Helsinki University of Technology

Laboratory of Telecommunications Technology

STREAM PROTOCOL, version 2+

ST2+

12.11.1996

Ove Strandberg

Nokia Research Center

ove.strandberg@research.nokia.com

Table Of Content

Acronyms

LRM	Local resource manager
RSVP	Resource Reservation Protocol
SID	Stream ID
SCMP	Stream Control Message Protocol
ST	Stream Protocol
ST2+	Stream Protocol version 2+
ST-II	Stream Protocol version 2
QoS	Quality of service

1 Introduction

The Stream Protocol (ST2+) is an experimental protocol for the Internet to handle real-time traffic. The ST2+ reserves resources along the IP network to establish a guaranteed real-time end-to-end connection, see figure 1.1. The applications on top of ST2+ can build multi-destination simplex data streams with selected quality of service (QoS). The stream is an unidirectional point-to-point or point-to-multipoint connection. *[RFC 1819]*

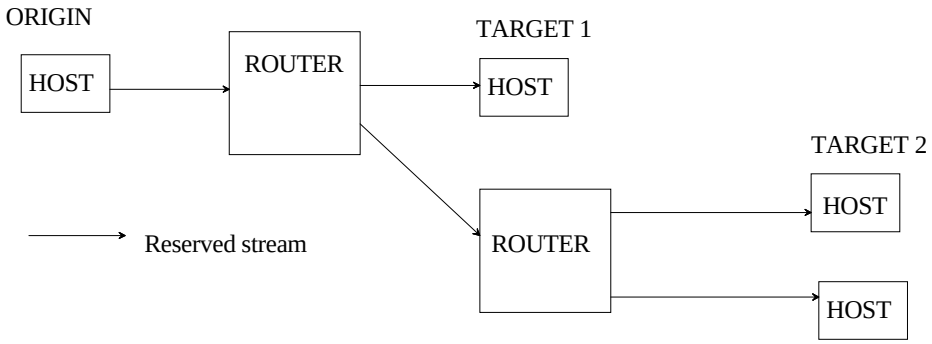


Figure 1.1, Basic ST2 realization.

The latest version of ST2+, RFC 1819, was issued as a RFC as late as August 1995 and has not yet found as widespread support as another reservation protocol, the RSVP, has received. The background of the ST protocol is though long and its developments started in the late 1970's. The prior version, Stream Protocol version 2 (ST-II), was issued as a RFC in October 1990. *[RFC 1190]*. Other versions have been ST version 1 and ST version 1,5.

No version of the ST protocol has ever been issued as an IP standard, but experimental networks has been set up on the IP to refine and validate the protocol.

This document will give a brief presentation of ST2+, starting with the basic principle in chapter 2. In chapter 3 is some technical specifications presented and in 4 will a network example highlight a typical case. Chapter 5 with conclusion will finish this document. This document is mostly based on RFC 1819. Other references are mainly extracted from this RFC to better pinpoint the real source.

2 Basic description of ST2+

The ST2+ protocol describes a way to establish connections over the Internet, but the protocol doesn't introduce a new routing method. The ST2+ uses other already developed routing protocols to realize the path for the connection. The ST2+ is really a pure resource reservation scheme, using a routing function in the IP that is totally independent of the ST2+ protocol.

The ST2+ is in a sense a parallel protocol to the IP as the ST2+ packets are different and that they have to be processed by a separate ST agent in the IP router. The four first bits in the IP packet indicate the IP version used, thus IPv4 has version number 4 and IPv6 has number 6. This version number is also used to distinguish between IP and ST packets i.e. if the packets first four bits shows number 5, it indicates that the packet is a ST packet. In no other way is the ST packet similar to the IP packet. *[RFC 1700]*

The ST2+ protocol can however use the IP to encapsulate the ST packet and make the ST packet traverse network parts that doesn't support ST traffic. The content of the IP packet is indicated with the Protocol field in the IP header and it should be set to 5. Thus the number 5 indicate that the IP packet contains ST traffic in the payload.

2.1 The building blocks

To be able to reserve resources there has to exist an entity in the routers that control the packet processing. In the ST2+ case is a ST agent specified to control ST traffic flows and control actions. The agent needs a local resource manager, LRM, to keep track on what can be done on a specific router. The example network in fig. 1.1 will then be developed into figure 2.1.

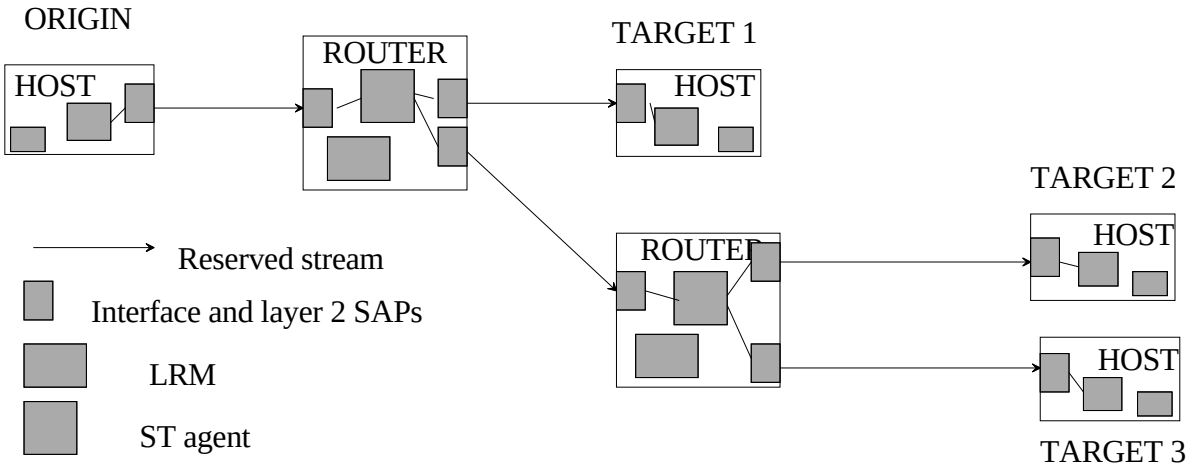


Figure 2.1, Building blocks in the network nodes.

Every host and router also include IP parts that support normal IP best-effort traffic. Note however that the ST model also relies on the IP protocol for the addressing scheme, routing and second layer SAP's. Other building blocks are the setup protocol, SCMP and the data transfer protocol, ST (see next chapter 2.2).

The application layer above the ST will not be TCP or UDP, but new real-time application transport protocols have to be developed. Application options are RTP, PVP and NVP to handle real-time, video and voice transport. *[Schu94], [Cole81], [Cohe81].*

2.2 The ST2+ packet

There are two ST2+ packet types, see also figure 2.2:

- Data packet, ST (Stream Protocol)
- Control packet, SCMP (Stream Control Message Protocol)

These packets have similar tasks as the IP and ICMP packets in the IP protocol suit; data and control. The ST2+ packets are distinguished by the D-bit in the ST header. If the D-bit is set to one, it indicates that the ST packet is a data packet. If the D-bit is set to zero, it indicates that the packet is a SCMP control packet.

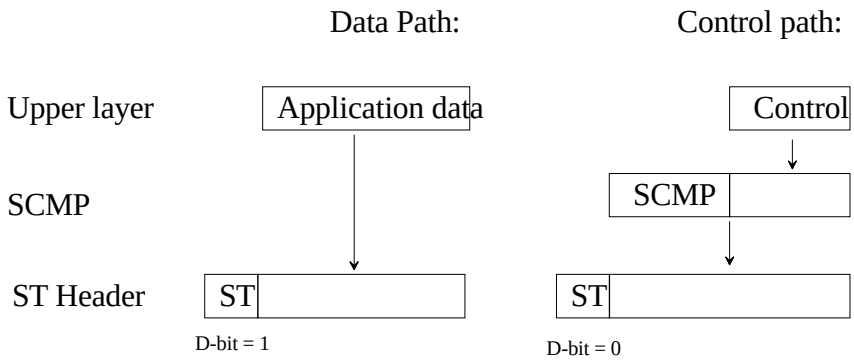


Figure 2.2, Data and control packets.

The header of the ST packet is shown in figure 2.3. The first 4 bits are set to value 5 as described in beginning of chapter 2. The header also contains a ST version number, the ST2+ version number is 3.

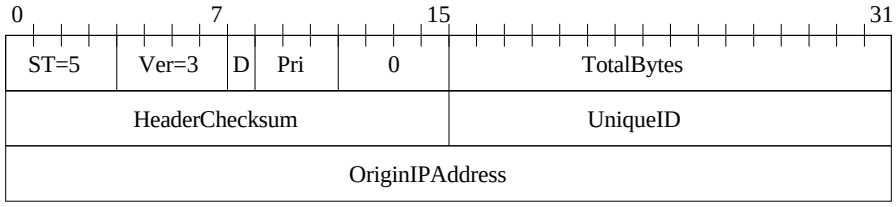


Figure 2.3, ST2+ header.

The priority field tells in which order packets may be dropped. The TotalBytes indicate the size of the ST packet, the maximum is 65535 bytes, same as for IP packets. The HeaderChecksum also works the same way as in IP packets, it is used to validate the correct header. The last 6 bytes are used to identify the stream i.e. specifies the stream ID (SID). Every packet belongs to a stream and is identified by an distinct and unique identifier that simplify packet processing. The UniqueID is an unique number at the stream origin (locally unique) while the OriginIPAddress gives the SID global uniqueness. The bits 12-15 is set to 0 in the QoS specifications for the default ST2+ case, but can have other values in other more subtle QoS specifications.

2.3 Principles

2.3.1 Setup and data transfer

The real-time communication is realized in two steps: connection set-up and data transfer. The set-up uses the control packets, SCMP, to reserve the resources at the routers along the path to the target (targets). The stream setup tasks are to select the route and to reserve the resources. After the setup is ready is the data transmitted along the established stream. Thus the data stream relies on the static path over preselected routers, i.e. there is a sharing of fate for the stream and the selected routers. Further will a duplex communication require two streams.

The data transfer is done under stringent real-time requirements. One strict rule for the end-to-end data transfer is that only one ST packet size is allowed, i.e. the section of the routes that require the smallest packet size specify the ST data packet size. This is determined during the stream setup. The reason is that ST2+ packet fragmentation is forbidden. No error correction is performed, even errored packets are forwarded to the application.

2.3.2 Stream

The most basic concept of the ST2+ is the definition of a stream. It is a unidirectional connection from the origin to the targets in a routing tree construction. Nodes in the tree is called ST2+ agents. The stream is identified with a globally unique stream ID, SID. The SID identifies the data packets so that minimal processing is needed and real-time requirements can easier be met. The stream is setup by the origin, but unspecified targets can after this subscribe and join the stream.

The state of the stream is kept updated with signaling between the ST agents so that stream status can be propagated along the stream path to both origin and target (targets). The signaling is done with SCMP packets and the signaling is protected with the use of strict acknowledgment of messages

2.3.3 LMR and flow specification

The real-time requirements are defined with a flow specification and ST2+ define one for inter-operability reason. Other specifications can be made, some for future needs. The flow specification is transparent to the SCMP as it only transfer it to the LRM's for action. However note that one stream is allowed to have only one flow specification in all parts of the stream. The local resource manager, LRM, is not specified by the ST2+, but the LRM is expected to service the ST2+ with correct responds. The LMR is responsible to check the flow specification, reserve local resources and to help supervise the stream according negotiated flow specification and QoS. Typical QoS is:

- Throughput; average and maximum.
- Delay; end-to-end and variance.

The QoS is set in the same way the packet size is determined (section 2.3.1), the location with the lowest QoS determine the QoS for the whole stream. If this QoS is not good enough will either that part or the whole stream be deleted.

2.3.4 Routing

The routing function is externally supplied and the ST agent will use any routing protocol that has been developed for the IP protocol. One exception is the source routing scheme, which can't be used. The routing is done hop-by-hop from one ST agent to another starting from the origin and finally reaching down to the target(s).

2.4 SCMP message exchange

The SCMP message exchange realize control between the ST agents, a strait example of the setup is shown in figure 2.4. The origin sends a "Connect.request" message to setup a connection. The message is sent to the next hop ST agent given by the routing function. This agent reserves the needed resources and forwards the "Connect.request" message to the next ST agent. When the "Connect.request" message is received at the target is the resources checked and indication given to the application, which considers the connect conditions. If everything is OK will a "Connect.accept" message be transmitted in the reverse direction. The accept message is traversing the intermediate ST agents along the same route and will result in the origin in a "Connect.accept-indication" to the originating application. When all accepts from all targets have arrived will the connection be ready for data transfer, only at this stage is ST data transfer allowed.

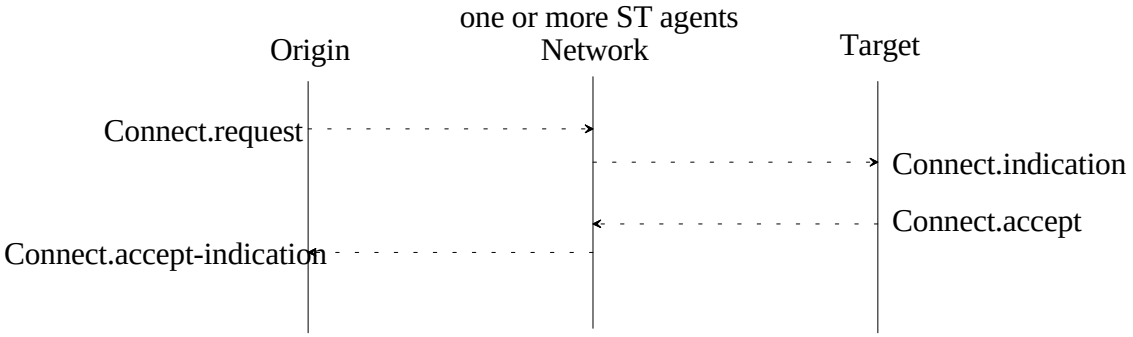


Figure 2.4, Primitives for the Connect stream operation

3 SCMP Protocol specification

The most important instrument in the execution of the ST2+ protocol is the SCMP control protocol. The SCMP defines commands and replies between the ST agents so that streams can be operated upon.

The routers shift states according exchanged SCMP messages. The routers act on SCMP messages according their states. As faulty states can mess the network is more needed to keep the network stable; the ST agents execute according states i.e. agents follow state diagrams, timers are needed to impose re transmit and to force agents change state (typical in error situations), etc. The bitmap format of the SCMP is shown in appendix 1, page 15.

3.1 Control PDU's

The ST agents communicate with the following commands and replies (control PDU's):

1. ACCEPT to accept a new stream
2. ACK to acknowledge an incoming message
3. CHANGE to change the quality of service associated with a stream
4. CONNECT to establish a new stream or add new targets to an existing stream
5. DISCONNECT to remove some or all of the stream's targets
6. ERROR to indicate an error contained in an incoming message
7. HELLO to detect failures of neighbor ST agents
8. JOIN to request stream joining from a target
9. JOIN-REJECT to reject a stream joining request from a target
10. NOTIFY to inform an ST agent of a significant event
11. REFUSE to refuse the establishment of a new stream
12. STATUS to query an ST agent on a specific stream
13. STATUS-RESPONSE to reply queries on a specific stream

With these SCMP messages are the stream state changed. The ACCEPT, CHANGE, CONNECT, DISCONNECT, JOIN, JOIN-REJECT, NOTIFY and REFUSE messages has to be acknowledge explicitely with an ERROR or ACK message. ACK, ERROR and STATUS-RESPONSE are never acknowledged.

3.2 States

The ST agents can be in one of the following states when a stream is under consideration:

- IDLE: the stream has not been created yet.
- PENDING: the stream is in the process of being established.
- ACTIVE: the stream is established and active.
- ADDING: the stream is established. A stream expansion is underway.
- CHGING: the stream is established. A stream change is underway.

Previous experience with ST has lead to limits on the operations of one stream that can be executed simultaneously. If the actions mentioned beneeth arrive at a ST agent, then it should queue the request in a FIFO manner until the prohibited condition is not valid any more. These restrictions for one specific stream are:

1. One at a time: A single ADD or CHG operation can be processed at one time. If an ADD or CHG is already underway, further requests are queued by the ST agent and

handled only after the previous operation has been completed. This also applies to two subsequent. The second operation is not executed until the first one has been completed.

2. Establish -> Delete: Deleting a stream, leaving a stream, or dropping targets from a stream is possible only after stream establishment has been completed. A stream is considered to be established when all the next-hops of the origin have either accepted or refused the stream. Note that stream refuse is automatically forced after timeout if no reply comes from a next-hop.

3. Establish -> Data: An ST agent forwards data only along already established paths to the targets, see also Section 3.1. A path is considered to be established when the next-hop on the path has accepted the stream. This implies that the target and all other intermediate ST agents are ready to handle the incoming data packets. In no cases will a ST agent forward data to a next-hop ST agent that has not explicitly accepted the stream. To be sure that all targets receive the data, an application should send the data only after all paths have been established, i.e., the stream is established.

4. Precondition valid CHGING & ADDING: It is allowed to send data from the CHGING and ADDING states. While sending data from the CHGING state, the quality of service to the targets affected by the change should be assumed to be the more restrictive quality of service. When sending data from the ADDING state, the targets that receive the data include at least all the targets that were already part of the stream at the time the ADD operation was invoked.

3.3 Timers

Every SCMP request waits for a respond, but before that is an acknoledgement expected. Thus there are two types of timers; acknowledgeTimer and responseTimer. The SCMP request and its acknowledgeTimer is typically run N times before aborting the request. After the timer has compeltely expired is a recovery action taken. The recovery action depends of course on the type of SCMP request. However retransmission after time outs is used to recover from lost or ignored messages.

3.4 Other definitions

The **Group of stream** concept is in ST2+ realization defined by four relationships:

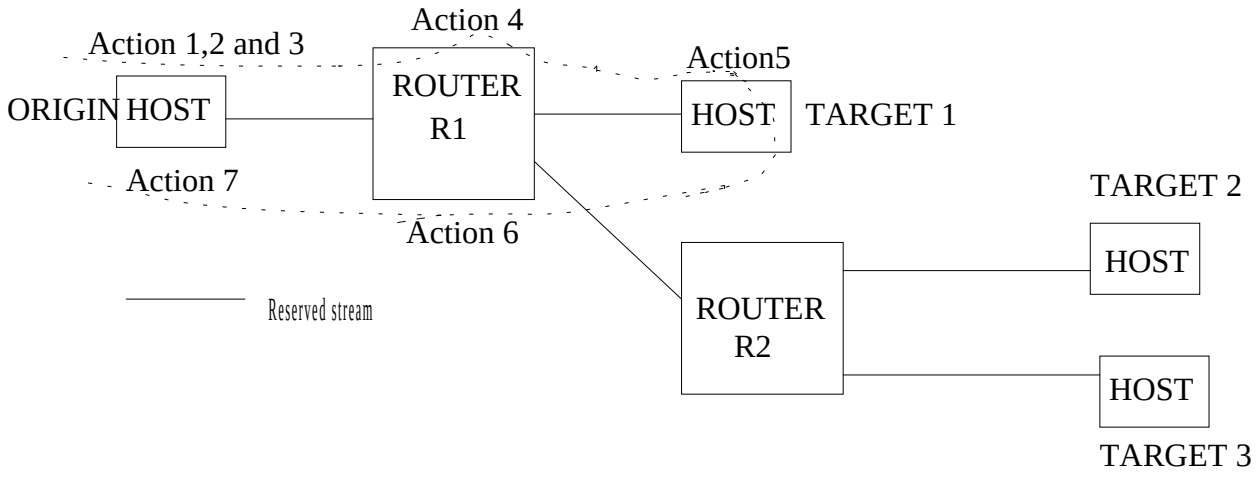
- Bandwidth sharing, bandwidth limit set for the group, not the individual stream.
- Fate sharing, ex. preemption deletes all in the group.
- Route sharing.
- Subnet resource sharing, ex. Ethernet address sharing in multicast.

Any combination of these are allowed to describe a group of streams.

4 Network case

The stream setup and tear down is here described for the network in figure 4.1. The case only shows successful progress of the SCMP protocol. This example is only one case out of many combinations possible.

4.1 Setup



4.1.1 Information from the Application

The application at the origin has to collect nessessary information before launching the setup. The info can be known by the application user or the info migth be collected over the IP network (normal IP best effort traffic). The application passes to the ST agent:

- the list of the stream's targets. The list may be empty, here T1, T2 and T3.
- the flow specification containing the desired quality of service for the stream
- information on the groups in which the stream is a member, if any
- information on the options selected for the stream.

4.1.2 Initial setup at the origin

The ST agent at the Origin has to do following actions:

- allocates a stream ID (SID) for the stream
- invokes the routing function to determine the set of next-hops for the stream
- invokes the Local Resource Manager (LRM) to reserve resources

The routing function is called and in this trivial case is the next hop ST agent located at the R1. The LRM is checked according the Flow specification from the application, the possible group of stream may affect the LRM allocation.

4.1.3 Sending CONNECT messages

The Origin ST agent sends a CONNECT message to the R1 ST agent. The proper SID, FlowSpec and TargetList is attached. TargetList may also be empty.

4.1.4 Connect processing by Intermediate ST agent

An ST agent receiving a CONNECT message, assuming no errors, responds to the previous-hop with an ACK. The ACK message must identify the CONNECT message

to which it corresponds by including the reference number indicated by the Reference field of the CONNECT message. The intermediate ST agent calls the routing function, invokes the LRM to reserve resources, and then propagates the CONNECT messages to its next-hops, as described in the previous sections. R1 sends the Connect to T1 and R2. R2 later also propagates the CONNECT to T2 and T3.

4.1.5 CONNECT processing by the Targets

An ST agent like T1 that gets a CONNECT message, assuming no errors, responds to the previous-hop with an ACK. The ST agent invokes the LRM to reserve local resources and then queries the specified application process whether or not it is willing to accept the connection.

The application on the T1 host is presented with parameters from the CONNECT message including the SID, Origin, FlowSpec, TargetList, and Group, if any, to be used as a basis for its decision. Subsequently received data packets will carry the SID.

Finally, based on the application's decision, the ST agent sends to the previous-hop, the router R1, from which the CONNECT message was received either an ACCEPT or REFUSE message. Since the ACCEPT (or REFUSE) message has to be acknowledged by the previous-hop, it is assigned a new Reference number that will be returned in the ACK. The CONNECT message to which ACCEPT (or REFUSE) is a reply is identified by placing the CONNECT's Reference number in the LnkReference field of ACCEPT (or REFUSE). The ACCEPT message contains the FlowSpec as accepted by the application at the target T1.

4.1.6 Accept processing by Intermediate ST agent

When the intermediate ST agent R1 receives an ACCEPT, it first verifies that the message is a response to an earlier CONNECT. If not, it responds to the next-hop ST agent of T1 with an ERROR message, with ReasonCode (LnkRefUnknown). Otherwise, it responds to the T1 ST agent with an ACK, and propagates the individual ACCEPT message to the previous-hop, the Origin, along the same path traced by the CONNECT i.e. in the reverse direction toward the origin.

The FlowSpec is included in the ACCEPT message so that the origin and intermediate ST agents can gain access to the information that was accumulated as the CONNECT traversed the internet. Note that the resources, as specified in the FlowSpec in the ACCEPT message, may differ from the resources that were reserved when the CONNECT was originally processed. Therefore, the ST agent presents the LRM with the FlowSpec included in the ACCEPT message. It is expected that each LRM adjusts local reservations releasing any excess resources. The RM may choose not to adjust local reservations when that adjustment may result in the loss of needed resources. It may also choose to wait to adjust allocated resources until all targets in transition have been accepted or refused.

4.1.7 Accept processing by the Origin

The origin will eventually receive an ACCEPT (or REFUSE) message from each of the targets T1, T2 and T3. As each ACCEPT is received, the application is notified of the target and the resources that were successfully allocated along the path to it, as specified in the FlowSpec contained in the ACCEPT message. The application may then use the information to either adopt or terminate the portion of the stream to each target.

When an ACCEPT is received by the origin, the path to the target is considered to be established and the ST agent is allowed to forward the data along this path

4.1.8 REFUSE processing

Let's take the situation that T1 won't accept the stream. An intermediate ST agent like R1, that receives a REFUSE message with ReasonCode (ApplDisconnect) acknowledges it by sending an ACK back to the T1 (the next-hop), invokes the LRM to adjusts reservations as appropriate, deletes the target T1 entry from the internal database, and propagates the REFUSE message back to the previous-hop ST agent, the Origin. When the REFUSE message reaches the origin, the ST agent at the origin sends an ACK and notifies the application that the target is no longer part of the stream and also if the stream has no remaining targets. If there are no remaining targets, the application may wish to terminate the stream.

An empty stream is also allowed to be kept. An empty stream will wait for targets that want to receive the stream. These targets will get the stream by sending join request to the origin. The origin will setup the path according requested QoS.

4.2 Tear down

A stream is usually terminated by the origin when it has no further data to send. A stream is also torn down if the application should terminate abnormally or if certain network failures are encountered. Processing in this case is similar to the previous descriptions except that the disconnect ReasonCode (ApplAbort, NetworkFailure, etc.) is different.

When all targets have left a stream, the origin notifies the application of that fact, and the application is then responsible for terminating the stream. Note, however, that the application may decide to add targets to the stream instead of terminating it, or may just leave the stream open with no targets in order to permit stream joins.

5 Conclusion

In this document was the ST2+ protocol introduced. The ST2+ can be seen as a signaling system to establish a connection over the Internet infrastructure *[Luoma96]*. With the ST2+ can applications rely on that the network will fulfill negotiated QoS guarantees. The ST2+ does this by setting up resources at the routers and relying on that the node status will remain. Thus the smooth operation of the ST2+ rely on that the routers can keep their states intact. This might be good from a efficiency point-of-view, but it is in contradiction to the end-to-end principle of the IP world (simple network, reliability done end-to-end). *[Huitema, p 295]*

A major drawback of the ST2+ protocol is the need of a parallel ST protocol next to the IP protocol. The double resource approach consumes processing and memory resources.

Another drawback is that a reliable service is guaranteed only if all the intermediate routers support ST2+. This QoS restrain will certainly be noticeable in first implementation cases were few routers will support ST2+. The use of network parts that don't support ST2+ i.e. network parts with only IP support, will work independent from the ST2+ and offer only best-effort QoS. A typical case is the different routing of individual IP packets across the only-IP network part.

The routing is provided by the IP protocoll, the ST2+ builds the route with the routing function during the setup. The fixed route is then never changed for a stream. The resulting routing can in a way be regarded as strict source routing as the data stream will follow the route that the setup process acomplished.

The new ST2+ packet types and the good penetration requirement of ST2+ routers make possible future use of the ST2+ protocol very uncertain. A first implementation case could be to implement ST2+ on some few host and only on bottleneck routers in between.

The prior ST version, ST2, has implementation available at least for Digital, IBM, NeXT, Macintosh, PC, Silicon Graphics and Sun. The BERKOM MMTS project in Berlin by Deutche Telecom uses ST2 as its core protocol for multimedia like conferencing and mailing. *[DeAl92]*

References

Cohe81	Cohen D., NVP - A Network Voice Protocol NVP-II, University of Southern California, Los Angeles, 1981.
Cole81	Cole R., PVP - A Packet Video Protocol, University of Southern California, Los Angeles, 1981.
DeAl92	Delgrossi L., The BERKOM-II Multimedia Transport System, version 1, BERKOM Working Document, October 1992.
Huitema	Huitema C., "Routing in the Internet", ISBN 0-13-132192-7, Prentice-Hall 1995, 319 pages.
Luoma96	Luoma M., Comments on the IP routing course, Helsinki University of Technology, 5. November 1996.
RFC 1190	Topolcic C., "Internet Stream Protocol version 2 (ST-II)", RFC 1190, CIP working group, October 1990.
RFC 1700	Reynolds J., Postel J., "Assigned numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.
RFC 1819	Berger L., Delgrossi L., Duong D., Jackowski S., Schaller S., "Internet Stream Protocol specification version 2+ (ST2+)", RFC 1819, Stream Protocol Working Group, August 1995.
Schu94	Schulzrinne H., RTP: A Transport Protocol for Real-Time Applications. Work in progress, 1994.

Appendix:

SCMP format:

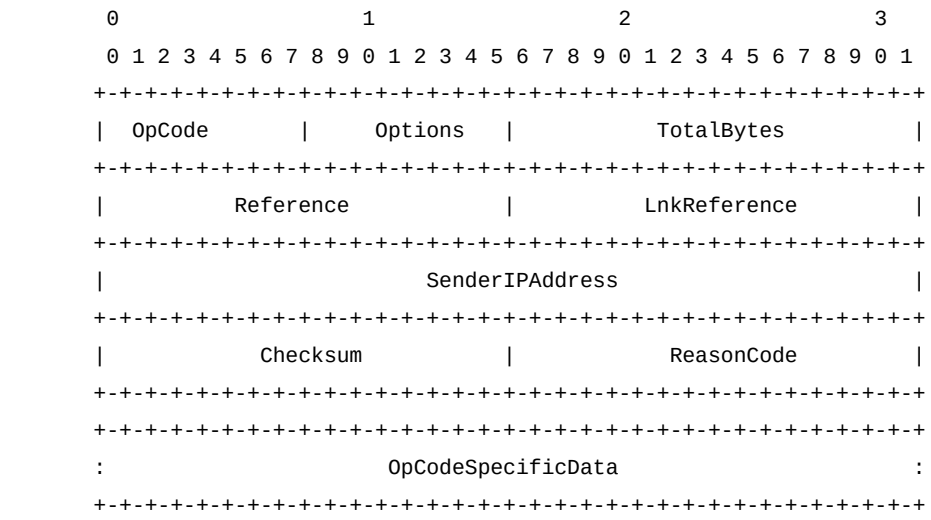


Figure 11: ST Control Message Format

o OpCode identifies the type of control message.

o Options is used to convey OpCode-specific variations for a control message.

o TotalBytes is the length of the control message, in bytes, including all OpCode specific fields and optional parameters. The value is always divisible by four (4).

o Reference is a transaction number. Each sender of a request control message assigns a Reference number to the message that is unique with respect to the stream. The Reference number is used by the receiver to detect and discard duplicates. Each acknowledgment carries the Reference number of the request being acknowledged. Reference zero (0) is never used, and Reference numbers are assumed to be monotonically increasing with wraparound so that the older-than and more-recent-than relations are well defined.

o LnkReference contains the Reference field of the request control message that caused this request control message to be created. It is used in situations where a single request leads to multiple responses from the same ST agent. Examples are CONNECT and CHANGE messages that are first acknowledged hop-by-hop and then lead to and ACCEPT or REFUSE response from each target.

o SenderIPAddress is the 32-bit IP address of the network interface that the ST agent used to send the control message. This value changes each time the packet is forwarded by an ST agent (hop-by- hop).

o Checksum is the checksum of the control message. Because the control messages are sent in packets that may be delivered with bits in error, each control message must be checked to be error free before it is acted upon.

o ReasonCode is set to zero (0 = NoError) in most SCMP messages. Otherwise, it can be set to an appropriate value to indicate an error situation as defined in Section 10.5.3.

o OpCodeSpecificData contains any additional information that is associated with the control message. It depends on the specific control message and is explained further below. In some response control messages, fields of zero (0) are included to allow the format to match that of the corresponding request message. The OpCodeSpecificData may also contain optional parameters. The specifics of OpCodeSpecificData are defined in Section 10.3.