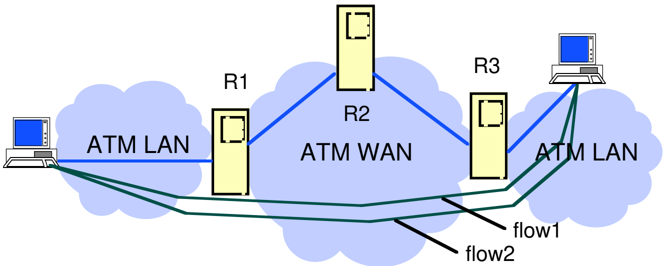
Application REQuested IP over Atm (AREQUIPA)

Overview

- Motivation
- What is Arequipa?
- How does it work?
- Implementation
- The Web and Arequipa
- Conclusions

Motivation

- We want to provide guaranteed QoS for IP over ATM today.
- With Arequipa, applications can request a dedicated SVC for all data exchanged between two sockets.



What is Arequipa

- Example:

```
s = socket(.,DATAGRAM,..);
set_atmaddr(&aa,...);
set_qos(&qos,ATM_CBR,peak,...);
arequipa_preset(s,&aa,&qos);
connect(s,&ip_addr,sizeof(ip_addr));
write(s,buf,BL);
```

```
s = socket(.,DATAGRAM,..);
bind(s,&local_addr,
     sizeof(local_addr));
arequipa_expect(s,1)
read(s,buf,BL);
write(s,buf1,BL);
```

How does it work?

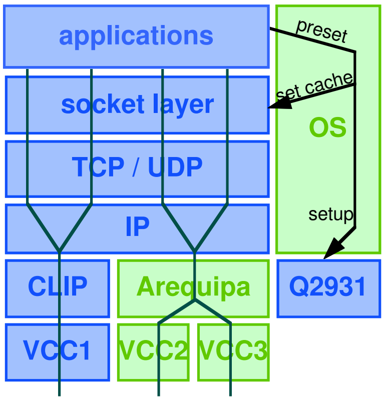
- Arequipa_preset
  - On Arequipa_preset a dedicated connection is established for a given socket.
  - Any data sent from the socket uses that connection
- Arequipa_expect
  - A_expect prepares a socket to use an incoming Arequipa connection. When a socket receives data from an expected incoming Arequipa connection data sent from the socket is sent over the incoming connection.
- To avoid confusion between dedicated and non-dedicated incoming connections a special BHLI is used.

Implementation

- In the protocol stack Arequipa is seen as one device.
- A pointer to this device is put in the ip_route cache of the sockets which use Arequipa. IP thus forwards the data to the Arequipa device.
- A pointer to the dedicated VPC is saved in the socket descriptor. It is used to send the data over the correct VPC.
- Signalling is done by the same entity as for IP over ATM

Implementation (cont'd)

- Arequipa is part of the linux-atm distribution:
  `http://lrcwww.epfl.ch/linux-atm`



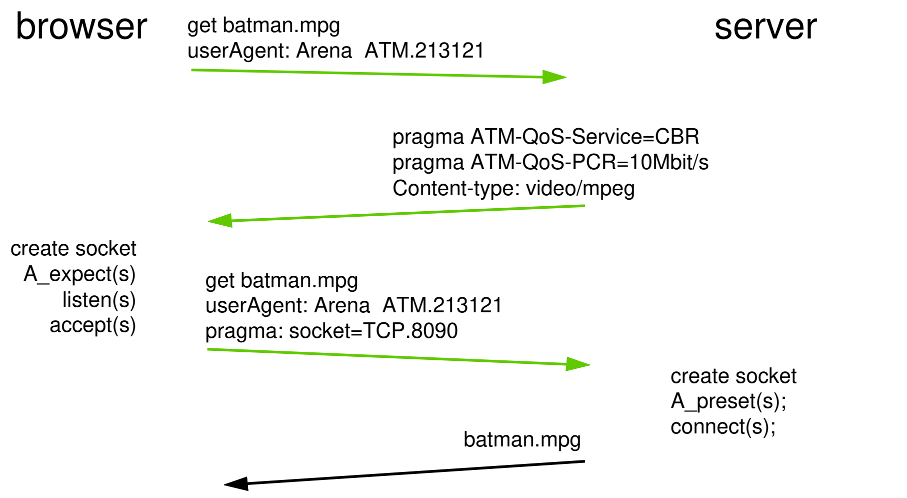
Arequipa and the Web

- Required QoS is specified in meta information of Web documents
- Web browsers put their ATM address in every request
- If a browser wants to receive a document over an Arequipa connection, it also indicates the socket on wich to send the document

Arequipa and the Web (cont'd)

- When a server sees QoS in a document, ATM address and socket in a request, it sends the document using Arequipa
- When a server sees QoS in a document, ATM address in request but *no* socket, it send header of document over normal connection.
- Given the header with the QoS info, a browser can choose whether it wants to use Arequipa or not.
- Arequipa capable servers and browsers are fully compatible with standard servers and browsers.

Example



Issues

- Given an mapping from INT-SERV flowspecs to ATM traffic descriptors, Arequipa can provide integrated services
- Address resolution is delegated to applications. They can exchange their ATM addresses prior to establishment of Arequipa connection (eg. during service negociation)
- SVCs can be used bidirectionally
- A specific BHLI is needed to tell dedicated connections from shared ones

Conclusion

- Arequipa can provide guaranteed QoS for point to point data flows within an ATM cloud.
  (Point to point can be ingress router to destination)
- Arequipa runs on current ATM networks (UNI 3.1) and needs no network support (eg. NHRP, RSVP).
- With Arequipa one can use a best effort network for common traffic and a specific network (ATM, FR, RBONE) for e2e connections with QoS guarantees.
- Arquipa allows to experiment today with INT-SERV over ATM. An experimental RFC would allow other developers to build compatible code.