A Uniform Host Protocol Framework Planning To Change

Tian Chen, Liu Wenyu, Wang Yi
Dept. of Electronics and Information Engineering
Huazhong University of Science and Technology
Wuhan, Hubei, China
{tianchen, liuwy, ywang}@mail.hust.edu.cn

Huaien Luo
Department of Electrical and Computer Engineering
National University of Singapore
Singapore
luo.huaien@gmail.com

***ABSTRACT*—as a substantial foundation of the Internet, traditional layered host protocol stack is found to be a framework "Not Planning to Change". This framework is hard to accommodate new layers and new protocols, hence hard to provide new services including security and mobility. What's more, it is also hard to share information among layers, which is critical for cross-layer optimization in MANET implementation. A novel protocol stack framework, altitude based architecture (ABA), is presented in this paper. The proposed framework is cater to framework extensibility, and by negotiation mechanisms, a universal communication model is also presented for backward-compatibility. ABA, which is planning to change, aims to build a uniform host protocol framework for future Internet and MANET. Analyses and simulation illustrations are also given in the paper.**

***KEY WORDS*—Protocol framework, protocol stack, re-configurable architecture, cross layer feedback**

I. Introduction

The layered framework of host protocol stacks plays a key role in Internet's success. This architecture has three main advantages: *simple*, communication functions are decomposed to several layers, each has a relatively independent function; *flexible*, a layer could involve independently by maintain interfaces to its adjacent layers; and *efficient*, layered packet formats guarantee both routing performance and processing efficiency in host. However, it is clear this strict layered protocol stack is insufficient for future Internet and MANET.

Various schemes are proposed to add new services (like security and mobility) to Internet, as these functions are not native to any traditional layer. On the other hand, sharing information among protocol layers (cross-layer feedback) is the basic requirement for MANET performance optimization. The problem is: traditional host protocol framework makes these requirements difficult to realize. Layered architecture define strict interfaces between layers, makes them couple with each other tightly. To maintain backward compatibility, new protocols have to be added to stack as patches: Mobile IP [1] adds a "bag" besides IP layer; HIP [2] inserts a new Host Identity layer between transport layer and network layer by faithfully inheriting the old interfaces; network layer is spitted to two sub-layers in SHIM [3], one for identity and the other for routing; ISP [4] and CLASS [5] add signal mechanisms for each couple of two layers to enable cross-layer feedback. Similar schemes are also presented for in [13-17]. Almost all these schemes are implemented in a factitious way, and their amendments to protocol stack sometimes even conflict with each other. To sum up, layered protocol architecture is a framework "Not Planning to Change".

A novel architecture, Altitude Based Architecture (ABA), is proposed in this paper. This approach eases the insertion of new protocol layers and new services, and supports full cross-layer feedback. An universal communication model is also presented for backward-compatibility. ABA, which is planning to change, aims to build a uniform host protocol framework for future Internet and MANET.

Next section presents related works and the design motivations of ABA. Main ideas of the proposed architecture are given in Section III. For illustration of ABA's effectiveness, simulations of scenarios are conducted in Section IV. Section V includes cost analyses and transition considerations. Conclusions and future works are presented in the final section.

II. Related Works and Design Motivation

The traditional layered framework is not ready to accommodate new layers or new requirements. There are works try to fundamentally revise the basic Internet architecture in [12, 18-20]. Hereby are some related works of host protocol framework.

Layered OON [6] architecture is composed of many separate service objects (SV). The framework is organized as a grid with layers and planes. A layer is a vertically defined function sets implemented by a group of SVs; a plane is a horizontally defined function sets implemented by a group of SVs. Positioned in the grid, a SV belongs to a specific layer and a specific plane and each represents a specific function like fragment or forwarding. Thus, a new protocol could be easily composed of some already existed SVs.

Role-based network architecture [7] is a non-layered architecture. Modular protocol unit is called a role, which is a description of a function block that performs some specific function relevant to forwarding or processing. Raw data are organized into chunks, or, role-specific headers; different roles could process the same chunk, and several chunks could be delivered to one role. New services could be easily merged into existing systems, and RBA is friendly to middle boxes [7].

OON modeling eases the design of extendable and reusable systems, while it is still hard for OON to add new layers.

978-1-4244-1645-5/08/$25.00 ©2008 IEEE

RBA is extremely extensible, but its packets forwarding performance in routers and processing performance in host are dissatisfied. Not only host protocol framework, they demand revolutions to entire Internet. Both two architectures are focus on extensibility; cross-layer feedback is not considered in them at all which is critical in MANET.

MobileMan [8] proposed a full cross-layer mechanism for MANET. By a vertically standby component named Network Status (NS), protocols could exchange network information through NS. MobileMan eliminates the need for direct interaction mechanism between each couple of layers, hence maintain layer independency. All these dedicated architectural efforts for cross-layer feedback including MobileMan [8] and ECLAIR [9] are not extendable.

To sum up, a new protocol framework for future network should not only ready to accommodate new requirements of Internet/MANET, but also should meet other requirements. Our design goals are listed below.

Extensibility is the primary objective. A robust and fully extensible protocol framework should be provided, to ease the insertion of new protocol layers and new protocols. And it's more meaningful to build a uniform framework for both future Internet and MANET

Full cross-layer design is the secondary objective. Dedicated managers should provide not only network status but also local status: network status includes geographic topology, physical link Signal-Noise-Ratio, network congestion etc; local status includes battery energy, service queue length etc. Cross-layer trigger and cross-layer cooperation should be taken into consideration. The new mechanism should not only be an information exchange centre of the layers, but also the centre of cross-layer design.

Backward compatibility is necessary to facilitate the evolution of host protocol stack in a nod-by-nod transition process. Together with the universal communication model to be mentioned later, newer nods and older nods could still communicate with each other without any obstacle. *Layer independence* should be kept for system modularity and processing performance consideration. *A Universal communication model* should be provided including: Automatic Discovery of Protocol Resources； Stack Synchronization among Communication Peers； Dynamic Configuration of Protocol Components; and maybe On-the-fly Reconfiguration of Protocol Components。

III. MAIN IDEAS OF THE PROPOSED FRAMEWORK

As a further enhancement of MobileMan [8], ABA provides *extensibility* in addition to other merits of MobileMan. As shown in Figure 1, protocol stack is still layered in ABA (*altitude number* and *protocol number* in hereby figures are illustration only). Each protocol belongs to a protocol layer, and an altitude value is specified to each protocol layer. Protocols in the same layer are distinguished by their own protocol ID value. A protocols administrator (PA) stands vertically besides the protocol stack. Every protocol mounts to PA according to its belonging layer's altitude and its protocol ID.

PA has three individuals. Stack Manager is in charge of protocols registration and interfacing. Cooperate Manager is the headquarters of cross-layer optimization with two main functions: Information sharing and inter-layer cooperates. Session Manger handles protocol stack synchronization in peers, which is essential for backward compatibility. Thus, the cross-layer design of Cooperate Manager is almost the same with MobileMan [8], with some extra consideration in local status. Other goals are provided by Stack Manager and Session Manger.

Protocols in adjacent layers never interface to each other directly, but to PA. A protocol of ABA should never assume its upper layer or lower layer, but only get/send its protocol frames from/to Stack Manager, as illustrated by packet flow in Figure 1. A protocol should only assume that it could communicate with its counterpart in peer. This greatly eases the insertion of new protocol layers and new protocols. For example when a new security service emerged, a new protocol layer E could be silently inserted between layer C and layer D without bother any of them. The Stack Manager simply change the packet processing flow route to pass though layer E. Figure 2 shows the addition of a new layer(E) and two new protocols(C3,E1).

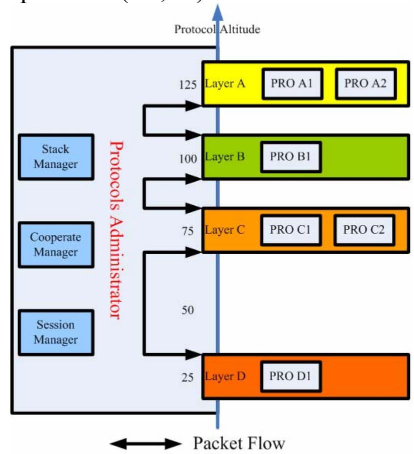

Figure 1 Packet Flow Chart of ABA

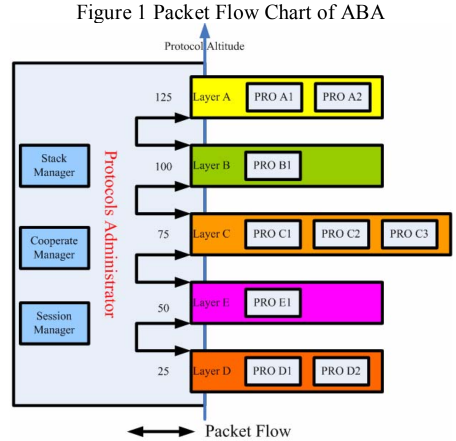

Figure 2 Packet Flow Chart of ABA with a new layer

By Session Manger's universal communication model, two nodes could always connect with each other whether their protocol stacks are identical or not. Before a session, peers' Session Managers negotiate and synchronize their stacks first. In Figure 3, node X and node Y both have security layer E, so they use 5 layers to communicate after their synchronization. In Figure 4, M is an old fashion node without layer E, thus X and M could still communicate by 4 layers. Easy layer/protocol insertion and universal communication model together are the bases of ABA and make it planning to change.

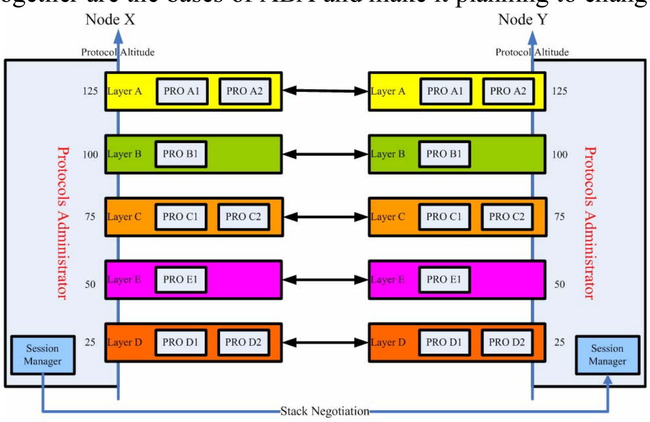

Figure 3 Communications between new nodes

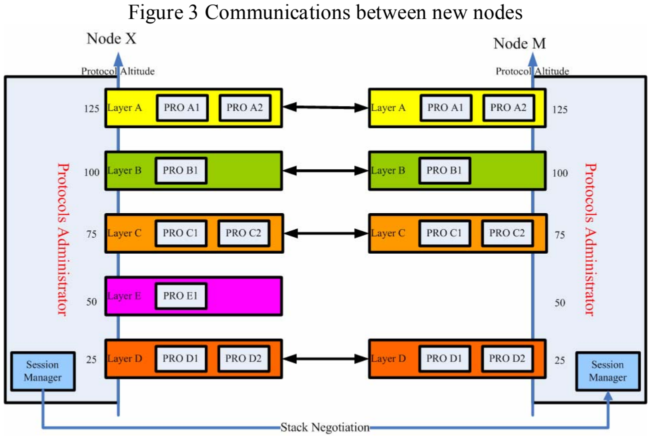

Figure 4 Communications between old node and new node

By maintaining layered nature, ABA guarantees both packet processing performance in host and routing performance in network. Packets are still processed layer by layer, protocol by protocol. Details of these processes are presented below:

a) Packet format and processing

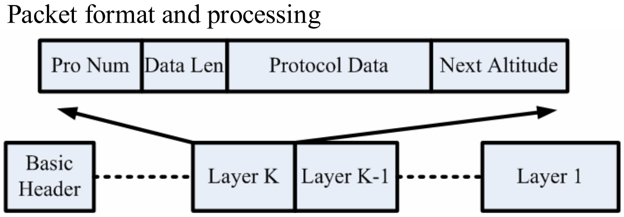

Figure 5 ABA Packet format

Packets are still processed layer by layer. "Pro Num" field is used to distinguish different protocols in the same layer (e.g. TCP, UDP in internetworking layer); "Data Len" field indicates total length of layer header except "Next Altitude" field; "Next Altitude" field indicates the next layer. Example in Figure 6 illustrates the processing sequence: two protocols in layer C both participate in the communication and each processes its own frames. However, formal standardisation of the packet format is not the emphasis of this paper.

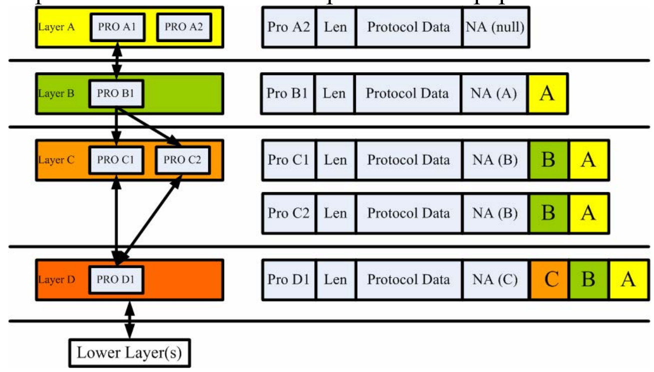

Figure 6 ABA processing sequence

b) Stack synchronization

Session manager use handshakes to achieve protocol stack synchronization. A preferred stack description and a minimum stack description should be included in the request. A typical negotiation needs 2 or 3 times handshakes. Use the node X,Y,M scenario again: node X wants A,B,C,D,E and node Y say yes; node X wants A ,B,C,D,E and node M reply sorry I only have A,B,C,D; then node X agreed. The two procedures are shown in Figure 7. This mechanism may also enable runtime protocol stack synchronization.

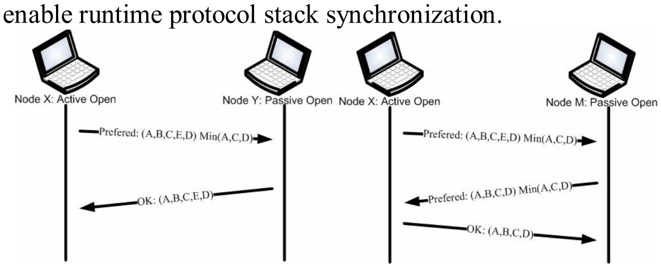

Figure 7 Stack synchronization

The problem is: if two nodes need to synchronize protocol stack by their Session Manager before their session could start, how could two Session Managers talk at the very beginning without a protocol stack agreement? It seems to be a contradiction. In fact for packet forwarding considerations, some lowest layers like internetworking should be left untouched. Thus, altitude management is built only above the internetworking layer, and Session Manager could do their job directly by using the internetworking layer service. A practical model is presented in section V.

IV. Scenario Simulations

It is hard to evaluate "ability to change" than to evaluate simple questions like "how fast" and "how much". For this reason, a dedicated event-driven simulator is developed by C++. Two application scenarios, which are hard to be supported in traditional framework or MobileMan, are implemented and verified in the simulator to illustrate the effectiveness of ABA.

a) Host Identity

This scenario is implemented to verify the extensibility and backward compatibility of ABA. For security concern, people want to authenticate himself to the communication peer uniquely. HIP [2] try to satisfy the requirement by introducing in a separate host identity layer between transport layer and IP internetworking layer. Because HIP has changed the 5-layered IP model, HIP nodes are not backward compatible with existing IPv6 nodes. Obviously, it is hard for HIP to be widely accepted in current framework.

As stated above, a new dedicated security layer could be easily inserted In ABA. As shown in Figure 8, an inserted host identity layer is implemented in the simulator. In the simulation, a node with new identity layer could receive packets simultaneously from two different neighbours: one with identity layer and the other not. Besides HIP [2], other security protocols like IPSEC [11] may also be fit into this layer.

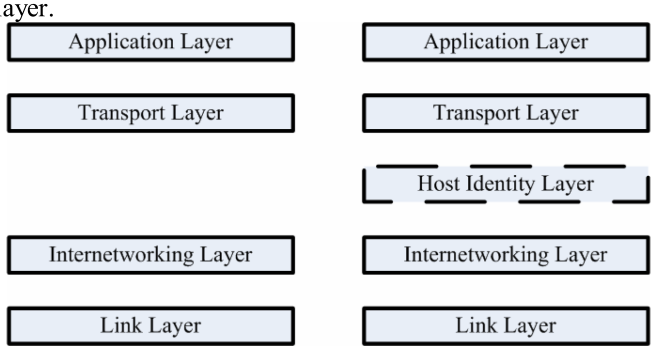

Figure 8 New Host Identity protocol layer

b) IP Multicast

The main problem of IP multicast in Internet is that not all routers in the network support multicast forwarding. It is inefficient to support multicast protocol in current framework. IP in IP tunnels are always deployed to transfer one multicast router to another.

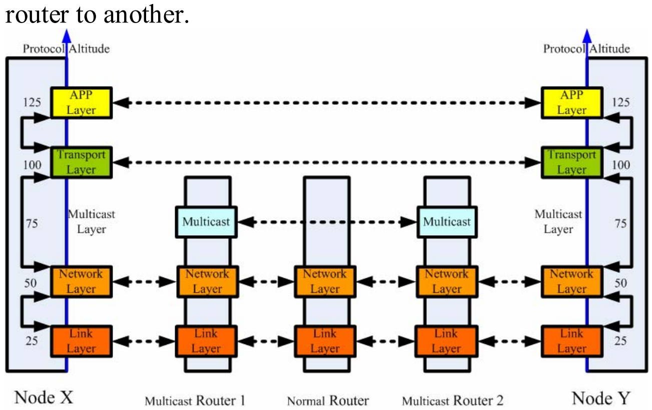

Figure 9 routing illustration of multicast packet

Figure 9 takes IP multicast as an example of a new IP service and work through its deployment under ABA. Multicast routers of the same multicast group form an overlay distributed network; when a multicast member sends a packet to an edge multicast router, a multicast layer header is inserted to the packet between network layer and transport layer headers,; by broadcasting packets in the overlay network, packets are forwarding to whole overlay network; before leaving the multicast network, edge servers peel off the multicast header from the packet; at last the packet arrives each group member. In the multicast layer, dedicated management and forwarding multicast protocols could be developed to maintain dynamic overlay network topology. This procedure could be totally transparent to multicast peers; they just send to/receive from a multicast IP address.

V. COST ANALYSIS AND TRANSITIOSN CONSIDERATION

Compared to its advantages, the cost of ABA is minimal. As computational costs to hosts and routers are limited to several machine instructions, our analyses are focused on communication cost.

Typical overhead of packets can be roughly estimated. Each layer of ABA may need only 4 extra bytes: 1 byte for Next Altitude field, 1 byte for Protocol Number field, 2 bytes for data/header length indication. Assume a typical video stream packet with IP-UDP-RTP encapsulation, 12 extra bytes are added to each packet. The length indication bytes could be further reduced if protocols could manage length indication by themselves, and put them into layer header enables nodes to skip some layers if they want to. What's more, peers could negotiate packet header compression for a long duration session. Extra communication delay is also minimal in ABA. To each session, only 1 or 2 round trips delay for stack synchronization is needed at the start of the session, typically far less than the session duration.

As a new architecture, it would be helpful to provide a series of mechanisms for transitioning between successive generations. The first choice is migrate legacy protocols to ABA and gain all benefits of it. Almost all existing protocols have to be revised, this is the must prices of obtain extensibility and cross layer optimization. They should accommodate new PA interfaces and fix themselves into specific layers. Some unnecessary couples between adjacent layers should also be cancelled, for example: TCP checksum calculation should be changed to not involve IP addresses. However, the works is trivial and the impact can be minimal.

It is impractical to set a Flag Day for protocol stack transition. One choice is mixed protocol stack as shown in Figure 10. Specific numbers could be allocated to new versions of protocols; legacy interfaces could be reserved for backward compatibility; old applications could still use the socket function call and bypass ABA stack and session management at all. A practical illustration model is shown in Figure 10. Traditional 5 layers of IP model and 3 new layers including stack negotiation layer are mounted on PA according to their altitude value (these values are for illustration only). As mentioned in section III, internetworking layer (IP layer) is treated as the default basic layer in ABA. Physical layer and link layer are considered to be "local" to peers. For illustration only, altitude value 08 is assigned to internetworking layer; protocol num 06 is assigned to IP protocol. This assignment corresponds to type value 0x0806 of IP in 802.11 MAC systems and also is used for illustration only.

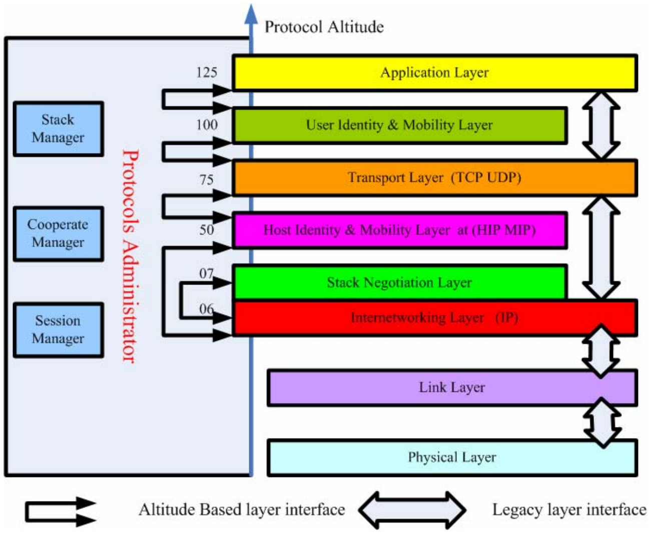

Figure 10 mixed protocol stack

Dual stack transition is another choice: traditional protocol framework still functions; new framework realized node by node. Just like transition from IPv4 to IPv6, old applications could work and new applications emerge.

VI. CONCLUSIONS

In this paper, a novel altitude-based host protocol framework is presented. By introducing protocols altitude management mechanism into traditional strict layered structure, ABA achieves extensibility, backward compatibility, full cross layer cooperation, etc. This proposal aims to build a uniform host protocol framework for both future Internet and MANET, and is planning to change. By maintaining layered nature, ABA also guarantees packet processing and routing performance. The cost is trivial: small improvement to existing protocol stack and protocols; a little delay before communication. Two application scenarios are verified by simulations to illustrate the effectiveness of ABA.

Our future works aim to build a software prototype to verify the architecture and to evaluate its performance.

ACKNOWLEDGEMENT

The project is supported by The National Natural Science Foundation of China (No.60572063) and Chinese National High-Tech Research and Development Plan (No.2007AA01Z223).

REFERENCES

[1] D. Johnson, et al., Mobility Support in IPv6, IETF RFC3775, June 2004
[2] R. Moskowitz and P. Nikander, Host Identity Protocol Architecture, draft-ietf-hip-arch-00 (work in process), Internet Draft, IETF, 16 October 2004
[3] Huston, G., Architectural Approaches to Multi-Homing for IPv6, Work in progress: Internet Drafts draft-ietf-multi6-architecture-04.txt, February 2005
[4] G. Wu et al., Interactions between TCP and RLP in Wireless Internet, IEEE GLOBECOM, Rio de Janeiro, Brazil, Dec. 1999, vol. 1B, pp. 661–66.
[5] Q. Wang and M. A. Abu-Rgheff, Cross-Layer Signaling for Next-Generation Wireless Systems, WCNC, vol. 2, Mar. 2003, pp. 1084–89.
[6] Stefan Bocking, Object oriented network protocol, Beijing, Machinery Industrial Press (ISBN 7-111-08076-9/TP), 2000.
[7] B. Braden, T. Faber and M. Handley. From Protocol Stack to Protocol Heap - Role-Based Architecture, First Workshop on Hot Topics in Networking, Oct 2002.
[8] M. Conti et al., Cross-Layering in Mobile Ad Hoc Network Design, IEEE Comp., vol. 37, no. 2, 2004, pp. 48–51.
[9] Raisinghani, V.T.; Iyer, S.; Cross-layer feedback architecture for mobile device protocol stacks, Communications Magazine, IEEE Volume 44, Issue 1, Jan. 2006 Page(s):85 - 92
[10] Christian Vogt; Martina Zitterbart, Efficient and Scalable End-to-End Mobility Support for Reactive and Proactive Handoffs in IPv6, IEEE Communication Magazine, Vol.44, No.6, June 2006
[11] Kent, S., "IP Encapsulating Security Payload (ESP)", draft-ietf-ipsec-esp-v3-10 (work in progress), March 2005.
[12] Sylvia Ratnasamy, Scott Shenker, Steven McCanne,"Towards an Evolvable Internet Architecture",SIGCOMM'05 August 22–26, 2005, Philadelphia, Pennsylvania, USA.
[13] Tian Chen, Liu Wenyu, "A novel IPv6 communication framework: mobile SHIM6 (M-SHIM6)",IEEE WiCOM 2006, September 22-24, 2006 Wuhan, China.
[14] T. Anderson, T. Roscoe, and D. Wetherall. Preventing Internet Denial-of-Service with Capabilities. In 2nd ACM Workshop on Hot Topics in Networks, Cambridge, MA, November 2003.
[15] Ion Stoica, Scott Shenker, and Hui Zhang. Core-stateless fair queueing: Achieving approximately fair allocations in high speed networks. In Proceedings of SIGCOMM, September 1998.
[16] Alex Snoeren, Hari Balakrishnan, and Frans Kaashoek. Reconsidering internet mobility. In Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VIII), May 2001.
[17] Dina Katabi, Mark Handley, and Charles Rohrs. Internet Congestion Control for High Bandwidth-Delay Product Networks. In ACM SIGCOMM, Pittsburgh, PA, August 2002.
[18] Ion Stoica, Dan Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet indirection infrastructure. In Proceedings of SIGCOMM, August 2002.
[19] David Clark, Robert Braden, Aaron Falk, and Venkata Pingali. FARA: Reorganizing the addressing architecture. In ACM SIGCOMM Workshop on Future Directions in Network Architecture, Karlsruhe, Germany, August 2003.
[20] Mark Gritter and David R. Cheriton. TRIAD: A new next-generation Internet architecture. http://www-dsg.stanford.edu/triad/, July 2000