NIC offloads at hyperscale

Introducing the OCP NIC Core Features Spec v1.0

Netdev 0x17, 2023

Willem de Bruijn
willemb@google.com

Difficult for operators to integrate new devices into fleet

Hardware bugs

- UDP zero checksum conversion

Disconnect to Workloads

- Custom protocols vs. protocol specific offloads
- Scale to tens of millions of connections

Inconsistent Interfaces

- Telemetry: which bytes does a byte counter count

Difficult for vendors to meet the needs of operators

Each RFP is written from scratch:

- incomplete: overlooking feature interplay, subtle details, performance aspects
- imprecise: "must have feature foo" but foo is nowhere defined unambiguously
- impossible to validate: no shared testsuites, let alone representative workloads

Why Document

- consistent behavior across devices
- correct behavior: warn about common implementation bugs
- apt behavior: share workloads and operating conditions

Why An Open Spec

- codify and share industry wide expertise, and iterate
- in a public format that is unencumbered by NDAs
- to create a broad market

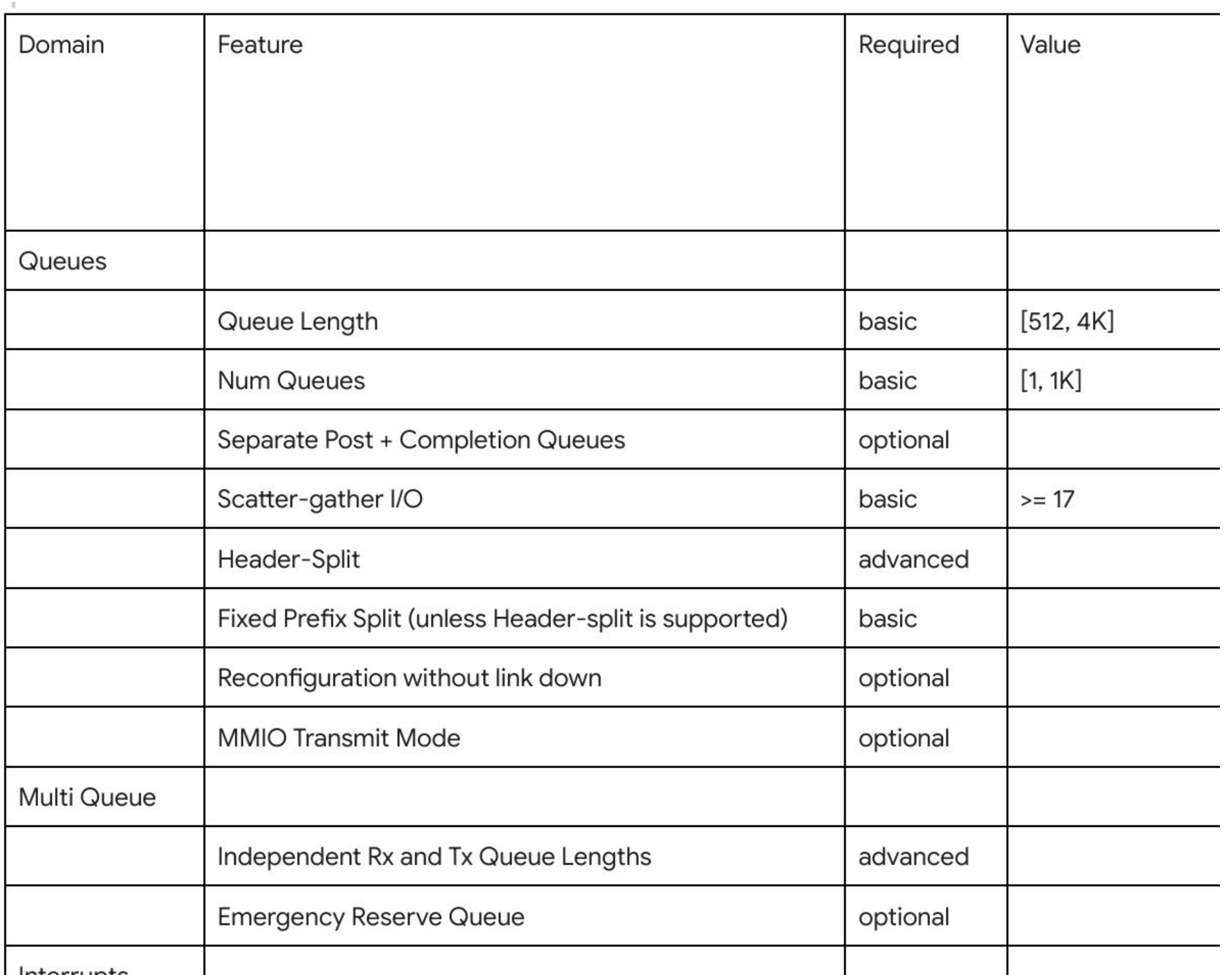
What

Spec

Testsuite

Checklist

Self Certification



Domain	Feature	Required	Value
Queues			
	Queue Length	basic	[512, 4K]
	Num Queues	basic	[1, 1K]
	Separate Post + Completion Queues	optional	
	Scatter-gather I/O	basic	>= 17
	Header-Split	advanced	
	Fixed Prefix Split (unless Header-split is supported)	basic	
	Reconfiguration without link down	optional	
	MMIO Transmit Mode	optional	
Multi Queue			
	Independent Rx and Tx Queue Lengths	advanced	
	Emergency Reserve Queue	optional	
Interrupts			

Target: Hyperscale Servers

- High End: 100s cores, 100s Gbps
- Large Scale: 10M+ active flows, 100K+ conn/s
- Heterogeneous Fleet
- Closed World: Custom Protocols
- Continuous Monitoring

Target: Hyperscale Servers

- High End: 100s cores, 100s Gbps
- Large Scale: 10M+ active flows, 100K+ conn/s
- Heterogeneous Fleet
- Closed World: Custom Protocols
- Continuous Monitoring

Scope: Core Features: Uncontroversial "Table Stakes"

Explicitly not: virtualization, smartnics

Interface: Driver behavior (net_device_ops / NDIS)

Not: implementation. Not: Device (PCI).

Open Compute Project (OCP) NIC Core Features Spec v1

opencompute.org/wiki/Networking/NIC_Software#Specs (pdf)

100% compliance is not a goal. Spec is a starting point. Report the diffs.

Contents

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Not Exhaustive: A Sample of Non Obvious Details

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Why Standardization

Target

- Hardware
- Scope
- Workload Model

Interface

Validation

- Self-certification

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Queues

Interrupts

Multi-Queue

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Queues

- Header Split
- 4K/9K MTU + Conserving Memory

Interrupts

Multi-Queue

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Design Principles

Checksum Offload

Segmentation Offload

Receive Segment Coalescing

Timestamping

Traffic Shaping

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Design Principles

- Stateless
- Protocol Independent (!= Programmable)

Checksum Offload

Segmentation Offload

Receive Segment Coalescing

Timestamping

Traffic Shaping

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Design Principles

- Stateless
- Protocol Independent (!= Programmable)

Checksum Offload

- Linear sum over defined range
- Only one sum per packet

Segmentation Offload

Receive Segment Coalescing

Timestamping

Netdev 1.1 Keynote, DaveM, "Hardware Checksumming: Less is More", 2016

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Design Principles

- Stateless
- Protocol Independent (!= Programmable)

Checksum Offload

- Linear sum over defined range
- Only one sum per packet

Segmentation Offload

- TSO, USO, PISO
- Jumbogram (BIGTCP)
- Details: FIN, PSH only on last segment

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Receive Segment Coalescing

Timestamping

Traffic Shaping

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Receive Segment Coalescing

Timestamping

- At Line Rate
- Applications: CC, Fleet monitoring, …

Traffic Shaping

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Receive Segment Coalescing

Timestamping

- At Line Rate
- Applications: CC, Fleet monitoring, …

Traffic Shaping

- Egress: Earliest Departure Time

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Protocol Support

- IPv6 First

Telemetry

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

Bitrate

- Scalability: 1 to M streams, 1 to N cores
- With and without CSUM/TSO/RSC/…
- Real world conditions: antagonists
- Peak, stress and endurance runs

Packet rate

Connection count & rate

Latency

Intro

I/O API

Offloads

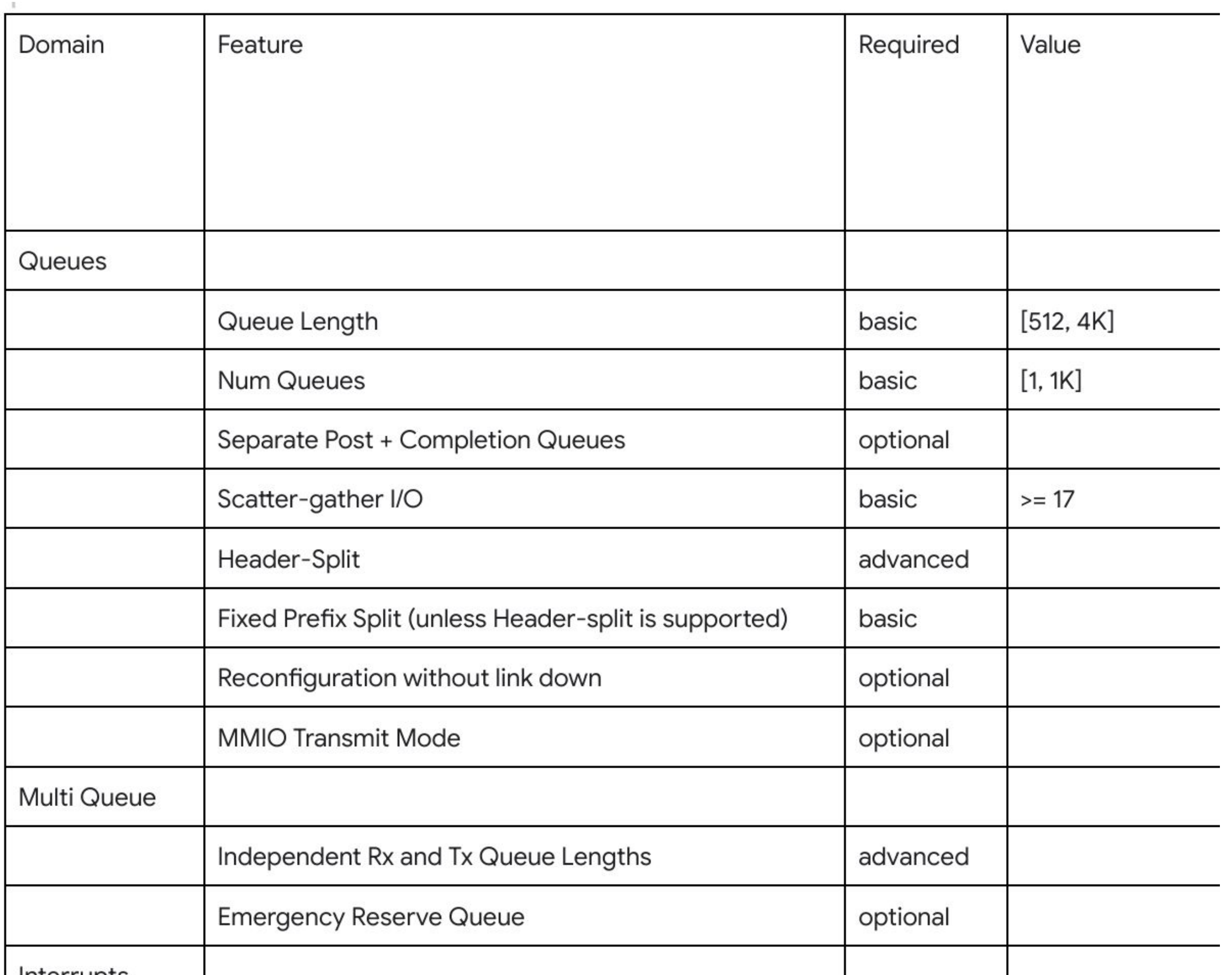
Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation



Domain	Feature	Required	Value
Queues			
	Queue Length	basic	[512, 4K]
	Num Queues	basic	[1, 1K]
	Separate Post + Completion Queues	optional	
	Scatter-gather I/O	basic	>= 17
	Header-Split	advanced	
	Fixed Prefix Split (unless Header-split is supported)	basic	
	Reconfiguration without link down	optional	
	MMIO Transmit Mode	optional	
Multi Queue			
	Independent Rx and Tx Queue Lengths	advanced	
	Emergency Reserve Queue	optional	
Interrupts			

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

- Configuration
- Functional
- Performance

Possible ways to measure

An appendix: *not* normative

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

- Configuration
- Functional
  - RSS
  - RSC
  - …
- Performance

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

- Configuration
- Functional
  - tools/testing/selftests/net/csum
  - tools/testing/selftests/net/gro
  - tools/testing/selftests/net/mmap
  - tools/testing/selftests/net/so_txtime*
  - tools/testing/selftests/net/toeplitz*
  - tools/testing/selftests/net/tso
  - tools/testing/selftests/net/udpgso*
  - github.com/wdebruij/kerneltools/blob/../tstamp.c
  - ip link
- Performance

Intro

I/O API

Offloads

Protocol Support

Telemetry

Performance

Appendix: Checklist

Appendix: Validation

- Configuration
- Functional
  - RSS (Toeplitz)
  - RSC
  - …
- Performance
  - neper tcp_rr, tcp_stream, udp_rr, ..
  - reproducible results
  - antagonists

Join the effort

How To Get Involved

- Certify Devices
- Contribute Tests + CI Infra
- Contribute Text: Review v1 for v1.1
- Contribute Text: Add features for v2

Certify Devices

- Validate with Test Suite
- Self Certify with Checklist
- Publish Certification
  - OCP Inspired
  - OCP Marketplace

100% compliance is not the goal. Spec is a starting point. Document differences

Community Ongoing Work: WIP

- Improve testsuite
- Help vendors certify devices
- Collect changes for v1.1
- Expand to new features for v2
  - PSP Inline Crypto, Other? (DDP, QUIC, ..)

- As First Party: As an OCP member, Sign CLA
- As Third Party: through OCP Networking mailing list + monthly call

The OCP Process (for this spec)

1. Find an OCP project: Networking
   - Present idea and get initial support
2. Form a group of contributors
3. Sign Contributor License Agreement
4. Develop Draft
   - Decide: in the open or closed
5. Share with Community for input
6. Present to OCP project
7. Present to OCP Incubation Committee
8. Sign Final Spec Agreement

1. The case for a NIC feature spec
2. OCP Spec v1 overview
3. Join

Questions

More Info: opencompute.org/wiki/Networking/NIC_Software
Contact: me, OCP networking mailing list, monthly call