I'd like to add a method: Socket.getifaddrs.  

This method is a wrapper to getifaddrs() function. The result is an array of instances of Socket::Ifaddr class as follows.

```ruby
% ./ruby -rpp -rsocket -e 'pp Socket.getifaddrs'
[#<Socket::Ifaddr lo UP,LOOPBACK,RUNNING,0x10000 [PACKET protocol:0 lo hatype:772 HOST hwaddr:00:00:00:00:00:00]>, #<Socket::Ifaddr eth0 UP,BROADCAST,RUNNING,_MULTICAST,0x10000 [PACKET protocol:0 eth0 hatype:1 HOST hwaddr:00:16:3e:95:88:bb] broadcast:[PACKET protocol:0 eth0 hatype:1 HOST hwaddr:ff:ff:ff:ff:ff:ff]>, #<Socket::Ifaddr sit0 NOARP [PACKET protocol:0 sit0 hatype:776 HOST hwaddr:00:00:00:00:00:00]>, #<Socket::Ifaddr lo UP,LOOPBACK,RUNNING,0x10000 [127.0.0.1] netmask:[255.0.0.0]>, #<Socket::Ifaddr eth0 UP,BROADCAST,RUNNING,_MULTICAST,0x10000 [221.186.184.67] netmask:[255.255.255.240] broadcast:[221.186.184.79]>, #<Socket::Ifaddr lo UP,LOOPBACK,RUNNING,0x10000 [::1] netmask:[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]>, #<Socket::Ifaddr eth0 UP,BROADCAST,RUNNING,_MULTICAST,0x10000 [fe80::216:3eff:fe95:88bb%eth0] netmask:[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]>]
```

This method can be used to choose multicast-enabled interfaces. Multicast applications needs to identify a such interface to sending a multicast datagram.

```ruby
% ./ruby -rpp -rsocket -e 'pp Socket.getifaddrs.reject {|ifaddr| !ifaddr.addr.ip? || (ifaddr.flags & Socket::IFF_MULTICAST == 0) }.map {|ifaddr| [ifaddr.name, ifaddr.ifindex, ifaddr.addr] }'
```

This method can be used to obtain broadcast addresses to send a broadcast messages.  

Also, the broadcast addresses can be used to receive broadcast messages.  

getifaddrs() is not standardized but many platforms have.

**BSDI, FreeBSD, NetBSD, OpenBSD, DragonFly, MirOS, GNU/Linux, MacOS X, SunOS 5.11 (OpenIndiana), Cygwin 1.7.15**

From GnuLib document, getifaddrs() is not exist on  

AIX 5.1, HP-UX 11, IRIX 6.5, OSF/1 5.1, Solaris 10, Cygwin 1.5.x, mingw, MSVC 9, Interix 3.5, BeOS.


I implemented new Socket::Ifaddr class for return value of Socket.getifaddrs, instead of constructing the result from currentn data structures such as Array. This choice make us possible to implement Socket::Ifaddr#inspect to show flags as its names (UP,...). Also, struct getifaddrs contains Address-specific data, ifa_data which may be extended by platforms. When we find a way to extract some data from ifa_data, we can add a method for that.

Socket::Ifaddr is different from Socket::Interface by  


They have no one-to-one mapping.  

For example, Socket.getifaddrs returns two or more elements for one interface.

Associated revisions
Revision b323d7d5 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@40639 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

Revision 40639 - 05/11/2013 08:32 AM - akr (Akira Tanaka)

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

History

#1 - 05/06/2013 12:00 AM - naruse (Yui NARUSE)
- Target version set to 2.1.0

I'm for this method.

#2 - 05/06/2013 01:23 AM - judofyr (Magnus Holm)

Shouldn't this be moved to CommonRuby?

#3 - 05/06/2013 01:40 AM - naruse (Yui NARUSE)

judofyr (Magnus Holm) wrote:

  Shouldn't this be moved to CommonRuby?

There's no such consensus yet, especially bundled libraries.

#4 - 05/06/2013 08:33 PM - akr (Akira Tanaka)

- File getifaddrs2.patch added

I updated the patch to fix compile error on environments which has no getifaddrs().
Also I refined the result of Socket::Ifaddrs#inspect.

% .ruby -ssocket -rpp -e 'pp Socket.getifaddrs'
[<Socket::Ifaddr lo UP,LOOPBACK,UNINITIAL,0x10000 PACKET]protocol=0 lo hatype=772 HOST hwaddr=00:00:00:00:00:00>,
[<Socket::Ifaddr eth0 UP,BROADCAST,UNINITIAL,0x10000 PACKET]protocol=0 eth0 hatype=1 HOST hwaddr=00:16:3e:95:88:bb]
#5 - 05/11/2013 05:32 PM - akr (Akira Tanaka)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r40639.
Akira, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- ext/socket: New method, Socket.getifaddrs, implemented.
  [ruby-core:54777] [Feature #8368]

#6 - 05/11/2013 05:33 PM - akr (Akira Tanaka)

It seems no one against for this method and naruse-san (the release manager of Ruby 2.1.0) supported.
I added Socket.getifaddrs.

Files
<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>getifaddrs.patch</td>
<td>31.6 KB</td>
<td>05/04/2013</td>
<td>akr (Akira Tanaka)</td>
</tr>
<tr>
<td>getifaddrs2.patch</td>
<td>32.7 KB</td>
<td>05/06/2013</td>
<td>akr (Akira Tanaka)</td>
</tr>
</tbody>
</table>