

(draft)

# ERRP/0.6: Experiment Resource Reservation Protocol Ver. 0.6

Ken-ichi Chinen, IRC, JAIST; HRC, NICT.  
Toshiyuki Miyachi, HRC, NICT; IRC, JAIST.

May 21, 2008.

This article describes ERRP (Experiment Resource Reservation Protocol) version 0.6. The protocol is used between RM (Resource Manager) and its client.

## 1 Overview

Experiment resource reservation protocol (ERRP) aims reservation and maintainance of resources for network experiments — hardware equipments (PC, switch, and others) and logical items (VLAN number, various network service and others). The protocol is a connection oriented protocol on TCP/IP, and it is designed for client/server style. Client and server communicate over the connection. Server is a program for management of resources, Resource Manager (RM). Client reserves resources of manager and controls those resources. So, client is a driving program of network experiments. We call it 'driver'.

Driver(client) and manager(server) communicate over the connection. Communication consists of pairs of request by driver and its response by manager. Driver requests retrievals of resource information, appending new resources, reservation of those resources.

By this protocol, you can do exclusive locks for resources. The driver use favorite resources for its experiment with search.

## 2 Terminology

**Resource:** It is most important thing both logical and physical.

**Category:** A category of resources. We expect 'node', 'vlan', 'switch', 'switch-port' and 'service'.

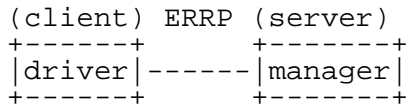
**(RM) Resource Manager:** A server program of ERRP.

**(ED) Experiment Driver:** A client program of ERRP.

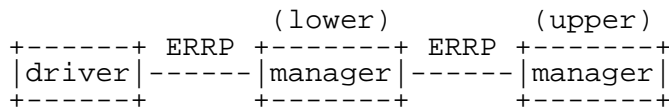
(ENCD) Experiment Node Configuration Driver: same as ED.

### 3 Connections

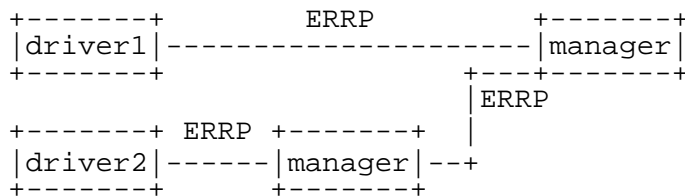
ERRP is a connection oriented protocol. Driver(client) and manager(server) communicate over the connection. Following figure depicts client, server and connection.



In case of using multi-user experiment facility or experiments with multi-facilities, the experiment requires to use several managers. Managers are connected serially. Closer manager to driver is called 'lower'. the other one is called 'upper'.



Manager can handle multiple client. Chain of ERRP often forms a tree structure. However, each client runs independently. Client does not care other clients.



### 4 Name Space

Since resources are stored in multiple managers, the way to identify resource is required. According to accant manner of e-mail and netnews, "!" separates resource and manager.

```

foo!A      resource 'A' on upper manager 'foo'.
C          if the manager has private resource 'C', it express this.
           otherwise, it express the resouce of upper manager.

```

Manager applies resource name resolving from closer to further. At first, the manager scan the resource into the resource database itselfs. When the resouces are found, the manager handle it. Otherwise (not found), the manager bypasses such command tu upper manager.

For example, from the view point of client 'apple' in below figure, left IDs in following samples mean right resources.

```

x          => bar!x
!x         => bar!x
y          => bar!y

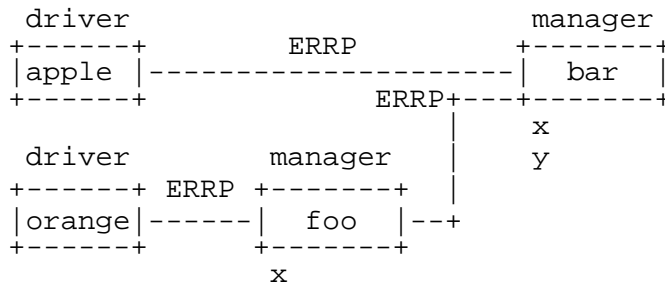
```

From that of 'orange', "!" effects to identify managers.

```

x          => foo!x
!x         => foo!x
y          => bar!y
bar!x     => bar!x

```



## 5 Resource Category

Resource manager have to handle various type resources. Then, the program handles using category.

category	description
cat	categories
conn	connection for upper managers
node	node; PC, PDA and others
switch	network switch (or router)
swport	network switch port
vlan	VLAN-ID; the number for VLAN
service	services; WoL and others

## 6 Property of Resource

Except following several literals, ERRP related program can use any literals as property of resources. Following literals are reserved properties:

name	name of the resource; stores string
owner	ownership of the resource; stores user and project
user	user of the resource; stores user and project
state	state of resource; depend on category

## 6.1 Node Properties

Currently, following properties are discussed and appeared. Most of them are hardware specification.

required	<b>owner</b>	ownership; stores user and project
	<b>name</b>	name; stores string
	<b>state</b>	state. see Section 7
recommended	<b>if-N</b>	N-th network interface
	<b>n_if</b>	the number of network interfaces
optional	<b>helth</b>	methods for helth check; IPMI, ICMP, SNMP and others
	<b>power</b>	methods for power control; IPMI, WoL, SNMP and others
	<b>n_mc</b>	the number of managment card
	<b>mc-N</b>	N-th managment card; IPMI, iLO and others
	<b>bootdisk</b>	bootdisk; IDE, SCSI and others
	<b>n_cpu</b>	the number of CPU
	<b>cpu-N</b>	N-th CPU
	<b>memsize</b>	the size of memory

For example, some manager holds following properties for node.

```
info node sintclf001
201 Okay
name: sintclf001
diskhint: IDE
bootdisk: IDE
Helth: ICMP,SSH,IPMI
Power: SNMP-NECMIB,IPMI
n_if: 6
n_experiment_if: 4
if-0: type=manage media=GigabitEthernet MAC='00:14:85:38:A2:66' \
phy-port='silaswc001,1/1' IP-addr=172.16.4.1
if-1: type=empty media=GigabitEthernet MAC='00:14:85:38:A2:67' \
phy-port='' IP-addr=0.0.0.0
if-2: type=experiment media=GigabitEthernet MAC='00:0E:0C:A7:81:0E' \
phy-port='silaswc002,1/1' IP-addr=0.0.0.0
if-3: type=experiment media=GigabitEthernet MAC='00:0E:0C:A7:81:0F' \
phy-port='silaswc002,2/1' IP-addr=0.0.0.0
if-4: type=experiment media=GigabitEthernet MAC='00:0E:0C:85:BE:00' \
phy-port='silaswc002,3/1' IP-addr=0.0.0.0
if-5: type=experiment media=GigabitEthernet MAC='00:0E:0C:85:BE:01' \
phy-port='silaswc002,4/1' IP-addr=0.0.0.0
n_mc: 1
mc-0: type=IPMI conn=Override MAC='' phy-port='' IP-addr=172.16.4.1
use: 1
owner: undef
user: undef
state: pooled
.
```

Moreover, location, country, price and other things are expected as properties. This framework allows adding of such properties when programs want to use them.

## 7 Node State

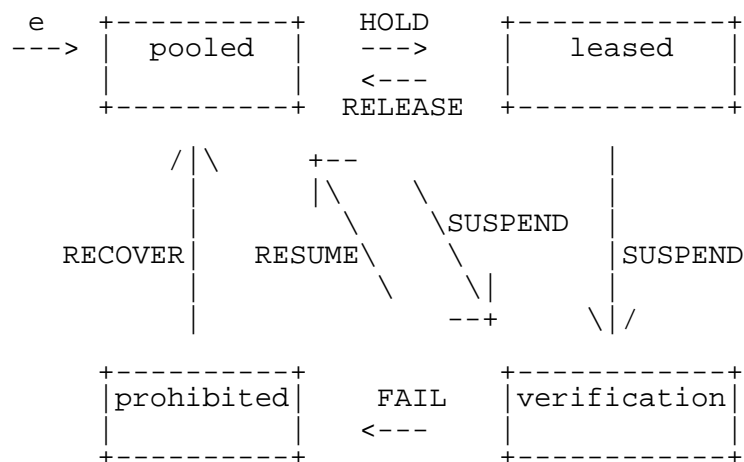
RM sets node's state according to client's request. Following 4 states are defined.

- pooled (idel, free, helth)
- leased (reserved)
- verification (meybe sick)
- prohibited (dead)

'leased' and 'pooled' means user occupied or not. Since node often broken, administrator have to verify its health. 'verification' means 'wait to administrator check' and/or unreliable. When administrator authorize the node to broken, the state of the node changes 'prohibited'. Otherwise, administrator shifts it to 'pooled'.

Following figure depicts transition of these states with client commands as input.

Note RESUME, FAIL and RECOVER commands are available for only administrator. See Section 11.



## 8 Communication Syntax

### 8.1 Generic Rule

```
<string> ::= [_A-Za-z0-9]+
<abstring> ::= [_A-Za-z][_A-Za-z0-9]*
<number> ::= [0-9]+
<white> ::= (" " | "\t")+
<nl> ::= "\n" | "\r\n"

<name> ::= <abstring>
<id> ::= <name> | <number> | <name> "!" <name>
<id-list> ::= <id-list> ", " <id> | <id>
<host> ::= <FQDN> | <IP-addr>
<FQDN> ::= [A-Za-z0-9\.]+
<IP-addr> ::= [1-2]*[0-9]*[0-9]+\.[1-2]*[0-9]*[0-9]+\.[1-2]*[0-9]*[0-9]+\.[1-2]*[0-9]*[0-9]+
<port> ::= <number>
<cat> ::= "service" | "node" | "vlan"
        | "switch" | "swport" | "conn" | "cat"
```

### 8.2 Request

```
<request> ::= <command> <arg-list>
<arg-list> ::= <arg-list> <white> <argument> | <argument>
<argument> ::= <abstring> "=" <string>
            | <abstring> | <number>
```

### 8.3 Response

Response consists of response-code by 3 digits and response-body.

```
<response> ::= <even-response-code> <arguments> <nl>
            | <odd-response-code> <string> <nl> <prop-chunk> "." <nl>
```

When response-code is even number, response forms multi lines. And it terminates a line which start by period (.). Otherwise, odd number, the response is single line. Detail of responses appeared other sections.

## 9 Commands

Most commands require login procedure with USER and PASSWD. The procedure aims avoiding conflicts and collision of resource usage. Moreover, to identify experiment, PROJECT command is ready. Client must issue PROJECT after login (USER and PASSWD).

The number of commands that not required login are few. HELP is prepared for usefulness in manual operation. VER and SYST is useful for checking manager ability.

Administrators can issue RESUME, FAIL, RECOVER and PERMITRANGE .

Following table shows a summary of them.

<b>command</b>	<b>p</b>	<b>a</b>	<b>description</b>
VER			version of protocol
SYST			system version
HELP			print commands
NOP			no operation
USER			user
PASSWD			password for user
PROJECT	X		project
QUIT			leave
CONNECT	X		connect upper manager
LIST	X		list of entities
INFO	X		information, list properties of entity
DISLIKE	X		mark as dislike resource
LIKE	X		unmark as dislike resources
FIND	X		find entities
HOLD	X		hold entities (be leased)
FINDHOLD	X		find and hold entities (for atomic)
RELEASE	X		release entities (be pooled)
SUSPEND	X		enter verification from pooled or leased
RESUME	X	X	return pooled from verification
FAIL	X	X	enter prohibited from verification
RECOVER	X	X	return pooled from prohibited
REGIST	X		register new resource
RANGE	X		choice range
PERMITRANGE	X	X	assign range v.s. users
PERMITCHECK	X	X	check permission

## 9.1 Generic Information

```
VER
  100 ERRP/0.6
```

```
SYST
  100 ERM/0.7
```

```
HELP
  201 OK
  <description>
  .
```

```
NOP
  100 NOP
```

```
<description> ::= <description> <desc-line> | <desc-line>
<desc-line>   ::= [^\.].* <nl>
```

Description is a chunk of line which consist of any charactors expect period starts. Because period indicates the end of response.

## 9.2 Login

```
USER <user>
  200 OK
  210 please send password
```

```
PASSWD <passwd>
  200 OK
  400 ignore
```

```
PROJECT <project>
  200 OK
  400 ignore
```

```
<user>      ::= <abstring>
<passwd>    ::= <string>
<project>   ::= <abstring>
```

## 9.3 Leave

```
QUIT
  100 OK
```

Manager closes the connection without response. After response manager may disconnect immediately. Some client program could not receive response.

Note QUIT does not means resource release. Because resource reservation is indepent from connection. Thus, drivers have to release those resources by RELEASE command.

## 9.4 Connect Upper Manager

```
CONNECT <manager-name> <host:port>
      <manager-name> ::= <abstring>
```



## 9.5 Retrieval of Resource Information

```
LIST <cat> <state>
LIST project
  201 OK
  node1
  ...
  nodeN
  .

400 ignore

INFO <cat> <id>
  201 Ok
  <prop-chunk>
  .

400 not found/ignore
```

```
<prop-chunk> ::= <prop-chunk> <prop-line>
<prop-line>  ::= <abstring> ":" <white> <prop-v-list> <nl>
<prop-v-list> ::= <prop-v-list> <white> <prop-v>
                | <prop-v-list> "," <prop-v>
                | <prop-v>
<prop-v>    ::= <string>
```

## 9.6 Find Resources

```
FIND <cat> <conds>
  201 Ok
  node1
  ...
  nodeN
  .

400 ignore request

410 no idle node

414 not enough idle nodes

420 no matched node
```

'conds' means conditions. The number of require node was specified like 'num=13'.

It is means single node that you not specified the number.

The condition about network interface is little complex. If you require fastethernet, you have to use 'if[media=fastethernet]'. If you don't care media type, only said 'if'.

```
<conds>    ::= <conds> "," <cond> | <cond>
<cond>    ::= "num" "=" <number>
                | "bootdisk" "=" ("IDE" | "SCSI")
                | "if"
                | "if" "[" <ifqlist> "]"

<ifqlist> ::= <ifqlist> "," <ifq> | <ifq>
<ifq>     ::= "media" "=" <string>
                | "type" "=" <string>
```

### 9.6.1 Example of FIND

```
find node num=4
```

— want any 4 nodes

```
find node num=9,if[media=fastethernet,type=experiment]
```

— want 9 nodes with 1 interface for experiment network

```
find node num=7,if[media=atm],if[media=fastethernet],if
```

— want 7 nodes with 3 interface(atm, fastethernet and any).

```
find vlan num=4
```

— want any4 VLANs

### 9.6.2 Dislike/like

Sometime, clients have dislike resources in some reason. The client can regist dislike resources by DISLIKE. The server should skip these resources when it replies in FIND or FINDHOLD.

```
DISLIKE <cat> <id>
200 Ok
210 Sure
```

```
LIKE <cat> <id>
200 Ok
210 Sure
```

Using LIKE, the client can cancel the effects of DISLIKE.

### 9.6.3 Algorithm

FIND dicides response according to following.

```
clear(cand-list)
foreach r <permitted resources> {
    if(<r is dislike>) {
        continue
    }
    if(<r is leased>) {
        continue
    }
    if(<r match conditions>) {
        add(cand-list, r)
    }
}
reply(cand-list)
```

## 9.7 State Transition

```
HOLD <cat> <id-list>
  200 Ok
  210 Ok some items are holded already

  400 ignore

  401 ignore
  node1
  ...
  nodeN
  .

  410 busy

  420 reserved

FINDHOLD <cat> <conds>
  201 Ok
  node1
  ...
  nodeN
  .

RELEASE <cat> <id-list>
  200 Ok
  210 Ok some items are released already

  400 ignore

  401 ignore
  node1
  ...
  nodeN
  .

  410 free entity

  420 not permitted, ownership misssmatch
```

Release all resources for this project.

```
RELEASE project
  200 OK
  400 ignore
```

Release all resources for this user.

```
RELEASE user
  200 OK
  400 ignore
```

```

SUSPEND <cat> <id>
  200 OK, to be verification
  410 unknown node
  420 that node is not 'pooled'
  430 you are not permitted

RESUME <cat> <id>
  200 OK, to be pooled
  410 unknown node
  420 that node is not 'verificationed'
  430 you are not permitted

FAIL <cat> <id>
  200 OK, to be prohibited
  410 unknown node
  420 that node is not 'verificationed'
  430 you are not permitted

RECOVER <cat> <id>
  200 OK, to be pooled
  410 unknown node
  420 that node is not 'prohibited'
  430 you are not permitted

```

## 9.8 Regist New Resource

```

REGIST <cat> <id>
<prop-chunk>
.
  200 Ok
  400 ignore properties
  402 broken request
      timeout or does not terminated
  410 ignore id

```

## 9.9 Access Control

```

RANGE <cat> <id-list>
  200 OK
  400 ignore
  401 unknown node
  node1
  ...
  nodeN
.

PERMITRANGE <cat> <user> <id-list>
  200 OK
  400 ignore
  410 ignore user
  401 unknown node
  node1
  ...
  nodeN
.

```

## 10 Sample of Command Sequence

### 10.1 Using Several Nodes

```
SYST
VER
USER foo
PASSED bar
PROJECT hawaii
LIST node pooled
INFO node n13
HOLD node n7,n43,n12,n3
FIND node num=4,if[media=fastethernet],if
{do experiment(s)}
RELEASE node n7,n43,n12,n3
QUIT
```

Use node-'n54' according to manager's oracle.

```
{login}
FINDHOLD node num=4,if[media=fastethernet],if
{use n54 as any role}
RELEASE node n54
QUIT
```

### 10.2 Releasing When You Meet Unexpected ENCD Trouble

RM is daemon program. So, the program must run anytime. If the contents of the program is broken, the program will restart. However, clients (almost ENCD) disconnect suddenly in rare case. In such case, RM is not wrong but the consistency of contents is broken. Since the client is down, user do not know what nodes are reserved to last project. Then, a releasing method without nodename is required.

```
{connect}
USER foo
PASSWD bar
PROJECT hawaii
HOLD node n3
{disonnnect peer}

{re-connect}
USER foo
PASSWD bar
PROJECT hawaii
RELEASE project
{run experiment, again}
HOLD node n3
```

## 11 Administrator

User 'admin' is a special user. He/she can issues special commands, like following:

```
RESUME
FAIL
RECOVER

PERMITRANGE
```

## 12 Literals

### 12.1 Interface Media

Following terms are defined for query.

ATM	ATM; asynchronous transfer mode
Ethernet	Ethernet (10Mbps)
FastEthernet	Fast Ethernet (100Mbps)
GigabitEthernet	Gigabit Ethernet (1Gbps)
10GigabitEthernet	10 Gigabit Ethernet (10Gbps)

FastEthernet means 100Mbps Ethernet. It does not care detail (e.g., 100BASE-TX, 100VG-ANYLAN and others).

### 12.2 Interface Type

Following terms are defined for query.

manage	managment network (probably 1 per host)
experiment	experiment network
empty	open circuit

## A History

ERRP was called SBRP (StarBED Resource Protocol) previously. Design and implementation of SBRP starts October 2003 by K. Chinen.

```
SBRP/0.1    early of Oct 2003
SBRP/0.2    Oct 14, 2003
SBRP/0.3    unknown
SBRP/0.4    Dec 2003
SBRP/0.5    Oct 2004
```

We change name SBRP to ERRP because the protocol has no restriction for StarBED. Most network testbeds can use that.

```
ERRP/0.6    just working
```

## B Changing from ERRP/0.5

- Registration of resource is available.
- Introduce the chain of managers and its name space.
- Introduce the concept of resource categories.
  - adding new category 'service', 'switch' and 'swport'.
- Merge the name of node states to recent papers.
  - 'verification' and 'prohibited'.
- Add properties 'n.cpu' and 'cpu-N'
- Add response-code 210 as positive ack for 'HOLD' and 'RELEASE'
- Add command 'DISLIKE' and 'LIKE'
- Add response-code 210 as positive ack for 'DISLIKE' and 'LIKE'
- Arrange response codes of REGIST

## C Changing from SBRP/0.4

**added command:**

```
USER
PASSWD
PROJECT
SUSPEND
RESUME
FAIL
RECOVER
LIST PROJECT
RELEASE PROJECT
RELEASE USER
RANGE
PERMITERANGE
```

**removed command:**

```
JOIN
```

## D People

Current ERRP is a result of corroboration of following people:

Ken-ichi Chinen\*  
Toshiyuki Miyachi  
Makoto Misumi  
Naoki Isozaki\*  
Shinsuke Miwa

Persons listed with star, joined the making of the reference implementation, **sbrm** and **erm**.