

Multi-address Interface in Socket API

draft-sarolahti-mptcp-af-multipath-01

Pasi Sarolahti

IETF-77, Anaheim, CA, USA

2010-03-26

Why Change Socket API?

- Socket API has stood the test of time well
 - Small set of generic operations
 - Serves various different applications in addition to usual transport protocols:
 - Unix domain, IPsec key management, routing, ...
- Leverage existing API to the extent possible with multi-address sockets
 - Keep the API generic
 - Don't change functions, change address format
 - Keep socket options as *options*

Proposed Approach

- Express addresses in the same place they have always been expressed: **socket address field**
- Address format indicates multipath support
 - AF_INET, AF_INET6: application does not understand multiple addresses per socket
 - AF_MULTIPATH: application understands multiple addresses per socket (and supports MPTCP)
- Kernel function interface remains unchanged

AF_MULTIPATH

- New address family
- Collects together multiple addresses
 - Possibly different address families
- No changes to socket operations

Len: 40	AF_MULTIPATH
Adrs: 3	
Len: 8	AF_INET
128.214.4.64,80	
Len: 8	AF_INET
193.229.9.132,80	
Len: 20	AF_INET6
2001:708:10:55::1234,80	

Usage

- Use normal socket calls as before

not compiled, not tested

```
hp = gethostbyname2("name.com", AF_MULTIPATH);  
/* if hp == NULL, retry with AF_INET */  
s = socket(PF_INET, SOCK_STREAM, NULL);  
i = connect(s, hp->h_addr, hp->h_length);
```

- Protocol MUST ignore addresses that point to different hosts
- Normal operation remains similar, but
 - Current list of addresses can be obtained using *getsockname*, *getpeername*
 - New addresses can be set using *connect*, *bind*

Discussion

- Applications can ignore socket address content
- Backwards compatibility
 - If application supports MPTCP, it uses AF_MULTIPATH
 - If stack does not support AF_MULTIPATH, error is returned
 - application can retry with traditional address families
- Could support IPv4/IPv6 migration
 - With AF_MULTIPATH, single gethostbyname2 call could trigger both A and AAAA DNS queries
- Differences to past semantics
 - Address content can change during socket lifetime
 - Connect and bind can be called multiple times for same socket

Next steps?

- Comments on overall idea?
- What are the follow-up actions, if any?
- Note:
 - Does not prevent enabling MPTCP with traditional address families
 - Is not specific to MPTCP