RIPE ncc

Routing Information Service
R.I.S.
Design Note

Antony Antony[*] *Henk Uijterwaal*[†]

RIPE NCC

October 14, 1999

Document: RIPE-200.ps

Abstract

This document discusses the Routing Information Service (RIS), a project proposed as a new activity of the RIPE NCC in the RIPE NCC activity plan for 1999 and 2000 [1, 2]. The document gives an overview of the project, discusses the implementation and mentions a couple of points that have to be addressed in future design documents. This document is intended to solicit input from interested parties.

[*]Email: antony@ripe.net
[†]Email: henk@ripe.net

Contents

1 Introduction

1.1 Outline of this document

This document discusses the Routing Information Service (RIS), a project proposed as a new activity of the RIPE NCC in the RIPE NCC activity plan for 1999 and 2000 [1, 2].

The outline of this document is as follows: section 1.2 gives the necessary background for this project. The next section 1.3 briefly discusses which tools are already available and show why the RIS is needed. The next section (1.4) gives an overview of the goals for the RIS. Section 1.5 shows the basic setup up the RIS.

In order to prove the concept and to get a feeling for the data-volumes and other problems involved in setting up this service, a prototype was developed. This prototype is discussed in section 2.

The design of the final system is discussed in section 3. The prototype also showed that there are a number of open questions that have to be answered before the design is finalized. These are discussed in section 4.

The project is closely related to the IRR/Reality Checking project [3] and section 5 discusses the interface between the two. Collaboration with other, related projects is discussed in section 6. Finally, the implementation schedule is discussed in section 7.

1.2 Background

In general, the Internet consists of large number of interconnected regional national and international backbones. The backbones often peer or exchange traffic and routing information with one another at Internet exchange points. These exchange points can be considered the “core” of the Internet. Backbone providers participating in the core must maintain a complete reachability information or default free routing table of all globally visible network-layer address reachable throughout the Internet.

From a routing point of view, however, the Internet can be considered to be partitioned into a number of independent sub-networks, the so-called Autonomous Systems (AS’s). The Autonomous System’s are connected to one or more Autonomous Systems. From this viewpoint, it follows that routing in the Internet can be divided into two domains: internal, or inside an AS, and external, or between AS’s. At the boundary of each AS, border routers exchange reachability information to destination IP address blocks or prefix, for both transit networks and networks originating in in that domain.

This project only deals with external routing, tools to study internal routing have been developed by a number of router vendors as well as others. This project also focuses on so-called default-free routers. A default-free router is a router that actively decides where to send packets with a destination outside the AS to which the router belongs, and not forward it, by default, to another router.

How IP packets are routed from one AS to another, is is based on propagating information about network entities (IP address prefixes) and routing policies between them through the network. The commonly used protocol for exchanging this information is the Border Gateway Protocol, version 4 (BGP4) [4]. BGP is an incremental protocol that sends update information only upon changes in network topology or routing policy. As a path vector routing protocol, BGP limits the distribution of a router's reachability information to its peers or neighbor routers.

A path is a sequence of intermediate autonomous systems between source and destination routers that form a directed route for packets to travel. Router configuration files allow the stipulation of routing policies that may specify the filtering of specific routers, or modification of path attributes sent to neighbor routers. Routers may be configured to make policy decisions based on both the announcement of routes from peers and accompanying attributes and local policies. The attributes in the announcements may serve as hints to routers to choose from alternate paths to a given destination.

Backbone border routers at core locations may have 50 or more external peers as well as large number of intra-domain peering sessions with internal backbone routers. After each router makes a new local decision on the best route to a destination, it will sent that route, or path information along with accompanying distance metrics and path attributes to each of its peers. As this reachability information travels through the network, each router along the path appends its unique AS number to a list in the BGP message. This list is the route's AS-PATH. An AS-PATH in connection with a prefix provides a specific handle for a one-way transit route through the network.

Routing information in BGP appears as a series of announcements that either update or withdraw a route within the network. A route update indicates a router either has learned of a new network attachment or has made a policy decision to prefer another route to a certain network destination. Route withdrawals are sent when a router makes a new decision that a network is no longer reachable. A BGP update may contain multiple route announcements and withdrawals. In stable routing environment routing updates should be very less except when policies changes and Addison of new physical networks. In reality the number of BGP updates exchanged per day in the Internet core is one or more orders of magnitude larger than expected [6].

From routing updates routers maintain a so-called routing table containing information about topology and policies (see [5] for more details). From the information in the routing table, a forwarding table is extracted every time a route is modified. The forwarding table tells the router to which interface a packet with a certain IP-prefix has to be sent.

Both the routing and forwarding tables are large, 64 k prefixes or more and can change very rapidly. Routers therefore only maintain the current tables, as they do not need more information for their operations. This set-up has several operational consequences.

First of all, it is easy to determine for a network operator of certain AS, if and how another AS can be reached. All he has to do, is look at the current routing table for the path from his AS to the target AS. This would be sufficient for one-way communication, but in practice, most communication is two-way and there is no way for the network operator to tell if the target AS has a route back to his AS.

Related to this is that reachability metrics are limited to what an observer at a certain AS can see. For example, it will be easy to define a metric to describe which part of the Internet can be reached from this AS but the equally interesting opposite metric, which part of the Internet can reach our AS, is impossible to extract.

Finally, the fact that only the current table is maintained, means that any history about the route will be lost forever. However, historic information is essential when trying to detect instabilities or debug problems that only show up every once in a while. For example, if a user complains that a certain site cannot be reached at certain times, looking at the current table is not going to be of any use, as the site is either reachable or it is not. The routing table does not give any clue when the route was last announced or withdrawn.

1.3 Existing tools

Over the years, several tools to access routing information have been developed, such as: BGP-routing table dump tools and the well-known *traceroute* and *ping* programs. Traceroute combined with path information shows how site can be reached, ping shows if end-to-end two-way communication to a site is possible. All these programs have in common that they only deal with the present situation and one way from observers network. In order get the complete picture of end-end-end communication we often have to look for reachability from remote location to your own network.

One possibility is to use a so-called Looking Glass. Looking Glasses provide access to the routers of the remote AS, allowing an observer to see how traffic from the remote AS is routed back to his network (similar to a traceroute on the remote AS) and thus determine the reachability of the observer's network from the remote AS (prefix and AS-PATH).

However, not all sites have installed this facility and the Looking Glass only provides information about the AS where it has been installed. In order to find which sites can reach a certain AS, one would have to loop over the Looking Glasses at all AS's. This is not practical. Finally, the Looking Glass, again, provides no information about the development of the routing over time.

Other tools include ASExplorer [9] and RouteTracker. Both are developed by Merit and are being used to collect routing announcements from 5 core locations in the United States. However, these are tools primarily intended for research purposes and not for network operators.

In short, it is time for a tool that combines information from several default-free core locations in RIPE region. This is the Routing Information Service.

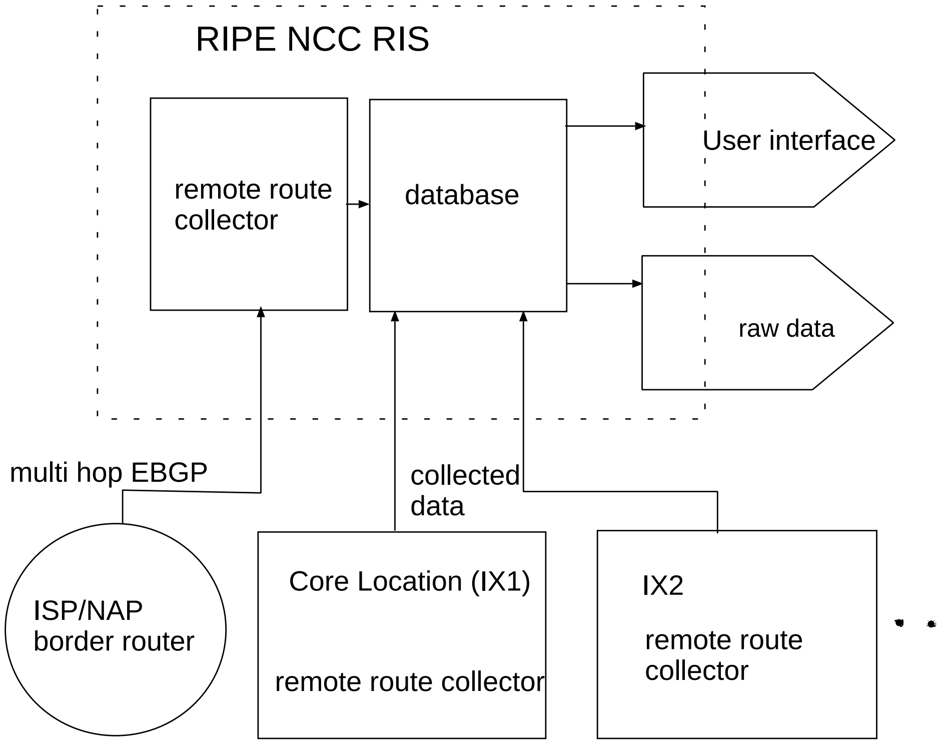

Figure 1: Basic setup of the Routing Information Service.

1.4 Goals of the Routing Information Service

The goal of the Routing Information Service (RIS) is to collect routing information between Autonomous Systems (AS) and their development over time from default free core the Internet. The RIS will collect and store default free BGP announcements as a function of time from several locations and provide that information to the users of the service, allowing them to see the full picture with all routes that are currently anywhere and their development over time. In other words, it can be regarded as one integrated Looking-Glass for the entire Internet that includes history information.

One important application for this data will be debugging. For example, if a user complains that a certain site could not be reached earlier, the RIS will provide the necessary information to discover what caused this problem. This is particularly useful for problems that occur periodically.

There are numerous other applications for the data, for example;

1. The routing table and its development over time can be used to check for local and global convergence of the table and routing flaps.

2. The routing table reflects the policies announced by the sites operating the routers. At a local level, the data can be used to verify the setup of the routers and correct any errors.

3. On global level, the RIS data can be used to compare policies registered

in the Routing Registry against the policies that are actually announced. The RIS will therefore provide essential input for the Reality Checking project for the IRR [3].

4. Related to this is that the RIS will provide information about fake routes inserted into the network, for example by spammers.

5. The routing table also reflects the addressable IP-networks that can be reached, which AS-numbers are actually being used and such. These are valuable statistics.

The project is aimed at NOC's and engineers at ISP's, though it is a priori not excluded that end users will have access to the data as well.

1.5 Setup of the RIS

The basic setup of the RIS is shown in figure 1. At the RIPE NCC a main collection machine will be installed. This machine runs 3 programs: a route collector, a database and user interface. At a limited number of core locations (IX1, IX2, ... in figure 1) a dedicated machine will be installed. This machine will run a copy of the route collector program. The route collector programs collects routes, either directly from the border routers at that site or via Multi-hop BGP peering from other nearby routers. All data is transferred to the RIPE NCC and stored in a database. The database can be accessed by the sites participating in the project and will provide them with both the raw data as well as derived statistics.

The NCC already has experience with running remote data-collection points for the test-traffic project [7]. It is planned to use this experience for the RIS.

2 Prototype Setup

In order to prove the concept and to get a feeling for the data volumes involved, a prototype of the RIS has been developed and has been used for data collection during the last months.

The setup of the prototype is shown in figure 2 and consists of a route-collector, a database and a rudimentary user-interface.

We tested two products as route-collector software: GNU-Zebra [8] (versions 0.74 and 0.76, under the Linux operating system) and the Multi Threaded Routing toolkit (MRT) [9] (versions 1.4.xa and 2.0.0a under the Linux and FreeBSD operating systems).

GNU-Zebra is a free software package that manages TCP/IP based routing protocols. It supports the BGP-4 protocol [4] as well as RIPv1, RIPv2 and OSPFv2. GNU-Zebra can be used with both IPv4 and IPv6. Zebra is intended be used as a Route Server and Route Reflector.

The Multi-threaded Routing Toolkit (MRT) is a project developed by Merit at the University of Michigan, to be used for routing architecture and protocol research. Software developed until now includes multi-protocol IPv4/IPv6 routing daemons as well as routing analysis and simulation tools.

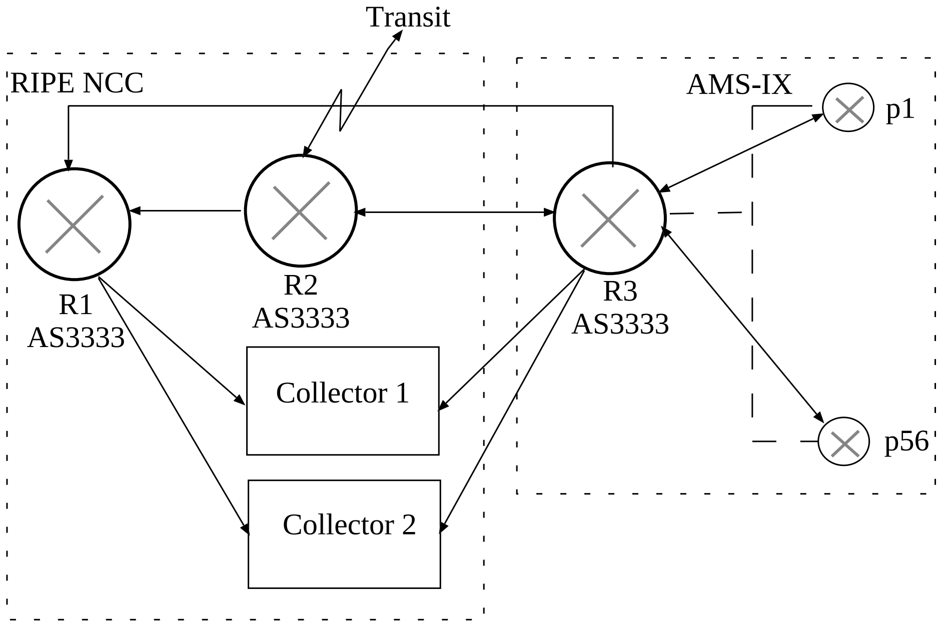

Figure 2: Setup of the prototype RIS.

At peak times, the route collector may have to deal with 70,000 prefix updates per second. The prototype shows that the route collectors can keep up with these these loads, however, the database is a point of concern. These studies show that generic database systems have problems coping with these numbers. The best solution at the moment seems to be store the data in regular files and only use a generic database product to maintain a summary file indicating which information is stored in which file. However, further studies are needed here.

The Multi-hop BGP4 peering mode has been tested with the prototype. This mode can be used to reduce the number of collection points.

The RIPE NCC is happy to make the software of the prototype available to interested parties [10]. However, the software will be made available on an "as-is" basis and the RIPE NCC, unfortunately, does not have the resources to provide more than basic installation support.

The conclusion that can be drawn from this prototype is that it is possible to collect routing information from routers using BGP. It also gave us a good idea of the data volumes involved. This experience will be used in the final design. Data collection with dedicated machines at remote points has not yet been tested but no special problems in this respect are foreseen.

3 Final Setup

Figure 3 shows the proposed setup for the RIS in more detail.

Data-collection machines (the "Remote Route Collector" (RRC) in figure 3) will be installed at a few major topologically interesting points on the Internet,

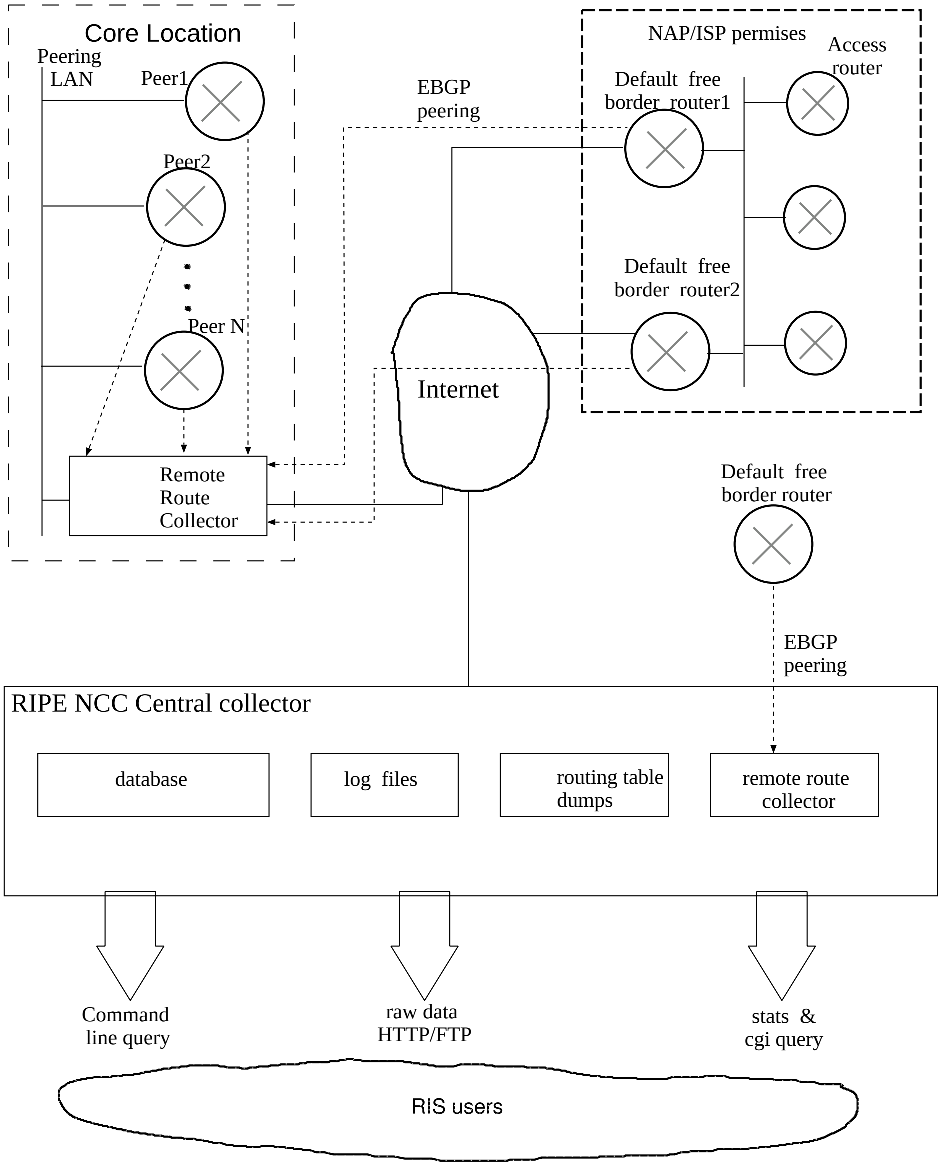

Figure 3: Overview of the final RIS setup.

such as major exchange points and the like. These Remote Route Collectors will collect routing information either directly over the exchange LAN of the site or using the BGP-multi-hop mode to nearby ISP's. All peerings will be set up by hand and only data from ISP's interested in this project will be collected.

The optimal number of Remote Route Collectors and their location still has to be investigated. In first order, it is expected that around 10 such RRC's will be installed in the RIPE service area.

In principle, all peerings via BGP-multi-hop could be done by the central machine at the RIPE NCC. However, that would involve the transfer of a large amount of data over the networks involved. One central collector will not scale

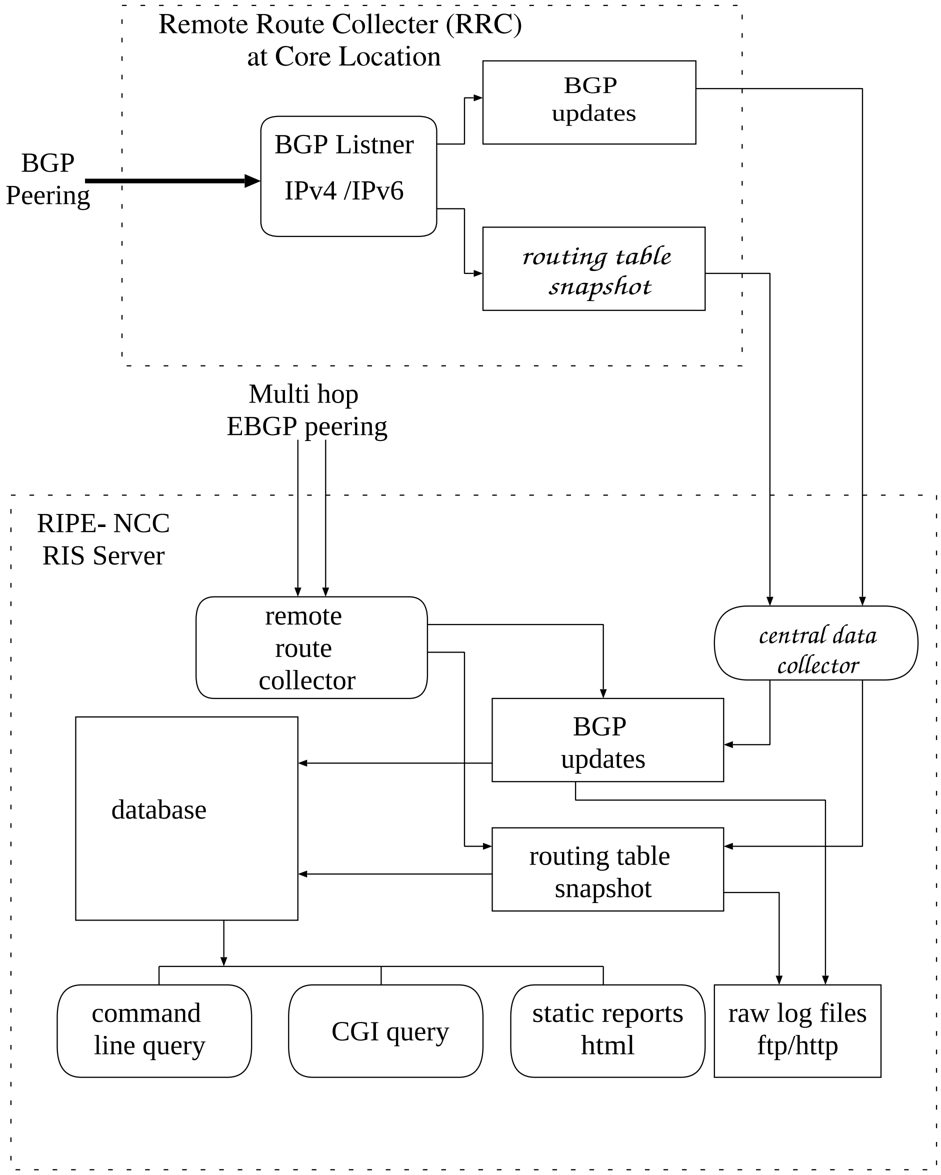

Figure 4: Detailed overview of the final set up.

to several hundred peers. It is therefore more efficient to do some of the BGP-multi-hop peerings with the RRC's, reduce and compress the data there and only transfer the results to the RIPE NCC.

The RRC's are discussed in more detail in section 3.2.1.

Data is then transferred to a central machine at the NCC and stored into a database. Design considerations for this database are discussed in section 3.1.2.

Users of the RIS can access the database through various interfaces discussed in section 3.1.3. It is a goal of the project to transfer the data as often as practically possible without overloading the networks or databases, in order to minimize the time between data-taking and availability of the data to the users.

3.1 Software for the RIS

An overview of the various software modules used in the final setup is shown in figure 4.

3.1.1 Software for the Remote Route Collectors

Remote Route collectors will run BGP listener which logs all updates received from its peers and also maintains a routing table. In the first instance, it is planned to run the MRT [9] and GNU-Zebra [8] software under a supported UNIX platform on each RRC. These products were described in section 2.

Depending on the development paths of MRT and Zebra, the RIPE NCC might decide to re-write parts of those software packages in order to add missing features or to optimize performance, or to use parts of other software packages. The parts of any external software essential for the RIS will be reviewed and documented according to appropriate RIPE NCC guidelines in order to remain maintainable in the future.

The BGP listener software will collect time stamped BGP4 announcements, including:

- The route prefix and length.
- The origin AS.
- The AS paths for which the announcements are visible.
- Any additional BGP attributes which are propagated through the system.
- Errors in the BGP announcements.
- Periodic dump of routing table.

As both Zebra and MRT supports IPv4 and IPv6, it will be easy to extend the RIS to include IPv6 routing information when native IPv6 networks start to be deployed on a large scale.

Receiving speed: Figure 5 shows the number of updates received by an RRC as a function of time. When the RRC is started, it receives some 65,000 initial prefix announcements in 90 seconds from a single peer. After the initial setup, the rate, averaged over a 1 week period, will drop to 2 updates/second/peer[1].

Assuming 60 to 100 peerings at a single collection point, the typical number seen with the prototype will translates to a peek of 50,000 to 80,000 announcements per second to be record by a RRC. Collector software will be resource intensive in terms of memory, to keep several views of routing table and I/O intensive to record the updates and snapshots.

During major instabilities, the number of updates might even be higher. In order not to interfere with operation of the routers, it will be considered to limit the data-flow from the routers to the RRC's.

[1] The slope of the right-hand side of figure 5 suggests a higher rate but appears to be a statistical fluke.

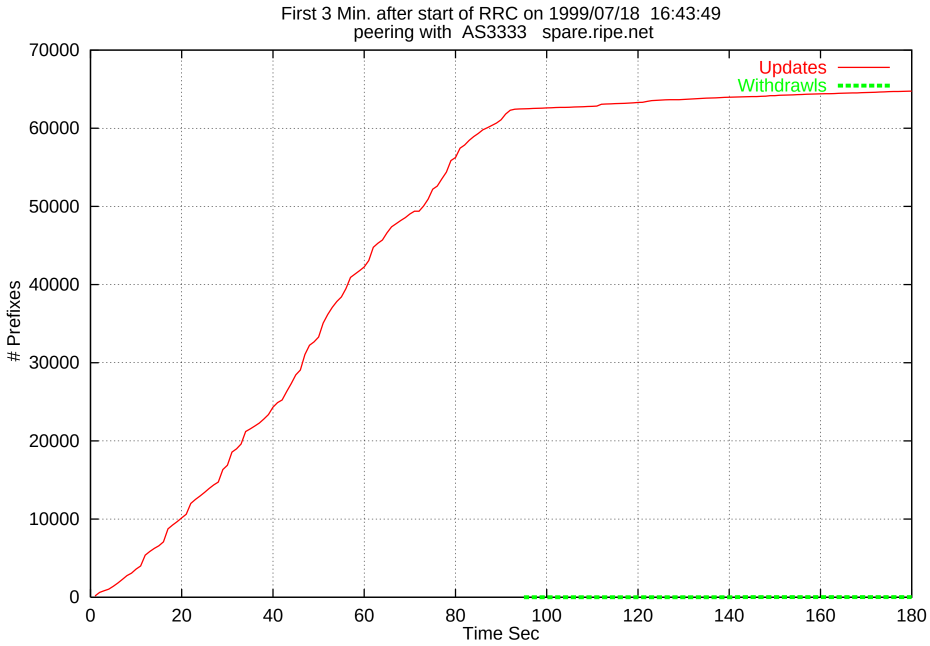

Figure 5: Number of updates seen by the prototype RRC peering with a single router during the first 3 minutes of operation.

3.1.2 Database software

The main requirements for the database are:

Insertion speed: Assuming all RRC's are restarted once on a same day total number of updates collected from 10 RRC's during one day will be about 110 M prefix updates. To summerize and insert this data into database in about 3 hours, required processing speed will be 10,000 prefixes/sec. The database will have to buffer incoming data and the situation where not all data has been inserted in the database will be flagged. The acceptable time elapsed between data-taking and availability in the data-base still has to be investigated.

Query speed: This should be such that an individual user will receive an answer to a simple query (e.g. what was the route from A to B at a given time) in $\mathcal{O}(10)$ s. If this takes much longer, extracting data will be too cumbersome and the RIS will not be used.

Merging data: The software should be able to merge data from remote points into database if they have been disconnected from the RIPE NCC while still collecting data without "falling behind" with the regular updates. The amount of data to be merged at once here may be of the order of the data collected during a week. (For example, an RRC that got disconnected from the Internet at the start of a long weekend, followed by a few days necessary to solve the problem.)

	RRC (GByte)			*NCC Central (GByte)*	
	Peer/Day	*Daily*	*Monthly*	*Monthly*	*Yearly*
BGP announcements					
• Uncompressed	0.007	0.490	14.3	143.0	-.-
• Compressed	0.005	0.350	-.-	-.-	1002.0
Routing table dumps					
• Uncompressed	0.021	1.430	43.0	430.0	-.-
• Compressed	0.002	0.143	-.-	-.-	430.0
Storage needed					
• Hard disk	0.028	0.633	57.3	573.0	-.-
• Tape	-.-	-.-	-.-	-.-	1700.0
Data to be transferred					
• Compressed	-.-	0.500	-.-	-.-	-.-

Table 1: Storage requirements for the RIS. The first 3 columns refer to the individual RRC collecting BGP-announcements and routing table dumps, the last 2 columns to the central collection point at the NCC. The top 3 rows show the size of the BGP-announcements, routing table dumps and total storage space needed at that stage. The last row shows the amount of data to be transferred. Data will be stored either compressed or uncompressed, "-.-" indicates that data will not be stored in the (un-)compressed format at that stage. A total of 10 collection points has been assumed. The numbers in this table are based on the prototype and will be further refined as we gain more experience. The numbers also do not include index-files and other overhead introduced by the data-base program.

Data volumes: The prototype shows that the data at a typical collection point consists of:

- 7 Mbyte/day/peer of BGP announcements when all logging options are switched on.

- 7 Mbyte for a full dump of the routing table for each peer. Assuming that the data is dumped every 8 hours (3 times a day), this translates into 21 Mbyte per peer, per day.However, the routing table can be compressed quite efficiently by a factor of ≈ 10 using standard file compression tools.

  The number of dumps each day is a trade-off between data-volume and CPU power. One can reconstruct the routing table at any point in time from an initial dump plus all updates received since the time the table was dumped. However, the longer the time elapsed since the routing table was dumped, the more updates will have been received and thus the more CPU power is needed to reconstruct the table. Dumping the table more often, means that less CPU power is needed but requires more storage space. 3 dumps a day appears to be a reasonable compromise.

The maximum number of collection points is of the order of 40, approximately 30 core locations in the RIPE region and about 10 core locations elsewhere in the world. However, there is a considerable overlap between the data that would be collected at all 40 points, so we assume that data collection at 10 points will be sufficient. Data-collection might also be done by exchanging data with similar projects (see 6), though this does not change the total data-volume.

An estimate of the upper limit of the total data volume based on these results can be found in table 1. The numbers in these table are based on our studies with the prototype and will be further refined as we gain more experience. Assuming that data will be available on disk for 1 month and is then moved to tape, the RIS will require some 600 Gbyte of disk-space as well as a tape storage device capable of handling some 2 Tbyte.

Handling such an amount of data is by no means trivial. The database should be able to handle this amount of data and be able to cope with future expansion.

Experience with the prototype has shown that the popular MySQL database cannot cope with these requirements. Other products are being investigated.

3.1.3 User Interface

Regardless of the method chosen to store the data, it is planned to implement the following ways to access the data (in this order):

1. All (or a selected subset of) the raw data (both the full routing tables and the individual updates) will be made available to the sites participating in the project via FTP or HTTP.

2. Command line queries, a user enters a command and the resulting output will be shown on the screen, similar to the well known RIPE NCC whois server. Typical queries that one can think of are, for example:

   - What was the route between two ASN's at a particular time, as seen at a particular site?
   - When was a route between those two ASN's first announced, which updates were made and when was the last route withdrawn?
   - Which prefixes were announced by a certain ASN
   - Who announced a route to a certain ASN.

   These queries will be refined and expanded based on the input from the users.

3. Queries through a cgi-script. An interface to command-line queries via a web-interface will be provided. This interface will also allow us to provide graphical output, for example, a map of ASN's and their connections.

4. Reports and statistics. The data will be collected and reduced to reports showing statistics of Internet routing. For example:

- Rates of updates and withdrawals for certain routes.
- Number of flaps

This will again be expanded based on user input.

In the first instance, it is foreseen that all queries will be done to the RIPE NCC RIS server. At a later stage, the possibility to do local queries to the individual RRC's might be added. The advantages of this are two-fold, it will reduce the load on the central server and it will allow the users to use the RIS data when a major network problem causes the central server to be unreachable.

3.1.4 Data collection at the Remote points

The RRC's will need careful monitoring and security since they are interacting with critical network elements like border routers, are located at important network points, and problems may trigger alarms at peer NOC's. *For example, if a box goes down NOC monitors may generate alarms equivalent to peer going down even though an RRC is not announcing any routes. This should be avoided.*

For security reasons, the RIS will get a separate AS-number. The RRC's will also not announce any routes though peers may wish to set up filters to block any announcements from the RRC's.

Although it is planned to control the RRC's from a central point with no operators or service required at the local sites, it is expected that each site that hosts a RRC appoints a local contact. This contact should take care of things that cannot be done remotely, such as rebooting the machine or copy information from the console in case of network or hardware problems, keep the RIPE NCC informed about planned maintenance of network and such. The local contact, obviously, has to be reachable by phone or email.

3.2 Hardware

3.2.1 Hardware for the RRC's.

The prototype shows that route collection is a memory and I/O intensive process, CPU power is less important. Approximately 256 Mbyte of memory is needed to run all software without significant swapping.

Although data will be pulled to the RIPE NCC at regular intervals, one should consider the situation where connectivity to the RIPE NCC is lost for some time and that data will have to be buffered at the RRC's. In order to cope with this situation, the RRC's will have to be equipped with at least 20 Gbyte of disk space.

Over the last year's the RIPE NCC has gained experience with the operation of remote machines for data-collection purposes [7]. It is planned to use that experience to run the RRC's, no particular problems are foreseen in this respect.

There is also considerable interest in the installation of Test-boxes at the same location as the RRC's and it has been suggested to use the same hardware for both projects. This idea will be investigated further.

3.2.2 Hardware for the central collection point

The central machine will collect data in the same way as the RRC's. Besides that, it will run the database where all collected data is stored as well as handle external queries. The machine will be equipped with sufficient disk-space to keep approximately one month of data online. Older data will be stored on tape.

3.3 Peering with the route collector

The RIS project will use AS Number 12654 for peering at all collection points. This ASN 12654 will not announce any routes[2]. Ideally, the RIS should peer with all members of the site for their entire default free routing table. These peerings will be set up using BGP4. The peering policy will be added to autonum objects in the RIPE whois database.

The RIPE NCC central collector will also setup a few multi-hop BGP4 peerings to ISP/NAP's border routers.

3.4 Bandwidth requirements for the RRC's.

Approximately 500 Mbyte (see table 1) will be transferred from each RRC to the RIPE NCC daily. This translates into a constant flow of about 6 kbyte/s. The central collection point at the NCC will see an incoming flow of about 60 kbyte/s. However, the data volume that has to be transfered can reduced only by sending the differences between the periodic dumps, rather than the full dumps. We will ensure that these data-transfers will not affect normal operations.

4 Research issues, Testing

4.1 Open Questions

While writing this proposal, a number of questions came up that need further investigation. These are listed in this section.

1. Empirically investigate the added value in collecting information at different places. Determine how one can best select the sites were data is collected. Determine how to merge identical data collected at different sites.

2. As can be seen from table 1, the RIS will have to handle large amounts of data. What is the best database to use for this project and how can its performance be optimized/tuned for this project?

3. Find out, in collaboration with the first users, how long the data should remain in the database?

[2] Assuming that the router software on the other side does not need any announcements from ASN 12654.

4. Is the full data still useful after a while, or can it be reduced to a smaller sample that is only sufficient to produce statistics?

5. Determine the optimal number of routing table dumps per day based on practical experience with the first RRC's.

6. Investigate if default routes should be collected as well. Default routes are routes that are used by routers if no information about a certain IP-prefix is known.

7. It will be necessary idea to maintain a mirror data-base of the aut-num objects (and their history) in the RIR databases. When analyzing data at a later point in time this information will be essential. The maximum number of ASN's is 64 k so this will not create a large volume of data. However, the data is stored into several whois-servers so a common format and tools to extract information from the servers will have to be defined.

8. Investigate what kind of support is necessary for RRC and Server machines at NCC on a "24/7"-basis, in order to make the RIS available to its users full time.

4.2 α- and β-testing

For a service as the RIS, it is essential that the potential users of the service are involved from the first stage of development. The RIPE NCC has therefore started to look for users of the α- and β-versions.

Inside the NCC, there is interest in using the first versions and work on the definition of the user interface, database queries and other services based on the data.

Outside the NCC, at least 4 sites have shown interest in providing us with data through peering sessions and it is hoped that these sites will also act as the first external users of the project.

In order to get optimal feedback from the first users, documentation and a user's manual will be written as the project is being developed. The documentation will describe the program, the user's manual will show how the data should be interpreted and how it can be used for operational purposes. It is also planned to give regular progress reports in the appropriate RIPE-WG's and solicit input from all interested parties.

When the project is in a more mature state, turning the RIS into a regular service will be discussed in the RIPE-WG's.

5 Interface to the IRR Reality Checking Project

One possible use of the RIS is in the IRR Reality Checking Project [3]. The RIS provides the actual routing and this can be checked against the routing announced in the DB for consistency. At the moment, the IRR Reality Checking Project seems to be most interested in the dumps of the routing tables and not in the individual BGP updates.

The interface between the two projects will be defined at a later stage.

6 Related projects

There are several ongoing projects similar to the RIS. The most important ones are:

- Merit/IPMA [11] has been doing BGP logging since 1996. BGP announcements are collected directly from 5 major exchange points in North America as well as via multi-hop BGP from 20 providers. More on this project can be found at: `ftp://ftp.merit.edu/statistics/ipma/data`
- The University of Oregon Route Views Project, see: `http://www.antc.uoregon.edu/route-views/`.

The RIPE NCC plans to collaborate as well as try to exchange data with those projects. The NLANR-group at SDSC [12, 13] has already shown interest in collaborating with the RIPE NCC on RIS-related research.

Other projects focus on the visualization of the data and this might be an other area for collaboration in the future.

7 Schedule, milestones and resources.

Based on discussions both inside the RIPE NCC and with the chairmen of the appropriate working groups, the following milestones have been set for the project:

July 1, 1999: Work started when AA joined the project.

October 14, 1999: Draft project plan presented to the community.

September 1999/RIPE-34:

- Presentation of the project plan in the Routing-WG.
- Demonstration of the prototype.

October 14, 1999: Final project plan presented to the community.

October 1999-February 2000: Development work, intermediate steps:

- Documentation of the prototype.
- Develop the first version of the RRC software.
- Set up peering sessions with interested parties and start collecting data with (prototype) RRC software. This data will be used to develop the user-interface and to finalize the estimates of the data-volumes.

- Make the raw data available for analysis.
- Build a prototype user interface and get user feedback on that. The prototype will be documented.
- Start to extract statistics from the raw data.
- Finalize and order hardware.

February 2000/RIPE-35:

- Present answers to the research issues mentioned in section 4.
- Have a more advanced system ready to be tested inside and outside the NCC. User documentation will be provided and show how the data can be used in day-to-day operations.

February - May 2000:

- Install several RRC's.

May 2000/RIPE-36:

- Add more features based on user feed-back.
- Study and implement analysis of the data over a longer term.
- Discuss implementing this as a regular service

September 2000/RIPE-37:

- Turn project into a regular service.

At the moment it is expected that 1 network engineer (AA) will work full-time on the project. The second author of this document will spend approximately 20% of his time on this project. It is likely that additional expertise or manpower will be needed during the various stages of the development of the service. These will be borrowed from other groups inside the RIPE NCC.

When the project is turned into a regular service, the manpower situation will be reviewed.

8 Conclusions

In this document, an overview of the RIS project was presented, the implementation has been discussed and a list of open questions has been given. The project was presented at the RIPE-34 meeting and feedback from the RIPE-community has been included in this design document. The RIPE NCC will now start with the implementation of the RIS.

Acknowledgments

The authors would like to thank: Pedro Figueiredo and João Damas (RIPE NCC) for their work on the prototype setup and the chairmen of the RIPE Routing, DB and IPv6 working groups for their input to this proposal.

References

[1] John Crain *et al.*, "RIPE NCC Activities & Expenditure 1999", RIPE-186.

[2] John Crain *et al.*, "RIPE NCC Activities & Expenditure 2000", RIPE-197.

[3] Joachim Schmitz and João Luis Silva Damas, "Reality Checking of the RIPE Routing Registry", RIPE-201.

[4] Y. Rekhter *et al.*, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, see `ftp://ftp.ripe.net/rfc/rfc1771.txt`

[5] T. Bates *et al*, PRIDE Users Guide, 1994.

[6] Craig Labovitz *et al*, "Internet Routing Instability", Proceedings of SIGCOMM'97, September 1997.

[7] Henk Uijterwaal, Olaf Kolkman, "Internet Delay Measurements using Test-Traffic, Design Note", RIPE-158.

[8] Zebra routing software, see `http://www.zebra.org`

[9] Multi Threaded Routing toolkit MRT, see `http://www.mrtd.net/`

[10] See: `http://www.ripe.net/ris`

[11] IPMA tools, see: `http://www.merit.edu/ipma/tools/`

[12] NLANR, see: `http://www.nlanr.net/`

[13] SDSC, see: `http://www.sdsc.edu/`