Bibliography on Persistent Network Connection

Keywords: Persistent Connection, TCP mobility, reliable network connections, TCP failover, Application Mobility, Session Mobility, Connection Migration

Chakchai So-In (cs5@cs.wustl.edu)
October 10th, 2006

Issues on Persistent Network Connection:

1. Client or Server moves (same process and both processes are running)
- Virtualized connections (Indirect layer):
  - Port mapping (virtual socket)
- Location Management:
  - Dynamic DNS, Rendezvous HIP Extension, SIP Mobility
- Connection Identifier (Authentication):
  - Diffe-Hellman key exchange
- Maintaining TCP connection states (seq/ack synchronization and data consistency):
  - TCP Freeze, TCP SUSPEND state, Circular-Buffer/Double Buffer, Go-back-N data retransmissions
- Control Signaling (key exchange, transfer state information):
  - TCP Option (inband), UDP protocol (outband)

2. Client or Server moves (different process and process may be up/down or migrate to another process)
- Checkpointing/ State Logging
- TCP Fault Tolerance/Fail over:
  - Backup Server running promiscuous mode

3. TCP handoff issues
- Single connection: MobileIP (within TCP timeout)
- Different connection:
  - TCP handoff for persistent connection
  - IP diversity: M-SCTP, DCCP, SIGMA, Cellular SCTP, pTCP (multi-homing)

Evaluation Criteria
- Application transparency
- Kernel Modification/ TCP stack modification
- Change in infrastructure
- IP diversity
- NAT/PAT supported

General problem:
- Persistent connection using Mobile SCTP/ SCTP migration option, internal splice connection, TCP handoff with long term disconnection (maintain the continuity)
- Transport Mobility Tunnelling
- Analyse location management schèmes etc. Secure Dynamic DNS/ SIP
- Transport layer approach on QOS

1. Victor C. Zandy and B.P. Miller, *"Reliable network connections,"* In Proceedings of the 8th ACM Intl. Conf. on Mobile Computing and Networking (MobiCom), pages 95--106, Atlanta, Georgia, USA, September 2002. http://www.cs.wisc.edu/~zandy/rocks/

2. Victor C. Zandy, *"Application Mobility",* Ph.D. dissertation, University of Wisconsin-Madison, 2004. http://www.paradyn.org/html/publications-by-category.html

Victor C. Zandy at WISC proposed reliable sockets (rocks) and reliable packets (racks). These schemes provide the transparent network connection mobility on user-level. These techniques support moving to another network (recover broken connections without loss of in-flight data), unexpected link failure, and laptop suspension. Enhancement Detection Protocol and timeout mechanism was proposed in order to enable remote detection for ordinary interoperation.

Rock uses *heartbeat probe* mechanism is used to detect link failure (outband UDP control connection). Rock layer is inserted below application and above transport layer. The buffer is created in order to synchronize in-flight buffer (TCP send buffer + size of TCP peer recv buffer). There are three more states for rock operations; CLOSED, CONNECTED, and SUSPEND. Once connected, the application use rock library as an ordinary socket library. If the link is disconnected, rock will move to SUSPEND state, and rock will attempt to reconnect automatically. Go-back-N retransmission is performed for any data that was in-flight during link failure. As the technique in TCP Migrate Option, the Diffe-Hellman key exchange protocol is used to authenticate the connection.

Theses schemes also support long term suspension, rock defines ABORT whether the application really want to end the connection or the connection is terminated because of the disconnection. With existing process checkpoint, these schemes can also handle parallel program such as MPI and PVM. With network proxy, these schemes can also support the fault tolerant of link failure.

The limitation of rock is that it can not perform properly if the other end has moved during the period of disconnection, it's blocked from establishing a new connection by a firewall, or the last know address was masqueraded by a NAT.

Rack uses a packet filter to insert itself in the flow between local socket and the network in order to prevent local TCP socket from aborting due to connection failure. Once the link is disconnected, rack sends to the local socket a zero-window packet in order to make the application stop sending the data. And window update is sent to resume operation when link is reconnected. The limitation of this is that since it can not change the binding of kernel sockets to other process, it will allow sockets to abort, so it could not recover the connection. Racks use TCP keepalive mechanism to detect link failure. The general ideas of Racks is almost the same as TCP splice, MSOCK (with proxy) and TCP Migration Option but it can support long term disconnection.

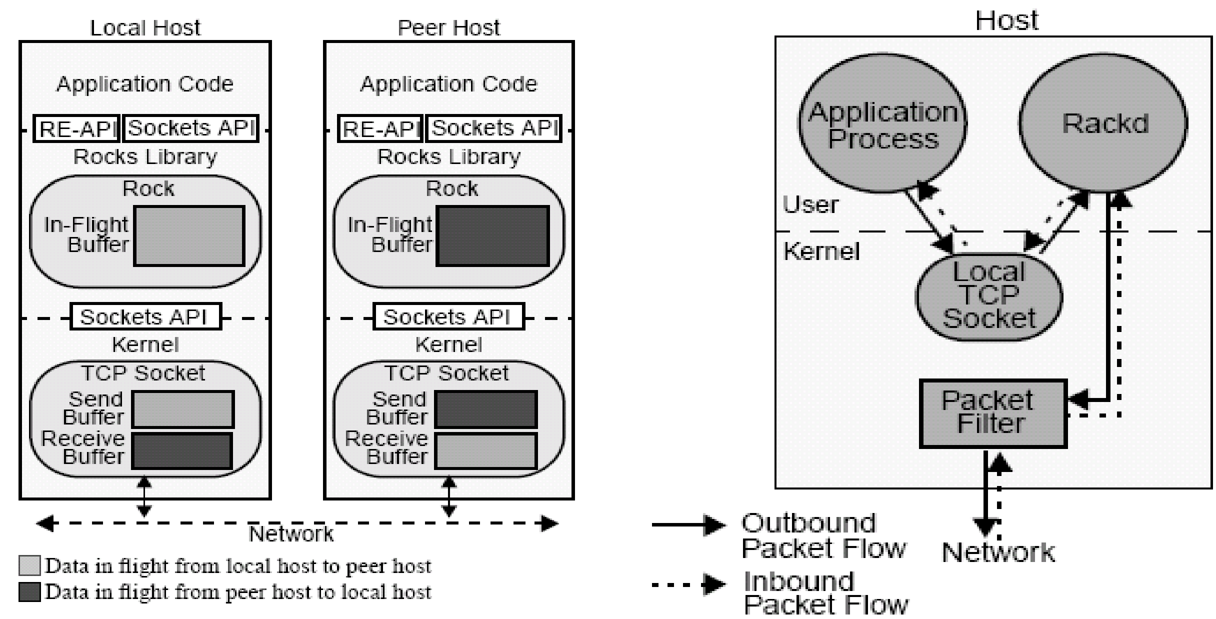


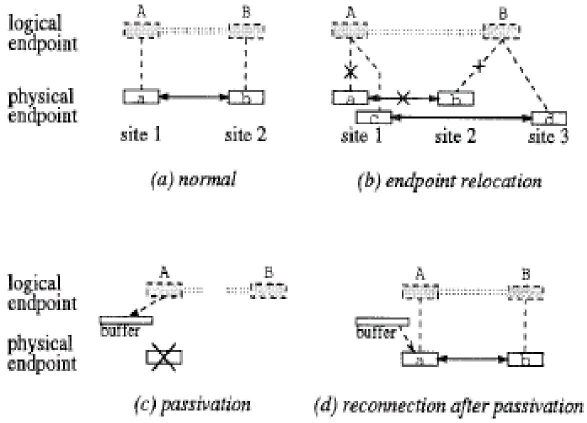
Figure 1.1 a) Rock architecture b) Rack architecture

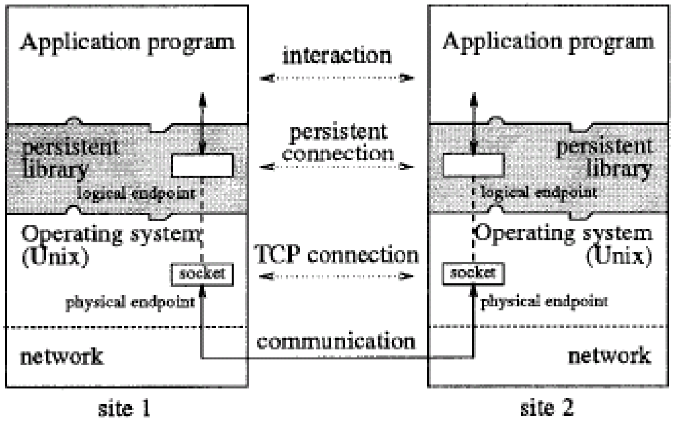
- Implementation: Linux and Window
- Performance: The overhead of racks is apparent on slower links, but not the overhead of rocks.

3. Yongguang Zhang and Son Dao, *"A "Persistent Connection" Model for Mobile and Distributed Systems,"* 1995. http://citeseer.ist.psu.edu/zhang95persistent.html

This paper describes the model of persistent connection for stream connection between two persistent processes. This model supports the temporary disconnection. This technique is to hide the disconnection and endpoint relocation to provide the process a virtual, always connected communication session as the logical endpoint vs. physical endpoint. The author also describes a address scheme in order to hide the change of address from the application. The virtual address has three tuples, (socket descriptor, IP, and Port such as s, 0.0.0.15:3). This address has the format as a typical IP address scheme. 0.x.x.x represent a persistent process, x.x.x is the persistent process id. The application program use this logical endpoint (virtual address) in the socket address and the persistent library will call the real destination IP address. The application will see only the virtual address although the real IP address is changed. The physical endpoint FIFO buffers (non-blocking write) or blocks (blocking write) the operation if there is no existed connection. The library acts as the logical-physical mapping.

- Implementation: Libps (persistent library)
- Performance: N/A

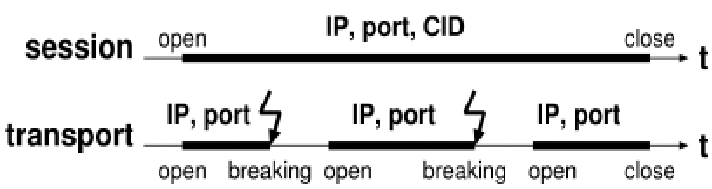


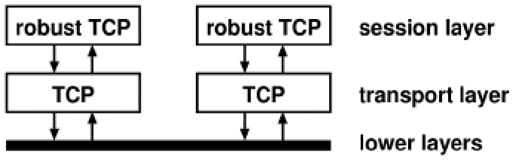


4. Richard Ekwall, Piter Urban, and Andri Schiper, *"Robust TCP Connections for Fault Tolerant Computing,"* ICPADS 2002. http://citeseer.ist.psu.edu/ekwall03robust.html

This paper proposes a new TCP scheme for TCP fault tolerance when TCP breaks the connection due to the some duration disconnection. In this scheme, there is timeout involved. So the connection seems to be connected as long as the application does not terminate the connection. The new layer is inserted on top of transport layer, session layer. This session layer hides the disconnection from the application layer. A unique connection identifier (CID) was introduced in order to identify a connection. It is chosen randomly by server and keeps increasing linearly (k++: 32 bits wrap around). When the link is resumed, new TCP connection is reestablished with the CID and the control information. The control message is UDP message which contains TCP states information such as next requested byte.

- Implementation: Java Library
- Performance: Less than 10% overhead





5. Alex C. Snoeren, *"A Session-Based Architecture for Internet Mobility,"* Ph.D. dissertation, Massachusetts Institute of Technology, December 2002. http://nms.csail.mit.edu/projects/migrate/

6. Alex C. Snoeren and Hari Balakrishnan, *"An End-to-End Approach to Host Mobility,"* In Proceedings of the 6th ACM MobiCom, August 2000. http://nms.csail.mit.edu/papers/migrate.html

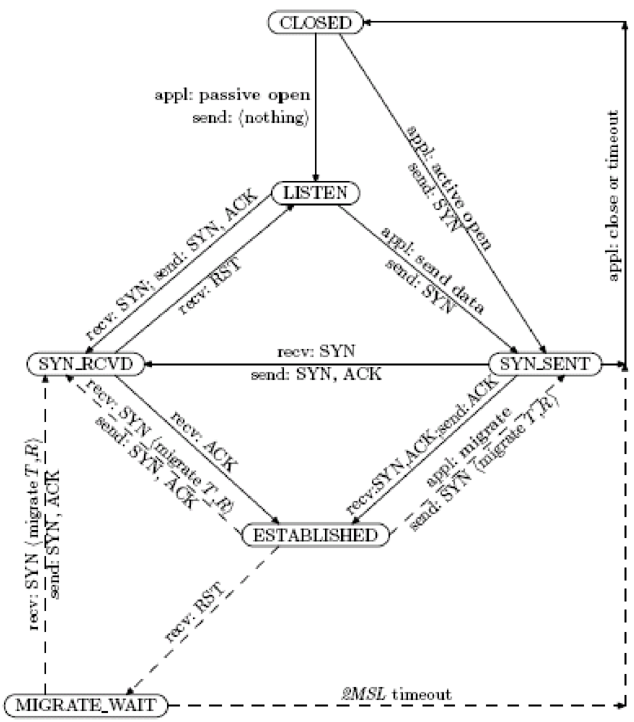
7. Alex C. Snoeren and Hari Balakrishnan, *"TCP Connection Migration,"* draft-snoeren-tcp-migrate-00.txt, November 2000. Work-in-progress, expired May 2001. http://nms.csail.mit.edu/papers/migrate.html

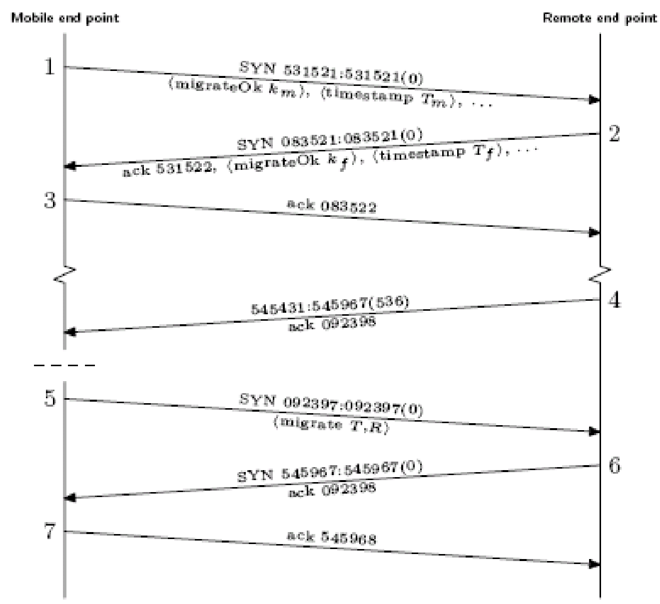
8. Alex C. Snoeren, David G. Andersen, and Hari Balakrishnan, *"Fine-Grained Failover Using Connection Migration,"* In Proceedings of the 3rd USENIX USITS, March 2001. http://nms.csail.mit.edu/papers/migrate.html

9. Jon Salz, Alex C. Snoeren, and Hari Balakrishnan, *"TESLA: A Transparent, Extensible Session-Layer Architecture for End-to-end Network Services,"* http://nms.csail.mit.edu/papers/migrate.html

These papers proposed the technique to preserve active TCP connection. DNS dynamic updated is used to identify mobile host location. The author added MIGRATE_WAIT state to a typical TCP states. When link is disconnected, TCP state is moved to that state (no send or receive data). A unique token is used to identify each connection. This token can be negotiated by Elliptical Curve Diffie Hellman key exchange during the initial setup (normal TCP setup with TCP migrate option). When a client changes IP address or link is broken, the extra layer hides the broken link from transport layer. It stops TCP transmission (going to WAIT state) and freeze the application data (or buffer the application data). Byte sent and received numbers are also stored. It might have to buffer some intermediate data while link is broken. Once link is resumed, the client uses the unique token to resynchronize a previous TCP connection with the right data sent and received and move TCP WAIT state to normal state.

- End to end mobility (preserve TCP semantic)
- This technique can coexist with traditional TCP. The host with migrate feature will negotiate with the other end if TCP migration will be enabled. TCP normal operation will be used instead if the other end does not support.
- This technique is needed to modify TCP stack by inserting MIGRATE_WAIT state.
- Both client and server TCP stack are modified.
- FTP and SSH are tested.
- Limitation: Multi-party conferencing, gaming, and multi-cast based content distribution. Also, No process migration feature (assume the process is up during connection migration)
- Implementation: Linux (Test on SSH/FTP)
- Performance: With TCP Migrate options, Migrate's overhead is almost negligible compared to the connection virtualization techniques which are large if high bandwidth and small block sizes.





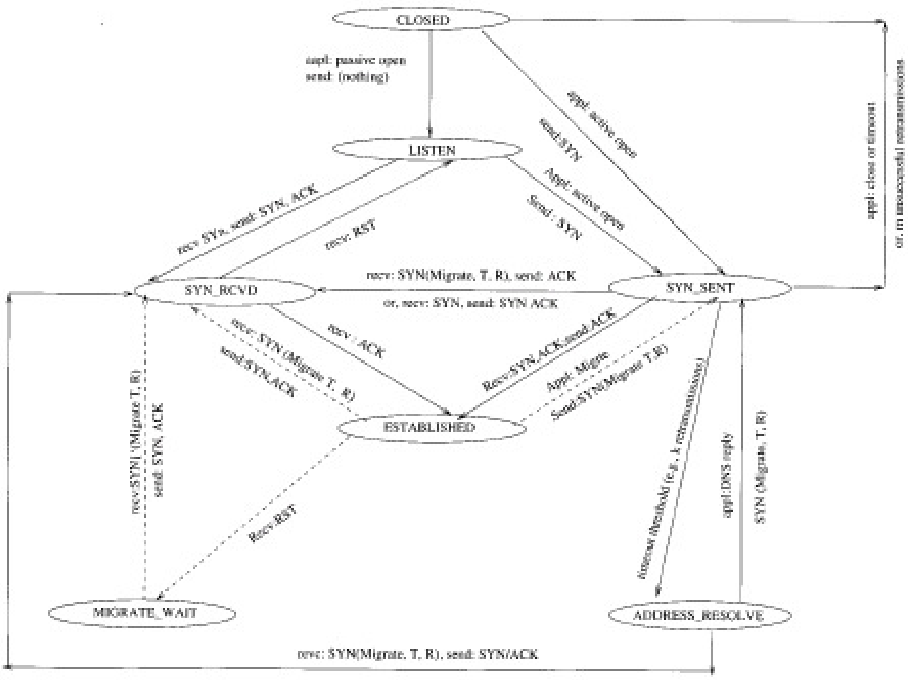
10. Alex C. Snoeren, Hari Balakrishnan, and M. Frans Kaashoek, *"Reconsidering IP Mobility,"* In Proceedings of the 8th HotOS, May 2001. http://citeseer.ist.psu.edu/438107.html

This paper describes the fundamental issues on mobility: locating the mobile host or service, reserving communication, disconnecting gracefully, hibernating efficiently, and reconnecting quickly. Also, the authors recommend good system mobility architecture such that eliminating the dependence of higher protocol layers upon lower-layer identifiers, working with any application-selected naming scheme, handling unexpected network disconnections, exposing its occurrence to applications, and providing mobility services at the mobile nodes not via proxies.

11. Sameer Tilak, Nael B, and Abu-Ghazaleh, *"A concurrent migration extension to an end-to-end host mobility architecture,"* Mobile Computing and Communications Review, 5(3):26–31, July 2001. http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=584053

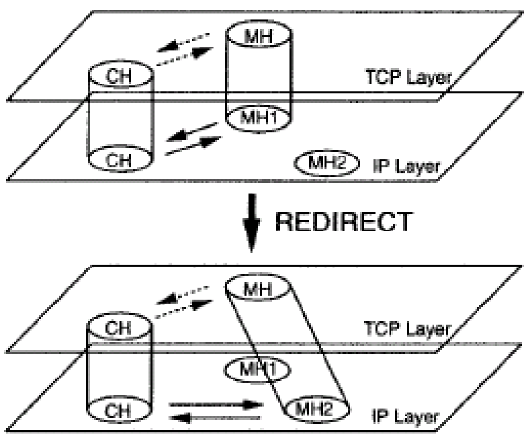
This paper proposed the extension of TCP migration technique by A.C. Snoeren. The extension is to support the concurrent migration of two end connections. ADDRESS_RESOLVE stage is introduced in order to determine if the peer has also moved (after *k* retransmissions). The stage is for resolving the new peer address by contact DNS server.

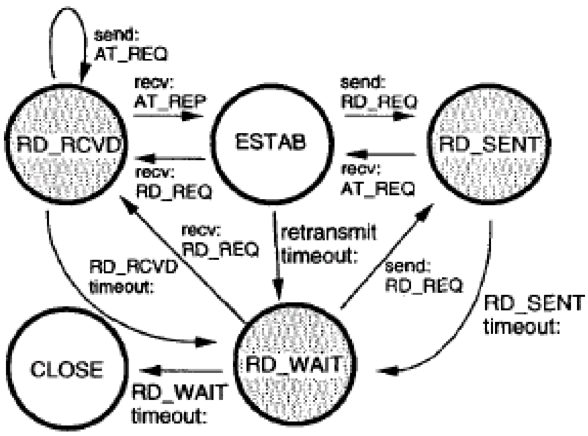
- No implementation yet.
- Race condition might be occurred.



12. Funato, D., Yasuda, K., and Tokuda, H., *"TCP-R: TCP mobility support for continuous operation,"* ICNP 1997. http://doi.ieeecomputersociety.org/10.1109/ICNP.1997.643720

This scheme is to maintain TCP connection during the change of IP address or network device. The idea of TCP redirection is to revise the pair of address during the change of IP address. New TCP options were introduced to check TCP-R compatibility, transfer the connection identifier, update the current IP address, and authentication purpose. Also, three more TCP states were added; RD_SENT, RD_RCVD, and RD_WAIT. RD_SENT state maintains the pair of address. RD_RCVD does the authentication. Once the connection faces the TCP transmission timeout problem which causes the connection tear down, TCP-R moves the state into RD_WAIT to let the host wait instead of drop the connection. When TCP-R detects the resumed connection, it moves the state back to RD_SENT state.





13. Tadashi Okoshi, Masahiro Mochizuki, Yoshito Tobe, and Hideyuki Tokuda , *"MobileSocket: Toward Continuous Operation for Java Applications,"* IEEE 8th International Conference on Computer Communications and Networks 1999. http://citeseer.ist.psu.edu/okoshi99mobilesocket.html

14. Teemu Koponen, Pasi Eronen, and Mikko Sarela, *"Resilient Connections for SSH and TLS,"* Helsinki Institute for Information Technology. http://infrahip.hiit.fi/papers/resilient.pdf#search=%22resilient%20connection%20for%20ssh%22

15. Zhong, X., Xu, C.-Z., and Shen, H., *"A reliable and secure connection migration mechanism for mobile agents,"* Distributed Computing Systems Workshops, 2004. In Proceedings. 24th International Conference, March 2004. http://www.ece.eng.wayne.edu/~czxu/paper/nsocket-mdc04.pdf

This paper describes the mechanism to make the connection migration reliable and secure for mobile agent called NapletSocket.. In the design, three main components; redirector, controller, and data socket, are used to make reliable connection.

mainly the author added SUSPEND and RESUME state transitions to typical TCP states

16. Ricardo Baratto, Shaya Potter, Gong Su, and Jason Nieh, *"MobiDesk: Mobile Virtual Desktop Computing,"* In Proceedings of the Tenth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, PA, September 26-October 1, 2004, pp. 1-15. http://www.ncl.cs.columbia.edu/research/move/

17. Gong Su and Jason Nieh, "*Mobile communication with virtual network address translation*," Tech. Report CUCS-003-02, Columbia University, February 2002. http://www.cs.columbia.edu/techreports/cucs-003-02.pdf#search=%22mobile%20communication%20with%20virtual%20network%22

18. Gong Su, *"MOVE: Mobility with Persistent Network Connections,"* Ph.D. dissertation, Department of Computer Science, Columbia University, October 2004. http://www.ncl.cs.columbia.edu/research/move/

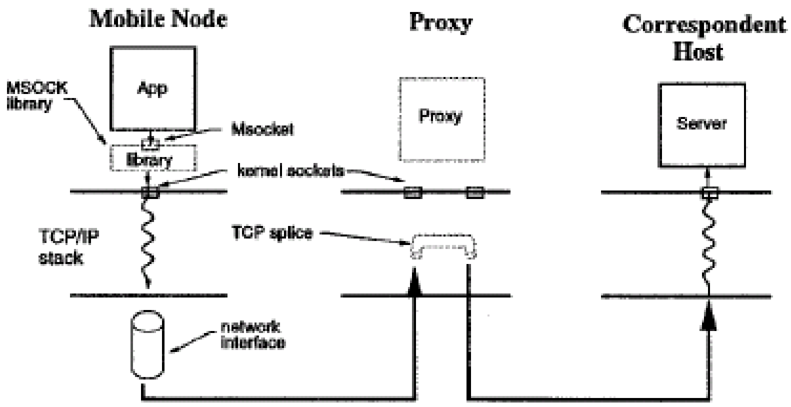
This thesis describes the seamless connection migration and process migration mechanism. A naming abstraction, ConnEcion virtualization and encapsulation (CELL) was introduced to represent the unique virtual name for the connection. Also, Host only HandOff was also proposed. Together with low-overhead security mechanism, the connection can migrate securely only in a single packet.

19. David A. Maltz and Pravin Bhagwat, *"MSOCKS: An Architecture for Transport Layer Mobility,"* INFOCOMM 1998. http://citeseer.ist.psu.edu/60438.html

20. Pravin Bhagwat, David A. Maltz, and Adrian Segall, *"MSOCKS+: an architecture for transport layer mobility,"* Computer Networks, July 2002 Volume 39 On pages: 385-403. http://www.cs.cmu.edu/~dmaltz/publications.html

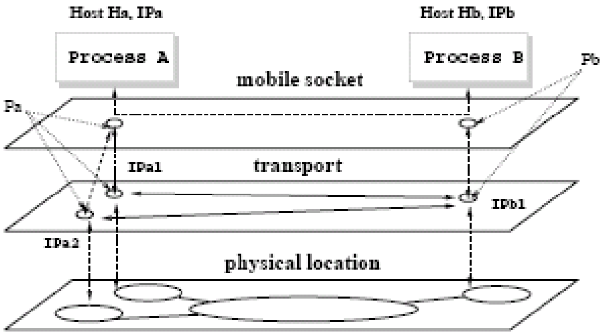
21. David A. Maltz and Pravin Bhagwat, *"TCP Splicing for Application Layer Proxy Performance,"* Journal of High Speed Networks, vol. 8, no. 3, 1999, pp. 235-240. http://www.cs.cmu.edu/~dmaltz/publications.html

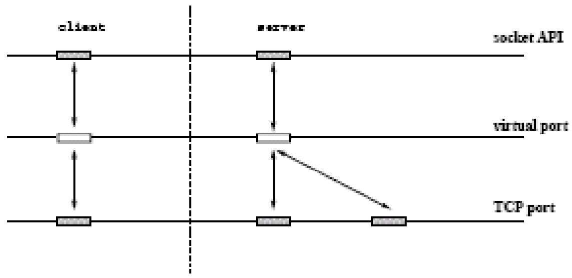
These papers describe how to use proxy to maintain the connection states (splitting TCP connection). The proxy can transparently splice the connection to alternate server where ensure the connection consistency (byte sent/received). The splice technique is to make it appear that there is only single direct TCP connection although there are two separate TCP connections. During the client disconnection, the proxy keeps the server connection open and splices the new client connection to that connection when the client link is resumed. In order to reconnect request, a unique connection ID is used to identify the connection. So the proxy can splice the connection to the previous connection by dynamic mapping. During the operation, the proxy acts as a relay, it does not send the acknowledgement packet on behalf of node; however, it maintains the TCP state synchronization.



22. Xun Qu, Jeffrey Xu Yu, and Richard P. Brent, *"A mobile TCP socket,"* Technical Report TR-CS-97-08, Computer Sciences Laboratory, RSISE, The Australian National University, Canberra, ACT 0200, Australia, April 1997. http://eprints.anu.edu.au/archive/00001586/

This paper introduced a special layer (mobile socket layer: MSL) below typical TCP socket. The mobile socket layer implements a naming abstraction. Virtual Port represents the additional port between socket and TCP port. Once the application binds the TCP port, the virtual port is bind instead. The mobile TCP association is established between two virtual ports. The virtual port is assigned to a home IP address and a local port. Thee sockets will see a TCP port with consistent permanent address. MSL maps these numbers with a real IP and TCP port number dynamically. The TCP connection for virtual IP and port are set up between two real IP and TCP port. In order to identify a unique connection, Vid is added to distinguish the reconnection request. This Vid is assigned by the server virtual port when the client request initially. The MSL can keep the TCP connection on until the sending buffer is full.



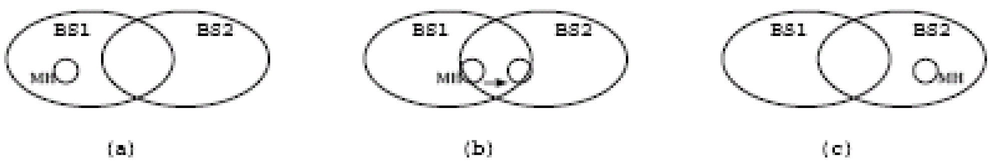


23. M. Riegel and M. Tuexen, *"Mobile SCTP,"* IETF Internet Draft; draft-riegel-tuexen-mobile-sctp-02.txt, Feb. 2003. http://www.potaroo.net/ietf/all-ids/draft-riegel-tuexen-mobile-sctp-06.txt

24. Wei Xing, Holger Karl, and Adam Wolisz, *"M-SCTP: Design and Prototypical Implementation Of An End-To-End Mobility Concept,"* 2002. http://citeseer.ist.psu.edu/xing02msctp.html

With the main feature of SCTP, "multi-homing: allow a single SCTP endpoint to support multiple IP address within a single association", the mobile node has at least two IP address in the existing association. The mobile node uses one of the IP addresses to prepare for actual handoff and tells the corresponding server using another transport layer connection that it can be reachable by another IP address. Mobile SCTP also makes use of ADDIP extension to muti-homing support so it can dynamically delete and insert IP address during the connection.

- Long term disconnection unsupported




```
                                  *******
                +--I        ********     **
       [2.0.0.2]|  A      **  **    **     **
                | / \____*    *    *******  *
#########       |       *   *********  *  ***           +-------+
#       #I------+       ****               **  [8.0.0.4]|       |
#       #I------+       *     Internet  ***  *----------|       |
#########       |      * *              **   * [8.0.0.5]+-------+
       [4.0.0.3]|      *  **   *    **   *   *
                +--I   *    ********     *  *
                   A    *        **     * **
                  / \_____*        *******
```

```
Mobile Client            *********                    Server
```

25. Seok Jon Koh, Hee Young Jung, and Jae Hong Min, *"Transport layer internet mobility based on mSCTP,"* Advanced Communication Technology, The 6th International Conference 2004 Volume: 1, On page(s): 329- 333. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1292884

26. Koh, S., J., and Xie, Q., *"mSCTP with Mobile IP for Transport Layer Mobility,"* Internet draft,version 3, IETF, February 2004, http://www.ietf.org/internetdrafts /draft-sjkoh-mobile-sctp-mobileip-03.txt

These papers describe the technique to use SCTP for transport layer mobility. This technique is the same as that in Mobile SCTP by M. Riegel and M. Tuexen. However, the paper also talks about how to use that over Mobile IP only for location management purpose (CN locates MN). Once the association is established, the data between MN and CN is based on only SCTP over IP. Mobile SCTP supports only CCOA for Mobile IPv4 and COA for Mobile IPv6.

27. S. Fu, L. Ma, M. Atiquzzaman, and Y. Lee, *"Architecture and performance of SIGMA : A seamless hanover scheme for data networks,"* IEEE ICC, Seoul, South Korea, May 2005. http://www.cs.ou.edu/~netlab/Pub/Kluwer-SIGMA.pdf, http://ieeexplore.ieee.org/iel5/9996/32113/01495024.pdf

This paper proposed the technique for seamless hanover. SIGMA is run on top of SCTP. The general concept is the same as mobile SCTP; allowing multiple addresses, add or delete IP address dynamically. Also, DNS is used for location management.

28. Ilknur Aydin, Woojin Seok, and Chien-Chung Shen, *"Cellular SCTP: A Transport-Layer Approach to Internet Mobility,"* University of Delaware. http://citeseer.ist.psu.edu/634772.html

29. E. Kohler, M. Handley, and S. Floyd, *"Designing DCCP: Congestion control without reliability,"* May 2003. http://citeseer.ist.psu.edu/kohler03designing.html

30. P. Yalagandula, A. Garg, M. Dahlin, L. Alvisi, and H. Vin. *"Transparent Mobility with Minimal Infrastructure,"* Technical Report TR-01-30, UT Austin, June 2001.

31. Hung-Yun Hsieh, Raghupathy Sivakumar, *"pTCP: An End-to-End Transport Layer Protocol for Striped Connections,"* IEEE International Conference on Network Protocols, Paris France, Pages: 24 -33, November 2002. http://citeseer.ist.psu.edu/hsieh02ptcp.html

32. Lorenzo Alvisi, Thomas C. Bressoud, Ayman El-Khashab, and Keith Marzullo, *"Wrapping Server-Side TCP to Mask Connection Failures,"* INFOCOM 2000. http://citeseer.ist.psu.edu/314228.html

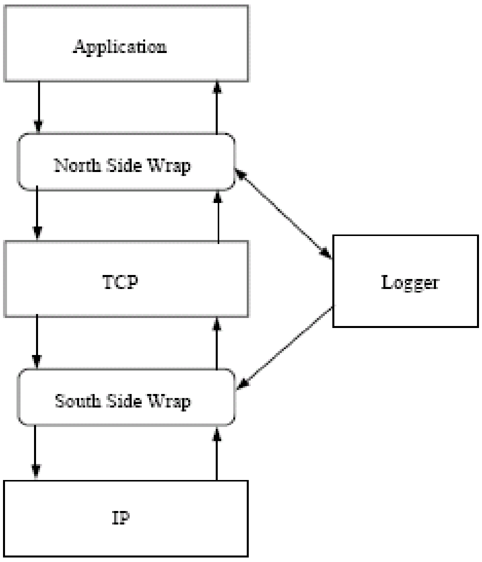
33. Dmitrii Zagorodnov, Keith Marzullo, Lorenzo Alvisi, Thomas, and C. Bressoud, *"Engineering Fault-Tolerant TCP/IP Servers Using FT-TCP,"* In Proceedings of IEEE Intl. Conf. on Dependable Systems and Networks (DSN) 2003. http://citeseer.ist.psu.edu/564162.html, http://www.cs.utexas.edu/users/lorenzo/lft.html

These papers introduced a server fault-tolerant technique for TCP/IP server (FT-TCP: fault tolerant TCP). It allows the faulty server still keep the TCP connection open till it recovers or the backup server is taking over. SSW (south side wrap) and NSW (north side wrap) are two wrappers which are inserted below and above TCP layer on TCP server. The wrappers keep storing the TCP connection state to the logger in order to maintain the TCP state consistency (TCP sequence number).

SSW intercepts data passing between the TCP&IP layer. For segments coming from the TCP layer to the IP layer, SSW maps the sequence number from the client's connection state to the server's current connection state. It does so to allow a recovering server's TCP to propose a new initial sequence number during the threeway handshake at connection establishment; SSW translates the sequence numbers to be consistent with those used in the original handshake. For packets going from the IP to TCP layer, SSW performs the inverse mapping on the ACK number. SSW also sends packets to the logger and either modifies or generates acks coming from the server to the client. It does so to ensure that the client side discards data from its send buffer only after it has been logged at the logger.

NSW intercepts read and write socket calls from the application to the TCP layer. During normal operation, NSW logs the amount of data that is returned with each read socket call. We call this value the *read length* for that socket call. When a crashed server is recovered, NSW forces read socket calls to have the same data and read lengths. It does so to ensure deterministic recovery. During recovery NSW also discards write socket calls to avoid resending data to the client.

- Slow failover (from ST-TCP)
- Backup server -> take over How??
- No process migration approach



34. Florin Sultan, Kiran Srinivasan, Deepa Iyer, and Liviu Iftode, *"Migratory TCP: Connection Migration for Service Continuity in the Internet,"* In Proceedings of the IEEE Intl. Conf. on Distributed Computing Systems (ICDCS), pages 469--470, Vienna, Austria, July 2002. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1022294

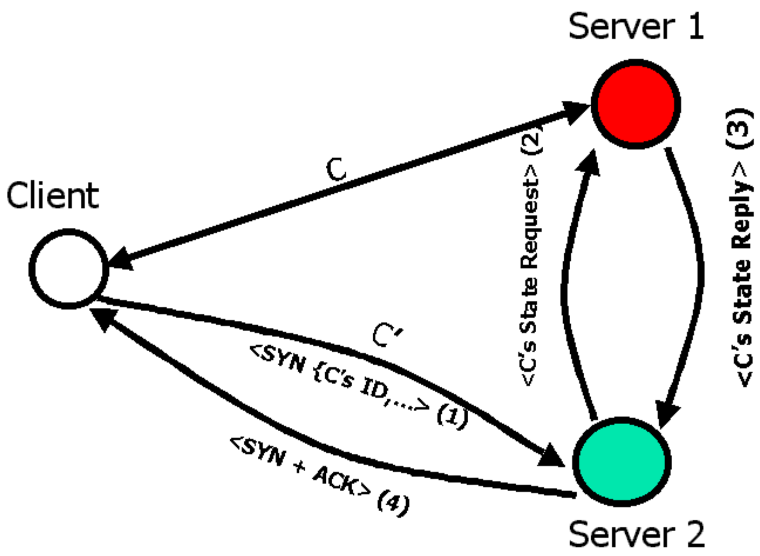
35. Florin Sultan, Kiran Srinivasan, Deepa Iyer, and Liviu Iftode, *"Migratory TCP: Highly Available Internet Services Using Connection Migration,"* 2002. http://citeseer.ist.psu.edu/sultan02migratory.html

36. Florin Sultan, Kiran Srinivasan, and Liviu Iftode, *"Transport Layer Support for Highly-Available Network Services,"* 2001. http://citeseer.ist.psu.edu/sultan00transport.html

37. Kiran Srinivasan, *"MTCP: Transport Layer Support for Highly Available Network Services,"* 2001. http://citeseer.ist.psu.edu/srinivasan01mtcp.html

All these papers mainly introduced TCP migration technique (M-TCP) for application server. During the migration, the destination and source servers have to be in operation and they also transfers all supporting state information (protocol state and application snapshot) when the client is going to migrate. The client can change different server voluntarily in order to obtain better performance. The connection ID (C) is introduced in order to identify the client connection. In this figure, the client prefers the server 2 connection. It send the connection ID to Server 2 to identify itself then the server 2 sends the request to acquire the client state from server 1.

- Limitation: lack of fault tolerance support, single connection migration, single-process server state



38. Manish Marwah, Shivakant Mishra, and Fetzer, C., *"A system demonstration of ST-TCP,"* Department of Computer Science, Colorado University, Boulder, CO, USA. Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on Publication Date: 28 June-1 July 2005 On page(s): 308- 313. http://ieeexplore.ieee.org/xpls/abs_all.jsp?tp=&arnumber=1467805

39. M. Marwah, S. Mishra, and C. Fetzer, *"TCP Server Fault Tolerance Using Connection Migration to a Backup Server,"* In Proceedings of the 2003 IEEE International Conference on Dependable Systems and Networks (DSN 2003), San Francisco, CA (June 2003).

These papers describe the design of ST-TCP (Server fault-Tolerant TCP) technique. This is done by having an active backup server that keeps track of the TCP state connection of the active server. There is no modification at the client (standard TCP) but be needed to modify TCP stack at the server. The server migration is transparent toward the client; the client can keep the same connection (maintaining byte sent/received) once the server is down (the backup server is taking over).

The backup server keeps tapping all packets (Ethernet tapping) flowing between main server and client in order to keep TCP connection state. Also, the main server and backup server keep sending the periodic hearbeat message (timeout mechanism) in separate connection (UDP over serial link) so that the backup server can detect if the main server is down or just its performance is degraded.

- - 1 Server<->1 Backup Server what if multi server <-> 1 BS (port??)
- - Backup Server & Server has to run all same application

40. Ronghua Zhang, Tarek F. Abdelzaher, and John A. Stankovic, *"Efficient TCP Connection Failover in Web Server Clusters,"* 2004. http://citeseer.ist.psu.edu/704561.html

41. Zhao Li, Ke Xu, Mingwei Xu, Lizheng Fu, and Jianping Wu, *"A New Method of Fault Tolerance TCP,"* In Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing 2003 Page: 320. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1243062

42. Koch, R.R., Hortikar, S., Moser, L.E. , and Melliar-Smith, P.M., *"Transparent TCP connection failover,"* Dependable Systems and Networks, 2003. In Proceedings of the 2003 International Conference on Publication Date: 22-25 June 2003 On page(s): 383- 392. http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=27228&arnumber=1209949&count=82&index=39

This paper describes the failover technique for TCP server. Only TCP/IP server stack is needed to modify (insert the layer between TCP and IP). Once the migration is required, TCP failover migrates the TCP server endpoint from the main server to backup server. Assume both server processes accept connections, handle requests and generate replies. They both go through the same state transitions.

All IP packets are redirected to backup server. The backup server keeps the TCP connection state synchronization with the client and server. The primary server can ack the packet back to the client only if it received the ack from the backup server. Also, the outgoing packets from the primary server have to be synchronized with the packet which the backup server will generate (packet size and sequence number).

43. Bryan Kuntz and Karthik Rajan, *"MIGSOCK. Migratable TCP Socket in Linux,"* Master Thesis, the Information Networking Institute, Carnegie Mellon University, 2002. http://www-cgi.cs.cmu.edu/~softagents/migsock.html

This thesis implemented a network socket migration in Linux kernel module and TCP stack toward the existing process migration (CRAK).

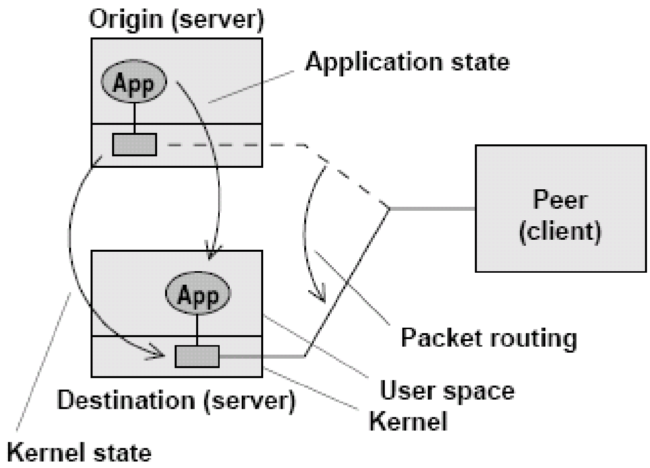
- Implementation: Linux kernel
- Performance: N/A

44. Maxim Orgiyan and Christof Fetzer, *"Tapping TCP Streams,"* IEEE International Symposium on Network Computing and Applications (NCA'01) page 0278. http://doi.ieeecomputersociety.org/10.1109/NCA.2001.96254

This paper describes one of the TCP failover techniques. The additional layer (above TCP layer) is added on server, backup server, and client. The backup server also tap all packets on both direction (server<->client). Once the main server is down, each additional layer of backup server and client reestablish new TCP connection which maintaining the same TCP states.

45. Werner Almesberger, *"TCP Connection Passing,"* Ottawa Linux Symposium, July 2004. http://tcpcp.sourceforge.net/

This article describes “tcpcp”, TCP connection passing mechanism that the application can pass their ownership of TCP connection endpoint across the Linux hosts. Without losing the socket connection, this technique can move a socket from one machine to another.



46. Shaya Potter and Jason Nieh, *“WebPod: Persistent Web Browsing Sessions with Pocketable Storage Devices,”* IW3C2 2005. http://citeseer.ist.psu.edu/734291.html

This paper introduced a portable system for mobile user. This technique keeps the same persistent, personalized web browsing session maintaining the user’s plugins, bookmarks, browser web content, open browser windows, and browser configuration options and preference. Also, with checkpoint/restarting mechanism, it can manually suspend and resume all web browsing states. This software is installed on a portable storage device such as USB.

1. Implementation: Linux (Apple iPod)
2. Performance: N/A

47. Mohammad, A. and Chen, A., *“Seamless mobility requirements and mobility architectures,”* Global Telecommunications Conference, 2001. (GLOBECOM '01) IEEE, 2001 Volume: 3, On page(s): 1950-1956 vol.3. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=965914

48. Shelley Zhuang, Kevin Lai Ion, Stoica Randy, and Katz Scott Shenker, *“Host Mobility Using an Internet Indirection Infrastructure,”* 2002. http://citeseer.ist.psu.edu/zhuang02host.html

49. B. Landfeldt, T. Larsson, Y. Ismailov, and A. Seneviratne, *"SLM, A Framework for Session Layer Mobility Management,"* In Proceedings of the IEEE ICCCN, Oct. 1999. http://ieeexplore.ieee.org/iel5/6535/17441/00805557.pdf#search=%22slm%20a%20framework%22

50. Fumio Teraoka, Yasuhiko Yokote and Mario Tokoro, *“A Network Architecture Providing Host Migration Transparency,”* SIGCOMM 1991. http://citeseer.ist.psu.edu/teraoka91network.html

51. Hari Balakrishnan, Karthik Lakshminarayanan, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Michael Walfish, *“A Layered Naming Architecture for the Internet,”* SIGCOMM 2004. http://citeseer.ist.psu.edu/balakrishnan04layered.html

52. Wedlund, E. and H. Schulzrinne, *"Mobility Support Using SIP,"* ACM/IEEE International Conference on Wireless and Multimedia (WOWMOM) 1999. www.cs.columbia.edu/IRT/papers/Wedl9908_Mobility.pdf

53. Wesley M. Eddy and Joseph Ishac, "*Location Management in a Transport Layer Mobility Architecture,"* NASA/TM-2005-213844 August 2005. http://gltrs.grc.nasa.gov/reports/2005/TM-2005-213844.pdf

54. H. Schulzrinne and E. Wedlung, *"Application-layer mobility using SIP,"* ACM Mobile Computer Communication Review 1(5), 1999. http://www.cs.columbia.edu/IRT/papers/Schu0007_Application.pdf#search=%22application-layer%20mobility%22

55. Kevin Brown and Suresh Singh, *"M-TCP: TCP for Mobile Cellular Networks,"* 1997. http://citeseer.ist.psu.edu/brown97mtcp.html

56. Takahashi, M., Kohiga, A., Sugawara, T., and Tanaka, A., *"TCP-Migration with Application-Layer Dispatching: A New HTTP Request Distribution Architecture in Locally Distributed Web Server Systems,"*
Communication System Software and Middleware, 2006. (Comsware 2006) First International Conference on Publication Date: 08-12 Jan. 2006.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=34859&arnumber=1665161&count=92&index=11

57. Zygmunt J. Haas, *"Mobile-TCP: An Asymmetric Transport Protocol Design For Mobile Systems,"* 1997. http://citeseer.ist.psu.edu/haas97mobiletcp.html

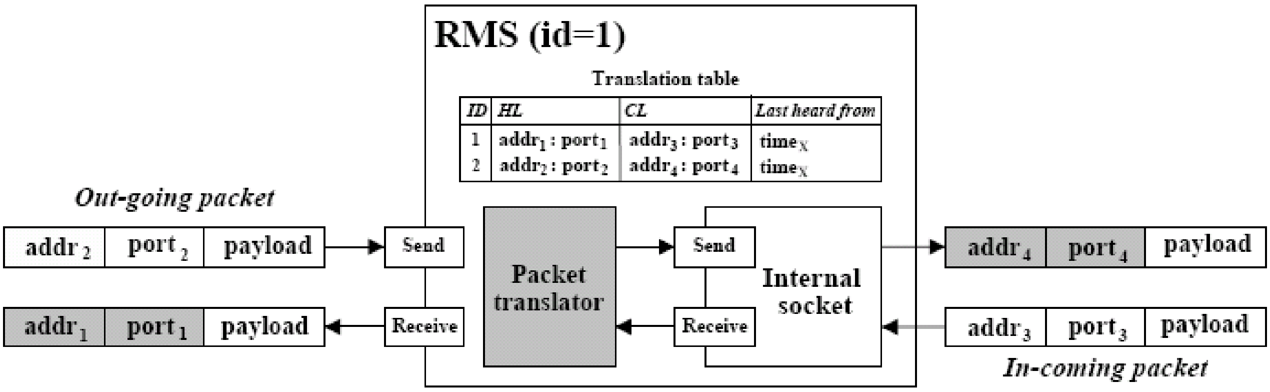
58. Peter Simons and Ralph Babe, *"Sustaining Idle TCP Connections with Linux IP Chains,"* CyberSolutions Research. http://cryp.to/publications/masquerading-idle-connections/

59. Timothy S. Mitrovich, Kenneth M. Ford, and Niranjan Suri, *"Transparent Redirection of Network Sockets,"* In Proceedings of the OOPSLA Workshop on Experiences with Autonomous Mobile Objects and Agent Based Systems.
http://ihmc.us:16080/research/projects/Nomads/mockets.pdf#search=%22transparent%20redirection%20of%20network%22

60. Navid Aghdaie and Yuval Tamir, *"Client-Transparent Fault-Tolerant Web Service,"* 20th IEEE International Performance, Computing, and Communications Conference 2001. http://citeseer.ist.psu.edu/aghdaie01clienttransparent.html

61. Johan Kristiansson and Peter Parnes, *"Application-layer Mobility support for Streaming Real-time Media,"* In IEEE Wireless Communications and Networking Conference (WCNC´04), Georgia, Atlanta, 2004. www.cdt.luth.se/~peppar/docs/rsocket/rsocket.pdf

This paper introduced a UDP socket extension for application layer mobility, Resilient Mobile Socket (RMS). This technique is like the socket translation as the Figure. The packet translator functions NAT like; to hide a change of the internal socket from applications, it converts the packets from internal socket to those on external socket. Unlike other transport layer mobility, it runs over UDP. In RSM architecture, in order to maintain the connection, SUSPEND mechanism by Handover manager is used to hibernate an on-going communication. An in band control signaling is used to transfer the socket identifier and exchange location information so RMS can identify the individual packet flows due to the change of IP address.



62. T. Koponen, A. Gurtov, and P. Nikander, *"Application mobility with HIP,"* In Proceedings of the 12th International Conference on Telecommunications, May 2005.
Extended abstract in Proc. of NDSS'05 Workshop, February 2005.
http://www.cs.helsinki.fi/u/gurtov/papers/

63. R. Moskowitz and P. Nikander, *"Host Identity Protocol,"* Jan. 2006. Internet-Draft draft-ietf-hip-base06.txt (work in progress). http://www.ietf.org/internet-drafts/draft-ietf-hip-base-06.txt

64. Atiquzzaman, M. and Reaz, A.S., *"Survey and classification of transport layer mobility management schemes,"* Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on Publication Date: 11-14 Sept. 2005
Volume: 4, On page(s): 2109- 2115 Vol. 4.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=34629&arnumber=1651818&count=141&index=6

TCP for wireless network:

1. Tom Goff, James Moronski, and D. S. Phatak, *"Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments,"* 2000. http://citeseer.ist.psu.edu/goff00freezetcp.html

2. Jian Liu, *"ATCP: TCP for Mobile Ad Hoc Networks,"* (1999) SUN Microsystems, Palo Alto, CA. http://citeseer.ist.psu.edu/liu99atcp.html

3. Ajay Bakre and B.R. Badrinath, *"I-TCP: Indirect TCP for Mobile Hosts,"* 1995. http://citeseer.ist.psu.edu/bakre95itcp.html

4. Swastik Kopparty, Srikanth V. Krishnamurthy, Michalis Faloutsos, and Satish K. Tripathi, *"Split TCP for Mobile Ad Hoc Networks,"* 2002. http://citeseer.ist.psu.edu/kopparty02split.html

5. [5] Scott Dawson, Farnam Jahanian, and Todd Mitton, *"Experiments on Six Commercial TCP Implementations Using a Software Fault Injection Tool,"* 1996. http://citeseer.ist.psu.edu/87468.html

This paper did the experiment to find out the characteristic of TCP connection on commercial TCP implementation; for example, TCP implementation on Window 95.

- **Retransmission Timeout (RTO)**, Win95 retransmitted the dropped segment five times. It increased the RTO exponentially. The upper bound of WIN95 RTO is 260 seconds.
- **Keepalive Connection Timeout**, if implemented, WIN95 sent 4 retransmission of the segment at 1 sec interval then close the connection. Win95 waits to send the first keep-alive around 7907 seconds. When it's dropped, Win95 retransmitted it a second later. (No exponential backoff)
- **Connection Timeout**, based on time units or a number of retransmissions.

However what they found was most machines used a number of retransmission for timing out. 5 was the number for Win95. For Zero Window Timeout, basically, as long as zero window probes are being ACK, TCP allows connection to be opened. If there is no ACK, for WIN95, it seems that it keeps open for a while then close unlike Solaris and others.

6. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, *"Stream Control Transmission Protocol,"* Request for Comments 2960, October 2000. http://www.rfc-editor.org/rfc/rfc2960.txt

7. Karunaharan Ratnam and Ibrahim Matta, *"WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links,"* Proc. Third IEEE Symposium on Computers and Communications (ISCC '98) 1998. http://citeseer.ist.psu.edu/ratnam98wtcp.html

8. Ming Zhang, Junwen Lai, Arvind Krishnamurthy, Larry Peterson, and Randolph Wang, *"A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths,"* 2004. http://citeseer.ist.psu.edu/656037.html

9. H. Balakrishnan, S. Seshan, E. Amir, and R.H. Katz: *"Improving TCP/IP Performance over Wireless Networks."* In Proceedings of the Mobicom, November `95. http://nms.lcs.mit.edu/~hari/papers/mcn.ps

10. K. Wang and S. Tripathi, *"Mobile-end transport protocol: an alternative to tcp/ip over wireless links,"* In Proceedings of the IEEE Infocom, 1998. http://citeseer.ist.psu.edu/wang98mobileend.html

11. Hari Balakrishnan, Srinivasan Seshan, and Randy H. Katz, *"Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks,"* ACM Wireless Networks 1995. http://citeseer.ist.psu.edu/balakrishnan95improving.html

Delay-Tolerant Networking:

1. Hooke, A. Torgerson, L. Fall, K. Cerf, V. Durst, B. Scott, K., and Weiss, H., *"Delay-tolerant networking: an approach to Interplanetary Internet,"* Communications Magazine, IEEE Publication Date: June 2003. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1204759
2. Kevin Fall, *"A Delay-Tolerant Network Architecture for Challenged Internets,"* SIGCOMM 2003. http://citeseer.ist.psu.edu/fall03delaytolerant.html

3. Robert C. Durst, Patrick D. Feighery, and Keith L. Scott, *"Why not use the Standard Internet Suite for the Interplanetary Internet?*," Interplanetary Internet Study Seminar, California Institute of Technology, 25 October 1999, http://www.ipnsig.org/reports/TCP_IP.pdf

4. Khaled A. Harras and Kevin C. Almeroth, *"Transport Layer Issues in Delay Tolerant Mobile Networks,"* http://citeseer.ist.psu.edu/742024.html

Process Migration:

1. Jeremy Casas, Dan Clark, Ravi Konuru, Steve Otto, Robert Prouty, and Jonathan Walpole, *"MPVM: A Migration Transparent Version of PVM,"* Department of Computer Science and Engineering, Oregon Graduate Institute of Science & Technology 1995.

2. Georg Stellner, *"CoCheck: Checkpointing and Process Migration for MPI,"* In Proceedings of the 10th International Parallel Processing Symposium (IPPS '96) 1996. http://citeseer.ist.psu.edu/stellner96cocheck.html

3. E. N. (Mootaz) Elnozahy, Lorenzo Alvisi, Yi-Min Wang, and David B. Johnson, *"A Survey of Rollback-Recovery Protocols in Message-Passing Systems,"* 1996. http://citeseer.ist.psu.edu/elnozahy96survey.html

4. Michael N. Nelson, Brent B. Welch, and John K. Ouster, *"Caching in the Sprite Network File System,"* ACM Transactions on Computer Systems 1988. http://citeseer.ist.psu.edu/nelson88caching.html

5. Fred Douglis and John K. Ousterhout, *"Transparent process migration: Design alternatives and the Sprite implementation,"* Software--Practice and Experience, 21(8):757-785, August 1991. http://citeseer.ist.psu.edu/douglis91transparent.html

6. Dejan S. Milojicic, Fred DougLis, Yves Paindaveine, Rechard Wheele, and Songnian Zhou, *"Process migration*," ACM Computing Surveys (CSUR) Volume 32, Issue 3 (September 2000) Pages: 241 – 299. http://www.hpl.hp.com/personal/Dejan_Milojicic/pm7.pdf#search=%22process%20migration%22

7. Michael Litzkow, Todd Tannenbaum, Jim Basney, and Miron Livny, *"Checkpoint and migration of UNIX processes in the Condor distributed processing system,"* Technical Report UW-CS-TR-1346, University of Wisconsin - Madison Computer Sciences Department, April 1997. http://www.cs.wisc.edu/condor/publications/doc/dobbs_95.ps

8. J. S. Plank, M. Bech, G. Kingsley, and K. Li, *"Libckpt: Transparent Checkpointing Under UNIX,"* In Usenix Winter 1995. http://www.cs.utk.edu/~plank/plank/papers/USENIX-95W.ps

9. S. Osman, D. Subhraveti, G. Su, and J. Nieh, *"The design and implementation of Zap: a system for migrating computing environments,"* SIGOPS Operating System Review, 36(SI):361-- 376, 2002. http://www.ncl.cs.columbia.edu/publications/osdi2002_zap.pdf#search=%22zap%3A%20a%20system%22

10. Ethan Solomita, James Kempf, and Dan Duchamp, *"XMOVE A Pseudoserver For X Window Movement,"* The X Resource 1994. http://citeseer.ist.psu.edu/solomita94xmove.html

11. M. J. Litzkow and M. Solomon, *"Supporting checkpointing and process migration outside the Unix kernel,"* In Proceedings of the Winter Usenix Conference, (San Francisco, CA), 1992. http://www.cs.wisc.edu/condor/publications.html#checkpoint

12. Robert Grimm, Janet Davis, Eric Lemar, and Brian Bershad, *"Migration for pervasive applications,"* Submitted for publication, 2002. http://cs.nyu.edu/rgrimm/papers/migration02.pdf#search=%22migration%20for%20pervasive%22

Research groups

1. Disco lab at Rutgers University, http://discolab.rutgers.edu/mtcp/

2. Networks and Mobile Systems at MIT, http://nms.csail.mit.edu/projects/migrate/

12. Parallel Performance Tools at WISC, http://www.cs.wisc.edu/~zandy/rocks

13. Network Computing Lab at Columbia, http://www.ncl.cs.columbia.edu/research/move/

14. Infrastructure for HIP (InfraHIP) at Helsinki Institute for Information technology, http://infrahip.hiit.fi/

15. The UC Berkeley BARWAN Research Project CDROM at UC berkeley, http://daedalus.cs.berkeley.edu/

16. Telecom and Network Research at OU, http://www.cs.ou.edu/~netlab/

Miscellaneous: