International Journal of Networking and Computing – www.ijnc.org
ISSN 2185-2839 (print) ISSN 2185-2847 (online)
Volume X, Number Y, pages 1–13, June 2014

Research Roadmap Driven by Network Benchmarking Lab (NBL): Deep Packet Inspection, Traffic Forensics, Embedded Benchmarking, Software Defined Networking and Beyond

Ying-Dar Lin, Fellow, IEEE
Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
ydlin@cs.nctu.edu.tw

Received: (received date)
Revised: (revised date)
Accepted: (accepted date)
Communicated by Editor's name

Abstract

Most researchers look for topics from the literature. But our research derived mostly from development, in turn driven by industrial projects or product testing. We spanned into the areas of cable TV networks, multi-hop cellular, Internet QoS, deep packet inspection, traffic forensics, embedded benchmarking, and software defined networking. Among them, our multi-hop cellular work was the first along this line and has a high impact on both academia and industry, with over 600 citations and standardizations in WLAN mesh (IEEE 802.11s), WiMAX (IEEE 802.16j), Bluetooth (IEEE 802.15.5), and 3GPP LTE-advanced. Side products from our research include a startup (L7 Networks Inc., in 2002), a test lab (Network Benchmarking Lab, NBL, since 2002), and a textbook Computer Networks: An Open Source Approach (McGraw-Hill, 2011). It is a perfect time to have my 20-year half-time report as we celebrate the 70th birthday of my Ph.D. thesis advisor, Prof. Mario Gerla. This report could serve as a reference for researchers in developing their own roadmap.

Keywords: research model, research roadmap, development and research, network research, deep packet inspection, traffic forensics, embedded benchmarking, software defined networking

1 Roadmap and Footprints

From Development to Research

Research topics in the academia are often drawn from three sources: literature repository, development projects, and industrial discussions. The literature repository accounts for the dominant percentage as it is the easiest way to find a topic by following a crowd of researchers. Your papers could also enjoy being well cited if you are slightly ahead of the crowd or the fever on the topic persists for many years. The only problem with this source might be minor improvement on existing problems defined by others or wasted resources on pseudo, instead of real, problems. On the other hand, deriving a research topic from a development effort is an expensive approach, where research is defined as the non-trivial parts within the development process. The virtue in return is a real problem with a feasible solution. The problem or the solution might be new to the academia and the industry. Researching a real problem from the industrial discussions is an inexpensive alternative. However, as there might not be real development involved, the research result might not be a feasible solution. How tight research and development should go together is a choice. I myself prefer a tighter relationship because after all the nature of data communications is engineering instead of pure science.

With the choice of a tighter relationship between research and development, over a half of my research topics derived from development projects. This is particularly true with the prevalence of Linux and open source resources since late 90s. A rule of thumb is if I dont know how to develop it I would not research on it. My 20-year research career at National Chiao Tung University (NCTU) has spanned into several areas, including cable TV networks, wireless, Internet QoS, deep packet inspection, traffic forensics, and network and embedded testing. In addition to 102 journal papers, 51 conference papers, and 31 filed patents, 165 industry-oriented articles (in Chinese) and 3 books were written.

Cable and Multi-hop Cellular

Triggered by the development of bi-directional coaxial cable TV networks in mid-90s and a project sponsored by a company were our research on hybrid fiber coaxial (HFC) networks, with some well cited works on minislot allocation and scheduling, including HFC protocol design [1], IEEE 802.14 standardization [2], combined allocation and scheduling [3], MPEG-aware scheduling [4], HFC protocol design and implementation issues [5], optimal minislot allocation [6], optimal ranging [7], uplink scheduling [8], and n-ary collision detection [9].

Inspired by the weakness and instability in the connectivity of ad hoc networks, we were the first to propose in the year 2000 the wireless architecture that combines cellular and ad hoc networking into multi-hop cellular. Multi-hop cellular [10] and multi-hop WLAN [11] have been cited over 600 times with many follow-up works, including two special issues and four main-stream industrial standards. Supported by an industry project, we later extended this direction of research into mesh networking, with a turn-key development [12], a design of multi-channel with fewer radios [13], and an experimental study [14].

Internet QoS

Research works on Internet QoS were fostered by the surge of Internet growth in late 90s and early 2000s. With the abundant Linux and open source resources, we were able to prototype a QoS-enabled router. On that router, we developed and experimented a series of algorithms for (1) admission control (bandwidth brokers [15] and measurement-based admission control [16]), (2) scheduling (preemptive DRR [17], applying fair queuing to WLAN [18], applying fair queuing to request scheduling [19], request scheduling for DiffServ [20], multi-resource request scheduling for DiffServ [21], scheduling for GPRS [22], scheduling for WiMAX [23], DiffServ over network processors [24]), (3) classification (lookup-and-bypass classification [25]), (4) queue management (benchmarking bandwidth management techniques [26], TCP rate shaping [27], link load balancing [28], codec-aware VoIP playout [29]), (5) QoS routing (QoS routing granularity in MPLS [30], service-sensitive routing in MPLS [31]), (6) multicasting (RP relocation in PIM-SM [32]), and (7) TCP-friendly congestion control (comparing TCP-friendly congestion control schemes [33], TCP-equivalent rate control [34]).

Deep Packet Inspection with Two Spin-offs

While bandwidth became abundant and security issues arose in early 2000s, we moved the focus to deep packet inspection mainly for Internet security. The previous prototyped QoS-enabled router was turned into a 7-in-1 security gateway with routing, bandwidth management, NAT (Network Address Translation), firewall, VPN (Virtual Private Network), IDS (Intrusion Detection System), and content filtering (or called application firewall). The latter two and some other new functions require deep packet inspection on application headers and payloads, which is much slower than handling TCP/IP headers. To speed up deep packet inspection, we profiled many security packages (profiling string matching [35]), changed software architectures (integrated security gateway [36], content security gateway [37], in-kernel P2P management [38], stream-based anti-virus [39], scalable one-to-many streaming [40]), designed new algorithms for string matching (string matching for deep packet inspection [41], sub-linear string matching [42], content filtering with early decision [43]), and implemented string or classification matching into network processor (core-centric network processor [44], memory-intensive network processors [45], thread allocation in network processors [46], VPN over network processors [47]) and FPGA/SoC hardware (sub-linear string matching hardware with bloom filters [48], string matching automata with root hashing [49], scalable automata with indexing and hashing [50], automata in SoC [51]). In this stage, we built the dR research model where Linux-based development (open source development [52], embedded Linux [53]) triggered research issues and the proposed solutions were evaluated through experiments on developed systems. The side effects of this research model include a start-up, L7 Networks Inc. (www.L7.com.tw) since 2002, and a test lab, Network Benchmarking Lab (NBL, www.nbl.org.tw) since 2002, examining and benchmarking security, switch/router, WLAN, and VoIP, and more recently LTE and handheld products.

Traffic Forensics at NBL

NBL operations were purely development efforts without research until we established an on campus beta site in the dormitory network. Research issues arose when we started to use real traffic to test network products. Real traffic has been proved to be effective in triggering product defects which would otherwise become customer found defects instead of lab found defects. However, understanding and manipulating real traffic is non-trivial. Thus, another series of research were conducted, including testbed design (on campus beta site [54], NAT compatibility testbed [55] [56] [57], IPv6 beta site [58]), traffic replay (Socket Replay [59], WLAN Replay [60], ProxyReplay [61], Multi-Port Replay [62]), test coverage analysis and optimization [63] , traffic forensics (PCAP Lib [64], bug traces [65]), intrusion analysis (taint tracker for buffer overflow detection [66], evasion through IDS [67], attack session extraction [68], false positive and negative analysis in intrusion detection [69], weighted voting [70]), malware analysis (secure malware analysis environment [71], active and passive malware collection [72], malware classification [73], botnet detection [74]), and security criteria [75]. Research along this track is still on-going and may continue for a few more years.

Embedded Benchmarking Lab (EBL)

In the meantime, to span from network devices to handheld client devices, we established another lab, Embedded Benchmarking Lab (EBL, www.ebl.org.tw) in 2011. EBL reviews smartphones and touchpads in terms of functionality, performance, power consumption, stability, and GUI smoothness. Another series of research works are being developed from EBL, which range from performance profiling (bottleneck analysis on Android applications [76], multi-resolution profiler on Android applications [77]), cloud offloading with time-and-energy awareness [78], Android malware detection [79], and smartphone GUI testing [80] [81]. This is a relatively young research area with potentials of good impact on embedded systems in general, smartphones, tablets, and other handheld or future wearable devices. The concerned issues are usually not on protocol aspects but on software and hardware components in embedded systems.

Software Defined Networking (SDN)

With the same process of research led by development, we are getting into an emerging area, namely software defined networking (SDN). We view SDN as the second wave of cloud computing happening to networking, with the control plane being centralized and virtualized into the cloud while leaving the data plane at the customer side. SDN deployment started from data centers and now expands to the model of networking as a service (NaaS) offered by the operators to enterprise and residential subscribers. By centralizing the control-plane software of routers and switches to the controller and its applications, and controlling the data-plane of these devices remotely, SDN reduces the capital expenditure (CAPEX) and operational expenditure (OPEX) because the devices become simpler and hence cheaper and number of administrators could be reduced. SDN also enables fast service orchestration because the data plane is highly programmable from the remote control plane at controllers and applications. It is deemed to bring the biggest change to the data communications industry in this decade.

We are in the process of developing an SDN solution to control and manage campus switches and Wi-Fi access points, a test lab with test capabilities on conformance, interoperability, performance, stability, and test tools. Through this development process, research issues are being identified and investigated. Among them, standardization plays the foundation role to evolve the OpenFlow, the southbound API between controllers and switches, converge the northbound API between controllers and applications, extend the basic SDN architecture by service chaining (SC) and network function virtualization (NFV) to accommodate value-added services, and test systems and products in terms of conformance, interoperability, performance, and functionality. Other advanced research issues include performance and scalability of switches, controllers, and applications, security of SDN itself and security services offered by SDN, and use cases in all possible domains from data centers, operators of wired and wireless infrastructures, enterprises, homes, down to smartphones, wearable computers, and machine-to-machine (M2M) systems. Though there are papers published or being published on SDN, generic architectures and algorithms, and solid modeling and analysis are yet to be researched.

The rest of this article is organized as follows. We highlight five results and their impacts in five short sections. Section 2 gives a closer look at multi-hop cellular. Section 3 expands the roadmap on deep packet inspection. Section 4 and Section 5 zoom into the operations of NBL and EBL. The textbook Computer Networks: An Open Source Approach [82] is briefed in Section 6. Learned lessons summarized in Section 7 could be useful career tips for junior researchers.

2 Multi-hop Cellular Communications

This work presents a new architecture, multi-hop cellular network (MCN), for wireless communications. MCN preserves the benefit of conventional single-hop cellular networks (SCN) where the service infrastructure is constructed by fixed bases, and it also incorporates the flexibility of ad-hoc networks where wireless transmission through mobile stations in multiple hops is allowed. MCN can reduce the required number of bases or improve the throughput performance, while limiting path vulnerability encountered in ad-hoc networks. In addition, MCN and SCN are analyzed, in terms of mean hop count, hop-by-hop throughput, end-to-end throughput, and mean number of channels (i.e. simultaneous transmissions) under different traffic localities and transmission ranges. Numerical results demonstrate that the throughput of MCN exceeds that of SCN, the former also increases as the transmission range decreases. The above results can be accounted for by the different orders, linear and square, at which the mean hop count and mean number of channels increase, respectively.

We were the first to propose the architecture and analyze the capacity of multi-hop cellular networking back in 2000. The concept of relaying within a cell started from our Infocom 2000 paper. We proposed the architecture that evolved from ad hoc and cellular networks. It has been proved mathematically that its capacity grows linearly as the transmission range decreases because the hop count and the number of channels grow linearly and quadratically, respectively. We also designed and implemented a WLAN prototype with multi-hop relaying to access points. Recently we combined the multiple channel concept with 802.11s mesh networking, where few radios switch between channels. The solution and its firmware were licensed to Realtek Semiconductor as a turn-key solution bundled with Realteks WLAN chipsets.

Since 2000, our Infocom paper has received over 600 citations from papers, patents, books, and special issues. It was included as a theme topic in at least two books: Next Generation Mobile Access Technologies (Haas and McLaughlin, Cambridge, 2007) and Ad Hoc Networks (Wu and Stojmenovic (editors), IEEE Computer Society, 2004). Two special issues have been dedicated to the concept of multi-hop cellular: IEEE Communications Magazine (2007) and EURASIP Journal on Advanced in Signal Processing (2008). The paper was cited by several patents (US 7,145,892 in 2006, EP 1,481,517 in 2006, etc.) and has served as the foundation of many other patents that utilize relaying within a cell. One recent Ph.D. dissertation in Finland (Doppler, 2010) investigated various relaying techniques within cellular systems, and started by citing our Infocom paper. The work on multi-hop cellular has had long lasting impact not only on academia but also on industry. Relaying within a cell or towards an access point or base station has been standardized in IEEE 802.11s (1.0 in 2006, 2.0 in 2008, 3.0 in 2009 and 2011), WiMAX (IEEE 802.16j-06/013r3 in 2007, IEEE C802.16m-08/1436r1 in 2008), Bluetooth (IEEE 802.15.5), and under development within 3GPP LTE-advanced.

3 Deep Packet Inspection

From 2000, we started an investigation of deep packet inspection (DPI) examining application headers and payloads of incoming packets for application-aware and malicious traffic management. In comparison with table lookup of destination IP address and 5-tuple (source/dest IP address and port number, protocol ID) done in routers and firewalls, DPI requires signature matching on the variable-length application header and payload to look for specific applications, intrusions, viruses, malware, and spam, a much heavier process than the traditional table lookup. We started from restructuring packet flows within Linux systems. Next we designed string matching algorithms that could scale well over tens of thousands of signatures, and then implemented the algorithms in hardware and SoC designs to scale to multi-Gbps in throughput. This research roadmap on DPI, software algorithm hardware SoC, has interleaved development with research. The Linux-based development fostered a startup in 2002, L7 Networks Inc. L7 addressed the market of content-aware networking with DPI, and was later acquired by D-Link Corp.

After developing and researching DPI engines, we moved on to apply DPI to traffic forensics, in particular for product testing at NBL. We established the first on campus beta site, where potential defects could be detected earlier from live traffic at the beta site or from replayed traffic at NBL than at customer premises. NBL has developed the techniques of Beta Site (with redundancy for fast recovery), PCAP Lib (a classified library of packet traces), ILLT (In-Lab Live Testing, replay framework and tools), etc. Compared to the other test labs that depend solely on artificial traffic generated by test tools, NBLs approach to use live and replayed real traffic, labeled RealFlow, is world-wide unique. It has opened a unique opportunity for traffic forensics research in academia and for real traffic testing in industry.

4 Network Benchmarking Lab (NBL)

Founded NBL in 2002, NBL started as a customized testing service provider, grew to be a test solution/tool provider from 2005, and added the world-wide unique RealFlow real traffic testing from 2007. It has served over 100 companies, tested over 600 products, grown to a staff of 23 full-time engineers plus 20 students, and has been 2/3-supported by industry and 1/3 by government agencies. Positioning itself as a real traffic test lab, NBL has also developed its research roadmap along beta site, packet trace library, in-lab replay testing, malware sample database, etc. Based on the local significance established in the first decade, NBL has a chance to establish its global significance in the next decade.

NBL is operated in a 3-line structure, where the 1st-line (mostly full-time engineers) test products, the 2nd-line (a mixture of engineers and students) develop tools, and the 3rd-line (mostly graduate students) research techniques. Students are arranged to help engineers in the 1st and 2nd lines for one year to get familiar with the products, tools, and development environments, which enables them to identify a research topic from the development work. Important milestones are listed as follows.

- 2001 Pre-NBL: public benchmarking events with an IT magazine (2001 2010: security gateway, bandwidth manager, Web switch, ISP QoS, e-commerce, WLAN, CDN, IPv6 router, L2/L3 switch, VoIP, IDS, VoWLAN, 10G, Android smartphone, etc.)
- 2002 Officially launched
- 2003 MOU signed with UNH-IOL
- 2004 First Plugfest (interoperability) in Taiwan
- 2007 NCTU Beta Site established
- 2009 First RealFlow certificate issued, Live SOHO launched
- 2010 Live Security launched, PCAP Lib and ILLT released
- 2011 ACTS (Automatic Control Test System) first version released, sister lab EBL (Embedded Benchmarking Lab) launched
- 2012 ISO 17025 certified lab, NCC certified lab, NCC security criteria developed

5 Embedded Benchmarking Lab(EBL)

Following the same philosophy and footprint of NBL, EBL digs into handheld devices, including smartphones and tablets. These devices are client-side devices instead of networking devices, which means the industry served by EBL would be different from the one served by NBL. We consolidated a series of test methodologies and tools into EBL Test Suite v1.0 in the first three years with efforts on benchmarking, profiling, and optimization. In most cases, benchmarking, profiling, and optimization treat the devices as black boxes, grey boxes, and white boxes, respectively.

The overall objective is to provide methodologies and tools to cover all layers of smartphones. In particular, for Android systems, this could range from Java apps, Dalvik virtual machine, runtime library, Linux kernel, down to drivers and hardware.

6 Computer Networks: An Open Source Approach

Computer Networks undoubtedly is one of the key technologies of Information Technologies. Many textbooks are available on the shelves which adopted quite different approaches, from traditional, and sometimes dry, protocol descriptions to the application-driven top-down approach and the system-aspect approach.

This book, as its title indicated, takes a different approach from that of previous books, i.e., an open source approach. Besides written with logic reasoning minds and emphasizing more on why a protocol is designed that way than how a protocol works, this book tries to fill the gap between knowledge and skills by tracing the source code such that readers could learn where and how the protocol designs could be implemented. We found this open source approach quite effective in building readers know-how on protocol implementation, which makes this book very unique.

This book adopts traditional bottom up approach when introducing the architecture of computer networks. It consists of eight chapters where chapter 1 covers network concepts and philosophies that even junior instructors might benefit from reading it, chapter 2 to chapter 6 covers the TCP/IP reference model. Chapter 7 and chapter 8 cover advanced topics on Internet QoS and security, respectively. The protocol description text is interleaved with 56 representative open source implementations, ranging from the Verilog or VHDL code of codec, modem, CRC32, CSMA/CD, and crypto, to the C code of adaptor driver, PPP daemon and driver, longest prefix matching, IP/TCP/UDP checksum, NAT, RIP/OSPF/BGP routing daemons, TCP slow-start and congestion avoidance, socket, popular packages supporting DNS, FTP, SMTP, POP3, SNMP, HTTP, SIP, streaming, P2P, to QoS features such as traffic shaper and scheduler, and security features such as firewall, VPN, and intrusion detection. In addition, each open source is explained in a systematic way, including overview, data structures, call flow, algorithm, and code tracing. Furthermore, each open source is followed by hands-on exercises to equip readers with system-awareness and hands-on skills.

At the end of each chapter, besides written exercises, this book also provides hands-on Linux-based exercises which echo its goal again. It also provides end-of-chapter FAQ to help readers identify key concepts of each chapter. It also embeds 69 sidebars of Historical Evolution (33), Principle in Action (26), and Performance Matters (10) to highlight evolutions, principles, and performance numbers, respectively.

As compared to the most popular textbook on computer networks written by Kurose and Ross, this book emphasizes less on socket programming and java programming on applications, and network simulations. Kurose and Rosss book also spends more pages on discussing the underlying rationale on a specific topic, such as reliable transmission, which makes their book more suitable for undergraduate students. On the other hand, this book provides wider coverage on current technologies, especially on physical layer, Internet QoS, security, and wireless technologies, which makes it more suitable for senior undergraduate and graduate students in Computer Science or Electrical Engineering. We have maintained a Facebook community for Q&A at www.facebook.com/CNFBs, which is a plus for both instructors and students.

Here are two quotes from the book reviews: The exposure to real life implementation details in this book is phenomenal...Definitely one of the better books written in the area of Computer Networks. I have never seen a book giving such details on explaining the design and implementation of such practical systems...Those open source implementations are excellent demonstrations for practical networking systems.

7 Lessons

There are several lessons accumulated over the past two decades and summarized as follows.

1. **Development vs. Research**

   (a) Build the depth of the research team with the front line on development and the back line on research, which helps identifying real problems and feasible solutions.

   (b) The best way to tightly couple both lines is to send researchers to the front line for quite a while before they do research in the back line.

   (c) Develop first, then research. Research is the non-trivial parts identified in the process of development.

   (d) The performance numbers on most (¿90%) papers are from analysis or simulation. Very few are from the experiments on real implementations. The solutions on papers might not be feasible, and their problems might not be real either. There are very few societies in IEEE with a good balance between development and research, and, unfortunately, the communications society is not one of them.

   (e) The industry needs big development (i.e., products) and small research (i.e., patents), while the academia needs big research (i.e., papers) and small development (i.e., prototypes). To collaborate better, the industry needs to grow its research and the academia needs to grow its development.

2. **Research Roadmap vs. Random Picks**

   (a) Compared to random picks of topics, it is certainly better to form a research roadmap with a series of works addressing related problems in the same area, which helps researchers to construct deeper understanding about domain knowledge and related works.

   (b) However, dont rule out the possibility of innovation out of imagination. The off-roadmap topics could be rewarding too as we often see more clearly what goes wrong than the existing players when we are newcomers to an issue.

3. **Conferences vs. Journal/Magazines**

   (a) In US, it is very common to clock research by conference deadlines. However, it is difficult in Taiwan due to the constraints on travel budget. One could publish a dozen of journal papers per year but not even three conference papers per year. Thus, in Taiwan, we are forced to abandon the conference-driven model and embrace the journal-driven model which does not have clear clock ticks.

   (b) The review process in journals and magazines has been shortened compared to last decade, due to the on-line processing. The time-to-publish in journals and magazines becomes more comparable to conferences. However, in the computer society and communications society, several top conferences appear to be more influential than journals and magazines.

4. **Academic Services vs. Academic Cooperation**

   (a) Academic services through editorial boards, program committees, or technical committees might or might not bring academic cooperation. But knowing the rules of the game certainly helps in planning the publication venues.

   (b) It takes extra effort to build and maintain the external or international cooperation. But it still pays to do so because it brings in new or different thoughts and resources.

5. **Other Lessons**

   (a) Duplicating others (e.g. UNH/IOL) has no value.

   (b) Real traffic testing is indeed unique.

   (c) A work with high impact on the industry might not have high impact on the academia, and vice versa.

   (d) A high-impact paper might be rejected in its early version.

   (e) Many papers in top journals or conferences have low impact eventually. The review process can screen regarding quality but usually not impact.

References

[1] Ying-Dar Lin, Chia-Jen Wu, and Wei-Ming Yin. Pcup: Pipelined cyclic upstream protocol over hybrid fiber coax. *Network, IEEE*, 11(1):24–34, 1997.

[2] Ying-Dar Lin. On ieee 802.14 medium access control protocol. *Communications Surveys & Tutorials, IEEE*, 1(1):02–10, 1998.

[3] Ying-Dar Lin, Chen-Yu Huang, and Wei-Ming Yin. Allocation and scheduling algorithms for ieee 802.14 and mcns in hybrid fiber coaxial networks. *Broadcasting, IEEE Transactions on*, 44(4):427–435, 1998.

[4] Ying-Dar Lin and Chun-Mo Liu. A timestamp-sensitive scheduling algorithm for mpeg-ii multiplexers in catv networks. *Broadcasting, IEEE Transactions on*, 44(3):336–345, 1998.

[5] Ying-Dar Lin, Wei-Ming Yin, and Chen-Yu Huang. An investigation into hfc mac protocols: mechanisms, implementation, and research issues. *Communications Surveys & Tutorials, IEEE*, 3(3):2–13, 2000.

[6] Wei-Ming Yin and Ying-Dar Lin. Statistically optimized minislot allocation for initial and collision resolution in hybrid fiber coaxial networks. *Selected Areas in Communications, IEEE Journal on*, 18(9):1764–1773, 2000.

[7] Frank Yeong-Sung Lin, Wei-Ming Yin, Ying-Dar Lin, and Chih-Hao Lin. Optimal ranging algorithms for medium access control in hybrid fiber coax networks. *IEICE Transactions on Communications*, 85(10):2319–2326, 2002.

[8] Wei-Ming Yin, Chia-Jen Wu, and Ying-Dar Lin. Two-phase minislot scheduling algorithm for hfc qos services provisioning. *IEICE Transactions on Communications*, 85(3):582–593, 2002.

[9] Wei-Ming Yin and Ying-Dar Lin. Interleaving collision resolution engines in n-ary tree protocols. *Communications Letters, IEEE*, 5(12):494–496, 2001.

[10] Ying-Dar Lin and Yu-Ching Hsu. Multihop cellular: A new architecture for wireless communications. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1273–1282. IEEE, 2000.

[11] Ying-Dar Lin, Yu-Ching Hsu, Kuan-Wen Oyang, Tzu-Chieh Tsai, and Dong-Su Yang. Multihop wireless ieee 802.11 lans: A prototype implementation. *Communications and Networks, Journal of*, 2(4):372–378, 2000.

[12] Ying-Dar Lin, Shiao-Li Tsao, Shun-Lee Chang, Shau-Yu Cheng, and Chia-Yu Ku. Design issues and experimental studies of wireless lan mesh. *Wireless Communications, IEEE*, 17(2):32–40, 2010.

[13] Chia-Yu Ku, Ying-Dar Lin, Shiao-Li Tsao, and Yuan-Cheng Lai. Utilizing multiple channels with fewer radios in wireless mesh networks. *Vehicular Technology, IEEE Transactions on*, 60(1):263–275, 2011.

[14] Ying-Dar Lin, Shun-Lee Chang, Jui-Hung Yeh, and Shau-Yu Cheng. Indoor deployment of ieee 802.11 s mesh networks: Lessons and guidelines. *Ad Hoc Networks*, 9(8):1404–1413, 2011.

[15] Ying-Dar Lin, Cheng-Hsien Chang, and Yu-Ching Hsu. Bandwidth brokers of instantaneous and book-ahead requests for differentiated services networks. *IEICE transactions on communications*, 85(1):278–283, 2002.

[16] Yea-Li Sun, Chih-Chiang Chuang, and Ying-Dar Lin. Dynamic resizing of utilization target in measurement-based admission control. *Computer Communications*, 24(11):1097–1104, 2001.

[17] Shih-Chiang Tsao and Ying-Dar Lin. Pre-order deficit round robin: a new scheduling algorithm for packet-switched networks. *Computer Networks*, 35(2):287–305, 2001.

[18] Wei Huan-Yun, Chiang Ching-Chuang, and Lin Ying-Dar. Co-drr: An integrated uplink and downlink scheduler for bandwidth management over wireless lans. *IEICE transactions on communications*, 90(8):2022–2033, 2007.

[19] Shih-Chiang Tsao, Yuan-Cheng Lai, Le-Chi Tsao, and Ying-Dar Lin. On applying fair queuing discipline to schedule requests at access gateway for downlink differential qos. *Computer Networks*, 52(18):3392–3404, 2008.

[20] Ying-Dar Lin, Ching-Ming Tien, Shih-Chiang Tsao, Shuo-Yen Wen, and Yuan-Cheng Lai. Request scheduling for differentiated qos at website gateway. *Journal of Internet Technology*, 9(3):238, 2008.

[21] Ying-Dar Lin, Ching-Ming Tien, Shih-Chiang Tsao, Ruo-Hua Feng, and Yuan-Cheng Lai. Multiple-resource request scheduling for differentiated qos at website gateway. *Computer Communications*, 31(10):1993–2004, 2008.

[22] Hsu Yu-Ching, Lin Ying-Dar, and Chiang Mei-Yan. Two-stage dynamic uplink channel and slot assignment for gprs. *IEICE TRANSACTIONS on Communications*, 86(9):2694–2700, 2003.

[23] Yi-Neng Lin, Ying-Dar Lin, Yuan-Cheng Lai, and Che-Wen Wu. Highest urgency first (huf): A latency and modulation aware bandwidth allocation algorithm for wimax base stations. *Computer Communications*, 32(2):332–342, 2009.

[24] Ying-Dar Lin, Yi-Neng Lin, Shun-Chin Yang, and Yu-Sheng Lin. Diffserv edge routers over network processors: Implementation and evaluation. *Network, IEEE*, 17(4):28–34, 2003.

[25] Ying-Dar Lin, Huan-Yun Wei, and Kuo-Jui Wu. Ordered lookup with bypass matching for scalable per-flow classification in layer 4 routers. *Computer Communications*, 24(7):667–676, 2001.

[26] Huan-Yun Wei and Ying-Dar Lin. A survey and measurement-based comparison of bandwidth management techniques. *Communications Surveys & Tutorials, IEEE*, 5(2):10–21, 2003.

[27] Huan-Yun Wei, Shih-Chiang Tsao, and Ying-Dar Lin. Assessing and improving tcp rate shaping over edge gateways. *Computers, IEEE Transactions on*, 53(3):259–275, 2004.

[28] Ying-Dar Lin, Shih-Chiang Tsao, and Un-Pio Leong. On-the-fly tcp path selection algorithm in access link load balancing. *Computer communications*, 30(2):351–357, 2007.

[29] Kuo-Kun Tseng, Yuan-Cheng Lai, and Ying-Dar Lin. Perceptual codec and interaction aware playout algorithms and quality measurements for voip systems. *Consumer electronics, IEEE transactions on*, 50(1):297–305, 2004.

[30] Ying-Dar Lin, N-B Hsu, and Ren-Hung Hwang. Qos routing granularity in mpls networks. *Communications Magazine, IEEE*, 40(6):58–65, 2002.

[31] Hsu Nai-Bin, Lin Ying-Dar, Li Mao-Huang, and Lee Tsern-Huei. Service-sensitive routing in diffserv/mpls networks. *IEICE transactions on communications*, 84(10):2871–2879, 2001.

[32] Ying-Dar Lin, Nai-Bin Hsu, and Ren-Hung Hwang. Rpim-sm: extending pim-sm for rp relocation. *Computer Communications*, 25(18):1774–1781, 2002.

[33] Shih-Chiang Tsao, Yuan-Cheng Lai, and Ying-Dar Lin. Taxonomy and evaluation of tcp-friendly congestion-control schemes on fairness, aggressiveness, and responsiveness. *Network, IEEE*, 21(6):6–15, 2007.

[34] Shih-Chiang Tsao, Yuan-Cheng Lai, and Ying-Dar Lin. A fast-converging tcp-equivalent window-averaging rate control scheme. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2012 International Symposium on*, pages 1–6. IEEE, 2012.

[35] Po-ching Lin, Zhi-xiang Li, Ying-dar Lin, Yuan-cheng Lai, and Frank C Lin. Profiling and accelerating string matching algorithms in three network content security applications. *IEEE Communications Surveys and Tutorials*, 8(1-4):24–37, 2006.

[36] Ying-Dar Lin, Huan-Yun Wei, and Shao-Tang Yu. Building an integrated security gateway: Mechanisms, performance evaluations, implementations, and research issues. *Communications Surveys & Tutorials, IEEE*, 4(1):2–15, 2002.

[37] Ying-Dar Lin, Chih-Wei Jan, Po-Ching Lin, Yuan-Cheng Lai, et al. Designing an integrated architecture for network content security gateways. *IEEE Computer*, 39(11):66–72, 2006.

[38] Lin Ying-Dar, Lin Po-Ching, Tsai Meng-Fu, Chang Tsao-Jiang, and Lai Yuan-Cheng. Kp2padm: An in-kernel architecture of p2p management gateway. *IEICE transactions on information and systems*, 91(10):2398–2405, 2008.

[39] Ying-Dar Lin, Szu-Hao Chen, Po-Ching Lin, and Yuang-Chen Lai. A stream-based mail proxy with interleaved decompression and virus scanning. *Journal of Systems and Software*, 81(9):1517–1524, 2008.

[40] Ying-Dar Lin, Chia-Yu Ku, Yuan-Cheng Lai, and Chia Hung. In-kernel relay for scalable one-to-many streaming. *MultiMedia, IEEE*, 20(1):69–79, 2013.

[41] Po-Ching Lin, Ying-Dar Lin, T Lee, and Y Lai. Using string matching for deep packet inspection. *IEEE Computer*, 41(4):23–28, 2008.

[42] Po-Ching Lin, Ying-Dar Lin, and Yuan-Cheng Lai. A hybrid algorithm of backward hashing and automaton tracking for virus scanning. *Computers, IEEE Transactions on*, 60(4):594–601, 2011.

[43] Lin Po-Ching, Liu Ming-Dao, Lin Ying-Dar, and Lai Yuan-Cheng. Accelerating web content filtering by the early decision algorithm. *IEICE transactions on information and systems*, 91(2):251–257, 2008.

[44] Yi-Neng Lin, Ying-Dar Lin, Yuan-Cheng Lai, and Kuo-Kun Tseng. Modeling and analysis of core-centric network processors. *ACM Transactions on Embedded Computing Systems (TECS)*, 7(4):41, 2008.

[45] Yi-Neng Lin, Yao-Chung Chang, Ying-Dar Lin, and Yuan-Chen Lai. Resource allocation in network processors for network intrusion prevention systems. *Journal of Systems and Software*, 80(7):1030–1036, 2007.

[46] Yi-Neng Lin, Ying-Dar Lin, and Yuan-Cheng Lai. Thread allocation in cmp-based multi-threaded network processors. *Parallel Computing*, 36(2):104–116, 2010.

[47] Yi-Neng Lin, Ying-Dar Lin, Yuan-Cheng Lai, Chuan-Hung Lin, et al. Vpn gateways over network processors: Implementation and evaluation. *Journal of Internet Technology*, 11(4):457–463, 2010.

[48] Po-Ching Lin, Yin-Dar Lin, Yuan-Cheng Lai, Yi-Jun Zheng, and Tsern-Huei Lee. Realizing a sub-linear time string-matching algorithm with a hardware accelerator using bloom filters. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 17(8):1008–1020, 2009.

[49] Kuo-Kun Tseng, Ying-Dar Lin, Tsern-Huei Lee, and Yuan-Cheng Lai. Deterministic high-speed root-hashing automaton matching coprocessor for embedded network processor. *ACM SIGARCH Computer Architecture News*, 35(3):36–43, 2007.

[50] Kuo-Kun Tseng, Yuan-Cheng Lai, Ying-Dar Lin, and Tsern-Huei Lee. A fast scalable automaton-matching accelerator for embedded content processors. *ACM Transactions on Embedded Computing Systems (TECS)*, 8(3):19, 2009.

[51] Ying-Dar Lin, Kuo-Kun Tseng, Tsern-Huei Lee, Yi-Neng Lin, Chen-Chou Hung, and Yuan-Cheng Lai. A platform-based soc design and implementation of scalable automaton matching for deep packet inspection. *Journal of Systems Architecture*, 53(12):937–950, 2007.

[52] Ming-Wei Wu and Ying-Dar Lin. Open source software development: an overview. *Computer*, 34(6):33–38, 2001.

[53] Chi-Hung Chou, Tsung-Hsien Yang, Shih-Chiang Tsao, and Ying-Dar Lin. Standard operating procedures for embedded linux systems. *Linux Journal*, 2007(160):10, 2007.

[54] Ying-Dar Lin, I-Wei Chen, Po-Ching Lin, Chang-Sheng Chen, and Chun-Hung Hsu. On campus beta site: architecture designs, operational experience, and top product defects. *Communications Magazine, IEEE*, 48(12):83–91, 2010.

[55] Ying-Dar Lin, Chien-Chao Tseng, Cheng-Yuan Ho, and Yu-Hsien Wu. How nat-compatible are voip applications? *Communications Magazine, IEEE*, 48(12):58–65, 2010.

[56] Cheng-Yuan Ho, Fu-Yu Wang, Chien-Chao Tseng, and Ying-Dar Lin. Nat-compatibility testbed: an environment to automatically verify direct connection rate. *Communications Letters, IEEE*, 15(1):4–6, 2011.

[57] Cheng-Yuan Ho, Chien-Chao Tseng, Fu-Yu Wang, Jui-Tang Wang, and Ying-Dar Lin. To call or to be called behind nats is sensitive in solving the direct connection problem. *Communications Letters, IEEE*, 15(1):94–96, 2011.

[58] Ying-Dar Lin, Ren-Hung Hwang, Raghavendra Kulkarni, Shiau-Huey Wang, Chin-Yang Henry Tseng, and Chun-Hung Hsu. On campus ipv6 beta site: Requirements, solutions, and product defect evaluation.

[59] Ying-Dar Lin, Po-Ching Lin, Tsung-Huan Cheng, I-Wei Chen, and Yuan-Cheng Lai. Low-storage capture and loss recovery selective replay of real flows. *Communications Magazine, IEEE*, 50(4):114–121, 2012.

[60] Chia-Yu Ku, Ying-Dar Lin, Yuan-Cheng Lai, Pei-Hsuan Li, and KC-J Lin. Real traffic replay over wlan with environment emulation. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2406–2411. IEEE, 2012.

[61] Chun-Ying Huang, Ying-Dar Lin, Peng-Yu Liao, and Yuan-Cheng Lai. Stateful traffic replay for web application proxies. *Security and Communication Networks*.

[62] Y Lin, P Lin, and Y Lai. On-the-fly capture and replay mechanisms for multi-port network devices in operational networks. *Network and Service Management, IEEE Transactions on*, 11(2), 2014.

[63] Ying-Dar Lin, Chi-Heng Chou, Yuan-Cheng Lai, Tse-Yau Huang, Simon Chung, Jui-Tsun Hung, and Frank C Lin. Test coverage optimization for large code problems. *Journal of Systems and Software*, 85(1):16–27, 2012.

[64] Y Lin, P Lin, S Wang, I Chen, and Y Lai. Pcaplib: A system of extracting, classifying, and anonymizing real packet traces. *IEEE Systems Journal*.

[65] Ying-Dar Lin, Chun-Nan Lu, Yuan-Cheng Lai, and Zongo Pawendtaore Eliezer. Bug traces: identifying and downsizing packet traces with failures triggered in networking devices. *Communications Magazine, IEEE*, 52(4):112–119, 2014.

[66] LAI Yuan-Cheng, LIN Ying-Dar, WU Fan-Cheng, Tze-Yau Huang, et al. Embedded taint-tracker: Lightweight run-time tracking of taint data against buffer overflow attacks. *IEICE TRANSACTIONS on Information and Systems*, 94(11):2129–2138, 2011.

[67] Tsung-Huan Cheng, Ying-Dar Lin, Yuan-Cheng Lai, and Po-Ching Lin. Evasion techniques: Sneaking through your intrusion detection/prevention systems. *Communications Surveys & Tutorials, IEEE*, 14(4):1011–1020, 2012.

[68] I-Wei Chen, Po-Ching Lin, Tsung-Huan Cheng, Chi-Chung Luo, Ying-Dar Lin, Yuan-Cheng Lai, and Frank C Lin. Extracting ambiguous sessions from real traffic with intrusion prevention systems. *IJ Network Security*, 14(5):243–250, 2012.

[69] Cheng-Yuan Ho, Ying-Dar Lin, Yuan-Cheng Lai, I-Wei Chen, Fu-Yu Wang, and Wei-Hsuan Tai. False positives and negatives from real traffic with intrusion detection/prevention systems. *International Journal of Future Computer and Communication*, 1(2), 2012.

[70] Ying-Dar Lin, Yuan-Cheng Lai, Cheng-Yuan Ho, and Wei-Hsuan Tai. Creditability-based weighted voting for reducing false positives and negatives in intrusion detection. *Computers & Security*, 39:460–474, 2013.

[71] Ying-Dar Lin, Tzung-Bi Shih, Yu-Sung Wu, and Yuan-Cheng Lai. Secure and transparent network traffic replay, redirect, and relay in a dynamic malware analysis environment. *Security and Communication Networks*, 7(3):626–640, 2014.

[72] Ying-Dar Lin, Chia-Yin Lee, Yu-Sung Wu, Pei-Hsiu Ho, Fu-Yu Wang, and Yi-Lang Tsai. Active versus passive malware collection. *Computer Magazine, IEEE*, 47(4):59–65, 2014.

[73] Ying-Dar Lin, Yi-Ta Chiang, Yu-Sung Wu, and Yuan-Cheng Lai. Automatic analysis and classification of obfuscated bot binaries. *International Journal of Network Security*, 16(6):506–515, 2014.

[74] Kuochen Wang, Chun-Ying Huang, Li-Yang Tsai, and Ying-Dar Lin. Behavior-based botnet detection in parallel. *Security and Communication Networks*, 2013.

[75] Ying-Dar Lin, Chia-Yin Lee, and Hao-Chuan Tsai. Redefining security criteria for networking devices with case studies. *Security & Privacy, IEEE*, 12(1):43–53, 2014.

[76] Ying-Dar Lin, Cheng-Yuan Ho, Yuan-Cheng Lai, Tzu-Hsiung Du, and Shun-Lee Chang. Booting, browsing and streaming time profiling, and bottleneck analysis on android-based systems. *Journal of Network and Computer Applications*, 36(4):1208–1218, 2013.

[77] LIN Ying-Dar, Kuei-Chung Chang, LAI Yuan-Cheng, and LAI Yu-Sheng. Reconfigurable multi-resolution performance profiling in android applications. *IEICE TRANSACTIONS on Information and Systems*, 96(9):2039–2046, 2013.

[78] Y-D Lin, ET-H Chu, Y-C Lai, and T-J Huang. Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds. *IEEE Systems Journal*, 2013.

[79] Ying-Dar Lin, Yuan-Cheng Lai, Chien-Hung Chen, and Hao-Chuan Tsai. Identifying android malicious repackaged applications by thread-grained system call sequences. *Computers & Security*, 39:340–350, 2013.

[80] Ying-Dar Lin, ET-H Chu, Shang-Che Yu, Y-C Lai, and T-J Huang. Improving accuracy of automated gui testing for embedded systems. *Software magazine, IEEE*, 2014.

[81] Ying-Dar Lin, Jose F. Rojas, Edward T.-H Chu, and Yuan-Cheng Lai. On the accuracy, efficiency, and reusability of automated test oracles for android devices. *Software Engineering, IEEE Transactions on.*

[82] Ying-Dar Lin, Ren-Hung Hwang, and Fred Baker. *Computer networks: an open source approach.* McGraw-Hill, 2012.

Biography

Ying-Dar Lin is Distinguished Professor of Computer Science at National Chiao Tung University (NCTU) in Taiwan. He received his Ph.D. in Computer Science from UCLA in 1993. He served as the CEO of Telecom Technology Center during 2010-2011 and a visiting scholar at Cisco Systems in San Jose during 20072008. Since 2002, he has been the founder and director of Network Benchmarking Lab (NBL, www.nbl.org.tw), which reviews network products with real traffic. He also cofounded L7 Networks Inc. in 2002, which was later acquired by D-Link Corp. In May 2011, he founded Embedded Benchmarking Lab (www.ebl.org.tw) to extend into the review of handheld devices. His research interests include design, analysis, implementation, and benchmarking of network protocols and algorithms, quality of services, network security, deep packet inspection, P2P networking, and embedded hardware/software co-design. He recently stepped into software defined networking (SDN) and was appointed as a Research Associate from June 2014 by Open Networking Foundation (ONF). His work on multi-hop cellular was the first along this line, and has been cited over 600 times and standardized into WLAN mesh (IEEE 802.11s), WiMAX (IEEE 802.16j), Bluetooth (IEEE 802.15.5), and 3GPP LTE-Advanced. He was elevated to IEEE Fellow in 2013 for his contributions to multi-hop cellular communications and deep packet inspection. He is also an IEEE Distinguished Lecturer for 2014 & 2015, and currently on the editorial boards of IEEE Transactions on Computers, IEEE Computer, IEEE Network, IEEE Communications Magazine - Network Testing Series, IEEE Wireless Communications, IEEE Communications Surveys and Tutorials, IEEE Communications Letters, Computer Communications, Computer Networks, and IEICE Transactions on Information and Systems; and the lead guest editor of several special issues of IEEE journals and magazines. He published a textbook "Computer Networks: An Open Source Approach" (www.mhhe.com/lin), with Ren-Hung Hwang and Fred Baker (McGraw-Hill, 2011).