not have per-pair peer-to-peer service-level contracts or agreements, or who may not trust one another. By providing the CH as a third-party service, or as a service maintained by a large organization into which details about an ISP's bearer capabilities are pushed by the respective ISPs themselves (and, the validity of whose signed tickets, ISPs trust), we remove the need for complicated peer-to-peer service contracts. Thus, the billing tickets allow the ISPs to redeem their service cost by unambiguously and securely providing information about a call. The CH can settle the charges of each ISP separately off-line, say, at the end of the month, when the ISP presents valid billing tickets to the CH. The billing tickets also serve the additional purpose of providing fine-grained control over the duration of a call: if the user finds he has some billing tickets left at the end of the call, he can simply return these to the CH at the end of the call (since the CH bills a user according to the tickets issued to him, the user loses if he does not return these to the CH). In this process, the CH is involved again only at the end of the call. On the other hand, the user may find that the call lasts longer than he has billing tickets for, and the call agent, on behalf of the user, can transparently order more authentication and billing tickets (the CH grants both of these after making appropriate reservations). The idea is that we don't want to penalize a user for not making an accurate demand at the start of the call; the user is unaffected by overestimating or underestimating the call, although he does not have an incentive in doing this either.

   A note about the scalability of generating authorization and billing tickets, and performing off-line settlements settlements with them. Tickets of both kinds can be generated in a fine granularity off-line (even, perhaps, online), fast enough to make the system handle the call-setup reasonably well. Hardware signers and encryptors are available that can do a ten thousand signature verifications per second, and the operations themselves are highly parallelizable. A more serious problem could be in terms of the long-term storage required: AT&T, for example, handles a few million minutes of long-distance calls a day. If the granularity of a ticket is set at one minute, this would mean that when a Clearing House does off-line settlements with AT&T, it must be prepared to handle a few gigabytes of billing tickets every month. The most promising solution seems to be to use the technique of **aggregation** to cut down the amount of data to be deal with. For example, it may be possible for an ISP to aggregate the authenticated and encrypted billing ticket after decrypting them and modifying certain fields. The issue of the general use of aggregation in cryptography is a research topic in itself, but with some simplifying assumptions, is fairly easily deployable. The effectiveness of aggregation can be affected by several factors, including success of the user or network in estimating the duration of a call, degree of trust mutual trust reposed by the CH and the ISP in each other, etc.

   One factor that affects the performance of the system, and may affect the scalability of the CH, is the number of times the CH would need to be involved in generating the tickets *during* the time a call is in progress, and the time of an average call for a person can probably be estimated or deduced from experimental studies or previous usage, thereby improving performance.

Another attractive implementation of authorization and billing that can perform much better is through use of shared secrets and symmetric key techniques. These techniques are specially attractive for authorization, where the goal is to just ensure that the call is matched with the resources reserved: the CH can setup the session secrets with individual ISPs during call setup, which are then used in generating and processing authorization tickets. However, if extra features such as non-repudiation are desired (as, maybe, in billing), one may have to fall back on asymmetric key techniques. Just as for aggregation, mutual trust plays a performance-affecting role here.

The authentication between the user and the CH itself, for the purposes of billing, etc., can be carried out using a secure protocol like SSL.

2. *Trust management.* As mentioned above, it is advantageous for the Clearing House to also implement trust management functions. The goal of trust management is to take a unified approach to specifying and interpreting security policies, credentials, and relationships; it allows direct authorization of security-critical actions (in our case, this corresponds to actions related to the measurement of a network). A trust-management system provides standard, general-purpose mechanisms for specifying application security policies and credentials. Trust-management credentials describe a specific delegation of trust and subsume the role of public key certificates; unlike traditional certificates, which bind keys to names, credentials can bind keys directly to the authorization to perform specific tasks, such as who has what extent of permissions to monitor a network. Thus, an external monitoring unit may be permitted by the CH to gather information that it wants, provided it presents keys consistent with its capabilities, which are verified in realtime by the trust management component of the CH.
3. *Privacy management.* As of now, we can envision the Clearing House providing privacy of the user with respect to the participating ISPs by providing anonymization in the form of indirect billing records.