Fast handover and fast failover mechanisms
based on cross-layer collaboration among the link layer, the network layer and the transport layer

Fumio Teraoka, Yunsop Han
Keio University/NICT, Yokohama, Japan
tera@ics.keio.ac.jp
hanstone@tera.ics.keio.ac.jp

Keywords: cross-layer collaboration, fast handover, fast failover, SCTP

This paper describes a fast handover mechanism in the network layer called L3-FHOX and a fast failover mechanism in the transport layer called SCTPfx. Both mechanisms are based on a cross-layer architecture called CEAL. CEAL enables the control information exchange between layers in a node with keeping the layering structure. L3-FHOX is an example of cross-layer collaboration between the link layer and the network layer. SCTPfx is an example of cross-layer collaboration among the link layer, the network layer and the transport layer. We implemented both mechanisms in FreeBSD. The entire handover time in L3-FHOX is approximately 10 msec plus the RTT between the mobile node and its location management server while the normal handover procedure in IPv6 takes more than 1 second. The failover time of SCTPfx is 122 usec plus the RTT between the two end nodes while the normal failover procedure in SCTP takes more than 31 seconds.

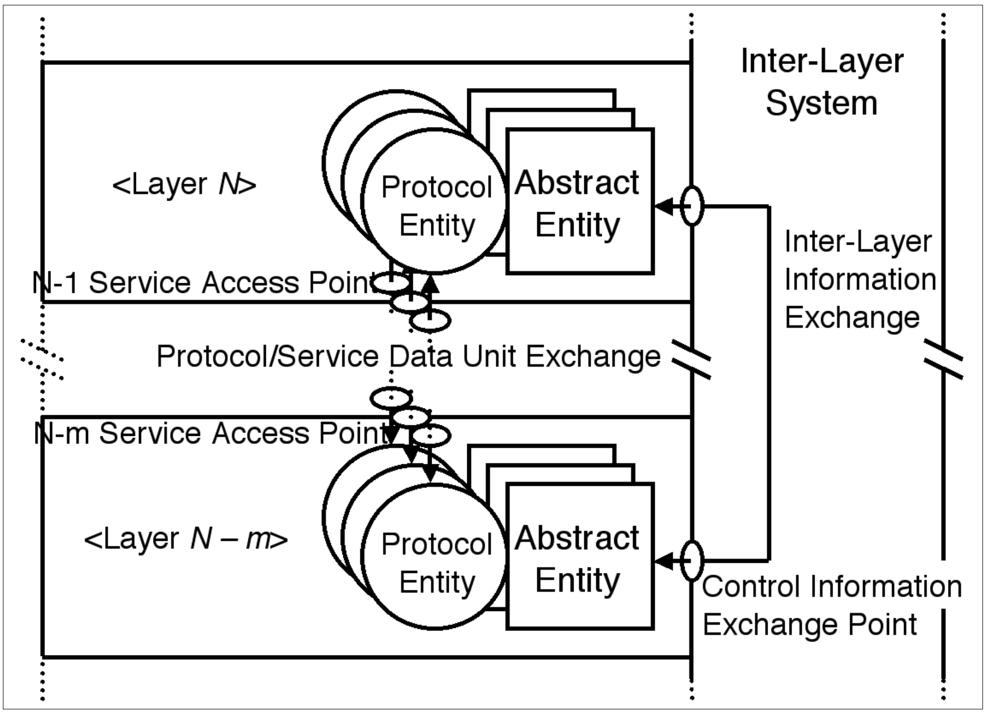
1. Introduction

The protocol layering model makes it easy to design protocols in a layer independently from other layers by concealing the details of a layer to other layers and providing abstract services to the upper layer. In some cases, however, a layer requires the information in other layers for efficient execution. Fast handover in the network layer and fast failover in the transport layer are typical examples that require the information in other layers for efficient execution. For fast handover in the network layer, the network layer requires the link layer information to predict handover and to select the most suitable access router to handover. For fast failover in the transport layer, the transport layer requires the lower layer information to know the failure event as soon as possible. Thus, the processing in a layer using other layer information is called *cross-layer collaboration* and the mechanism that enables the exchange of information between layers is called *cross-layer architecture*.

There are a lot of proposals that utilize cross-layer collaboration such as CLASS [1], ECLAIR [2], MobileMan [3], and Hydra [4]. We are proposing a cross-layer architecture called *CEAL* (Cross-layer control information Exchange between Arbitrary Layers) [5] and a fast handover mechanism in the network layer called *L3-FHOX* (L3-driven Fast HandOver mechanism based on X-layer arhitecture) [6] based on CEAL. In the transport layer, we are proposing a fast failover mechanism and a fast handover mechanism in SCTP [7]: SCTPfx [8] and SCTPmx [9], respectively.

This paper describes L3-FHOX as an example of collaboration between the link layer and the network layer and SCTPfx as an example of collaboration among the link layer, the network layer, and the transport layer.

Figure 1. Interaction model between layers in CEAL



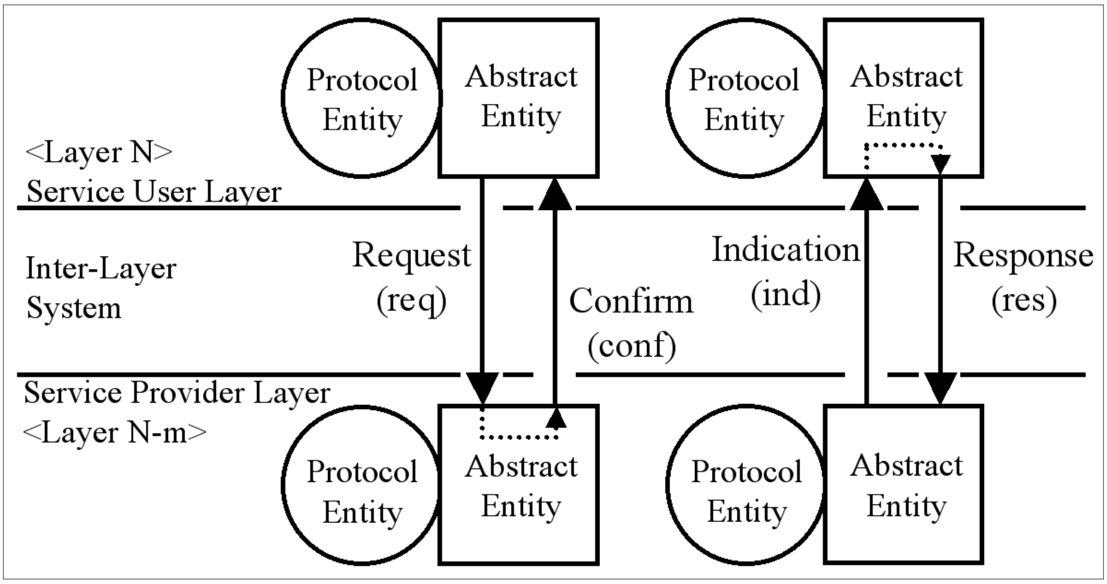


Figure 2. Four classes of primitives in CEAL

2. Overview of CEAL

In the OSI layering model, the *protocol entity* (PE) is the entity that processes a protocol. For cross-layer collaboration, the PEs in distinct layers must be able to exchange information. If a PE in a layer provides another PE in another layer with the protocol-specific information, each PE must be able to understand the information specific to all protocols. Therefore, CEAL introduces the *abstract entity* (AE) that transforms the protocol-specific information to the protocol-independent information as shown in *Fig. 1.* An AE is attached to a PE. CEAL also introduces the *inter-layer system* (ILS) that penetrates across all layers. Thus, the protocol-specific information of a PE is first transformed to the protocol-independent information by the AE attached to the PE; the protocol-independent information is carried to the AE attached to the PE that requires the information via the ILS.

In CEAL, each layer offers its services in the form of primitives. Four classes of primitives are defined as shown in *Fig. 2. Request* (req) is issued by the AE that wants to get the services or information from another AE, and *Confirm* (conf) is the acknowledgment of the request. *Indication* (ind) is the notification of the information to the AE that requested the service, and *Response* (res) is the acknowledgment of the indication. CEAL also defines three different usages of primitives. Type 1 is to provide the information in a layer to another layer immediately and consists of a Request and a Confirm. Type 2 is to notify another layer of events of a layer asynchronously. In Type 2, first a Request and a Confirm are exchanged; when the expected event occurs, an Indication is called. Type 3 is to control actions of a layer from another layer and consists of a Request and a Confirm.

CEAL defines nine L2 primitives as shown in *Tab. 1.* For example, *L2-PoAList* is used to acquire the list of available access points or base stations (i.e., points of attachment (PoA)). *L2-LinkStatusChanged* is used to receive a notification that the link status changed beyond the specified threshold.

3. L3-FHOX

3.1 Normal Handover Procedure in IPv6

Fig. 3 shows the normal handover procedure in IPv6. When the link status is getting worse, the link layer (L2) of the mobile node (MN) executes handover in the link layer (Fig. 3 (1)). This handover is called the L2 handover. However, the network layer (L3) of the MN does

Table 1. L2 primitives

Type	Name	Description
1	L2-LinkStatus	acquire link status
1	L2-PoAList	acquire list of points of attachment (PoA)
2	L2-LinkUp	notification that a link becomes available
2	L2-LinkDown	notification that a link becomes unavailable
2	L2-LinkStatusChanged	notification that the link quality changed beyond threshold
2	L2-PoAFound	notification that a new PoA is found
2	L2-PoALost	notification that a PoA disappeared
3	L2-LinkConnect	command to connect to specific PoA
3	L2-LinkDisconnect	command to disconnect

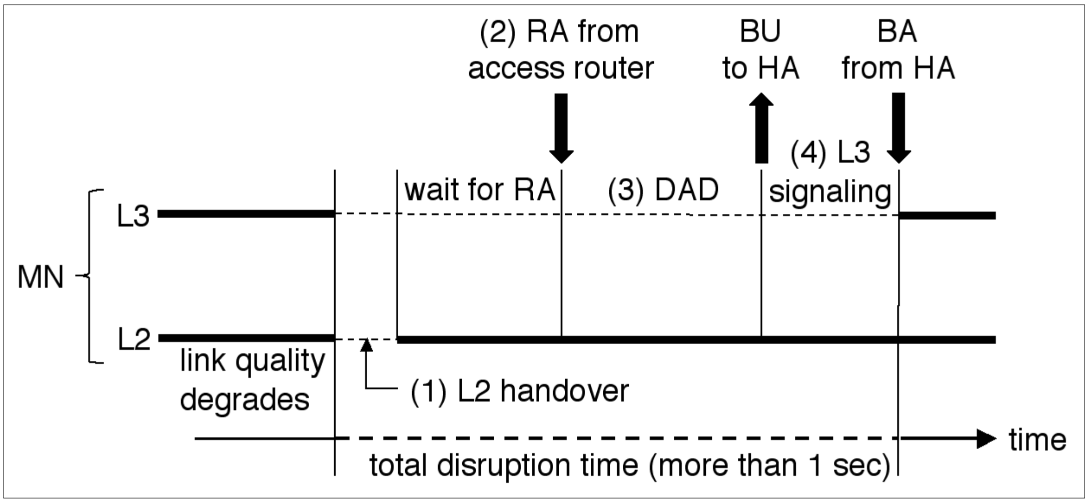


Figure 3. Conventional handover procedure

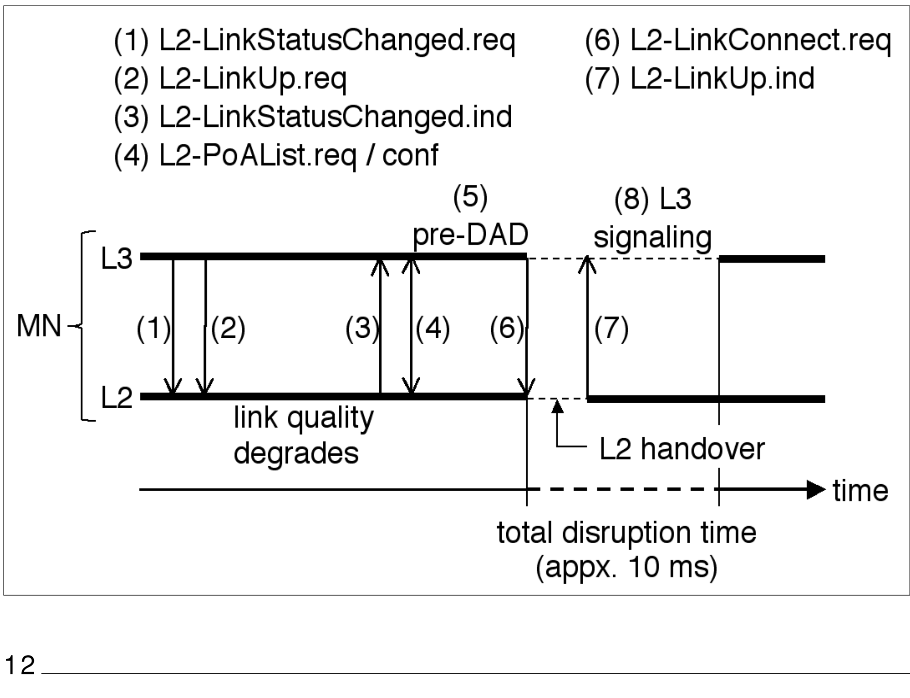
not perceive the *L2 handover*. In the network layer, the access router (AR) periodically sends the Router Advertisement message (RA). Upon receiving the RA, the L3 detects the L2 handover and starts the L3 handover procedure (Fig. 3 (2)). This wait time is one of the factors that make the entire handover time long. Next, the L3 generates a new IPv6 address and executes the duplicate address detection procedure (DAD) (Fig. 3 (3)). Since DAD makes use of timeout to detect duplicate addresses, it takes about 1 second. This is also one of the major factors that make the entire handover time long. Next, the L3 executes the L3 signaling, e.g., the exchange of the Binding Update message (BU) and the Binding Acknowledgement message (BA) between the mobile node and its home agent (HA) (Fig. 3 (4)). Thus, the entire handover takes more than 1 second.

3.2 L3-FHOX Procedure

L3-FHOX makes use of the L2 primitives defined in CEAL. *Fig. 4* shows the L3-FHOX handover procedure. First, the L3 of the MN issues *L2-LinkStatusChanged.req* and *L2-LinkUp.req* to the L2 to receive the notifications that the link status changes beyond the specified threshold and that the link becomes available again, respectively (Fig. 4 (1),(2)). When the link status is getting worse beyond the threshold, the L2 notifies the L3 of *L2-LinkStatusChanged.ind* (Fig.4 (3)). The L3 searches for the candidate access point for handover by exchanging *L2-PoAList.req/conf* (Fig. 4 (4)). Upon deciding the next access router (NAR), the L3 requests the pre-DAD to the NAR (Fig. 4 (5)). In the pre-DAD, since the NAR knows the all IPv6 addresses used in its cover area, it immediately responds to the MN. At this point, the L3 finishes the handover preparation. Next, the L3 issues *L2-LinkConnect.req* to the L2 to make the L2 start the L2 handover (Fig. 4 (6)). Upon finishing the L2 handover, the L2 notifies the L3 of *L2-LinkUp.ind* (Fig. 4 (7)). Next, the L3 starts the L3 signaling (Fig. 4 (8)).

As shown in Fig. 3, the disruption time due to the handover is the L2 handover time plus the L3 signaling time. In our experiment, the former is less than 10 msec in case of WiFi. The latter depends on the round trip time (RTT) between the MN and the HA; usually the order of 10 msec.

Figure 4. L3-FHOX handover procedure



3.3 Field Experiment of L3-FHOX

We implemented L3-FHOX in FreeBSD-5.4 and had a field experiment of L3-FHOX. On a circle road, we arranged eight access points / access routers each of which provides a distinct IPv6 subnet to the mobile node. The access point has a WiFi interface for the mobile node. The length of the circle road is approximately 1 km. The mobile node is on a car and sends real-time streaming data to the correspondent node (CN). The car runs at 40 km/h. The application used in this experiment is DVTS (Digital Video Transfer System) [10]. Although DVTS consumes approximately 35 Mbps bandwidth, we selected the halfrate mode of DVTS due to the bandwidth limit of WiFi. LIN6 [11] is used as a mobility support network layer protocol, which is based on ID/Locator split architecture. As a result, the entire handover time is approximately 10 msec. Upon a handover, there is almost no bad effect on the play-backed movie on the CN.

4. SCTPfx

4.1 Normal Failover Procedure in SCTP

SCTP is a new transport layer protocol. It has several new features such as multihoming support and partial reliability. For multihoming support, SCTP can have multiple paths in a single association between two end nodes. Among multiple paths, SCTP uses a single path as the primary path for data communication and reserves other paths as secondary paths. If the primary path fails, e.g., due to crash of a router on the primary path, one of the secondary paths is engaged as the primary path. This procedure is called *failover*. Even if a failover occurs, the association is kept available. According to the specification of SCTP, SCTP detects path failure by five times of timeout of the ACK for the transmitted data. Similar to TCP, since SCTP employs binary back off to calculate the timeout value, it takes at least 31 seconds to detect path failure.

4.2 Fast Failover in SCTPfx

SCTPfx detects path failure as soon as possible by collaboration among the link layer, the network layer, and the transport layer. There are several possible causes of path failure. For example, the cable is accidentally unplugged from the end node, a router on the path crashes, and connectivity to the destination is lost due to routing change. Due to the page limit, this paper focuses only on the case that the cable is unplugged in an end node and the case that the connectivity is lost in the network core.

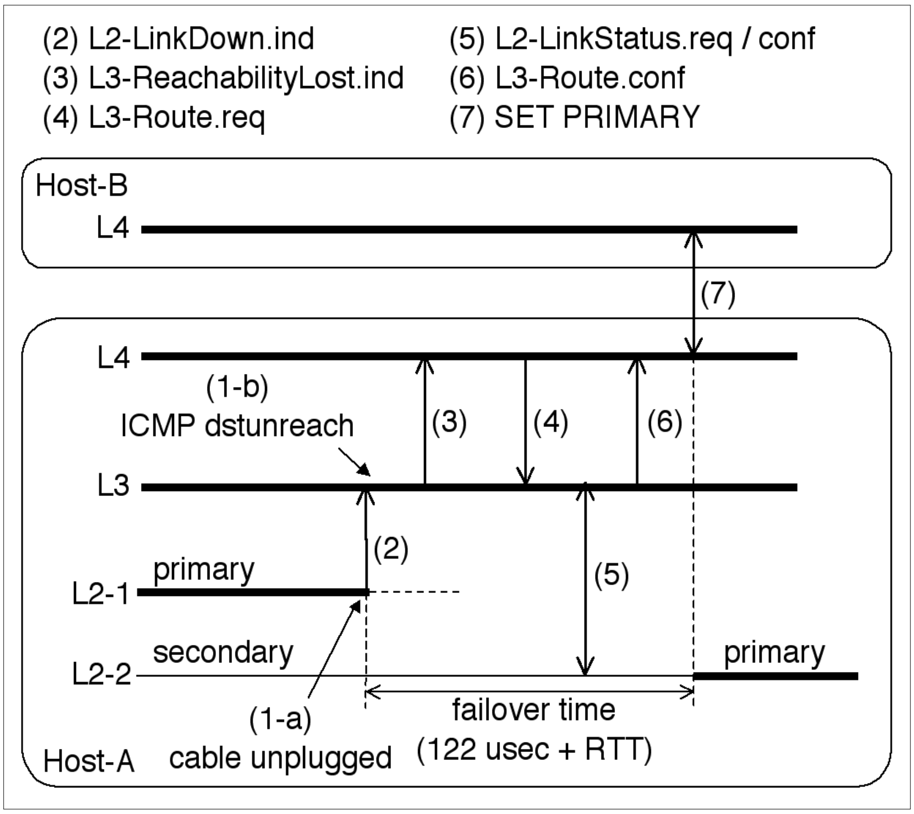
Fig. 5 shows the failover procedure in SCTPfx. In this example, the end node has two interfaces, L2-1 and L2-2. Suppose that L2-1 is the current primary path and L2-2 is the secondary path. In case that the cable is unplugged in the end node (Fig. 5 (1-a)), the L2 notifies the L3 of *L2-LinkDown.ind* (Fig. 5 (2-a)). Next, the L3 notifies the L4 of *L3-ReachabilityLost.ind* (Fig. 5 (3)). In case that the connectivity to the destination node is lost, e.g., the L3 of the end node receives the ICMP destination unreachable message (Fig. 6 (1-b)), the L3 notifies the L4 of *L3-ReachabilityLost.ind* (Fig. 5 (3)). The remaining procedure is the same for both cases. Upon receiving *L3-ReachabilityLost.ind*, the L4 (i.e., SCTP) issues *L3-Route.req* to the L3 to find an alternative route to the destination (Fig. 5 (4)). Upon receiving *L3-Route.req*, the L3 issues *L2-LinkStatus.req* to each interface and obtains the link information such as availability and the bandwidth (Fig. 5 (5)). Next, L3 selects the available routes to the destination and returns this result to the L4 by *L3-Route.conf* (Fig. 5 (6)). Upon receiving *L3-Route.conf*, the L4 selects the new primary path. Finally, the L4 of Host-A sends a *SET PRIMARY chunk* to the L4 of Host-B to switch the primary path (Fig. 5 (7)). Although SCTPfx also defines a fast recovery procedure, its description is omitted due to the page limit.

4.3 Evaluation of SCTPfx

Fig. 6 (on the next page) shows our test network. We implemented SCTPfx in the FreeBSD-6.1 kernel. We installed SCTPfx on two machines: Host-A and Host-B, both of which are IBM ThinkPad X40 with an Intel PentiumM 1.1 GHz CPU and 512 MB memory. At first, interface-1 of Host-A is the primary path. Next, we unplug the cable from interface-1, and then the fast failover procedure is executed. As a result, interface-2 becomes the new primary path.

In this environment, we measured the failover time. The result was 122 usec plus the RTT between Host-A and Host-B. This value is extremely small compared to the normal failover time (i.e., 31 seconds).

Figure 5. Failover procedure in SCTPfx



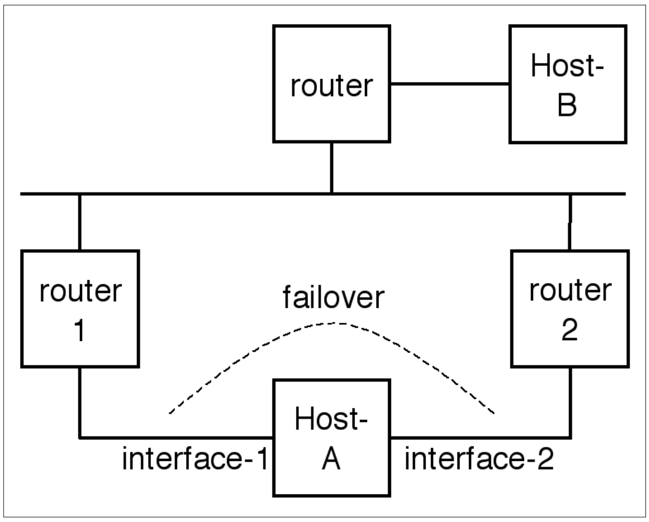


Figure 6. Test network of SCTPfx

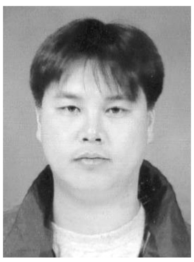
5. Conclusion

This paper shows effectiveness of cross-layer collaboration among the link layer, the network layer, and the transport layer by taking the fast handover mechanism in the network layer (L3-FHOX) and the fast failover mechanism in the transport layer (SCTPfx) as examples.

As the architecture of cross-layer collaboration, we are proposing CEAL. The current model of CEAL focuses only on cross-layer collaboration within a node. We plan to extend the current CEAL model so that it can deal with cross-layer collaboration between nodes.

Authors

FUMIO TERAOKA received a master degree in electrical engineering and a Ph.D. in computer science from Keio University in 1984 and 1993, respectively. He joined Canon Inc. in 1984 and moved to Sony Computer Science Labs., Inc. in 1988. Since April 2001, he is a professor of Faculty of Science and Technology, Keio University. He received the Takahashi Award of JSSST (Japan Society for Software Science and Technology) and the Motooka Award in 1991 and 1993, respectively. He also received the Best Paper Award in 2000 from IPSJ (Information Processing Society Japan). His research interest covers computer network, operating system, and distributed system. He was a board member of IPSJ from 2000 to 2002. He was a board member of JSSST from 2005 to 2009. He is a member of ACM, IEEE, JSSST, IPSJ, and IEICE.



YUNSOP HAN received his M.S. and Ph.D. degrees in Computer Science from Keio University, Yokohama, Japan, in 2005 and 2009 respectively. In September 2009, he joined The Keio University as a postdoctoral researcher. His research interests include wireless networks, cross-layer optimization, congestion control and transport protocols.

References

[1] Wang, Q., Abu-Rgheff, M.A., "Cross-Layer Signaling for Next-Generation Wireless Systems," In Proc. of IEEE WCNC 2003, pp.1084–1089, March 2003.

[2] Raisinghani, V.T., Iyer, S., "Cross-Layer Feedback Architecture for Mobile Device Protocol Stacks," IEEE Communications Magazine, Vol. 44, No. 1, pp.85–92, Jan. 2006.

[3] Borgia, E., Conti, M., and Delmastro, F., "Mobileman: Design, Integration, and Experimentation of Cross-Layer Mobile Multihop Ad Hoc Networks," IEEE Communications Magazine, Vol. 44, No. 7, pp.80–85, July 2006.

[4] Choi, S.H., Perry, D.E., Nettles, S.M., " A Software Architecture for Cross-Layer Wireless Network Adaptations," In Proc. of IEEE/IFIP WICSA 2008, pp.281–284, Februar 2008.

[5] Teraoka, F., Gogo, K., Mitsuya, K., Shibui, R., Mitani, K., "Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover," RFC 5184, May 2008.

[6] Gogo, K., Shibui, R., Teraoka, F., "An L3-Driven Fast Handover Mechanism in IPv6 Mobility," In Proc. of SAINT 2006 Workshop, pp.4–7, Januar 2006.

[7] Stewart, R., "Stream Control Transmission Protocol," RFC 4960, September 2007.

[8] Han, Y., Teraoka, F., "SCTPfx: A Fast Failover Mechanism Based on Cross-Layer Architecture in SCTP Multihoming," In. Proc. of AINTEC 2008, pp.113–122, November 2008.

[9] Han, Y., Teraoka, F., "SCTPmx: A Fast Handover Mechanism Using a Single Interface Based on a Cross-Layer Architecture," IEICE Trans. on Communications, Vol. E92-B, No. 9, pp.2864–2873, September 2009.

[10] DVTS Consortium, *http://www.dvts.jp/en/index.html*

[11] Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H., Teraoka, F., "LINA: A New Approach to Mobility Support in Wide Area Networks," IEICE Trans. on Communications, Vol. E84-B, No. 8, pp.2076–2086, August 2001.