Table of contents

Introduction

FRIP: What for?

The reason behind FRIP is really simple. I don't like to do myself anything that is not fun and can be done with the help of computer or some other machine. For example, I don't like to keep routing tables for my FIDO node manually since it doesn't look funny for me, can be done automatically and, hopefully, automatic routing is more efficient. Second reason is ability of current FRIP to route files, which is virtually impossible and dangerous[1] by standard FTN means.

Assumptions

You are not too paranoid. :-)

You have a decent computer, such as 486/8M/500HD. Even 386 will do, but it better should be running some multitasker to not to keep your mailer from answering a ring when mail processing takes place. FRIP has a DOS version, but it will be withdrawn in a half of year or so, cause there's only 5% of DOS-based FIDO nodes even here in Russia - that's in August of 1996. FRIP is distributed with sources, so everybody can port it to any OS he likes.

You are able to modify sample config file to suit your system.

You have mailer either with Binkley-style outbound, T-Mail style fileboxes or some program that can create a file-attach for your mailer.

Basics

There is two tightly coupled parts in FRIP today. The FRIP itself and so-called FARA - frip Aided Router of Attaches. Both of them are in one executable, and it is done on purpose. It makes me sure that if FRIP works, FARA is operational too. Here's what each one does.

FRIP

Does the routing analysis and builds routing table for FARA. Frip is able to produce text-form routing table for other software too, but it is not a primary function. FRIP builds routing table with the help of RIP-files, or, to be short, RIPs. Such a files are sent on a regular basis (one in a two weeks by default) to all FRIPs around and relayed through all the FRIP-ified nodes. Those nodes, in turn, gather information from RIPs coming through and discover the shortest ways to other nodes, networks, zones and domains around. The description of FRIP operation here is VERY brief. Please, refer to page to read more about it. As a user you don't have to know anything about it, though.

FARA

FRIP-Aided Routing of Attaches is a FRIP-based data (mail, files) delivery mechanism. FRIP is able to encapsulate data into a RIF files and transfer them through the FRIP-able nodes. FARA was brought to life after half a year testing of FRIP, which shown that a lot of people easily sets up

[1] There was real cases here in Moscow when two files with same names were forwarded via one node. In addition, it is impossible, for example, to hatch one file to more than one file echo at once if both of echoes go through same nodes.

FRIP, but does not let it produce routing tables for main system router. With the help of FARA FRIP can be used as a separate, add-on mail delivery system. FARA is more sophisticated than usual file routing software - it does not overwrite files with the same names, transfers file descriptions from *files.bbs* and *descript.ion* files, if one of them is found in a directory we sending file from and sets some extended attributes[2] of received file[3].

Minimal setup & usage

To be able to use most of the FRIP features you need to perform an easy set up: put FRIP in a separate directory and specify minimum of parameters in a config. File. The most complex parameters is those which let FRIP send files to your direct links. Things are real easy if your mailer supports Binkley-style outbound. Most current mailers, including The Brake, Binkley, Bink/+, T-Mail, Xenia and others do support such outbound. If your mailer supports Binkley-style outbound, define BinkOutBound and BinkOutBoundDefault parameters. First must specify path to your outbound, excluding last subdirectory, and second - that last directory name, whithout path. If your mailer does not support Binkley outbound but supports T-Mail style fileboxes (including tobesent.$$$ file feature), you can use TmailFileBoxesFAT or TmailFileBoxesHPFS for FAT (DOS) or HPFS (long name) style fileboxes. If both ways are not for you, still there is a workaround exist: DeliverCommand parameter can be used to tell FRIP which command it has to execute to send file to some address. Note that file must be sent only directly and in *kill file sent* mode. Apart from this, you should specify Address (one per your AKA), Dir (where you put FRIP), Inbound (your secure inbound directory), FripMail (NetMail, transferred by FRIP[4]) and Log (log file name) parameters. In a directory, specified by FripMail parameter your mail, received by FARA will be put, and all netmail written in that folder will be sent by FARA, if possible.

```
Address 2:5030/16.555
Address 666:777/888.999
Dir c:\net\Frip
Inbound c:\Net\Inbound\Secure
FripMail c:\Net\Mail\Frip
Log c:\Net\Logs\Frip.Log
; The following parameters assume that your main Bink
outbound is c:\Net\Outbound
BinkOutBound c:\Net
BinkOutBoundDefault Outbound
```

In addition to frip.cfg file links.cfg must be defined too. It is even more easy to create this file. For the first time just list all your direct links with neighbor nodes using FRIP, one node per line, together with your main aka first:

```
<your main address> <first link node address>
<your main address> <second link node address>
...
```

That's all. Now just run frip.exe when you need to pack fripmail, unpack received RIPs, RIFs and RIZs, and at least daily.

[2] OS/2 version only.

[3] Right now, *.subject* and *.history* attributes are set to file description and delivery path.

[4] Don't forget to add this directory as *.MSG-type netmail folder to your mail editor's configuration! If your mailer is arcmail-attach, this directory must not be the same as your mailer's netmail directory.

Common errors

To be gathered. :)

Simple tweaking

Total Control

Configuration File Reference

Lines, beginning with ':' and empty lines are ignored.

Boolean parameters can have value of `Yes/No/On/Off`.

Configuration file keywords reference

Keyword	***Example***	***Description***
Address	address 2:5020/32.555 address 7:1130/32.555 address 7:7/0.555	Your addresses.
Dir	Dir c:/net/frip	Frip home dir
Inbound	Inbound c:/net/in	Mailer (secure!!) inbound
FripMail	FripMail c:\net\FripMail	Frip will send netmail from this directory using FARA (see page) and receive FARA netmail to this directory.
Log	Log c:\net\logs\Frip	Log file name
BinkOutBound	BinkOutBound c:/net	Path to Binkley-style outbound EXCLUDING LAST DIRECTORY. NB! Select and define one of BinkOutBound+BinkOutBoundDefault, HPFSOutbound, TMailFileBoxes or DeliverCommand.
BinkOutBoundDefault	BinkOutBoundDefault	Last Bink outbound

	outbound	directory name (excluded above)
HPFSOutBound	HPFSOutBound c:/net/HPFS_outbound	Path to HPFS outbound (test version)
TMailFileBoxesFAT	TMailFileBoxesFAT ./filebox	Path to T-Mail fileboxes - FAT format (DOS mode)
TMailFileBoxesHPFS	TMailFileBoxesHPFS ./filebox	Path to T-Mail fileboxes - HPFS format (OS/2 & NT mode, long names)
DeliverCommand	DeliverCommand Squish Send "^%0" to %1	Command to send file %0 to node %1, DIRECT flavor and kill file sent ONLY!
Hold	Hold 333 5030/* -5030/222	Send RIPs with HOLD flavor to given nodes. Does not work with DeliverCommand.
AnnouncePeriod	AnnouncePeriod 15	Time in days between automatic announces. Default - 14 days. NB! **DO NOT SET IT TO LESS THAN A WEEK.**
PingPeriod	PingPeriod 2	Time between pings. Default - 2 days.
ZipCmd	ZipCmd zip -mj1	zip or pkzip command. Puhleeease! Do NOT use pkzip under OS/2, Win/NT or Win/95. Use native OS/2 or Win32 zip/unzip instead. This parameter is for special cases only - don't specify it if don't have to. Frip has reasonable defaults.
UnZipCmd	UnZipCmd unzip -Cjn	unzip or pkunzip command. See comments to zip command.

CmdLimit	CmdLimit 400	Max. length of OS command line. Default is 100 for DOS, Win/95 and NT (because of Win95!), 400 for OS/2.
Ansi	Ansi Off	Enable/disable ANSI coloring.
Debug	Debug On	Debugging information switch.
Map	Map n:\temp\MapDir	Directory to build routing map in. FRIP does not use that map at all, and is able to build it for sysop's use only.
MapFormat	MapFormat (%s)_%s_%s/%4s.%-3s	Map file name printf format. Each %s from left ot right substituted with: Domain, Zone, Net, Node, Point, Input rip file name (usually not needed).
AnnounceFromAdFiles	AnnounceFromAdFiles Off	Turn on processing of .ad files. For those who really needs it. Be careful! Don't use, if not understand what you are doing.
ProcessedFlag	ProcessedFlag c:/net/Frip_routing_changed.!!!	Frip will create this file if routing possibly changed

The following table lists keywords, used in external router mode. This mode is obsolete, not recommended and can be used only if neighbor FRIP nodes are support this mode too.

Obsolete Keywords Reference

Keyword	***Example***	***Description***
StripDomain	StripDomain Off	If turned on, FRIP will strip domain from addresses it puts in external route file.
PointRouting	PointRouting On	If turned on, FRIP will put routes to/via points to external routing file.
WildWord	WildWord *	Wildcard character or word for external routing file
RouteWildAfter	RouteWildAfter Yes	Whether to put wildcarded routes after or before detailed ones
RouteFilePath	RouteFilePath c:\net\squish	Routing file directory
RouteFileName	RouteFileName route.cfg	Routing file NAME ONLY, with no path. Leave empty if no external routing file to be created. It is recommended to use FRIP without external routing.
RouteFileSuff	RouteFileSuff Suffix.rou	Contents of this file will be added to output routing file at the end.
RouteFilePref	RouteFilePref Prefix.rou	Contents of this file will be prepended to output routing file
RouteLine	RouteLine Route %1 to %0	Routing file line format
ViaSelf	ViaSelf Yes	Put routing of node via itself to route file. For example, »Route 2:5020/666 to 2:5020/666«

Filtering

There is a way to control FRIP more precisely, if you want it. Definitions of your FRIP links accept parameters you can use to restrict routing in a special way or to add weights to links. The complete syntax of link definition is:

```
;/**
; *
; * The software included, file formats and basic algorithms are
; * copyright (C) 1995 by Dmitry Zavalishin. All rights reserved.
; *
;**/
;
; $Id: links.cfg 1.1 1996/01/07 15:15:23 dz Exp $
;
; this file lists your DIRECT links (those you call or called by)
; definition looks like this:
;   myaddr node [IN <parameters>] [OUT <parameters>]
;
; where parameters are:
;   {-mask|+mask}                        - enable/disable ads for
;                                  this address to come or go here
;   WILD=Domain|Zone|Net|Node|Point|None  - disable ads with wildness less
;                                  than given here.
;   TIME+=##d|h|m|s                      - make ad to be older by this number
;                                  of days/hours/minutes/seconds.
;
; example:
;   2:5020/32 2:5020/0 OUT +2:5020/* -2:5020/128
;
; æ»¿ß«¬ óáÞ¿Õ »p´¼ÙÕ ½¿¡¬«ó - ¡á ¡¿Õ íÒñÒÔ «ÔßÙ½áÔýß´
; «íÛ´ó½Ñ¡¿´ « p«ÒÔ¿¡úÑ.
;
; Å«ß½Ñ ß½«óá IN »¿ÞÒÔß´ óÕ«ñ¡ÙÑ õ¿½ýÔpÙ - ¬á¬¿Ñ p¿»Ù »p¿¡¿¼áÔý,
; ¬á¬¿Ñ - ¡ÑÔ. Å«ß½Ñ OUT - óÙÕ«ñ¡ÙÑ, ß««ÔóÑÔßÔóÑ¡¡«.
;
; îáß¬¿:
;   +áñÓÑß   - »Ó«»Òß¬áÔý á¡«¡ßÙ ß Ôá¬¿¼¿ áñÓÑßá¼¿
;   -áñÓÑß   - ¡Ñ »Ó«»ÒßáÔý á¡«¡ßÙ ß Ôá¬¿¼¿ áñÓÑßá¼¿
;
;   WILD=Domain|Zone|Net|Node|Point|None
;           - »Ó«»Òß¬áÔý á¡«¡ßÙ ß« ºóÑºñ«þ¬á¼¿ ó ÝÔ«® ¿ ß½ÑñÒ¯Ú¿Õ »«º¿µ¿´Õ
;   TIME+=##d|h|m|s
;           - ÒÕÒñÞ¿Ôý »Ó«Õ«ñ´Ú¿® á¡«¡ß ¡á ## ñ¡Ñ®/þáß«ó/¼¿¡ÒÔ/ßÑ¬Ò¡ñ
;             æ«¬ÓáÚÑ¡¿´:
;               WEEK - 7 ñ¡Ñ®
;               DAY  - ñÑ¡ý
;               ZMH  - ñÑ¡ý («Ôºó«¡¬á Ô«½ý¬« ó ZMH)
;               HOLD - Óáº ó ñóá ñ¡´
;               !CM  - ñóáªñÙ ó ñÑ¡ý
;               CM   - ºáñÑÓª¬á - 0

;
; ÅáÓá¼ÑÔÓ TIME+= ÓÑ¬«¼Ñ¡ñÒÑ¬Ôß´ ßÔáó¿Ôý Óáó¡Ù¼ »«½«ó¿¡Ñ ßÓÑñ¡Ñú« óÓÑ¼Ñ¡¿
; ¼ÑªñÒ ¬«¡¡Ñ¬Ôá¼¿ ß ñá¡¡Ù¼ Òº½«¼ - ¡Ñ ºáíÙóá®ÔÑ, þÔ« Ñú« ¡áñ ßÔáó¿Ôý ¿ ó
; IN, ¿ ó OUT ßÑ¬µ¿¿.
; ùÑ¼ «¡ í«½ýÞÑ, ÔÑ¼ ¼Ñ¡ýÞÑ óÑÓ«´Ô¡«ßÔý, þÔ« ¡á ñá¡¡Ù® ½¿¡¬ óÙßÔáó¿Ôß´
; Ó«ÒÔ¿¡ú. é ßÑ¬µ¿¿ IN «¡ ó½¿´ÑÔ ¡á »Ó¿¡¿¼áÑ¼ÙÑ Ó¿»Ù, Ô« ÑßÔý ¡á Ó«ÒÔ¿¡ú
; ÄÆ éÇæ. é ßÑ¬µ¿¿ OUT - ß««ÔóÑÔßÔóÑ¡¡«, ¡á ÒÕ«ñ´Ú¿Ñ Ó¿»Ù, Ô« ÑßÔý ¡á
; Ó«ÒÔ¿¡ú è éÇî.
;

; ôº½á¼ ¿º º«¡Ù 2 «ÔñáóáÔý óßÑ «íÛ´óÙ, ¬p«¼Ñ 7-® º«¡Ù ¿ 256-ú« Òº½á
; Ç¡«¡ßÙ «Ô/¬ Òº½Ò 5020/50 ÒÕÒñÞáÔý ¡á 4 þáßá, Ôá¬ ¬á¬ Ò ¡ÑÑ ÑßÔý
; »ÑÓÑÓÙóÙ ó Óáí«ÔÑ »«Ó´ñ¬á 4-Õ þáß«ó.
2:5020/32.555   2:5020/50        OUT -7:*/* -2:5020/256 TIME+=4 IN TIME+=4
2:5020/32.555   2:5020/269        OUT -7:*/* -2:5020/256

; Ò ¼Ñ¡´ »½«Õá´ ßó´ºý ß 50/0, »«ÝÔ«¼Ò ¡Ñ ßÔ«¿Ô «íÛ´ó½´Ôý Ñ¼Ò Ô«, þÔ« «¡
; ´ó¡« ß¼«ªÑÔ «Ôp«ÒÔ¿Ôý í«½ÑÑ ¬áþÑßÔóÑ¡¡« þÑpѺ ñpÒú¿Ñ Òº½Ù. ê ¡á«í«Ó«Ô,
; óßÑ «íÛ´ó½Ñ¡¿´ «Ô ¡Ñú« íÒñÒ ÒÕÒñÞáÔý ¡á 4 ñ¡´, þÔ«íÙ «¡¿ ºáóÑñ«¼«
; »Ó«¿úÓÙóá½¿ Ó«ÒÔ¿¡úÒ þÑÓѺ ñÓÒú¿Ñ Òº½Ù. Äñ¡á¬«, ß«óßÑ¼ «Ôõ¿½ýÔÓ«óÙóáÔý
; ¿Õ ¡Ñ íÒñѼ, þÔ«íÙ ¿ß»«½ýº«óáÔý ÝÔ«Ô ½¿¡¬ ó ¬áþÑßÔóÑ íÑ¬á»á - Ñß½¿
; ñÓÒú¿Ñ ß«óßѼ ßÔÒÕ¡ÒÔ.
;
; (é »Ó¿¡µ¿»Ñ, ÒÕ«ñ´Ú¿Ñ Ó¿»Ù Ô«ªÑ ½ÒþÞÑ íÙ½« ÒÕÒñÞ¿Ô ¡á ¡Ñß¬«½ý¬« ñ¡Ñ®,
; á ¡Ñ õ¿½ýÔÓ«óáÔý, ¡« ´ ºñÑßý Õ«þÒ »«¬áºáÔý Óáº¡ÙÑ ß»«ß«íÙ.)
;
```

```
2:5020/32.555   2:50/0           OUT -7:*/* -2:5020/* IN -2:5020/* TIME+=4d

; ØÔ¿ Òº½Ù ÑßÔý ¿ ó 7-® º«¡Ñ, Ôá¬ þÔ« õ¿½ýÔp«óáÔý ¡Ñ ¿¼ÑÑÔ ß¼Ùß½á
2:5020/32.555   2:5020/144       OUT -2:5020/256
2:5020/32.555   2:5020/180       OUT -2:5020/256 TIME+=!CM IN TIME+=!CM
2:5020/32.555   2:5020/69        OUT -2:5020/256

; ôº½á¼ ¿º 7-® º«¡Ù ß½áÔý óßÑ á¡«¡ßÙ
7:1130/32.555   7:1130/256
7:1130/32.555   7:1130/144
7:1130/32.555   7:1130/0
7:1130/32.555   7:1130/69

; ÝÔ« - ½«¡ú½¿¡¬, íÒñÑ¼ ß½áÔý Ñ¼Ò Ô«½ý¬« ¿¡õ«Ó¼áµ¿¯ « ßÑÔ´Õ,
; ¿ ¡Ñ íÒñÑ¼ ¡áúÓÒªáÔý ½¿¡¬ Ó¿»á¼¿ ß Ó«ÒÔ¿¡ú«¼ ¡á «ÔñÑ½ý¡ÙÑ Òº½Ù.
; èÓ«¼Ñ Ô«ú«, ¡Ñ íÒñÑ¼ ñáªÑ »ÙÔáÔýß´ ß««íÚáÔý Ñ¼Ò « Ó«ÒÔ¿¡úÑ ¡á
; Ñú« ªÑ ßÑÔý - ¡ÑºáþÑ¼.
; Å«ß¬«½ý¬Ò íÙóáÑÔ, þÔ« «¡ ¡Ñ »Ó«ºóá¡¿óáÑÔß´ ¬ ¡á¼ »« ñó«Ñ ßÒÔ«¬,
; óßÑ¼ »Ó«Õ«ñ´Ú¿¼ ¿ ÒÕ«ñ´Ú¿¼ Ó¿»á¼ ñ«íó¿¼ ¬ óÓÑ¼Ñ¡¿ »Ó«Õ«ªñÑ¡¿´ ñÑ¡ý.
; (»«½«ó¿¡Ò ¼á¬ß. ¿¡ÔÑÓóá½á »Ó«ºó«¡¬¿)
2:5020/32.555   2:5030/5555  OUT Wild=Node -2:5030/* TIME+=2d IN TIME+=1d
```

Aggregates

One of the most powerful features of FRIP is its ability to annonce routing not only to a single nodes or points[5], but even to whole nets, zones and domains[6]. Please, don't misuse this feature. It is assumed by FRIP, that only NC can announce default for a net, RC - for a region and ZC for a zone. All other aggregate announces are marked as non-approved and will allways be treated as bad when compared with approved ones. So, annonce of 2:5020/* by node 2:5020/3773 will only be taken in account if 2:5020/0 did not annonced this aggregate. It is wise move for *Cs to announce aggregates for their nets, regions and zones, because it lowers RIP traffic alot. Note that FRIP is designed so that aggregate routing does nothing inside of corresponding part of net. It is only transferred from node to node, and will not go to routing table, will not override detailed routes, etc. For example, route for 2:5020/* will be ignored by nodes in 2:5020, and just forwarded further. Nodes of other networks, on the other hand, will take it in account, use for routing and it will override detailed routes there, if happens to be better.

[5] Note that annonce of node address is automatically assumed to be announce of all the points of that node.

[6] There is no region aggregate because no single rule exist that can tell which net/node belongs to which region.

Protocol

For those who curious

This is the simple description of FRIP operation.

RIP, or RIP file, is a baisc data structure used in FRIP protocol. RIPs are used to pass routing and other info between FRIP-based nodes. Structure of RIP is described on page .

Frip operation is based on routing builder subsystem, which itself can be divided into the announcing and relaying parts. Announcer is a simple module wich sends AD-type[7] RIPs (or ADs) to all the frip links of our node once a two weeks[8] or so. Such a RIP is a message, that means »I'm here, and it will take XX hours to get from you to me this way« to all the FRIP nodes around. RIP is transferred to the next FRIP node and picked up by relaying part of FRIP there. Relayer first subtracts current time from the RIP timestamp to calculate time RIP spent to get here. This time is compared with delivery time, stored in routing database. If new time is better (less) than time in database, it is assumed, that this RIP brings us better routing, than we know yet. In such a case database is updated according to received RIP: delivery time is set to newly calculated, and routing destination is set to node we got this RIP from. Then RIP is relayed to our neighbours, with the exception of those listed in RIP seenby field. On the other hand, if RIP shos worse delivery time than we know, it is ignored and will not be relayed.

For those who real interested

For those who's serious to implement

Details, data formats, etc.

RIP structure

RIP file is an ASCII text file with the name beginning with letters »rip«, extension beginning with ».ri« and ending with one letter or digit. Lines in RIP file can end with either Unix-style newline character or RSX-style CRLF. Last line must end with newline or CRLF too. Each line begins with keyword and possibly contains a value. There are a few types of RIPs, distinguished by value of TYPE line.

RIP types

[7] AD means »Advertisement« here.

[8] Period can be changed but should NEVER be too small. Once a week is a good minimum. There is a reasons, believe me. We'll discuss 'em later.

Type	*Version*	*Obligatory keywords*	*Description*
AD	1	Ad, from, to, created, hops, path, seenby	Routing announce. Used to track routing
OFF	20	Off, from, to, created, hops, path, seenby	Routing loss announce.
PING	20	From, to, created	Used to gather average link delay information.
HI	?		Not implemented.
HACK	?		Not implemented.
RRQ	?		Rescan request. Should be used by a new FRIP node to get all available routing information quickly. Not implemented.
RACK	?		Rescan acknowledge. Not implemented. I'm not really sure it has to be implemented at all.
SEARCH	24?		Not implemented.

Here goes description of all possible RIP fields. Note that FRIP passing all the unknown fields through, so all the new fields that will be invented are to come through unnoticed by old FRIPs.

RIP fields

Field name	*Version*	*Data type*	*Description*
From	1	Address	Address of system, that sent this particular RIP-file, i.e. our neighbor.
To	1	Address	Address of system this RIP is intended for. Used to prevent processing of mistakenly routed RIPs by intermediate nodes. NB! RIPs should never be routed, as well as

			RIFs and RIZs.
Type	?	String	Type of RIP, see above. Not case-sensitive.
Capas	20	Flags	Capabilities of system, which generated this RIP. Used to keep track of neighbors abilities. Was introduced together with FARA-able FRIPs.
Created	1	Time/Date	Time/date to subtract from now to get route quality estimate. It is NOT always a real time/date RIP was created, don't treat as such.
Creator	?	Address	Todo: add version/os type here.
Flags	?	Flags	Special info. Used by AD-type RIPs to distinguish between routing types and carry additional info.
Hops	1	Integer	Hops this RIP made
Ad	1	Address	Announced address
Off	20	Address	Address, we lost routing for
Search	?	(Complex)	
Path	1	Address	Primary addresses of all systems RIP passed through
Seenby	1	Address	All AKAs of all systems RIP passed through

RIF structure

RIZ file is a binary file with the name beginning with letters »rif«, extension beginning with ».ri«

and ending with one letter or digit. RIF is used to carry mail, and beginning of it is a data carried. All the envelope info including addresses, path/seenby info, etc. etc. is kept in a »tailer«.

RIZ structure

RIZ file is a regular ZIP archive (InfoZIP 2.1 level), with the name beginning with letters »riz«, extension beginning with ».ri« and ending with one letter or digit. There can be RIPs and RIFs in RIZ file.

Final Words

Thanks!

I'd like to thank Kirill Lebedev, Kirill Shirokov, Anton Tsarevsky, Alex Tutubalin, Mike Bravo, Serge Troffimovsky, Dmitry Efimov, Andrew Iwanov, Ivan Munitsyn, Alexander Erlikh, Fyodor Ustinov, Basil Dolmatov, Yuri Safronov, Dmitry Morozovsky, Pavel Bourdanov, Vladimir Kopiev and all the other people, who took part in protocol discussions, betatesting, etc. etc.

Plans

Time zones must be used in RIPs.

Rescan requests and responses are not done.

Direct RIZ processing - without pre-unzip. Tricky, but possible.

Fatal routing loss. If some database record was not updated in a two months or so, remove it at all and generate routing loss event (OFF-type RIPs to all).

Make FRIP know does it rule the main netmail router or not. If not, add a special flag to a routes coming through to make sure these routes will not be used in generation of external routing tables as well.

Bugs

Not that FRIP has real serious bugs which are known but not fixed. Here is just things that can be done better:

FRIP should discover that routing base is empty and request links for rescan.

Inability to send mail because of route absence should force routing loss (OFF) request, but NOT TOO FREQUENT! I have to be careful here and not to bring OFF-storms to life.

In addition to Deliver command FRIP must understand arcmail-attach style mailers by itself.

Contacts

Yours truly Dmitry Zavalishin aka dz can be reached via:

E-mail (preferred): dz@phantom.ru, dz@kiae.su, dz@virgin.relcom.eu.net, Dmitry_Zavalishin@f32.n5020.z2.fidonet.org, fidonet#2:5020/32.

Phone: +7 (095) 110-6728. If somebody non-English speaking answers your call and you can't speak Russian just keep saying my name. :) Try German, if you can - my mother used to it. :)

Snail-Mail: Arteckovskaya 7-4-260, Moscow, Russia.