Extending ACNET Communication Types to Include Multicast Semantics

R. Neswold, C. King
FNAL, Batavia, IL 60510

When multicasts were first used in Fermilab's ACNET control system, they were handled different than normal ACNET communication. Rather than using the familiar ACNET API, programmers had to use specialty libraries. This meant that applications would communicate via ACNET for normal data acquisition needs, but would additionally manage multicast resources (even if hidden behind a library interface) for the few protocols requiring multicast reception.

As more services required multicast communications, more infrastructure was added – some of it resembling the connection management already done by the ACNET process. It became apparent that, with a few tweaks, multicasts could be cleanly incorporated into ACNET and, with it, some very interesting behaviour can be implemented.

Now that we have multicasted requests at our disposal, we found they have very useful applications. They give us further ways to reduce network resources as well as techniques that provide **high−availability** to our network services.

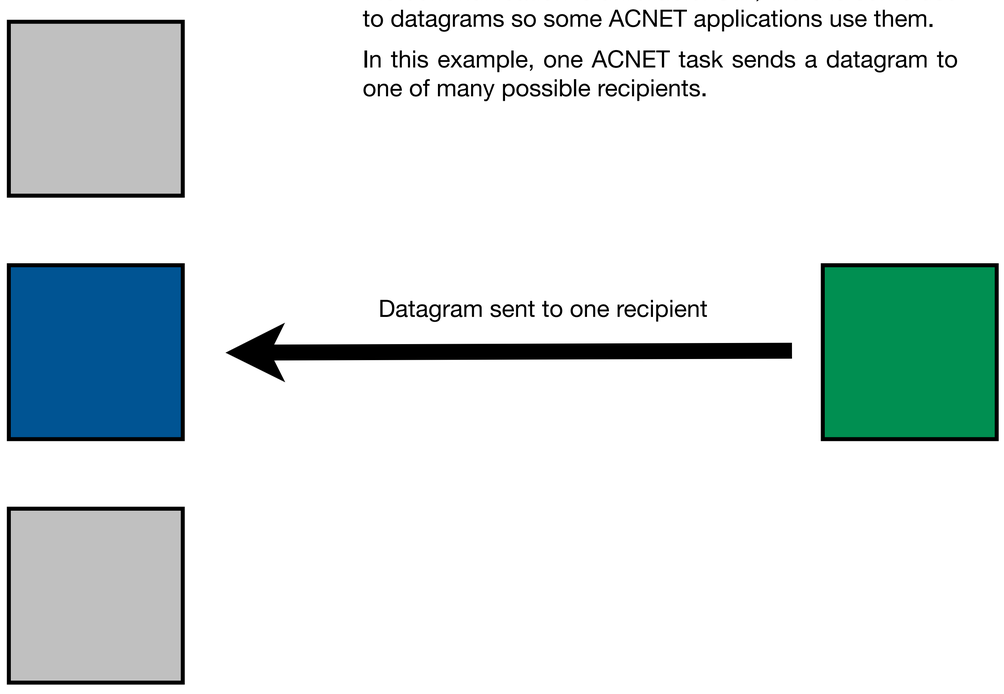
Unicast

Multicast

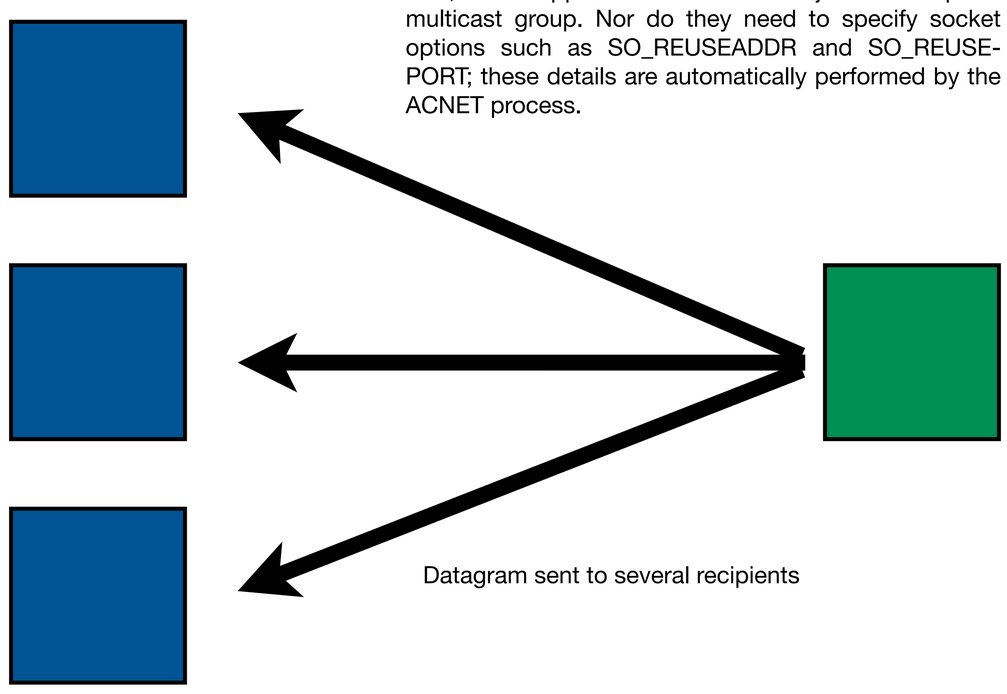
Datagrams

ACNET datagrams are very similar to TCP/IP datagrams; there is no state associated with the packet and the communication is unreliable. Still, there is some use to datagrams so some ACNET applications use them.

In this example, one ACNET task sends a datagram to one of many possible recipients.



ACNET multicast datagrams are also very similar to their TCP/IP counterparts. One major difference, however, is that application don't need to join and drop the multicast group. Nor do they need to specify socket options such as SO_REUSEADDR and SO_REUSEPORT; these details are automatically performed by the ACNET process.

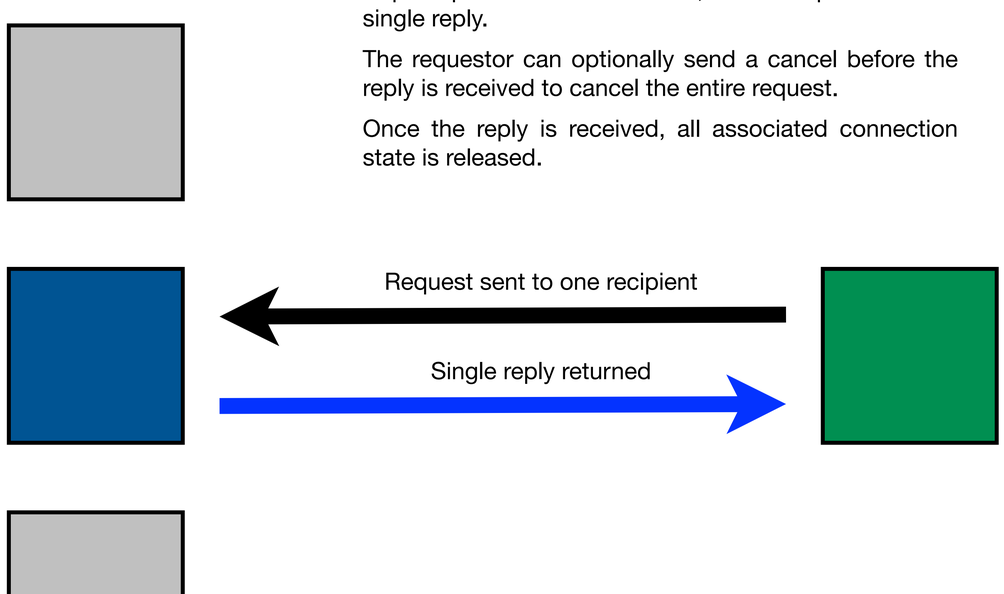


Request w/Single Reply

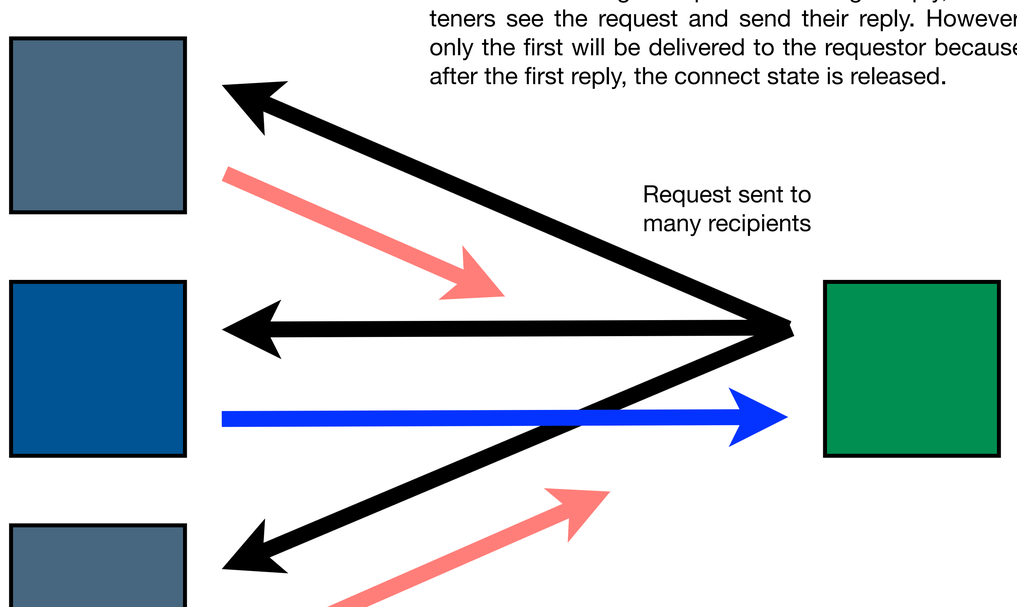
In a request for a single reply, an ACNET task sends the request packet to another task, which responds with a single reply.

The requestor can optionally send a cancel before the reply is received to cancel the entire request.

Once the reply is received, all associated connection state is released.



When multicasting a request for a single reply, all listeners see the request and send their reply. However, only the first will be delivered to the requestor because after the first reply, the connect state is released.



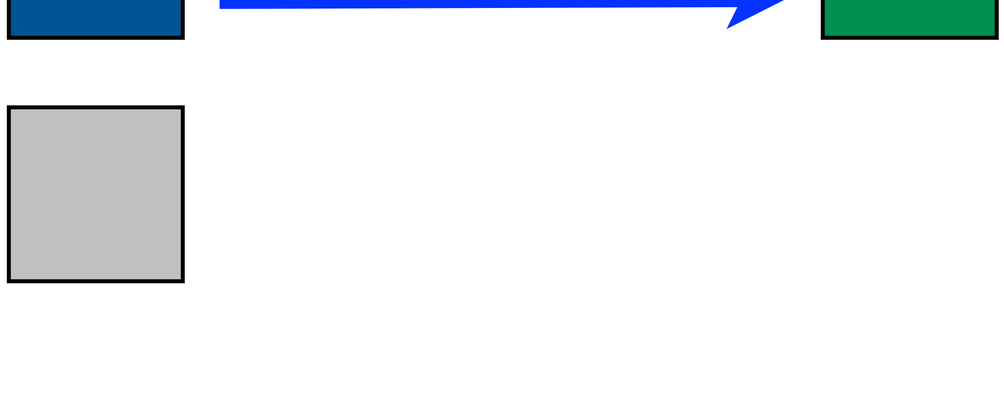
This can be used for **load balancing** a group of servers: a client sends a multicast requesting the next available server. Each server delays their response proportionately to their load. The first to respond is the least loaded.

Request w/Multiple Replies

In a request for a multiple replies, an ACNET task sends the request to another task, which responds with a series of replies.

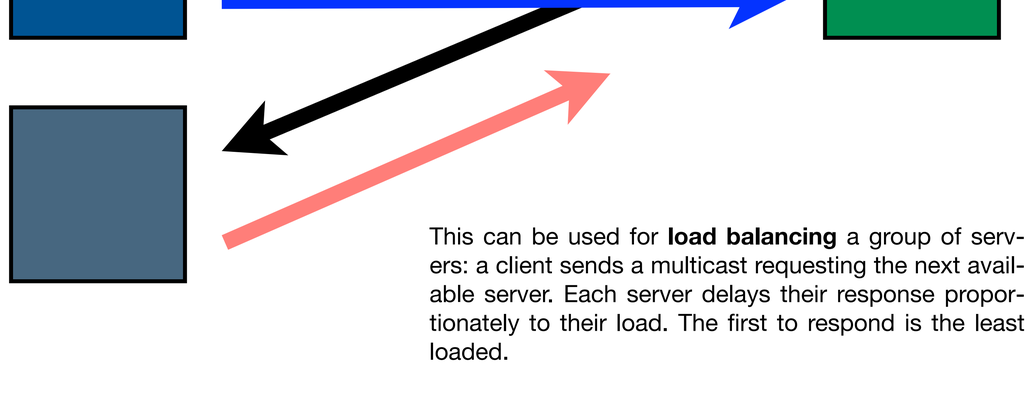
If the request specifies a finite number of replies, the server will indicate in the last packet that it is, in fact, the last packet.

The requestor can cancel the request by sending a cancel message.



When multicasting a request for multiple replies, all listeners see the request and send their replies. "Last Packet" status is ignored to prevent one server from closing the connection.

The only way to close this type of connection is for the requester to send a cancel message (which gets sent to all servers.)



This can be used for **service discovery**: the client sends a multicast query to see if any servers are available. All servers send a reply and the client receives them all.

It can also be used to poll multiple servers with one connection.