ARM File - Active Reliable Multicast File Protocol

Dmitri Kondratiev

dkondr@bigfoot.com

Table of Contents

Abstract

This document provides design overview of the ARM File - Active Reliable Multicast File transfer protocol. ARM File uses IP Multicast to reliably deliver a set of files to a group of recivers engaged in one-to-many or many-to-many session.

Protocol Description

File Session using Shared recovery with Multicast NAKs and Retransmissions

ARM File protocol is used to transfer one or more files in context of one distinct session. File session has unique ID contained in the packet header as well as info about number of files being transfered, file names and file sizes. File data is sent in packets that also have current data block number and file size info.

Reliable file data transfer is based on "Shared Recovery with Multicast NAKs and Retransmissions" mechanism. To avoid the NAK implosion receivers share recovery responsibilities. When a receiver detects missing data and sends a NAK, another receiver that has received the data can potentially respond to the NAK by retransmitting the missing data.

NAKs and retransmissions are sent by receivers to a multicast group address. The ARM File algorithm resembles Host Membership Queries in IGMP.

- Multicast NAKs and Retransmissions Algorithm in BSG cloud.

  In BSG cloud, BSGs of all levels (level 1 and 2) play the same role of receivers sending both NACKs and answering retransmission requests as well. Thus, no matter what level, all BSGs are equal in that they perform the same roles in ARM File protocol. This assumes some file storage and/or file cache on every BSG to retransmit data.

  Below the description of retransmission algorithm follows with ’receiver’ used in place of ’BSG’ and vice versa.

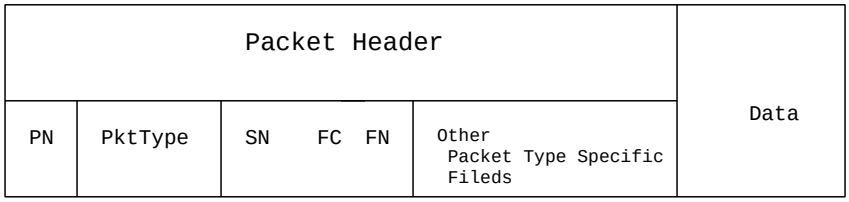
  - 1. When a receiver detects missing data, it starts a (random) timer.
  - 2. When the timer expires and receiver hasn’t yet received a NAK for the missing data from the group (multicast) address, it sends a NAK to the group (with a limited scope).
  - 3. All receivers within the scope see the NAK, start a (random) timer, and watch for the requested data to be retransmitted.
  - 4. If retransmitted data is seen, receivers stop the timer. Else, if their timer expires they retransmit the data to the multicast group address (with a limited scope) if they have it, or send another NAK if not.

- Advantages
  - Off-load the recovery burden from the sender, and provide faster recovery (since NAK and retransmitted data have a shorter distance to travel).
  - Other receivers that also missed the same data can see the NAKs, and don’t need to send them to benefit from the retransmitted data.
  - Reduced recovery latency.
  - Reduced network traffic in the path between sender and receiver.

- Caveats.

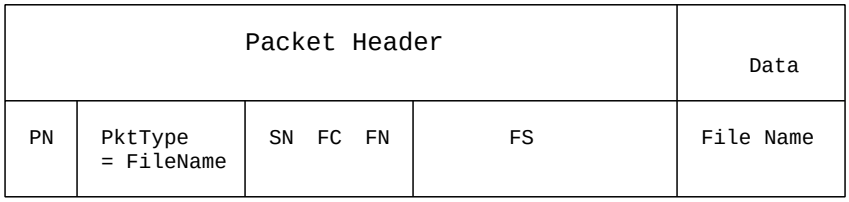
- May increase overall network traffic.
- Receivers must be capable of gracefully handling redundant data.
- Retransmission must be done in the limited scope.
- Data delivery is delayed because of the time-outs to send NACKs.
- Will not work on network topologies that don’t support many-to-many multicast.

Packet Types and Format



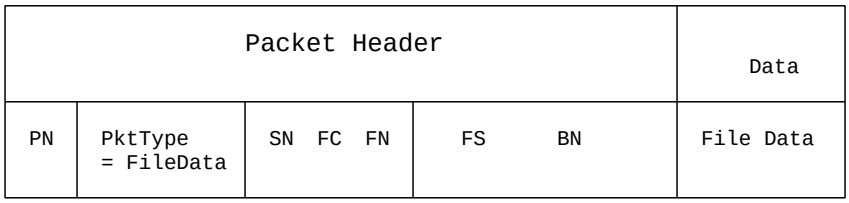
PN - Packet Number
PktType - Packet Type
SN - Session Number
FC - File Count
FN - File Number

Fig. 1.1 ARM File Protocol Packet Format



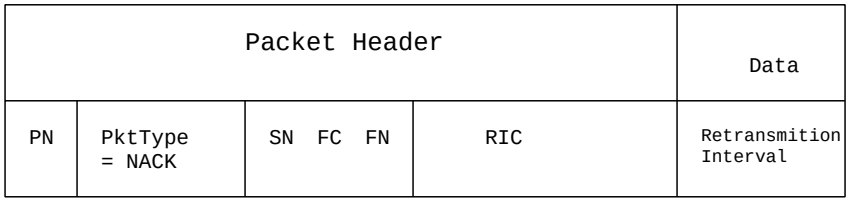
FS - File Size

Fig. 1.2 File Name Packet



FS - File Size
BN - Block Number

Fig. 1.3 Data Packet



RIC - Retransmission Interval Count (number of intervals)

Fig. 1.4 NACK Packet

Figure 1.