

Wenda Ni, Ph.D.

27220 SE 18TH PL
Sammamish, WA 98075
United States

M: +1 (425) 984 9332
E: wonda.ni@gmail.com

EDUCATION

- **Tsinghua University** Beijing, P. R. China
Ph.D., Dept. of Electronic Engineering Sept. 2005 - Jul. 2010
 - Topics: Network design, operation, and management with a focus on survivability, service differentiation, and two-layer traffic engineering in telecom transport networks
 - Thesis: Resource optimization and service quality in WDM optical networks
- **Tsinghua University** Beijing, P. R. China
Bachelor Degree of Engineering, Dept. of Electronic Engineering Sept. 2001 - Jul. 2005
 - Major: Physical electronics and Optoelectronics
 - GPA: 88.8/100 Rank: 5/80

WORK EXPERIENCE

- **ByteDance Inc.** Bellevue, WA, USA
Network R&D engineer, ByteDance Networking Nov. 2019 - present
 - Topics: Lambda network operating system, a customized version of SONiC, over world's bleeding-edge switching chips (25G, 100G, 400G, 800G)
 - Extended sub port router interface support over link aggregation group (LAG) parent port, and developed virtual switch (vs) unit test <https://github.com/Azure/sonic-swss/pull/1235>
 - Translated production Top-of-Rack (ToR) switch configuration of various vendors into Lambda configuration schema. Developed the first config file for Lambda with control-plane protocols, BGP, LLDP, NTP, etc., up and running. Implemented script to auto-generate default S0 (ToR) switch config.
 - Designed and developed buffer & QoS profile for RDMA and non-RDMA traffic forwarded over Trident3 (TD3) chip. No TD3 buffer profile was available for reference in public community as of the design was made.
 - Customized NTP service config to run NTP service in default virtual route and forward (VRF) instance and over loopback interface.
 - Proposed host interface ICMP and ICMPv6 trap types in OCP switch abstraction interface (SAI). <https://github.com/opencomputeproject/SAI/pull/1044>
 - Brought up Lambda OS on Celestica Questone platform. Developed fan control and monitoring script to remedy fan daemon crash.
 - Brought up Lambda OS on Quanta IX8E-56X and Ruijie B6510-48VS8CQ platforms; Save the system from kernel panic and oops.
 - Developed standard workflow to code submission and to embrace community code updates. Pioneered and accomplished the first code updates to synchronize with community 201911 branch, filling a total gap of around 180 commits behind.
 - Ported Broadcom SAI v3.6.7.1 (OCP SAI v1.4.1, Broadcom SDK v6.5.16) to run on SONiC 201911 branch image (Broadcom SAI v3.7.3.3-2, OCP SAI v1.5.1, Broadcom SDK v6.5.16): Hacked Makefile to enable incremental SAI-only build to avoid rebuilding the underlying SDK codes; Developed workaround to SAI package build failure on Debian 9 Stretch (Broadcom SAI package has to be built against Stretch as its running environment relies on the libprotobuf10 library (v3.0.0-9), which is not available on earlier Debian 8 release—Jessie); Debugged syncd crash on original Broadcom SAI release, and fixed SAI codes.

- Upgraded OCP SAI header to v1.5.1, v1.5.2, and enabled Lambda image build against Broadcom SAI v3.6.7.1
- SAI change without support from Broadcom: Suppressed global pause ability advertisement on TD3 server-facing ports (23 insertions); Implemented host interface ICMP and ICMPv6 trap, rate limiting, and stat counter attachment (171 insertions); Enabled ingress priority group (PG) packet and dropped packet accounting for generic traffic class (TC) to PG mapping (127 insertions, 12 deletions); Identified through code reading and fixed incorrect queue to TC reverse mapping cache content in a TC to queue mapping update; Identified through code reading and fixed potential PFC counter value mismatch in a get port stats API call; Restricted flex-counter-based egress TC packet accounting to unicast packets only (62 insertions, 22 deletions); Implemented EFP-based queue IPv6 stats (1703 insertions, 630 deletions); Fixed uninitialized multicast queue entry (MCQE) pool for lossy traffic (189 insertions, 90 deletions); Fixed incorrect PG to service pool mapping in a PG to pool mapping set operation; Implemented ingress packet sampling slow-path (packet samples destined to CPU) pipeline (1183 insertions, 276 deletions); Implemented egress packet sampling slow-path (packet samples destined only to CPU on TD3) pipeline (436 insertions, 263 deletions); Identified from diag shell that field entry active in PFC storm mistakenly matches and drops packets mapped on a particular TC and destined to CPU (e.g., RDMA packet sampling destined to CPU) post PFC storm restoration, and fixed field entry status at PFC storm restoration event; Designed and implemented ACL table VLAN egress binding, whose hardware behavior was incorrect with regard to SAI specification in original Broadcom SAI implementation (888 insertions, 254 deletions); Developed generic control flow for buffer profile attribute set, and implemented set operation for PG/queue min, PG headroom, PG XON offset (195 insertions, 89 deletions); Implemented runtime buffer headroom pool resizing (82 insertions, 40 deletions); Removed unnecessary DSCP to TC map entry reset when binding the map to a port (29 insertions, 8 deletions); Implemented runtime PG or queue buffer profile migration (252 insertions, 48 deletions).
- Designed and implemented ACL table VLAN egress binding, from command line interface (CLI) down to Broadcom SAI. Optimized hardware resource in binding ACL table to VLAN ingress. Extended critical resource management (CRM) to support per stage (ingress/egress) per bind point type ACL resource warning messaging and message accounting. Designed and implemented ACL table VLAN egress binding in Broadcom SAI, whose original implementation was incorrect. The uniqueness as well as the challenge of this feature is that when binding an ACL table to VLAN egress, an ACL rule in it cannot be functionality-wise fulfilled by only one field entry, but requires more than one field entry to accommodate. Identified and confirmed by vendor the same issue on commercial switches in production, which triggered fixes by network engineers in a timely manner. Contributed design practices to community as pull request review comments <https://github.com/Azure/sonic-swss/pull/1218>
- Added ICMP and ICMPv6 packet type support to control plane policing (CoPP).
- Customized zero touch provisioning (ZTP) with a thorough grasp of its internal execution flow and state machine transition (849 insertions, 594 deletions). Developed configuration changes to Kea DHCP server to offer options 66 and 67 for ZTP support. Enabled robust switch configuration update scenario with and without image upgrade. Unveiled the significance of management VRF in our unique test setup. Contributed some of the code changes to community as review feedback <https://github.com/Azure/sonic-ztp/pull/12>
- Proposed custom port and queue IPv6 stats in SAI. Gained deep insights into Broadcom TD3 stat counter capability as well as SAI stat counter implementation architecture, which leverages chip native statistics, flexible counters, and field processor (FP). Concluded from Broadcom document study (further confirmed by Broadcom) that egress flexible counters cannot resolve IPv6 packets and does not support user defined fields. Proved TD3 egress field processor (EFP) does not support TCAM slice sharing among field groups through code implementation and platform test.

- Designed and implemented EFP-based queue IPv6 stats in Broadcom SAI, syncd, swss, and up to show CLI. Designed mechanism to enable EFP TCAM slice sharing between user egress ACL tables (L3, L3v6, PFC watchdog drop action, sFlow rate limiting) and queue IPv6 stats.
- Enabled separate buffer pools for CPU traffic at both ingress and egress. Expanded XGS-device memory management unit (MMU) knowledge and understanding to non-unicast traffic in terms of packet flow, resource accounting, and admission control. Identified and fixed the issue of uninitialized MCQE pool for lossy traffic (pool shared limit being zero) in SAI MMU setting. Identified and fixed incorrect PG to service pool mapping caused by SDK API misuse in original SAI.
- Pioneered GitLab continuous integration (CI). Enabled GitLab CI pipeline for automated Lambda image build and Broadcom SAI v3.6.7.1 debian package build at merge request creation and updates.
- Enabled and customized sFlow with a full grasp of original community design and implementation, which, however, only support ingress sFlow (ingress packet sampling slow path and ingress sFlow v5 encapsulation). Identified code gap in Broadcom SAI v3.6.7.1 to enable ingress packet sampling slow-path pipeline. Filled the gap by implementing SAI logic of genetlink host interface creation, knet rx filter creation, samplepacket host interface trap creation, CPU COS MAP table entry creation (to direct packet samples to a specific CPU queue), host table entry creation for samplepacket trap id, and packet sampling enable/disable on physical port. Enabled configurable sample truncation size by user at knet rx filter level.
- Extended sFlow functionality from ingress only to egress: Designed and implemented egress packet sampling slow-path pipeline in SAI; Designed and implemented egress-sampled packet processing in knet rx filter; Designed and implemented egress sFlow v5 encapsulation in host sflow daemon (hsflowd); Designed and implemented CONFIG_DB schema to configure egress sFlow and allow sFlow enable/disable at per port per direction level; Contributed bug fixes and design improvements to community as pull request review comments (<https://github.com/Azure/sonic-swss/pull/1012>, 1224, 1427, <https://github.com/Azure/sonic-utilities/pull/592>).
- Brought up virtual switch (vs) tests in swss and fr GitLab CI pipelines. Fixed vs tests on speed buffer set, portchannel oper status down, ACL rule redirect action, ACL mirror table ingress & egress creation in separated v4 & v6 mirror table mode, and sFlow default global, global all, and config deletion. Contributed fixes to community as code commit (<https://github.com/Azure/sonic-swss/pull/1452>) and review comments (<https://github.com/Azure/sonic-swss/pull/1030>, 1397, 1095). Reduced vs docker image build time by half (~10 minutes) through incremental docker image build at CI pipeline build stage. Reduced vs test run time from over 120 minutes to ~28 minutes through test parallelization, which breaks and classifies vs tests into 6 parallel pipelines, at CI pipeline test stage.
- Brought up utility unit tests in utilities GitLab CI pipeline. Fixed drop counter test failure at more than 1 test run in persistent docker container.
- 2020 mid-year performance review: **Performance E (exceed expectations)**—“Often exceeds requirements: Often exceeds requirements in terms of efficiency and quality of work with significant contributions. Gains positive feedback from most internal and external clients”; **Engagement E (exceed expectations)**—“Highly engaged; has strong ownership as a role model”.
- Enhanced sub port router interface: Inherited parent port MTU changes at both SAI and kernel levels, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1479>); Accelerated ECMP hardware convergence at parent port oper status changes, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1492>); Enabled linking interface ingress (i.e., enslaving interface) to a non-default VRF, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1521>); Updated Port object reference count, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1712>, 1767); Enabled mirror monitor port via a local sub port router interface, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1712>, 1767);

[//github.com/Azure/sonic-swss/pull/1727,1725](https://github.com/Azure/sonic-swss/pull/1727,1725)); Developed vs unit tests to validate physical port host interface VLAN tag attribute (<https://github.com/Azure/sonic-swss/pull/1634>), sub port interface ingress to a VNET (<https://github.com/Azure/sonic-swss/pull/1642>), and APPL_DB processing sequence of sub port interface keys (<https://github.com/Azure/sonic-swss/pull/1663>).

- Designed KLISH CLI on QoS (mapping, scheduler, WRED/ECN), MMU, and PFC watchdog config commands, and watermark show & clear commands. Implemented KLISH CLI XML on QoS (show, WRED/ECN config), MMU (show), ACL (show & config), counters (port, queue, PFC, IPv6) (show & clear), watermark (PG, buffer pool) (show & clear), and ZTP (show & config). Developed or customized underlying utility scripts to support KLISH CLI usage, most notably, runtime buffer profile update on PG min, PG XON offset, and PG headroom, runtime PG/queue buffer profile migration, runtime buffer pool headroom resizing, runtime scheduler type and/or weight change, runtime PFC enable/disable on TC, and runtime DSCP to TC remapping.
- Designed and implemented ACL table port and/or vlan binding and unbinding on the fly to accommodate KLISH ACL CLI behavior. Introduced ACL rule id to priority mapping to flip rule priority ranking. This is to conform to vendor CLI convention that smaller rule id is of higher rule priority.
- Designed and Implemented PFC watchdog self-adaptation to PFC enable status change on TC. This removes the need to toggle PFC watchdog stop and start by user at the occurrence of such a change. Contributed to community <https://github.com/Azure/sonic-swss/pull/1612,1691,1696,1697>, <https://github.com/Azure/sonic-sairedis/pull/814>
- Backported Broadcom SAI DSCP to TC map set operation from v3.7.5.2 to v3.6.7.1 to enable DSCP to TC runtime remapping. Backported host interface trap action from v3.7.5.2 to v3.6.7.1 to refactor SAI trap action implementation from {copy to CPU, drop} combination to unmodified packet redirect port. Identified PFC back pressure remaining in effect during PFC storm in Broadcom SAI v3.6.7.1 to cause lossless packets not drained from buffer, and backported fix from v3.7.5.2
- Designed and implemented runtime PG or queue buffer profile migration (from lossy to lossless, and vice versa).
- Co-designed link flap error-disable, and conducted intensive review on code implementation.
- Designed control flow to enable resource availability check at KLISH CLI ACL table creation and ACL table binding to protect against orchagent crash arising from resource insufficiency.
- Fixed kernel call trace on enabling sFlow—“WARNING: CPU: 2 PID: 22 at /sonic/src/sonic-linux-kernel/linux-4.9.189/net/core/skbuff.c:658 skb_release_head_state+0x98/0xa0”—caused by original Broadcom knet driver implementation.
- Identified via core dump analysis stack overflow caused by recursive calls to nonblocking `write` characterized by `EAGAIN` error return in the original hsfowd implementation of Redis async command, and fixed by introducing `sleep` when such a call to `write` is captured.
- Customized host sflow daemon by hacking source codes: Aligned input and output interface indices in sFlow v5 header with interface-name-to-index mapping used by SNMP (Our approach is simpler than and compatible with the community one in <https://github.com/Azure/sonic-buildimage/pull/4794>); Designed and implemented control flow to enable configurable source IP address by user in assembling a sFlow v5 datagram; Designed and implemented control flow to enable user-configurable source interface to automatically derive source IP address in assembling a sFlow v5 datagram; Enabled IPv6 agent address population in sFlow v5 header, and contributed code change to community as review comment (<https://github.com/sflow/host-sflow/pull/43>); Enabled flexible raw packet header size, determined by user-configurable sample truncation size, for a flow sample.
- Delegated hsfowd process control and monitor to supervisord for automatic restart at process exit.

- 2020 annual performance review: **ByteStyle E (exceed expectations)**—“Always practices ByteStyle and has a positive influence as a role model”; **Engagement E (exceed expectations)**.
- Enhanced ACL: Proposed custom attribute, priority, to SAI ACL table object as a hint for TCAM slice sharing, designed swss processing schema & flow, and implemented attribute handling in Broadcom SAI. Enabled TCAM slice sharing among orthogonal ACL table types (e.g., between L3 and L3v6 tables) via the notion of logical tables on TD3 to double the number of ACL tables user can create at ingress that need not satisfy mutual exclusion requirement among them. Reduced TCAM usage from triple-wide to double-wide mode for host interface application. Enabled stage migration and ACL rule add/removal in a previously-bound, currently-unbound ACL table.
- SAI change without support from Broadcom: Enabled logical tables on TD3 for TCAM slice sharing among mutually exclusive field groups (74 insertions, 7 deletions); Created but not installed ACL entry in an unbound ACL table (75 insertions, 36 deletions); Designed and implemented ACL rule packet sampling action to a mirror session (433 insertions, 97 deletions); Loaded CUNCUN v6.1.3, which is upgraded from v5.3.3 with the corresponding SDK patch, at SAI initialization (115,415 insertions, 56,146 deletions).
- Enabled truncation of mirrored packet (fixed truncation size on TD3). Contributed to community fixes to mirror orch <https://github.com/Azure/sonic-swss/pull/1761>, 1800
- Designed and implemented ACL rule packet sampling action to a mirror session starting from orchagent down to Broadcom SAI and up to KLISH CLI.
- Designed software architectural change to enable packet mirror and/or packet sampling action installation to ACL rules of all ACL table types (e.g., L3 and L3v6), and conducted intensive code review.
- Unlocked ERSPAN IPv6 underlay capability on TD3 SDK 6.5.16, which requires CUNCUN upgrade from v5.3.3 to v6.1.3, together with SDK patch. Enabled user to configure DSCP value in underlay IPv6 header by fixing SDK. Contributed to community <https://github.com/Azure/sonic-swss/pull/1817>
- Enhanced mirroring: Observed LAG member status change at runtime to update mirror monitor port selection. Contributed to community <https://github.com/Azure/sonic-swss/pull/1827>, 1846
- Enhanced sub port router interface: Fixed kernel admin status, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1873>); Fix kernel MTU inheritance from parent port, and developed vs unit test (<https://github.com/Azure/sonic-swss/pull/1874>).
- Hacked SDK (200 insertions, 31 deletions) to enable fine-grained TCAM slice sharing among ACL tables of the same type (e.g., among mutually exclusive L3 tables). Identified 4 SDK bugs on ACL logical tables, and fixed 3 of them. Hacked SDK (221 insertions, 21 deletions) to bypass hardware limitation on the number of ACL tables user can create (32 on TD3) via logical table select TCAM entry sharing among ACL tables of the same type and priority. With this hack, the number of ACL tables user can create goes unlimited (set limit to 1024 in software), no longer constrained by the number of physical entries in logical table select TCAM.
- **Earned July-August 2021 ByteStyle Award—Aim for the highest (最有字节范奖之追求极致奖):** Raise the bar; Find the best solutions by widening one’s own perspective; Distill ideas down to their fundamental truths; Keep learning and growing. **Earned July-August 2021 Outstanding Team Award (优秀团队奖) being a team member.**
- Fixed Broadcom SAI 3.6.7.1 on binding ACL table to physical port and VLAN at the same time, in which case the original Broadcom SAI incorrectly implements a logical AND operation to match both physical port and outer VLAN ID in one single entry, defining a hardware behavior not compliant with SAI specification (263 insertions, 233 deletions).
- 2021 mid-year performance review: **Performance E (exceed expectations); Engagement E (exceed expectations).**

- Enabled DSCP match in L3 and L3v6 ACL tables. Enabled IP type match in user-defined ACL table. Disabled ACL rule counter action by default, and enabled runtime counter attachment/detachment to ACL rules.
- Hacked SDK to enable flex counter pool sharing among ACL tables that share TCAM slices. This eliminates the limit from original SDK on the number of field groups with counters at ingress being upper bounded by the number of flex counter pools (92 insertions, 7 deletions).
- Developed ACL table resource limit check at KLISH ACL table binding on the number of tables per priority and the total number of tables and at KLISH ACL rule creation on the number of tables with counters.
- Introduced severity level NOTICE between WARN and INFO into Broadcom SDK logging (BSL) system to log SDK messages to `/var/log/debug` but not to `/var/log/syslog` (53 insertions, 38 deletions).
- Developed ACL entry resource limit check for egress ACL tables at KLISH ACL table binding and rule creation.
- Identified and fixed bug in processing packet samples that are marked both ingress and egress sampled.
- Enabled configuring mirror session source interface to derive underlay source IP.
- Developed vs test coverage for CRM divide by zero processing path, and contributed to community <https://github.com/Azure/sonic-swss/pull/2028>, 2030
- Gained deeper insights (limitation, packet drop events inclusive to drop stats, interference) into TD3 egress counter system (MMU, ACL, and flex counter) via experimenting with counter and ACL use cases put all together. Redefined egress ACL rule default counter status (enable counter action for drop rule and disable counter action for forward rule) to leave EFP-based queue IPv6 stats to be accurate.
- Released TCAM occupancy when unbinding ACL table from all bind points. Issued DEL op before SET for rule update to ACL table not present in ASIC to have correct match fields to be installed to TCAM. This is to accommodate `Consumer::addToSync` behavior on absent field inheritance among consecutive SET ops.
- Developed vs test for ACL table stage change, priority change, pending bind points, unbound ACL table, rule action change, default counter action.
- Enabled sending sFlow v5 datagram over IPv4 or IPv6 via management VRF.
- Identified and fixed SDK 6.5.16 bug on slice reference count in the case of ACL table auto expansion to third TCAM set.
- Fixed mirror session status update on destination IP change within directly connected subnet.
- Developed mirror session limit check. Developed TCAM slice limit check at KLISH ACL table binding.
- On Tomahawk4 (TH4) and Trident4 (TD4), designed egress ACL table layout that hosts system usages such as PFC watchdog, packet sample rate limiting, and mirror policing, while accommodating user ACL applications.
- Redesigned queue IPv6 counter implementation that leverages flex counter 2.0 to reduce ACL entry consumption from 896 to 2 entries on TH4, and to 56 entries on TD4 (with the capability to further reduce ACL entry consumption to 1 entry on TD4).
- Enabled hardware logical table for TCAM slice sharing between IPv4 and IPv6 ACL table types at TD4 egress.
- Proposed the notion of user-level logical table at TD3 and TH4 egress, which does not support hardware logical table, to enable TCAM slice sharing among mutually-exclusive ACL tables of the same type. This lifts the ACL table limit from 1 per table type to 1024 among all table types (limit set in software) at egress.

- Proposed the notion of shadow table and shadow key to enable TCAM slice sharing between IPv4 and IPv6 table types at TD3 egress. This increases the IPv4 physical entry capacity from 8 entries only to up to 512 entries (64 times). Implemented the feature by hacking SDK 6.5.16 as well as changes in SAI.
- On TD4, proposed dynamic flex counter object reuse and flex counter action reuse among ACL tables that share the same set of TCAM slices at ingress. This eliminates the limit on user ACL table creation arising from the limit on the number of flex counter objects and actions available to users. At egress, fixed flex counter object allocation scheme used in Broadcom SAI that causes counter index delivery conflict when a packet hits entries in both IPv4/IPv6 and L2 tables, even if they are installed in different TCAM sets. Flex counter object represents hardware metadata bus resource that delivers counter index to flex counter modules sitting at the end of ingress and egress pipelines.
- Designed MMU profiles for TH4 and TD4, leveraging the feature of cooperative buffer update at ingress and egress available on LTSW devices. Designed and implemented near-zero headroom pool, i.e., headroom pool recycle, for better buffer resource utilization under lossy-only traffic.
- Redesigned and refactored KLISH ACL table resource limit check at new ACL table creation to reduce the computation time from over 30 seconds down to 1 second when the number of ACL tables created reaches over 600.
- 2021 annual performance review: **ByteStyle E (exceed expectations); Engagement E (exceed expectations)**.
- Proposed system port class ID reuse among different packet processing pipes to overcome hardware limitation on class ID resources.
- Fixed alpha set & get in Broadcom SAI 5.0.0.8.
- Initialized MMU profile for multicast packets in terms of multicast cell queue entry (MCQE) min, MCQE service pool, and queue to service pool mapping.
- Implemented PFC pause status get in HSDK 6.5.25 to support queue pause status query in SAI, a foundation to software PFC watchdog storm detection. Pause status is not exposed to logical table, but is only available in physical table.
- TH4/TD4 MMU & QoS validation and fix on Broadcom SAI 5.0.0.8: Applied separate PG to service pool mapping profile to CPU port under TH4/TD4 profile-id-based mapping architecture; Set lossless flag to PFC-enabled PGs; Set PFC refresh time on TD4; Initialized PG to PFC priority mapping in TH4 config yml to enable PG to generate PFC frames in XOFF state; Set ECN marking on TH4.
- Observed significant behavioral changes to default ACL action resolution on TD4, and restored the same behavior as TD3 and TH4 by using entry level strength based resolution at field group creation.
- Designed and validated egress counter system on TH4 and TD4. Included MMU drop, pipeline drop, and ACL drop into queue drop stats, which only contain MMU drop in original Broadcom SAI implementation.
- Designed and implemented LAG stats, ingress received packets/bytes, ingress discarded packets, egress transmitted packets/bytes, and egress discarded packets, using flex counter 2.0.
- Designed and implemented LAG ingress ACL bind under system port bitmap on TH4.
- Designed and implemented LAG egress ACL bind under the notion of interface class port on TH4/TD3 and L2 interface on TD4.
- Figured out 3 methods to keep packets residing in buffer for testing purpose on LTSW devices. Corrected original Broadcom SAI implementation on `SAI_PORT_ATTR_PKT_TX_ENABLE`.
- Proposed and validated single buffer pool design that enables traffic-hitless buffer profile change (from lossy to lossless).
- Changed MAC default behavior to consume pause and PFC frames on LTSW devices to prevent frame broadcast in VLAN domain.

- Redesigned and rewrote PFC to queue map to ensure correct behavior under PFC storm on LTSW devices, which employ profile-id-based map attachment to port.
- Implemented user-level logical table at TD3 and TH4 egress on Broadcom SAI 7.1.0.0. This lifts ACL table limit from 1 to many, and enables multiple lossless TCs with drop action under PFC storm. User-level logical tables are hosted by global system ACL table of the same type, and reuse the corresponding flex counter action on TH4 to eliminate ACL table limit arising from the number of flex counter actions.
- Designed and implemented PFC watchdog hardware offload. Our design is backward-compatible to accommodate all hardware offloading scenarios: no hardware offload, full hardware offload, and partial hardware offload (hardware detection & software restoration, and software detection & hardware restoration).
- Enhanced PFC watchdog with in-storm config change.
- Fixed the blind spot of ACL entry resource limit check at the very first bind point bind for egress ACL tables.
- Redesigned KLISH CLI to support per-port buffer profile.
- Designed and implemented ACL user defined field (UDF) from orchagent down to sairedis and up to KLISH CLI. Developed vs test that enables smooth integration with Broadcom SAI.
- Fixed mirroring: Removed VLAN tag if mirror packets are sent out of VLAN, and VLAN ID is monitor port native VLAN.
- Handled out of order ACL tasks: physical port ACL table bind before LAG leave; physical port LAG join before ACL table unbind.
- Designed control flow for Inband Flow Analyzer (IFA) 2.0 transit functionality on TH4 and TD4, compliant with SAI TAM model and object definition. Implemented software from orchagent down to Broadcom SAI and up to utility and KLISH CLI.
- Developed Broadcom SAI support for ERSPAN type III, and runtime mirror session encap change between ERSPAN type I and type III.
- Defined new hostif trap type for mirror-to-CPU packets. Designed and implemented rate limiting on mirror-to-CPU packets in Broadcom SAI, covering TD3, TH4, and TD4. Designed and implemented CPU-side receive scheme that inhibits kernel network stack forwarding to avoid duplicate packet forwarding. Employed the idea of tunneling (RSPAN) to further distinguish ingress-mirrored and egress-mirrored packets, and direct them to different netif/net_device/hostif destinations.
- **Earned 2022 Network Systems (NS) All-Hands Topic Most Popular Award (网络系统与硬件主题分享最受欢迎奖)**
- Designed and implemented sFlow hardware offload, which accommodates all hardware offloading scenarios: full ingress & egress offload, partial offload (ingress only or egress only), and no hardware offload.
- Designed and validated hardware offload solution to egress sFlow on TH4, which does not have native hardware offload support for egress sFlow.
- Extended inband telemetry functionality to IFA 1.0 transit on TH4 with minor code changes, and runtime shift between IFA 2.0 and IFA 1.0.
- Lifted ACL table limit at ingress by implementing the notion of user-level logical table.
- Evaluated and validated hardware capability to match VxLAN overlay 5 tuples on TH4 and TD4.
- In truncated packet sample case, pinpointed the root cause of incorrect frame length population in sFlow v5 raw packet header down to Broadcom knet psample callback (kernel 4.19.0-12-2), and developed fix.
- Enabled hosting user ACL tables in system COPP/trap field group to fully exploit hardware resource capacity.

- Enhanced SAI ACL with new match fields and actions: Match next hop/next hop group; Action to redirect packet to dynamic load balancing (DLB) next hop group; Action to enable adaptive routing and switching (ARS); Match RoCEv2 BTH reserved field using UDF yet not consuming additional container (i.e., leveraging the notion of UDF container sharing among orthogonal packet types).
- Pinpointed the root cause of transient packet match on all ports at new ACL rule installation down to Broadcom SAI, and designed remedies to inhibit such a match for both ingress and egress ACLs.
- Ported ACL custom changes from Broadcom SAI 7.1 to 9.2, which undergoes fundamental architectural refactoring in ACL implementation.
- Validated egress sFlow hardware offload on TH5, leveraging egress flex sampler and egress pipeline recirculation (EPRC) newly introduced to Tomahawk chip family, and designed SAI implementation.
- Extended IFA 2.0 transit functionality to TH5.
- In PFC watchdog hardware detection software restoration scenario, migrate PFC storm action from software-managed ACL to hardware native storm action (with reference to public implementation). The migration incurs minor changes (10 insertions, 4 deletions) on top of public implementation cherry-pick.
- **Earned 2023 H2 Data-SYS Excellent Project Award (优秀项目奖) being a team member of project white-box switches for network infrastructure 4.0 (4.0 网络架构自研交换机项目).**

• Microsoft Corporation

Redmond, WA, USA

Software engineer II, Azure Networking

Jan. 2019 - Nov. 2019

- Topics: SONiC network operating system over world's cutting-edge datacenter switching chips and switches (40G, 50G, 100G); RDMA over SONiC with full engineering responsibility of all RDMA-related components—PFC (DSCP-based and DOT1P-based), PFC storm watchdog, memory management unit (MMU), buffer watermark, ECN marking, WRED dropping, QoS mapping (DSCP to TC, DOT1P to TC, TC to PG, TC to queue), scheduling (WRR, DWRR, SP), and PFC storm watchdog warm-reboot; sub port router interfaces
- Lead the high-level design and the ground implementation of sub port router interfaces. Presented the high-level design at OCP SONiC community meeting. Revised SAI specification on sub port router interface creation.
- Designed, implemented, and test automated PFC watchdog warm-reboot feature, with ramp-up on the warm-reboot architecture from scratch. Resolved the issue of PFC watchdog not started after (both cold- and warm-) reboot by fixing the initialization dependency. Design archive can be found at https://github.com/wendani/sonic-architecture/blob/master/PFC_watchdog_warm_reboot.pdf
- Executed full development cycle of buffer pool watermark feature (design, implementation, test automation, and validation on various ASIC platforms). Improved the scalability of flex counter polling infrastructure with full sleep wait. Established synchronous counter clear path all the way down to ASIC to probe the hardware capability.
- Designed and implemented DOT1P to TC mapping support to the QoS mapping subsystem. Automated the feature test covering both control plane test (by using virtual switch and adding to the virtual switch test suite) and data plane test (by running on real ASIC platforms).
- Designed and implemented runtime DWRR weight change test. Captured and fixed (approved by Broadcom) weight change bug in Broadcom SAI 3.3 QoS scheduler codes. Developed a solid understanding of the scheduling hierarchies in both Trident2 and Tomahawk/Tomahawk2 architectures, together with the corresponding SONiC modeling and handling method.
- Gained an in-depth understanding of SONiC architecture and SAI switch pipeline model through feature development, code reading, SAI specification study, device play, and live site investigation. Design note can be found at <https://github.com/wendani/sonic-architecture>

- Awarded two tickets to Seattle Sounders match in the ScottGu Cloud + AI Red Zone Suites for going above and beyond in performance.

• Microsoft Corporation

Vancouver, BC, Canada

Software engineer 2, Azure Networking

Dec. 2017 - Jan. 2019

- Topics: white-box datacenter switches (40G, 50G, 100G)
- Developed fundamental insights into the memory management, scheduler, QoS, and flow control units of Broadcom XGS devices through theory of operation study and hand-on experiments with various chips.
- Characterized and test automated the DSCP mapping, ingress admission control (flow control XOFF/XON threshold, pool size), egress admission control (drop state, pool size), ECN marking, and WRED dropping of MMU on Broadcom BCM56850, BCM56960, BCM56970, and Mellanox Spectrum switch chips for various traffic types (lossy, lossless, data, and CPU). Captured all buffer setting deviations from the target design, proven by production deployment at scale. While being open-source, this part of the codes remains proprietary due to its intensity in design and methodology innovation.
- Designed, implemented, and validated WRED/ECN on/off feature within two weeks to eliminate the production rollout blocker. Started from zero coverage of the SONiC code base, got to the heart of the codes, and got the feature done within a stringent timeline. Feedback from Manager at feature delivery is “very impressive”.
- Built a solid foundation on Broadcom XGS switch chip pipeline from an exhaustive study of Broadcom University training materials and hand-on practice.
- One-year performance highlight from Manager—“I am fortunate to have you to manage this part” (“this part” refers to buffer, “the most esoteric feature in the ASIC” in his view).

• Pleora Technologies Inc.

Ottawa, ON, Canada

Senior embedded software designer

Jan. 2017 - Dec. 2017

- Topics: real-time video backhaul over GigE Vision standard protocol suite
- Designed and developed embedded software in a C and C++ mixed environment. Summoned by the core project team within six months since onboarding. Accomplished (passed product verification) five features in C++ with design pattern application in the first month of joining the core project.
- Improved the USRT (serial port communication) latency performance by over six times through batch data push and bug fixing. Identified architectural bottleneck that prohibits further performance improvements.
- Grasped the agent-side SNMP implementation in depth by learning the SNMP fundamentals from scratch and leading through the code base. Pinpointed and fixed the bug within a week.
- Ported third-party SNTpv4 module into our network stack and interfaced it with our embedded software for configuration, control, and application. Spotted bug (confirmed by third party) in code reading on the third day of taking over the source code.
- Proposed a lazy address conflict resolution method to reduce device boot-up time by ~7 seconds. Verified the feasibility of the method via experimental trials. Proposal adopted and implemented in Pleora’s latest devices for boot-up time optimization.
- Reduced the completion time of SNMP Get- and Set-Request by up to over 15% through agent-side MIB lookup acceleration using binary search.

• Viscore Technologies Inc.

Ottawa, ON, Canada

Principal R&D engineer on network architecture, software, and hardware

Sept. 2014 - Dec. 2016

- Topics: reliable data replication over optical broadcast medium (optical coupler); prototype of HEDA protocol on NetFPGA-10G board, funded in part by OCE TalentEdge fellowship; low-latency data communication over RDMA

- Setup NetFPGA-10G board on Fedora 14 through package installation, kernel module loading, and troubleshooting.
- Demonstrated UDP unicast, broadcast, and multicast over optical coupler. The uniqueness of the system is unidirectional communication links.
- Grasped in-depth knowledge of NIC hardware data plane and control plane architecture and of the role each module plays with respect to Ethernet physical layer and link layer standards. Enabled burst-mode transmission—a key foundation of communication over an optical coupler—at the physical layer through disabling link fault signaling at the RS sublayer and masking fault alarms at the PCS sublayer.
- Developed a control-plane method to perform burst transmission—an enabling technology of prototyping HEDA protocol on NetFPGA-10G board. Designed and implemented a user-space token-passing protocol in C to demonstrate burst data transmission between two hosts over an optical coupler. This milestone eliminates the biggest challenge to prototype the HEDA protocol.
- Learned exhaustive knowledge of Linux kernel architecture, Linux kernel programming, Linux network device driver, and Linux network stack. Architected the integration and interface solution of embracing HEDA protocol in the Linux networking system (kernel 2.6.35.14). Designed and implementing software modules of HEDA protocol in C at the kernel network device driver level.
- Led a team of three at the hardware development phase and a team of two at the software development phase.
- Acquired a deep understanding of various RDMA technologies—InfiniBand, RoCE, and iWARP—through standard study (IBTA specification and IETF RFCs), programming practice, and hands-on experience playing with various NICs on our platform. Refactored OFA (OpenFabrics Alliance) libverbs example code to work with Mellanox OFED.
- Gained insights into the internal workings of RoCE data path and control path through code reading of open source Soft-RoCE implementation.
- Peeled Soft-RoCEv2 off Linux source tree, and ported it to work with kernel 4.2.0-rc8 as a kernel module.

• **Source Technologies International, Ltd.**

Ottawa, ON, Canada

Consulting software engineer

Jun. 2014 - Aug. 2014

- Developed a mobile scenario manager app—a commercial product for iPhone iOS—in 1.5 months with both Objective-C and iOS programming learned from scratch. The key techniques employed include UIStoryboard, Model-View-Controller (MVC) pattern, target-action pattern, delegation pattern, singleton pattern, core data framework to interact with a local SQLite database, JSON text parsing, and HTTP/HTTPS POST method to communicate with a remote server for credential authentication and scenario fetching, update, and launching.
- Tested app on iOS simulator and real iPhone device through Xcode. Created and Distributed iOS App Store Package to testers and clients for beta testing.
- Developed an Android counterpart in Java fulfilling the same requirements with Android programming learned from scratch. Gained insights into Activity/Fragment lifecycle, and managed state transitions through callback methods. Designed user interface layout using XML. Debugged Java code using LogCat.

• **NEC Laboratories America, Inc.**

Princeton, NJ, USA

Visiting researcher, Dept. of optical networking

Jan. 2011 - Feb. 2011

- Topic: Xen hypervisor and virtual machine (VM) live migration
- Setup Xen 4.0.1 pv-ops Ubuntu 10.10 through kernel configuration & building and kernel module building & installation/loading. Setup a network file system server to store the disk image that can be mounted concurrently by multiple VM clients. Demonstrated VM live migration between two physical hosts connected by a D-Link gigabit switch (DGS-2205).

- **Ericsson (China) Communications Co., Ltd.** Beijing, P. R. China
System design engineer intern, Dept. of broadband networks *Jul. 2009 - Sept. 2009*
 - Topic: IEEE 1588v2 (precision clock synchronization protocol) over Ethernet
 - Developed a solid understanding of the clock synchronization mechanism. Finished one technical report on protocol review. The knowledge acquired becomes critical in later designing a link-layer polling protocol for optical datacenters (**project with Viscore Technologies Inc., Canada**), where the issue of clock synchronization among ports was resolved using a similar yet simpler scheme.

R&D EXPERIENCE

- **Carleton University** Ottawa, ON, Canada
Senior researcher, Dept. of Systems and Computer Engineering *Jun. 2011 - Aug. 2014*
 - Topics: 1) Datacenter networking; 2) Optical datacenters; 3) Protocol design; 4) Survivable telecom network design and analysis; 5) Decomposition methods, in particular, Lagrangian relaxation and column generation, for solving network optimization problems; 6) Software-defined networks; 7) Passive optical networks; 8) Optical node architecture
 - Designed optimal link capacity to enable full bandwidth communication in emerging datacenter networks using fat-tree and Microsoft VL2 topologies
 - Proposed a low-cost, power-efficient, and reliable datacenter network architecture using passive optical devices. Resolved physical-layer scalability issues by employing advanced interconnection techniques. Designed a fully distributed link-layer communication protocol, termed HEDA, to enable collision-free frame transmission. Described the protocol using finite state machine and message sequence chart. Developed an analytical model to compute lower and upper bounds on the expected packet delay of our protocol. (**This line of achievements are outputs of a six-month industry research project with Viscore Technologies Inc., Canada.**)
 - Studied a novel layered network architecture—Valiant load balancing (VLB) networks over optical networks. Developed a network-level availability model to compute the probability that a VLB network is congestion-free under all traffic patterns. The main challenges in such a context arise from the unique routing and protection scheme that goes beyond the definition of conventional connection-level service availability as well as the logical link failure correlation that prohibits the use of traditional analytical methods. **Our work was honored as a runner-up of the 2013 Fabio Neri Best Paper Award.** One winner and three runners-up were selected from all papers published in *Elsevier Optical Switching and Networking* journal throughout year 2013. Official announcement: <http://www.journals.elsevier.com/optical-switching-and-networking/awards/2013-fabio-neri-best-paper-award/>
 - Developed analytical models to accurately compute the availability of upper-layer connections in two-layer networks, where dedicated path protection is deployed at either the lower layer or the upper layer. The accuracy of our model is validated through OPNET simulation. A real-life case of such two-layer networks is IP over optical networks.
 - Learned to be versatile on various network topics. Developed mental toughness in demanding R&D environments.
- **State University of New York (SUNY) at Buffalo** Buffalo, NY, USA
Senior research scientist, Dept. of Computer Science and Engineering *Oct. 2010 - May 2011*
 - Topics: network function virtualization; virtual network embedding; reliable network design and analysis
 - Studied service availability in virtual infrastructure mapping and cloud computing such as Amazon EC2. Implemented a numerical method in MATLAB to compute the probability that the target availability is not fulfilled over a finite service subscription period. Since the service duration is finite, conventional steady-state analysis cannot be applied. Rather, the numerical calculation is a method that performs transient analysis. Our study can help service providers to work out service level agreements that avoid significant penalty risk.

- Learned to get along with supervisors and managers

• **Fraunhofer Heinrich-Hertz-Institut (HHI)**

Berlin, Germany

Visiting student, Dept. of Photonic Networks and Systems

Oct. 2008 - Mar. 2009

- Topic: survivability of WDM networks
- Evaluated connection availability in optical networks that employ shared path protection and backup path reprovisioning after failure occurrence. Compared the impact of revertive and non-revertive operations after failure repair on connection availability.
- Proposed and studied a new logical topology mapping problem, which incorporates the lightpath-based physical-layer failure localization capability into the traditional mapping design of IP over transparent optical networks. **I, as the first author, was selected as a semi-finalist in the 2010 Corning Outstanding Student Paper Competition for this work.** Over 430 student submissions were received that year, of which 10-12 papers were selected as semi-finalists.

• **Dept. of Electronic Engineering, Tsinghua University**

Beijing, P. R. China

Research assistant, Lab. of Optical Networking and Microwave Photonics

Feb. 2005 - Jul. 2010

- Kept updated with the cutting-edge research in optical networking
- Responsible for project implementation, including topology and routing algorithm design and optimization, differentiated service provisioning, protection and restoration, service availability analysis, multi-period network planning, etc.
- Carried out joint research work with peers
- Took major roles in drafting research proposals based on the industry trends (proposal “Resource allocation and optimization in optical networks” funded by National 863 High Tech Program, year 2007-2008; proposal “Path-computation-element-based (PCE-based) network architecture and testbed for multi-layer multi-domain transport networks” funded by National 863 High Tech Program, year 2009-2010)
- Networked with world-class research groups and scientists through conferences and emails

COMPUTER SKILLS

Programming Languages: C (const, volatile, strncpy, strchr, strstr, strtoul, %z), C++ in C, assembly (Nios II, x86 & x64 (Intel & AT&T syntaxes), MicroBlaze, ARM), inline assembly (gcc), C in C++, C++, Java, Objective-C, Python, Jinja2 (template, custom filter, macro, include, filter (combine)), YAML (sequence, mapping, scalar, |, >), AMPL, MATLAB, scripts, Fortran, Verilog, VHDL, SQL

C++: mutable, nested class, struct, constructor (top-down, inside out), default, member initializer list, explicit, direct initialization, copy initialization, destructor (bottom-up, outside in), address of reference, pointer to member function, operator(), operator type, cast (dynamic_cast), swap, implicit interface, virtual function & default argument, initializer_list, sequential container (iterator, emplace, insert, const_iterator, empty, size, operator[], at, operator=, front, emplace_back, push_back, pop_back, erase, clear), string (NULL, rational operators, compare, length, find, substr, stoul, npos), stringstream, vector (data, capacity, reserve, resize), vector<bool>, list (splice), queue (push, pop), stack (top, push, pop), priority queue (top, push, pop), std::greater, associative container (iterator, emplace, insert, empty, size, find, lower_bound, upper_bound, count, erase, clear), pair, map (piecewise_construct, operator[], at (const)), unordered_map, set (contains), unordered_set, bitset, std::make_reverse_iterator, std::next, std::base, shared_ptr (make_shared, get, reset, swap), unique_ptr, dynamic_pointer_cast, weak_ptr, lvalue, rvalue, rvalue reference, move, ref, move constructor, move assignment operator, std::find, std::find_if, lambda expression, std::function, std::bind, strtol, getline, tolower, nullptr, noexcept, override, sort, typename, parameter pack, pack expansion, std::thread (join, detach), std::mutex, std::unique_lock, std::condition_variable, std::lock_guard, atomic operations (atomic_bool (load, operator=), atomic_flag), memory ordering, modification order, happens-before, synchronizes-with, postponement, RAII, exception-safe (copy and swap)

Python: sequence type (`len`), list (`remove`), tuple, mapping type, dictionary (`items`, `get`, `setdefault`, `update`, `pop`), `all`, `enumerate`, `natsorted`, `sorted`, `map`, `filter`, `lambda`, `isinstance`, `ternary`, `not`, `in`, `import`, `from`, parameter passing, `global`, nested function, `scoping`, `assert`, `hex`, `format`, `except`, `finally`, `module`, `argparse`, `pickle`, `re` (`compile`, `match`), `subprocess`, `binascii`, `click` (`argument`, `option`)

Scripts: `bash` (`-e`, `-x`), `tcsh`, expansion (`variable`, `arithmetic`), `command substitution`, `subshell`, `process substitution` `<()`, `read`, `eval`/`^``, `source`/`.`, `if then`, `for do`, `while do`, `until do`, `continue` (`n`), `break`, `case in` (`::`), `test`/`[]`, `[[]]`, `&&`, `||`, `$?`, `$#`, `!`, `$*`, `$@`, `shift`, `getopt`, `getopts`, `set` (`-x`, `+x`, `--`, `-e`), `${ : : }`, `${ :? }`, `Lua` (`and`, `or`, `ternary operator`)

GNU Makefile: expansion (`simple :=`, `recursive =`, `conditional ?=`), `automatic variable` (`$@`, `$<`, `$^`), `static pattern rule` `: :`, `secondary expansion` `.SECONDEXPANSION`, `$(eval)`, `$(call)`, `phony target` `.PHONY`, `conditional directive` (`ifeq`), `conditional expansion` `$(if)`, `$(foreach)`, `$(strip)`, `$(addprefix)`, `$(wildcard)`, `.ONESHELL`, `SHELL`, `.SHELLFLAGS`,

Operating Systems: `Linux` (`Debian`, `Ubuntu`, `CentOS`, `Fedora`, `Red Hat`, `Raspbian Lite`), `FreeRTOS`, `Windows`, `Windows Subsystem for Linux (WSL)`, `Mac OS X`

Networking: `IEEE 802.3`, `global pause`, `auto-negotiation`, `EEE`, `EPON (802.3ah)`, `10GEPON (802.3av)`, `flow control (802.3x)`, `LAG (802.3ad)`, `FEC (Reed-Solomon, FireCode)`, `IEEE 802.1`, `bridge (.1D, .1Q)`, `VLAN`, `PFC (802.1Qbb)`, `Q-in-Q`, `GRE`, `VXLAN (bridging, centralized routing (gateway), distributed routing (gateway), asymmetric routing, symmetric routing)`, `control plane (ioctl, AXI4-Lite, Avalon-MM, MDIO (802.3 Clause 22, 45), I2C)`, `network device driver`, `ARP (probe, announcement, gratuitous, proxy, RFC 5227)`, `NDP`, `IPv4`, `private address space (RFC 1918)`, `LLA (RFC 3927)`, `DSCP`, `ECN marking (RFC 3168)`, `WRED dropping`, `ICMP (RFC 792, echo, echo reply)`, `TCP`, `BGP (quagga, free range routing (FRR), peering, per-VRF peering, remote-as (internal, external), eBGP, AS_PATH loop detection, AS_PATH replacement, AS_PATH prepend, maximum-paths, best path selection, bestpath as-path multipath-relax, neighbor, peer group, route-map, address family, redistribute connected, network, EVPN, RT-2, ARP/NDP suppression, RT-3, RT-5)`, `UDP`, `DHCP (RFC 2131, 2132)`, `ISC DHCP`, `Kea DHCP (option-data)`, `SNMP (agent, ASN.1 encoding/decoding, trap, mib)`, `SNTPv4 (RFC 4330)`, `NTPv4 (RFC 5905)`, `sFlow (InMon v5)`, `host sflow daemon (hsflowd)`, `NicheStackTM IPv4 (arp, udp, snmpv2c, sntp4)`, `InfiniBand`, `RoCEv1`, `RoCEv2`, `iWARP (RDMA/DDP/MPA, RFC 5040, 5041, 5044)`, `RDMA`, `Verbs programming`, `1588v2`, `GigE Vision (WRITEREG, READREG, WRITEMEM, READMEM, EVENTDATA, DISCOVERY, PENDING_ACK, FORCEIP, heartbeat)`

Software for Open Networking in the Cloud (SONiC): `Table (COUNTERS_DB)`, `SubscriberStateTable (CONFIG_DB, APPL_DB, STATE_DB)`, `ProducerTable (ASIC_DB, FLEX_COUNTER_DB)`, `ConsumerTable (ASIC_DB, FLEX_COUNTER_DB)`, `ConsumerStateTable (APPL_DB)`, `ProducerStateTable (APPL_DB)`, `NotificationConsumer (APPL_DB, COUNTERS_DB, ASIC_DB)`, `NotificationProducer (ASIC_DB)`, `SelectableTimer`, `flex counter architecture`, `sairedis`, `synchronous SAI API call`, `PFC storm watchdog`, `buffer watermark`, `buffer config manager`, `ACL`, `warm-reboot/in-service software upgrade (ISSU)`, `scheduler`, `QoS mapping`, `neighbor`, `route`, `COPP`, `critical resource management (CRM)`, `sFlow`, `FDB`, `port`, `LAG`, `router interfaces`, `mirror`, `virtual switch (vs)`, `proxy ARP`, `ZTP`, `C++ unit test on Orches`

OCP switch abstraction interface (SAI): `pipeline`, `buffer`, `host interface`, `ACL`, `sFlow`, `Broadcom implementation`, `python-saithrift RPC library`

Switching: `ingress admission control`, `virtual output queuing`, `egress admission control`, `scheduling (strict priority (preemptive, non-preemptive), round robin (work-conserving, native (packet), weighted (packet), deficit (byte), weighted deficit (byte)))`, `policing/traffic shaping (average rate (packet, byte), average rate + burst size (packet, byte), average rate + burst size + peak rate (packet, byte))`, `output-queued shared memory`, `input-output-queued`

Linux kernel: kernel architecture, interrupt context, interrupt & interrupt handler, softirq, tasklet, kernel synchronization, RCU, memory barrier, process context, work queue, kernel thread and sleep (wait queue), kernel preemption, hrtimer, memory management, char driver, network traffic control (TC) (token bucket filter), Tx & Rx path below layer 3, new API (NAPI), l3mdev, psample, syscall, sysfs, tracepoint, kprobe, kdump, kbuild, kallsyms, live patch

FreeRTOS kernel: interrupt service routine (ISR), binary semaphore, mutex, queue, event group, task management & scheduling, software timer, heap memory management, stack overflow checking

Data structures (program): array, vector, linked list (singly, doubly), queue, stack, hash table (3sum), lookup table/map/dictionary/associative array, C++ multimap, tree, binary tree, heap, binary search tree, AVL tree, prefix tree/trie, bitmap/bit vector/bitset, set (bitmap, array, linked list, hash table, binary search tree), suffix array

Algorithms: searching (linear/sequential search, binary search, breadth-first search, depth-first search), iteration, recursion, sorting (bubble sort, selection sort, insertion sort, shell sort, quick sort, merge sort, radix sort, heap sort), selection (K^{th} smallest/largest), random sampling (sorted output, random output), bit manipulation, divide and conquer (quick sort), dynamic programming, greedy, invariant, minimum spanning tree, Dijkstra's algorithm

Applications: code reading, regular expression (basic, extended, [:space:], [:alnum:]), supervisor, systemd, system debugging, Linux kernel programming, Linux system programming (select, poll, epoll, getenv, timerfd,fcntl), netlink (rt, link, neigh), generic netlink, interprocess communication (IPC), process scheduling, concurrency, multithreading (POSIX), synchronization & condition synchronization, file access, socket programming, iOS programming, Android programming, OPNET simulator, AMPL/lp_solve, AMPL/CPLEX, Kusto query, Redis (-n, --scan, --pattern, get, hget, set (ex), hset, expire, del, hdel (multiple fields), dump (-d, -k, -y), load, ping), P4 packet test framework (PTF) (dataplane), Broadcom SDK (BST, KNET, rx filter, QoS mapping, FP, flex counters, scheduler, LAG, hashing, logging (BSL)), Cancun, flex flow APIs, OpenNSL, Azure Express-Route, KLISH (Kommand Line Interface Shell) 2.1.4, 2.2.0 (PTYPE, SUBCOMMAND (value, mode (switch)), COMMAND (lock, interrupt, hidden), PARAM (test, hidden), ACTION (builtin, shebang), CONFIG (operation, pattern, sequence), VAR, MACRO, VIEW, NAMESPACE, STARTUP (default_shebang)) OpenConfig YANG (ACL)

Design patterns: observer (one to many), bridge (separate an abstraction from its implementation), decorator (add responsibilities incrementally & dynamically), singleton, publish-subscribe, composite (composition of objects in tree structure)

Tools: gdb (bt, f, p, x, info (f, registers)), crash (vmlinux), gprof, OProfile, iperf, strace, ftrace, SystemTap, Scapy (ether, dot1q, ip, arp, icmp), Wireshark, autoconf (./autogen.sh, ./configure (--bindir, --libdir, --includedir, --enable-static, --enable-debug), make, make install), cygwin, cross toolchain (OpenWrt, Altera), JTAG, SnmpB, lftp, ssh (localhost), sftp, scp, SecureCRT (SSH, SFTP), PuTTY, telnet, Jira, CodeFlow, jumpbox, Dnsmasq, syslog, log analyzer

Linux command line: lsblk, fdisk, df, du, ps, mkfs, fstrim, sync, rsync, less, head (-n), tail (-n), grep (-r, -i, -n, -v, -q, --color), find (-type), xargs, ln, augtool (-r, set), mount (--bind), umount, /etc/fstab, file, chmod (+x), nm, tree (-L), lockfile, exec, wait, trap, export, source/., tar (.tar, .tar.gz), gzip, gunzip (.gz), zcat, zgrep, ifconfig, ifupdown2, ip addr, ip link, ethtool, arp, ip neigh, ip netns (add, exec), tcpdump (-ni, -vx, -w, -r), ip route, ip rule, ip monitor wget, curl (-o), md5sum, chown, disown, adduser, usermod, passwd, id, chroot, ischroot, pkg-config, dpkg (--root, --remove, --purge), dpkg-buildpackage (-rfakeroot, -b, -us, -uc, -j, -nc), fakeroot, debhelper (dh_clean), which, lsusb, virsh, diff, comm, echo (-n), tee, cat (-A, ctrl-D, <<token), awk (/pattern/action, -F, -f, print \$1), cut (-d, -f, -c), sort (-t, -k, -nr, -u), uniq (-u, -d, -c), sed (-n, -e, -i, .), tr (-s), date, time (-f), top, bc, expr, seq, systemctl (timer), systemctl (-w), nsenter, cgexec, xxd, crontab

Hardware: NetFPGA-10G, Xilinx, Virtex-5, Altera UDP/IP core, network interface card (Mellanox, Chelsio, Emulex), switch (Arista 7124SX, 7050QX-32S, 7060CX-32S, 7260CX3-64, Dell S6000, S6100-ON, Mellanox SN2700, Celestica Fishbone-48, Seastone-DX010, Questone2BD, Quanta IX8C-56X, IX8E-56X, Ruijie RG-B6510-48VS8CQ, H3C TCS81-120F, ByteDance B3010, B4010, B4020, Ubiquiti EdgeSwitch16, Netgear M4200, GS108PE, GS110TP), PCIe, SR-IOV, DMA, processor microarchitecture (Pentium, Pentium Pro, II, III, PowerPC 601, 603e, 604, 750, 7400), computer, microcontroller, and SoC architecture (Nios II, Cortex-M3/M4, MicroBlaze, ARM926EJ-S), exclusive access, memory barrier, write buffer, MAC core (MorethanIP 10/100/1000Mbps, Xilinx 10G), PHY (Broadcom AEL2005, Marvell 88X3310, 88E1512, Microsemi VSC8541), Camera Link interface, sensor (Maxim Integrated LM75, Altera Arria 10 internal temperature sensing diode & IP core), LED, FireWire card (SYBA)

Switch chips: Broadcom XGS devices, BCM56850 (Trident2), BCM56873 (Trident3), BCM56960 (Tomahawk), BCM56970 (Tomahawk2), BCM56788 (Trident4 X9), BCM56990 (Tomahawk4), BCM78900 (Tomahawk5), ingress pipeline (L2 unicast & multicast, L3 unicast & multicast), egress pipeline (L3), flexible pipeline, memory management unit (MMU) (resource accounting for unicast & non-unicast packets), replication and queueing engine (RQE), buffer management, buffer statistics tracking (BST), oversubscription buffer, transient capture buffer, cross-point (XP) architecture, ingress traffic management (ITM), scheduler, shaper, QoS mapping, PFC, ECN, WRED, flex QoS, mirroring (ERSPAN, local SPAN), mirror on drop (MOD), field processor (legacy, new, logical table selection, range checker, compression, hint), VRF, VXLAN (initiation, termination, bridging, riot), hashing (static, robust, flex, consistent, resilient), dynamic load balancing (DLB), flex digest, ECMP (weighted, hierarchical), LAG, user defined field (FP, hashing, flex hashing, flexible counter), flexible counters (1.0, 2.0, 2.2), flex monitoring, trace event, drop event, packet sampling (ingress, egress, ingress flex sampler, egress flex sampler), sFlow, sFlow hardware offload, IFA (1.0, 2.0), EP recirculation, Broadcom DNX devices, BCM88670 (Jericho), BCM88690 (Jericho2), packet processor architecture, traffic management architecture, Cisco SiliconOne, Cisco Lacrosse

Virtualization: veth, macvlan, netns, tap, hypervisor (VirtualBox, VMware, Hyper-V, Xen), container, Docker (`info`, `build context`, `login`, `pull`, `push`, `Dockerfile` (FROM, RUN, COPY; WORKDIR, EXPOSE, ENTRYPOINT), `build` (-t, -f), `load`, `images`, `tag`, `rmi`, `inspect` (--type, --format), `image` (`ls`, `history`, `build`, `tag`), `ps`, `run` (-ti, --privileged, -v, -d, -w, -pid), `exec`, `start`, `attach`, `wait`, `stop`, `rm`, `cp`)

Automation: Ansible (`ansible-playbook`, `name`, `with_items`, `with_dict`, `loop`, `when`, `rescue`, `include`, `include_vars`, `register`, `become`, `connection`, `async`, `poll`, `async_status`, `action plugin`, `module` (`set_fact`, `debug`, `fail`, `copy`, `template`, `command`, `shell`, `script`, `supervisorctl`, `lineinfile`), `custom module`), Jenkins (`template`, `pipeline`), GitLab Continuous Integration (CI) (`gitlab-runner`, `.gitlab-ci.yml`, `stages`, `stage`, `when`, `only`, `variables` (GIT_SUBMODULE_STRATEGY), `cache`, `allow_failure`, `artifacts`, `tags`, `script`)

Test: pytest, Google test

Version Control: Git (`local`, `remote`, `add --patch`, `branch --contains`, `tag -a`, `fetch --tags`, `push` (--force, --delete, --tags, tag), `show`, `rebase` (-i, --preserve-merges, commit split), `format-patch`, `apply` (-3), `am`, `cherry-pick`, `submodule` (`update --recursive`, `sync`, `add`) `bisect`, `blame --line-porcelain`, `archive`), StGit (`series`, `show`, `diff`, `refresh` (-p), `pop`, `push`, `import`, `export`, `uncommit`, `commit`), GitHub workflow, GitLab workflow (`rebase`), Bitbucket, Visual Studio Team Services, CodeFlow, CVS, TortoiseCVS, SVN, TortoiseSVN, Source Depot

IDE: Eclipse, Xcode

Text and Graphs: Vim, GVim, Emacs, Confluence, SharePoint, OneNote, L^AT_EX, PSTricks, Gnuplot, Origin, SMIV1, SMIV2, JSON, XML

Garage: 3D printing (Ultimaker 3 Extended, Cura), laser cutting (Universal laser systems VLS4.60, LibreCAD, AutoCAD, Adobe Illustrator), plywood, living hinge, safe box, piano, treasure box, ring box, phone stand, drink coaster, anodized aluminum, dog tag, vinyl cutting (Silhouette Cameo), Raspberry Pi Zero W, Arduino Uno (button, LED, potentiometer, speaker), sewing machine (Brother SE-400)

HONORS AND AWARDS

Network Systems (NS) All-Hands Topic Most Popular Award	2022
July-August 2021 ByteStyle Award—Aim for the highest (最有字节范奖之追求极致奖)	2021
Recognized reviewer (completing two or more scientific paper reviews) by OSA Publishing	2020
O-1 Visa—Individuals with Extraordinary Ability or Achievement	2016
Outstanding reviewer for Elsevier Optical Switching and Networking	2016
OCE (Ontario Centres of Excellence) TalentEdge Fellowship for industry R&D (CAD\$32,500)	2015
OCE (Ontario Centres of Excellence) TalentEdge Fellowship for industry R&D (CAD\$32,500)	2014
Recognized reviewer for Elsevier Computer Communications	2014
Runner-up of the 2013 Fabio Neri Best Paper Award	2013
Eligibility as NSERC Visiting Fellow in Canadian Government Laboratories	2011
Eligibility as NSERC Visiting Fellow in Canadian Government Laboratories	2010
Feng Deng conference travel grant (IEEE/OSA OFC 2010, San Diego, CA, USA, 12,000RMB)	2009
Semi-finalist of the 2010 Corning Outstanding Student Paper Award (10-12 out of over 430)	2009
Feng Deng conference travel grant (IEEE/OSA OFC 2009, San Diego, CA, USA, 10,000RMB)	2008
MPS (Monolithic Power System Inc.) Scholarship for excellent students (second class)	2008
DAAD (Deutscher Akademischer Austausch Dienst) research scholarship for young doctoral candidates up to six months (5,000€)	2008
Feng Deng conference travel grant (IEEE/OSA OFC 2008, San Diego, CA, USA, 10,000RMB)	2007
Excellent individual in social practice	2007
Feng Deng conference travel grant (IEEE GLOBECOM 2007, Washington D.C., USA, 10,000RMB)	2007
Excellent graduate of Tsinghua University, an honor entitled to top 10% graduates	2005
Shunde Wu Couple Scholarship for excellent students (second class)	2004
Shunde Wu Couple Scholarship for excellent students (second class)	2003
Individual prize in electronics process technology practice	2003
Citic Bank Scholarship for excellent students (second class)	2002

PROFESSIONAL AND VOLUNTEER ACTIVITIES

- Optical Society of America (OSA) Young Professionals (YP) Program, April 2011 - December 2013
- **Review Editor** on the Editorial Board of Optical Communications and Networks, specialty section of Frontiers in Communications and Networks
- **Technical program committee member** of OSA Photonic Networks and Devices (Networks) 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020
- **Technical program committee member** of IEEE Global Communications Conference (Globecom), Optical Networks and Systems Symposium 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023
- **Technical program committee member** of IEEE International Conference on Communications (ICC), Optical Networks and Systems Symposium 2012, 2013, 2014, 2015, 2016, 2017, 2019, 2020, 2021
- **Technical program committee member** of IEEE International Conference on Optical Network Design and Modeling (ONDM) 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024

- **Technical program committee member** of IEEE International Conference on Computing, Networking and Communications (ICNC), Optical and Grid Computing Symposium 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2023, 2024
- **Technical program committee member** of OSA Asia Communications and Photonics Conference (ACP), Track 3 Network Architectures, Management and Applications 2016, 2018, 2020
- **Technical program committee member** of IEEE Asia-Pacific Conference on Communications (APCC) 2016
- **Technical program committee member** of IEEE International Conference on Networks (ICON) 2012, 2013
- **Organization committee member** of the 56th Canadian Operational Research Society (CORS) Annual Conference, Telecommunications Cluster 2014
- **Industrial membership committee member** of Asia-Pacific Signal and Information Processing Association (APSIPA) 2018-2019
- **Volunteer** (paper curator, dispatcher, and reviewer; on-site assistant) of the Technical Conference on Linux Networking 2015 (Netdev 0.1)
- **Reviewer** of IEEE/ACM Transactions on Networking (1), IEEE/OSA Journal of Lightwave Technology (8), IEEE Journal on Selected Areas in Communications (1), IEEE/OSA Journal of Optical Communications and Networking (14), IEEE Transactions on Communications (1), IEEE Communications Letters (1), Elsevier Computer Networks (3), Elsevier Computer Communications (1), Elsevier Optical Switching and Networking (13), Springer Photonic Network Communications (8), Springer Telecommunication Systems (3), Wiley International Journal of Communication Systems (1), Wiley Networks Journal (1), Wiley Transactions on Emerging Telecommunications Technologies (2), IEEE China Communications (1), De Gruyter Journal of Optical Communications (1), Chinese Optics Letters (1), Acta Electronica Sinica (2), Journal of Beijing University of Posts and Telecommunications (4), and many good conferences such as IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, IEEE DRCN, IEEE ONDM, etc.

IMMIGRATION STATUS

Permanent resident of Canada, landed on 8 April, 2014
 Citizen of Canada, 3 August, 2018

MISCELLANEOUS

GitHub: <https://github.com/wendani>

Google Scholar: <https://scholar.google.ca/citations?user=KX0P6HsAAAAJ&hl=en>

PATENTS

1. Yunqu Liu, Kin-Wai Leong, **Wenda Ni**, and Changcheng Huang, "Methods and systems for passive optical switching," publication no. WO/2014/078940, published 30 May, 2014.
2. Yunqu Liu, Kin-Wai Leong, **Wenda Ni**, and Changcheng Huang, "Parallel optoelectronic network that supports a no-packet-loss signaling system and loosely coupled application-weighted routing," U.S. patent pending, application no. 61/992,570, filed 13 May, 2014.

JOURNAL PAPERS

1. **Wenda Ni**, Changcheng Huang, and Jing Wu, “Integrated design of fault localization and survivable mapping in IP over transparent WDM networks,” *Springer Photonic Network Communications*, vol. 28, no. 3, pp. 287–294, 2014.
2. **Wenda Ni**, Changcheng Huang, Yunqu Leon Liu, Weiwei Li, Kin-Wai Leong, and Jing Wu, “POXN: a new passive optical cross-connection network for low-cost power-efficient datacenters,” *IEEE/OSA Journal of Lightwave Technology*, vol. 32, no. 8, pp. 1482–1500, Apr. 15, 2014. (First-round review decision is “accept pending minor revisions”.)
3. **Wenda Ni**, Changcheng Huang, and Jing Wu, “Provisioning high-availability datacenter networks for full bandwidth communication,” *Elsevier Computer Networks, Special Issue on Communications and Networking in the Cloud*, vol. 68, pp. 71–94, Aug. 2014. (A thorough study with 24 pages in double-column format. 12 papers were accepted out of over 60 submissions. Decision from the first-round review is “conditional accept” subject to “moderate revisions”.)
4. **Wenda Ni**, Changcheng Huang, Jing Wu, and Michel Savoie, “Availability of survivable Valiant load balancing (VLB) networks over optical networks,” *Elsevier Optical Switching and Networking, Special Section on Cross-Layer Innovations*, vol. 10, no. 3, pp. 274–289, Jul. 2013. (Runner-up of the 2013 Fabio Neri Best Paper Award—one winner and three runners-up were selected from all papers published in this journal in year 2013)
5. **Wenda Ni**, Jing Wu, Changcheng Huang, and Michel Savoie, “Analytical models of flow availability in two-layer networks with dedicated path protection,” *Elsevier Optical Switching and Networking, Special Issue on Advances in Optical Networks Control and Management*, vol. 10, no. 1, pp. 62–76, Jan. 2013.
6. **Wenda Ni**, Xiaoping Zheng, Chunlei Zhu, Yanhe Li, Yili Guo, and Hanyi Zhang, “Achieving resource-efficient survivable provisioning in service differentiated WDM mesh networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 16, pp. 2831–2839, Aug. 15, 2008.
7. Yanwei Li, **Wenda Ni**, Heng Zhang, Yanhe Li, and Xiaoping Zheng, “Availability analytical model for permanent dedicated path protection in WDM networks,” *IEEE Communications Letters*, vol. 16, no. 1, pp. 95–97, Jan. 2012.
8. Qingshan Li, **Wenda Ni**, Yanhe Li, Yili Guo, Xiaoping Zheng, and Hanyi Zhang, “Incremental survivable network design with topology augmentation in SDH/SONET mesh networks,” *Springer Photonic Network Communications*, vol. 18, no. 3, pp. 400–408, 2009.
9. Chunlei Zhu, **Wenda Ni**, Yu Du, Yanhe Li, Xiaoping Zheng, Yili Guo, and Hanyi Zhang, “New and improved approaches for wavelength assignment in wavelength-continuous optical burst switching (OBS) networks,” *SPIE Optical Engineering*, vol. 46, no. 9, 090504, Sept. 2007.
10. Qingshan Li, Xiaoping Zheng, **Wenda Ni**, Yanhe Li, Hanyi Zhang, “Incremental survivable network design against node failure in SDH/SONET mesh networks,” *Springer Photonic Network Communications*, vol. 23, no. 1, pp. 25–32, Feb. 2012.
11. **Wenda Ni**, Qingshan Li, Yanhe Li, Hanyi Zhang, Bingkun Zhou, and Xiaoping Zheng, “Survivability in optical transport networks,” *Acta Electronica Sinica*, vol. 41, no. 7, pp. 1395–1405, Jul. 2013. (in Chinese)
12. **Wenda Ni**, Chunlei Zhu, and Xiaoping Zheng, “Design and implementation of all optical networks supporting differentiated services,” *ZTE Communications*, vol. 12, no. 6, pp. 10–13, Dec. 2006. (in Chinese)

1. **Wenda Ni**, Changcheng Huang, and Jing Wu, “On capacity provisioning in datacenter networks for full bandwidth communication,” in *Proc. IEEE International Conference on High Performance Switching and Routing (HPSR)*, July 2013, pp. 62–67.
2. **Wenda Ni**, Jing Wu, Changcheng Huang, and Michel Savoie, “Flow availability analysis in two-layer networks with dedicated path protection at the upper layer,” in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2012, CQR-P2.
3. **Wenda Ni**, Changcheng Huang, Jing Wu, Qingshan Li, and Michel Savoie, “Optimizing the monitoring path design for independent dual failures,” in *Proc. IEEE International Conference on Communications (ICC)*, Jun. 2012, ONS03.
4. **Wenda Ni**, Erwin Patzak, Michael Schlosser, and Hanyi Zhang, “Availability evaluation in shared-path-protected WDM networks with startup-failure-driven backup path reprovisioning,” in *Proc. IEEE International Conference on Communications (ICC)*, May 2010, ON02.
5. **Wenda Ni**, Yabin Ye, Michael Schlosser, Erwin Patzak, and Hanyi Zhang, “Survivable mapping with maximal physical-layer failure-localization potential in IP over transparent optical networks,” in *Proc. IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC)*, Mar. 2010, OWH1. (Semi-finalist in the 2010 Corning Outstanding Student Paper Competition—only 10-12 papers were selected as semi-finalists out of over 430 student submissions)
6. **Wenda Ni**, Erwin Patzak, Michael Schlosser, Yabin Ye, and Hanyi Zhang, “On operating shared-path-protected WDM networks non-revertively by using backup path reprovisioning,” in *Proc. IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC)*, Mar. 2010, OWH4.
7. **Wenda Ni**, Chunlei Zhu, Yabin Ye, Michael Schlosser, and Hanyi Zhang, “Reducing burst loss probability in multi-class optical burst switching networks by successive minimal incremental routing,” in *Proc. IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC)*, Mar. 2009, OWA6.
8. **Wenda Ni**, Michael Schlosser, Qingshan Li, Yili Guo, Hanyi Zhang, and Xiaoping Zheng, “Achieving optimal lightpath scheduling in survivable WDM mesh networks,” in *Proc. IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC)*, Feb. 2008, OWN5.
9. **Wenda Ni**, Chunlei Zhu, Xiaoping Zheng, Yanhe Li, Yili Guo, and Hanyi Zhang, “On routing optimization in multi-class optical burst switching networks,” in *Proc. IEEE International Conference on Communications (ICC)*, May 2008, ON03-4.
10. **Wenda Ni**, Xiaoping Zheng, Chunlei Zhu, Yili Guo, Yanhe Li, and Hanyi Zhang, “An improved approach for online backup reprovisioning against double near-simultaneous link failures in survivable WDM mesh networks,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Nov. 2007, ONSS05-2.
11. Yanwei Li, **Wenda Ni**, Heng Zhang, Nan Hua, Yanhe Li, and Xiaoping Zheng, “Availability analytical model for permanent dedicated path protection in service differentiated WDM networks,” in *Proc. IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC)*, Mar. 2013, JW2A.01.
12. Jing Wu, **Wenda Ni**, and Changcheng Huang, “Flow availability in two-layer networks with dedicated path protection,” in *Proc. OSA/IEEE/SPIE Asia Communications and Photonics Conference (ACP)*, Nov. 2012, ATh4D. (Invited talk)

Guohan Lu, Ph.D.

Partner software engineering manager, Azure Networking
Microsoft Corporation
Tel: +1 (206) 643 8829
E-mail: gulv@microsoft.com

Changcheng Huang, Ph.D., P. Eng.

Professor, Department of Systems and Computer Engineering
Carleton University
Tel: +1 (613) 260 7387
E-mail: huang@sce.carleton.ca

Daniel Wong

VP Product, Viscore Technologies Inc.
7 Bayview Road
Ottawa, ON K1Y 2C5, Canada
Tel: +1 (613) 252 7388
E-mail: danielhwong@gmail.com

Victor Yu Liu, Ph.D.

Chief Network Architect, Visa
Foster City, CA, USA
Tel: +1 (510) 384 3811
E-mail: viliu@visa.com, packerliu@gmail.com, yuliu@ieee.org
URL: <http://www.sis.pitt.edu/~yliu/>

Hanyi Zhang

Past head, Laboratory of Optical Networking and Microwave Photonics
Professor, Department of Electronic Engineering
Tsinghua University
E-mail: zhy-dee@tsinghua.edu.cn

Michael Schlosser

Project Manager
Department of "Photonic Networks and Systems (PN)"
Fraunhofer-Institute for Telecommunications, Heinrich-Hertz-Institut
Einsteinufer 37, D-10587 Berlin, Germany
Tel: +49 (30) 31002 346, Fax: +49 (30) 31002 250
E-mail: michael.schlosser@hhi.fraunhofer.de

Yabin Ye, Ph.D.

Senior Researcher, European Research Center, Huawei Technologies
Riesstr. 25, 80992 Munich, Germany
Tel: +49 (89) 158 834 4052
E-mail: yeyabin@huawei.com; yabin.ye@ieee.org

Robert Xin Liu

CEO, Source Technologies International, Ltd.

Room 102, Unit 3, Building No. 1, Section 3, Shang He Cun, Haidian District
Beijing 100097, China

Tel: +86 (10) 8849 3060; +86 137 0119 4002

E-mail: robert.liu@srctek.com