ARM Session Recovery and Load Balancing

Dmitri Kondratiev

dkondr@bigfoot.com

Table of Contents

This document describes different cases of multicast session recovery and load balancing in Active Network (AN) running Active Reliable Multicast family of protocols (ARM).

Problem statement

Initial setting is a group of active nodes (GSNs) participating in data transfer session with one designated active node as primary data source. This primary data source called GSN1 in this paper receives data from the data source external to GSN network. The use cases include bulk data transfer and media streaming. Regardless of data type being transfered both cases deal with multicasting data from one GSN1 source to many GSN2 receivers, which in turn send session data to their data clients - Retail Stores (RSs) or media clients. All GSN2 receivers, as well as original GSN1 sender, maintain local session data cache and equally participate in data recovery from their local caches.

The first goal of this paper is to provide solution that will allow to recover from these failures:

- Failure of any of GSN2 nodes.
- Failure of GSN1 node.
-

The second goal is to find approaches to load balancing problem in GSN group of active nodes.

Problems and limitations of traditional recovery with stand-by nodes

Traditional approach to recovery of single node failure suggests using redundant co-located node in stand-by mode that picks up functions of failing node as soon as this node failure is detected.

Traditional approach has the following problems and limitations:

- Redundant node must maintain the exact session state of the of the node that it mirrors. In GSN network this state includes:
  - Session cache state. Redundant node cache must have the same data as cache on failed node.
  - Other session parameters, such as session ID and session info.
- Questionable return of investment (ROI) and inefficient use of hardware resources. Stand-by node just sits doing no useful work in GSN network waiting for the failure.
- Problem of correct detecting of the node failure.
- Problem of signalling the node failure.
- Problem of starting redundant node.

Group session recovery in case of any GSN failure with N active nodes

Multicasting data to GSN group naturally supports node recovery and inter-node load balancing. Recovery in a group may be customised in many ways and can be more robust then scheme with one stand-by redundant node.

First of all instead of one stand-by node, N recovery nodes (RN) may be efficiently used to provide redundancy. In this approach additional N recovery nodes are added to the existing group that actively participate in the current session.

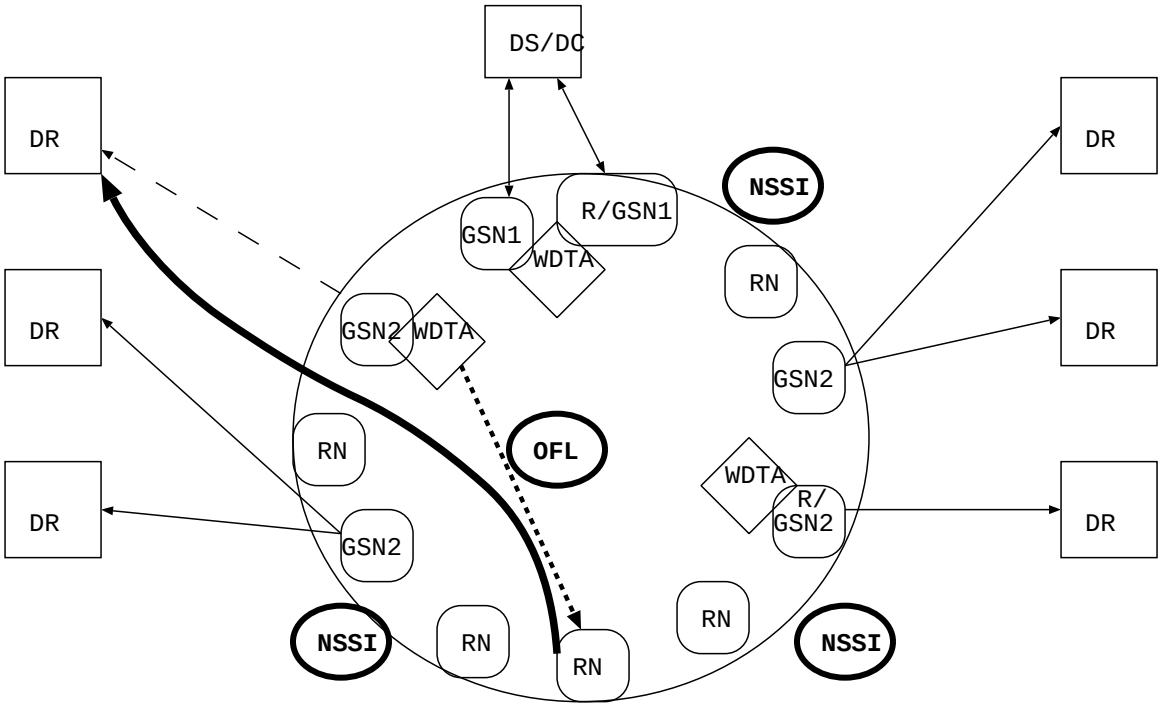

Fig1. Group Recovery / Load Balancing

DS/DC - Data Sender /Data Center
DR - Data Receiver
R - 'stand-by' Recovery Node
R/GSN2 - active Recovery Node
WDTA - Watch Dog Traffic Analyzer
OFL - offload request
NSSI - Node Session Status Info

data transfer in progres

failed / congested link

OFL (offload) request

session recovery

Figure 1. Fig1. Group Recovery / Load Balancing

Functions of active recovery nodes may vary. Group recovery may work in the following modes :

- Recovery nodes take some load from other nodes by servicing existing data clients (RS or media receivers). When some GSN node fails recovery node takes the failing

node clients and starts servicing new clients together with his original set of clients.

- Dedicated Recovery nodes only cache incoming sessions and don't serve any clients until some regular GSN node fails. In this case Recovery nodes work in "stand-by" mode only.

Group recovery may support different recovery conditions detection and initiation schemes as well:

- Automatic recovery conditions detection and initiation by GSN nodes.
- Explicit recovery condition detection and initiation with specialised Watch Dog Traffic Analyser (WDTA) nodes.

Automatic recovery conditions detection and initiation by GSN nodes.

In this mode all GSNs in multicast group periodically send Node Session Status Info (NSSI) packet to the group using ARM Info packet as a vehicle. Node Session Status Info packet contains the following information:

- ARMTP Session ID
- GSN IP
- GSN Cache Info. This includes the latest block written into cache on this GSN.
- Data Receivers Status (DRS). DRS is an array of records where each record describes current state of one Data Receiver (DR) (client of this GSN). DR record includes: DR IP, number of the last data block successfully received by this DR.

All GSN nodes receive NSSI packets from all GSN nodes in the group at some parametrised rate. Every node maintains a table where it keeps latest session status info from all other nodes. Every node starts a timer to detect time-out if some NSSI packet was not received in time. If one or more NSSI packet from some node is missing this in most cases will indicate the failure of the corresponding GSN.

When failure is detected, active nodes start bidding / competing for the right to serve DR clients of the failed node. Here many different bidding decisions and mechanisms may be implemented. For example, only nodes that currently have sufficient resources (adequate work load) can bid. Or all nodes can participate in bidding. In both cases the 'best' candidate is selected to serve clients of the failed node. Selection of the best candidate can be done with several algorithms described later.

Explicit recovery condition detection and initiation with specialised Watch Dog Traffic Analyser nodes.

In this mode separate dedicated Watch Dog Traffic Analyser (WDTA) node is coupled with every GSN node. WDTA does not maintain its own packet cache, its only goal is monitoring master GSN and session health and make intelligent recovery decisions. WDTA is a machine that joins the same session (or sessions) that GSN node is a member of which WDTA monitors. WDTA monitors alive status of its GSN by hardware means and can detect its failures such as power-down or reboot. WDTA also naturally monitors all packets sent to the session multicast group. Monitoring session packets allows WDTA observe the following state from all nodes in the session:

- Latest retransmission requests send by each node.
- Latest data blocks / packets received by each node.

In accordance with the above session state observations at any given moment during session life time, WDTA knows the best candidate GSN that can be used to take over failed GSN. In simplest case, WDTA knows at least about live dedicated recovery nodes (RN) if these participate in the session.

WDTA may detect both hardware failure and congestion conditions of its master GSN. When any of these failures happen WDTA sends explicit OFL request to offload his master to the ARM group. Offload (OFL) request includes:

- IP address of GSN to take over the failed node.
- Failed GSN Node Session Status Info (NSSI) described above.

The main advantages of this scheme include faster failure detection and recovery as well as better ROI, as WDTA node does not use external memory for caching.

Recovery of active node sending data (GSN1) to session group

This case requires redundant sending nodes operating in 'stand-by' mode. That requires data source (DS), such as data center (DC) or media server sending the same data both to active GSN1 and redundant, 'stand-by' GSN1 at the same time.

The failure of original GSN1 may be detected by coupled with it WDTA node. GSN1 failure detection and session redirection can be achieved in the same manner as described for the case of using WDAT in explicit session recovery in GSN2 group.

This case will also require additional ARM Info packets to be sent to GSN1 group indicating the fact of changing the group sender for the current session.

Load balancing in GSN group

The main goal described here is to balance the load implied by the number of data receiving (DR) clients and off-loading congested GSNs.

Similar to recovery scenarios load balancing may be divided into two main cases: automatic load balancing and load balancing on explicit request.

Automatic load balancing

In this mode all GSNs in multicast group periodically send Node Session Status Info (NSSI) packet to the group using ARM Info packet as a vehicle. All GSN nodes receive NSSI packets from all GSN nodes in the group at some parametrised rate. Active nodes decide when they have enough free resources to take off some load from other nodes. When this happens active nodes start bidding / competing for the right to serve extra DR clients of others possibly suffering from congestion or lack of memory resources. Again many different bidding decisions and mechanisms may be implemented here. For example, nodes can put the 'weight value' approximating free resources of the node in its bid. Node with the highest bid is selected to take extra clients from congested node.

Load balancing on explicit request

In this mode WDTA node coupled with congested node selects the best candidate to take part of his master load. Selection is made after monitoring all active nodes in the group after some sufficient period of time. WDTA then sends OFL request to ARM session group.