COSH
Centre for Open-source
Software and Hardware

FQ-PIE

CONTRIBUTING TO OPEN STANDARDS BY CONTRIBUTING TO FOSS

Vishal Kamath

Agenda

COSH
Centre for Open-source Software and Hardware

1 The WHAT	A Open Standards	B The IETF	C FQ-PIE	
2 The WHY	A Importance	B Opportunity	C How to Contribute	
3 The HOW	A FQ-PIE vs FQ-CoDel	B go-tc	C QoSMate	D Class Stats in Linux Kernel

FOSS ✅ … What are Open Standards?

COSH
Centre for Open-source Software and Hardware

1. Publicly available **technical specification**, or protocol that anyone can **freely access, use, and share** without restrictions

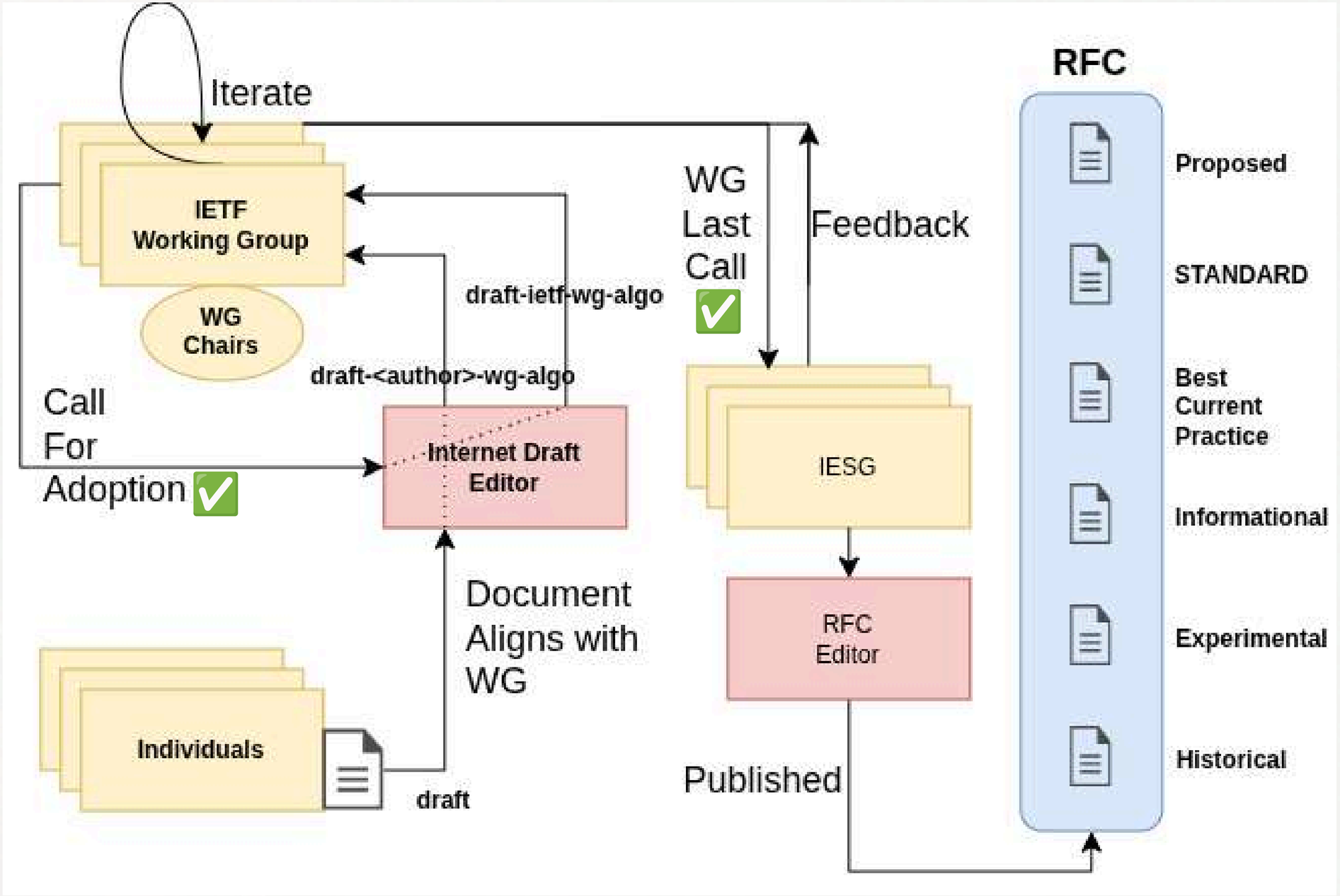
2. For software, provides reference point for any implementation - being generic enough - while also includes specific details and quirks

3. Example: IETF RFCs

The IETF

1. Internet Engineering Task Force
2. Standards for the Internet; Responsible for technical standards documents of the Internet Protocol suite = **RFCs**
3. Ex: DNS, TCP, IPv6, FQ-CoDel
4. All Communication & Work: Mailing List, Open Forum for each **Working Group**
5. **Meetings** (Ex: IETF 126) → Tri-annual hybrid meets to further WGs goals and tasks

COSH
Centre for Open-source Software and Hardware



– IETF Standards Timeline

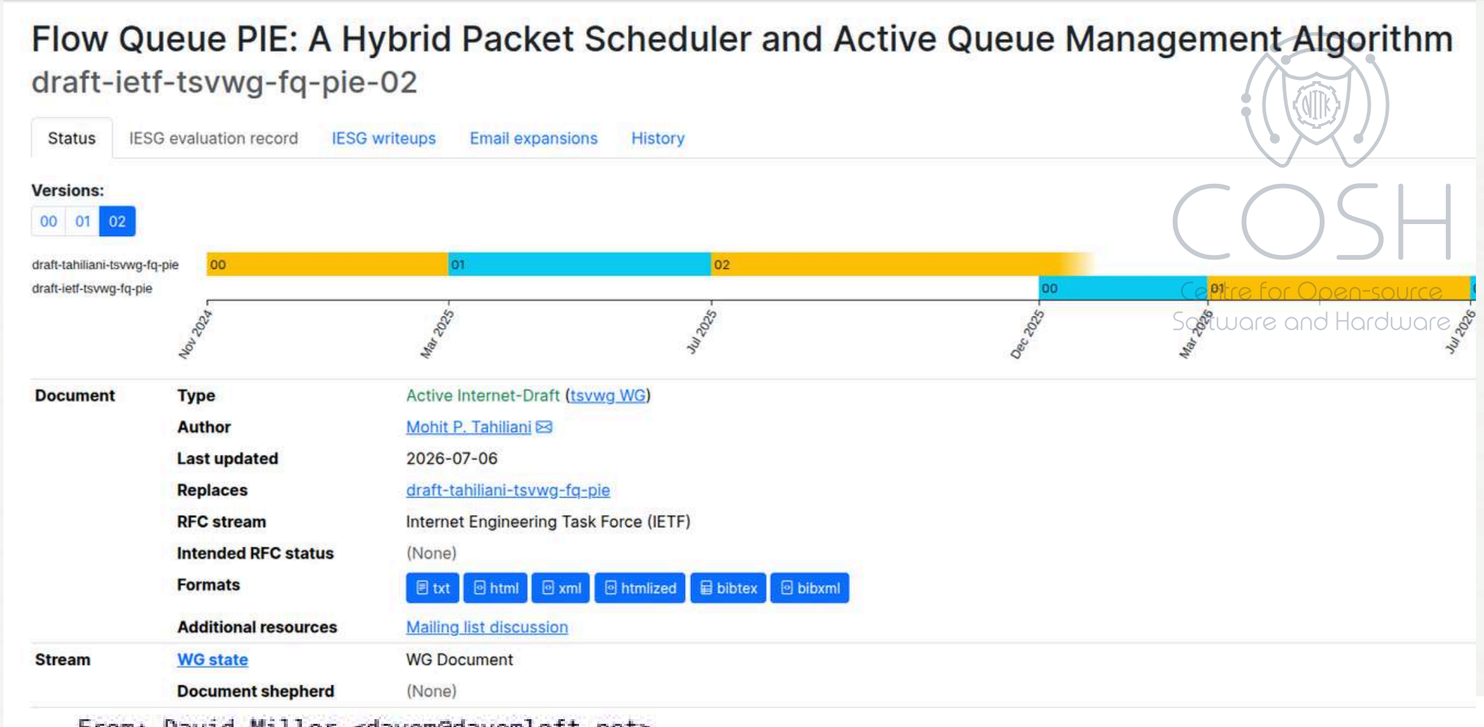
[Credit: Columbia University]

COSH
Centre for Open-source
Software and Hardware

What is FQ-PIE?

FQ-PIE

- Networking algorithm, developed by a research team at NITK, Surathkal
- Merged in Linux Kernel since January 2020
- Active Internet Draft in the **Transport and Services** WG
- Purpose: **REDUCE LATENCY** and improve fairness for Internet traffic
- **Active Queue Management** Algorithm, **Queueing** Discipline
- Current Default: **FQ-CoDel**


Flow Queue PIE: A Hybrid Packet Scheduler and Active Queue Management Algorithm
draft-ietf-tsvwg-fq-pie-02

Status | IESG evaluation record | IESG writeups | Email expansions | History

Versions: 00 01 02

draft-tahiliani-tsvwg-fq-pie: 00 01 02
draft-ietf-tsvwg-fq-pie: 00 01

Nov 2024 Mar 2025 Jul 2025 Dec 2025 Mar 2026 Jul 2026

Document	Type	Active Internet-Draft (tsvwg WG)
	Author	Mohit P. Tahiliani
	Last updated	2026-07-06
	Replaces	draft-tahiliani-tsvwg-fq-pie
	RFC stream	Internet Engineering Task Force (IETF)
	Intended RFC status	(None)
	Formats	txt html xml htmlized bibtex bibxml
	Additional resources	Mailing list discussion
Stream	WG state	WG Document
	Document shepherd	(None)

```
From: David Miller <davem@davemloft.net>
To: gautamramk@gmail.com
Cc: netdev@vger.kernel.org, jhs@mojatatu.com, dave.taht@gmail.com,
        toke@redhat.com, kuba@kernel.org, stephen@networkplumber.org,
        lesliemonis@gmail.com, tahiliani@nitk.edu.in
Subject: Re: [PATCH net-next v7 00/10] net: sched: add Flow Queue PIE packet scheduler
Date: Thu, 23 Jan 2020 11:39:03 +0100 (CET)     [thread overview]
Message-ID: <20200123.113903.125339377712611659.davem@davemloft.net> (raw)
In-Reply-To: <20200122182233.3940-1-gautamramk@gmail.com>

From: gautamramk@gmail.com
Date: Wed, 22 Jan 2020 23:52:23 +0530

> From: Gautam Ramakrishnan <gautamramk@gmail.com>
>
> Flow Queue PIE packet scheduler
>
> This patch series implements the Flow Queue Proportional
> Integral controller Enhanced (FQ-PIE) active queue
> Management algorithm. It is an enhancement over the PIE
> algorithm. It integrates the PIE aqm with a deficit round robin
> scheme.
> For more information:
> https://tools.ietf.org/html/rfc8033

Series applied, thank you.
```

COSH
Centre for Open-source Software and Hardware

Queue Management ~ Toll Booths!

COSH
Centre for Open-source Software and Hardware

Single Lane Booth



Credit

Multi Lane Booth



Credit

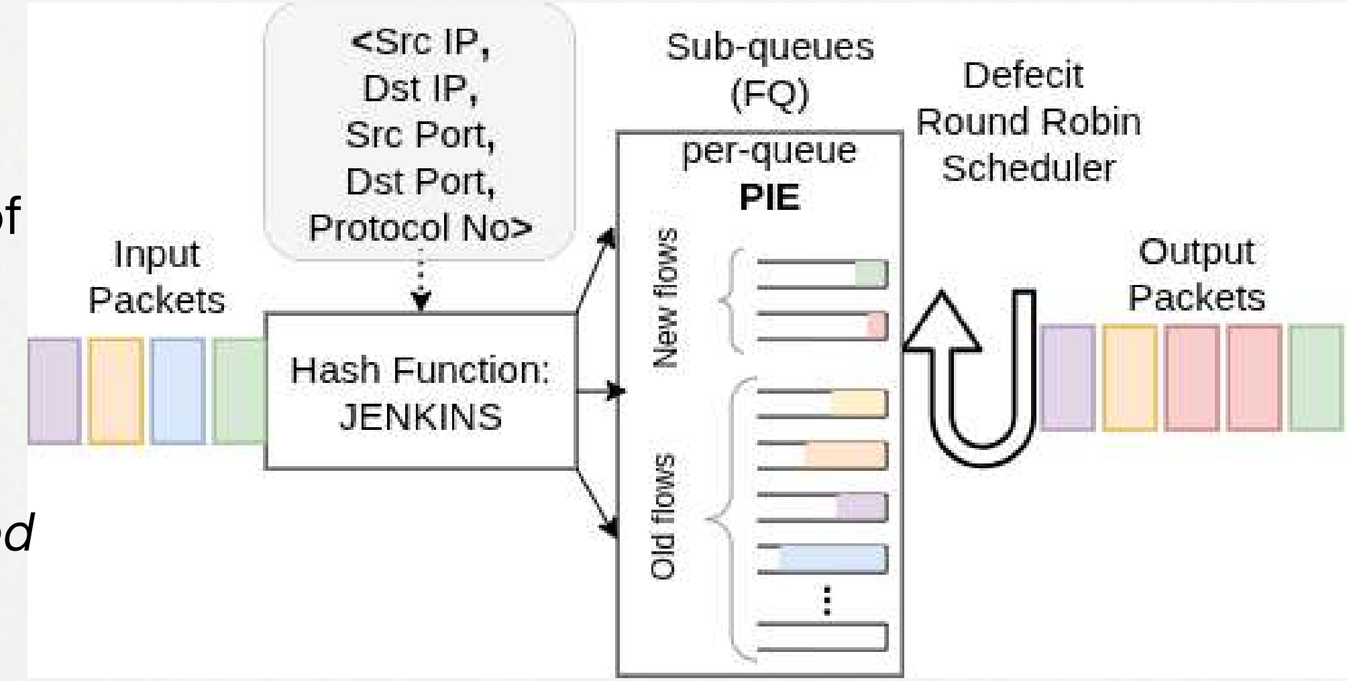
QoS: FASTag



Credit

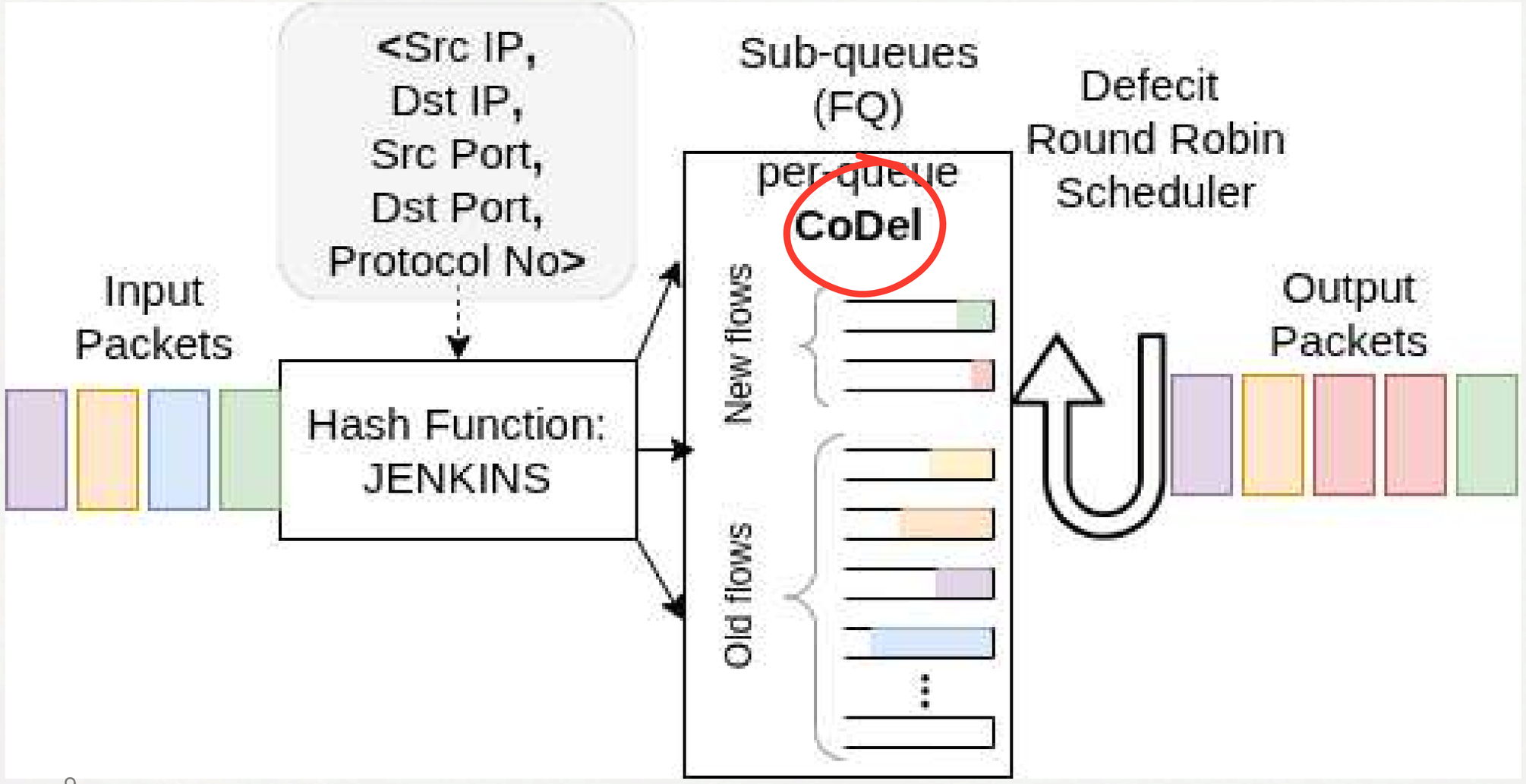
How FQ-PIE Works

- Packet 5-tuple: Source, Dest IP, Source, Dest Port, Protocol No. → **F**low **Q**ueue
- Use Jenkins Hash to put packet into one of 1024 buckets
- Each bucket managed by **PIE** algorithm: enqueue packet, or bucket too full?
- **PIE** uses *probability*, ***E**xponential **W**eighted **M**oving **A**verage* to make above decision
- Deficit *Round Robin* scheduler used to dequeue packets!
- Analogous to **default qdisc FQ-CoDel**
- FQ-CoDel (RFC 8290), PIE (RFC 8033)



Credit

COSH
Centre for Open-source Software and Hardware



– FQ-CoDel (default)

Importance of Open Standards

COSH
Centre for Open-source Software and Hardware

1. Gated, proprietarily developed protocols DO NOT enable community consensus, best practices

2. Protocol and specification growth, stability, and evolution can only prosper in THE OPEN

3. Open & FREE / Libre Standards form the Source of Truth for FOSS libraries, projects, etc.

Opportunity from Open Standards

COSH
Centre for Open-source Software and Hardware

1. Standards Development is First and Foremost: Engineering!

2. Collaborate with Engineers, Researchers, Scientists, Developers from around the Globe!

3. Build the Open Protocols that power systems all over world => Contribute to FOSS!

How to Contribute!

COSH
Centre for Open-source Software and Hardware

Example: The IETF

Review Internet Drafts!

- Provide feedback
- Identify improvement in clarity, wording, typos
- Ask questions!

Contribute to Reference Implementations:

- Various layers of the Network Stack
  - Operating System
  - Networking Library
  - CLI tool / Web Interface
- Check implementation compliance
- Open PRs to fix mismatches!

Test the Protocols!

- Plan physical and virtual topologies
- Design experiments
- Generate results, analyze data and plots
- Observe and suggest improvements, using appropriate evidence

Contributing to FQ-PIE: Timeline

COSH
Centre for Open-source Software and Hardware

Dec 2024 - Nov 2025

- Testing FQ-PIE vs FQ-CoDel in multiple Envs

Dec 2025

- Add FQ-PIE to go-tc
- Present at FOSS Mangalore meetup

May 2026 - Present

- Add class stats for FQ-PIE in Linux Kernel

Nov 2025

- Presented Work @ CCWG IETF 124

March 2026

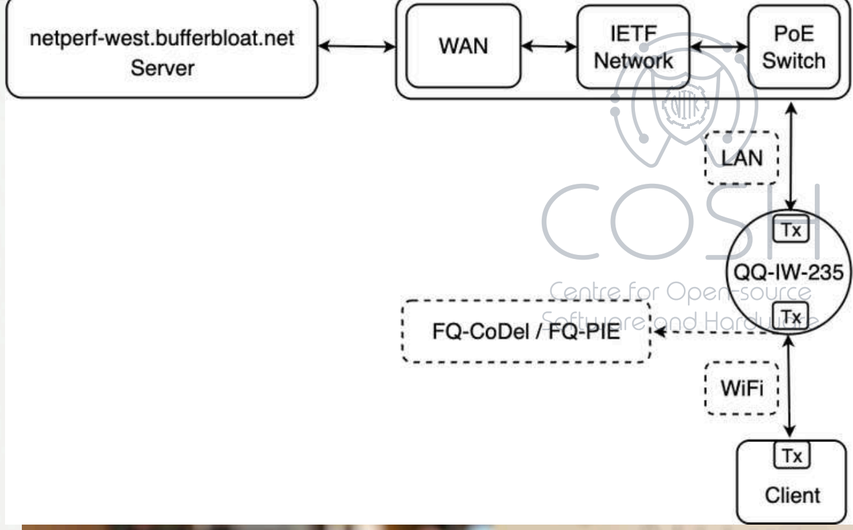
- Add FQ-PIE to QoSMate
- Inform @ tsvwg IETF 125

July 2026

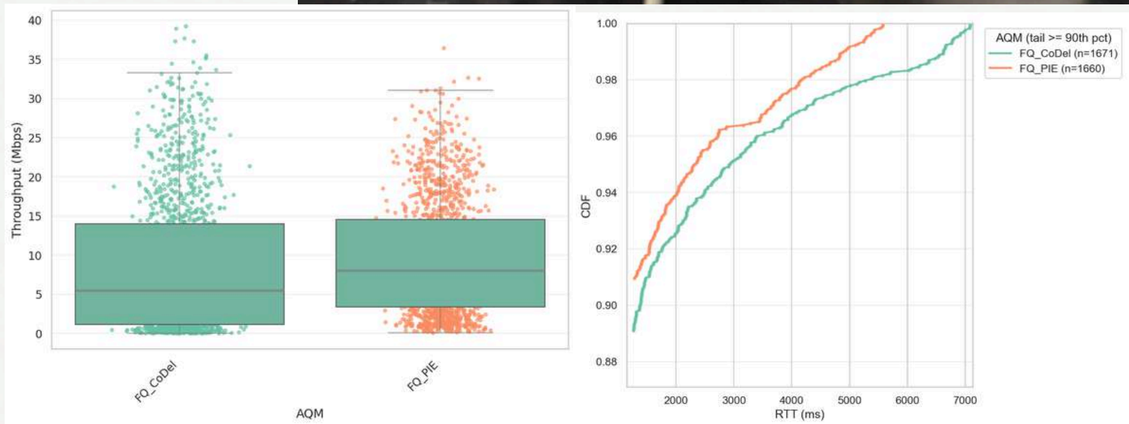
- Inform @ tsvwg IETF 126

FQ-PIE vs FQ-CoDel

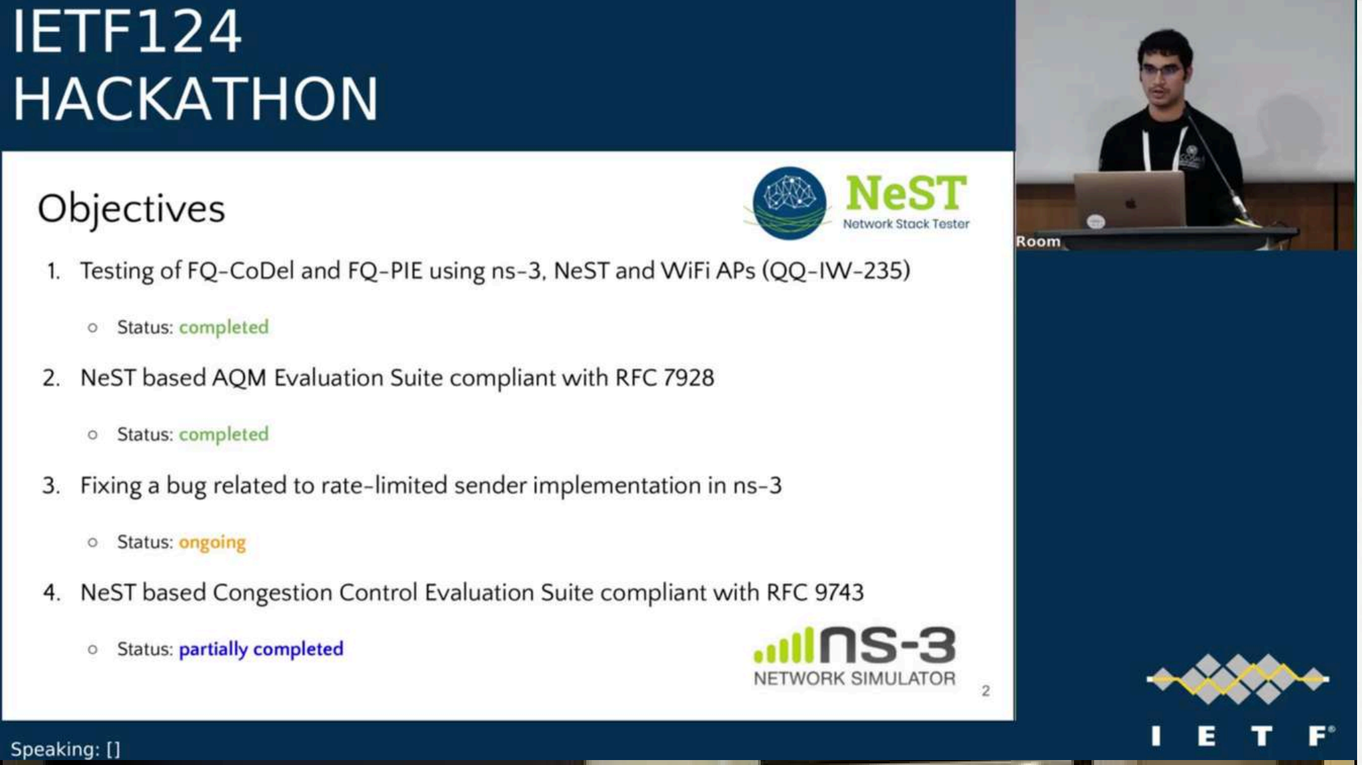
- Started as 5th Semester course assignment.
- Carried out comparisons between FQ-CoDel & FQ-PIE on latency and throughput.
- Conducted experiments across testbeds in:
  - Linux Network Namespaces
  - OpenWRT Routers
  - Mobile Hotspots
- Tested in networks like NITK campus network, IETF 124 hackathon network, etc.







Thanks to Quantum Networks for providing their APs!





IETF 122, 124

COSH
Centre for Open-source Software and Hardware

- Work presented at IETF 122 by Prof. Mohit P. Tahiliani
- Ran experiments onsite and presented results @ IETF 124 in:
  - Hackathon
  - CCWG
  - Hack Demo Happy Hours
- Received feedback on what we could improve, what else to test

FQ-PIE: go-tc

- Network library to configure these Qdiscs (normally use CLI)
- FQ-PIE implementation absent!
- Implement the NETLINK API for FQ-PIE
- Marshall and Unmarshall methods for FQ-PIE requests / messages
- Unit and integration tests to go-tc

README | MIT license | Security

tc reference | go report A+ | Go passing | coverage 92%

This is a work in progress version of `tc`. It provides a C-binding free API to the netlink based traffic control system of rtnetlink.

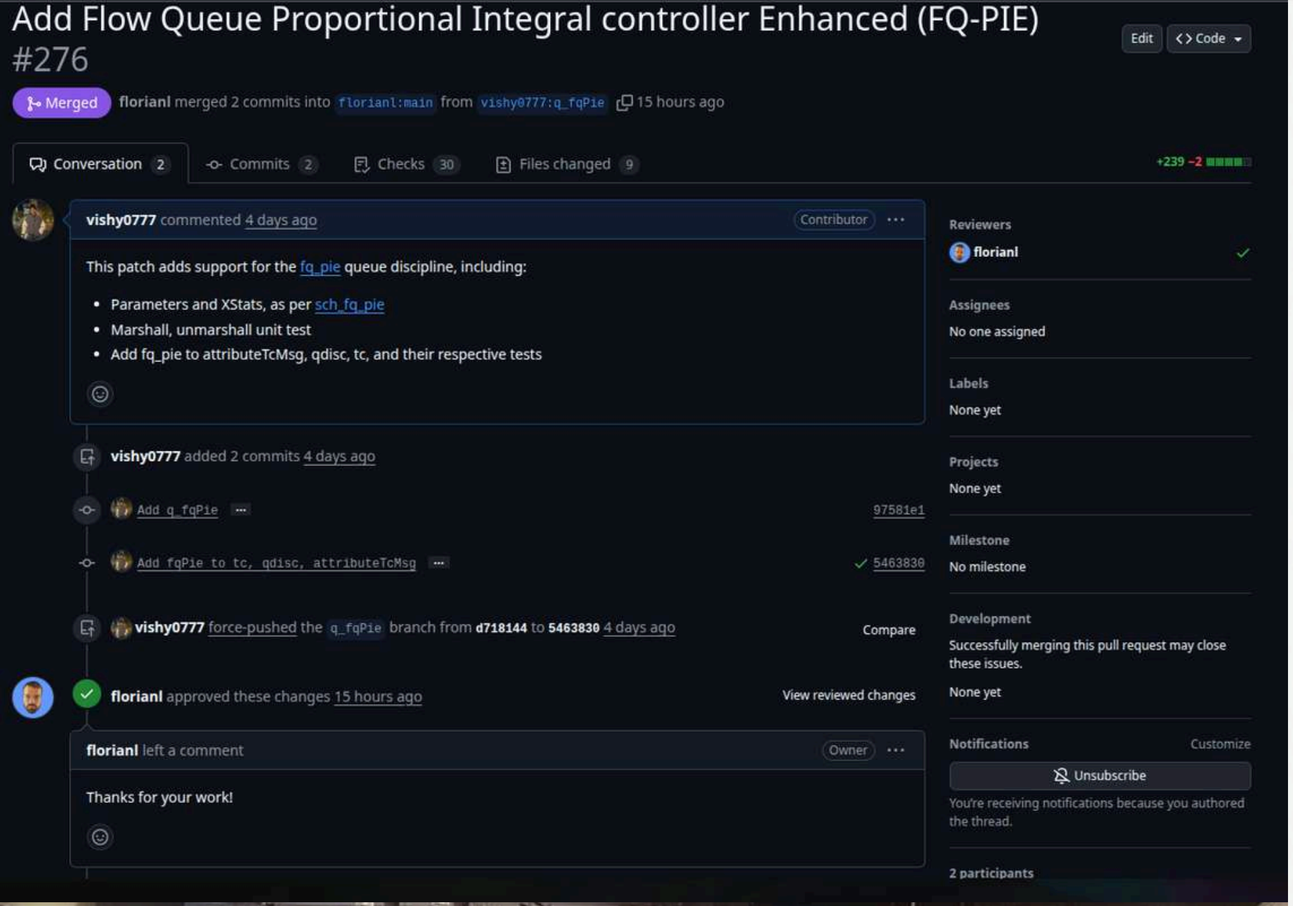
Example

```
const (
    tcaFqPieUnspec = iota //
    tcaFqPieLimit
    tcaFqPieFlows
    tcaFqPieTarget
    tcaFqPieTUpdate
    tcaFqPieAlpha
    tcaFqPieBeta
    tcaFqPieQuantum // sch_fq
    tcaFqPieMemoryLimit
    tcaFqPieEcnProb
    tcaFqPieEcn
    tcaFqPieBytemode
    tcaFqPieDqRateEstimator
)
```

```
type FqPie struct {
    Limit           *uint32
    Flows           *uint32
    Target          *uint32
    TUpdate         *uint32
    Alpha           *uint32
    Beta            *uint32
    Quantum         *uint32
    MemoryLimit     *uint32
    EcnProb         *uint32
    Ecn             *uint32
    Bytemode        *uint32
    DqRateEstimator *uint32
}
```

```
// marshalFqPie returns the binary encoding of FqPie
func marshalFqPie(info *FqPie) ([]byte, error) {…
}

// unmarshalFqPie parses the FqPie-encoded data and st
func unmarshalFqPie(data []byte, info *FqPie) error {
}
```





Merge Request & FOSS United Mangalore

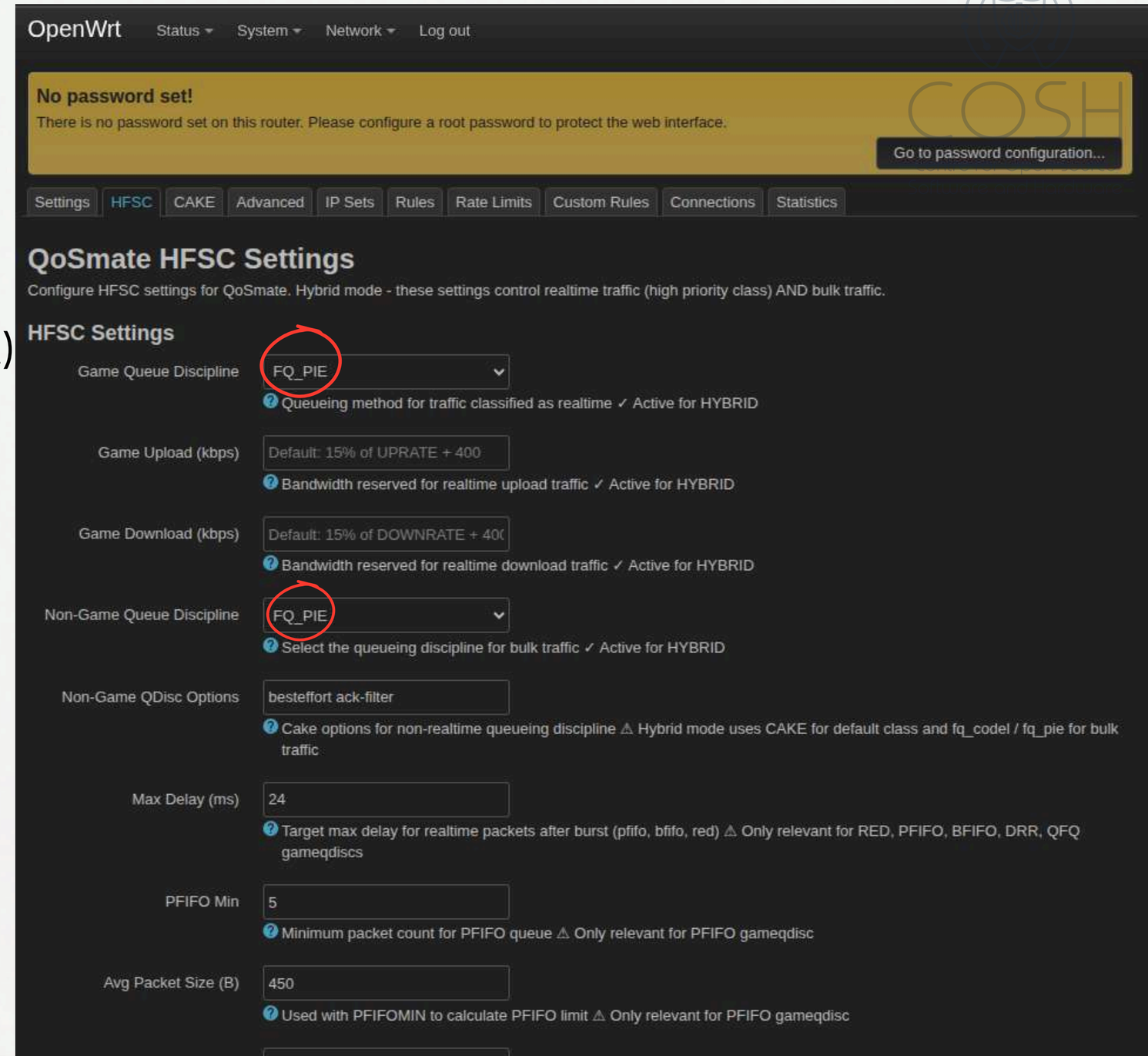
COSH
Centre for Open-source Software and Hardware

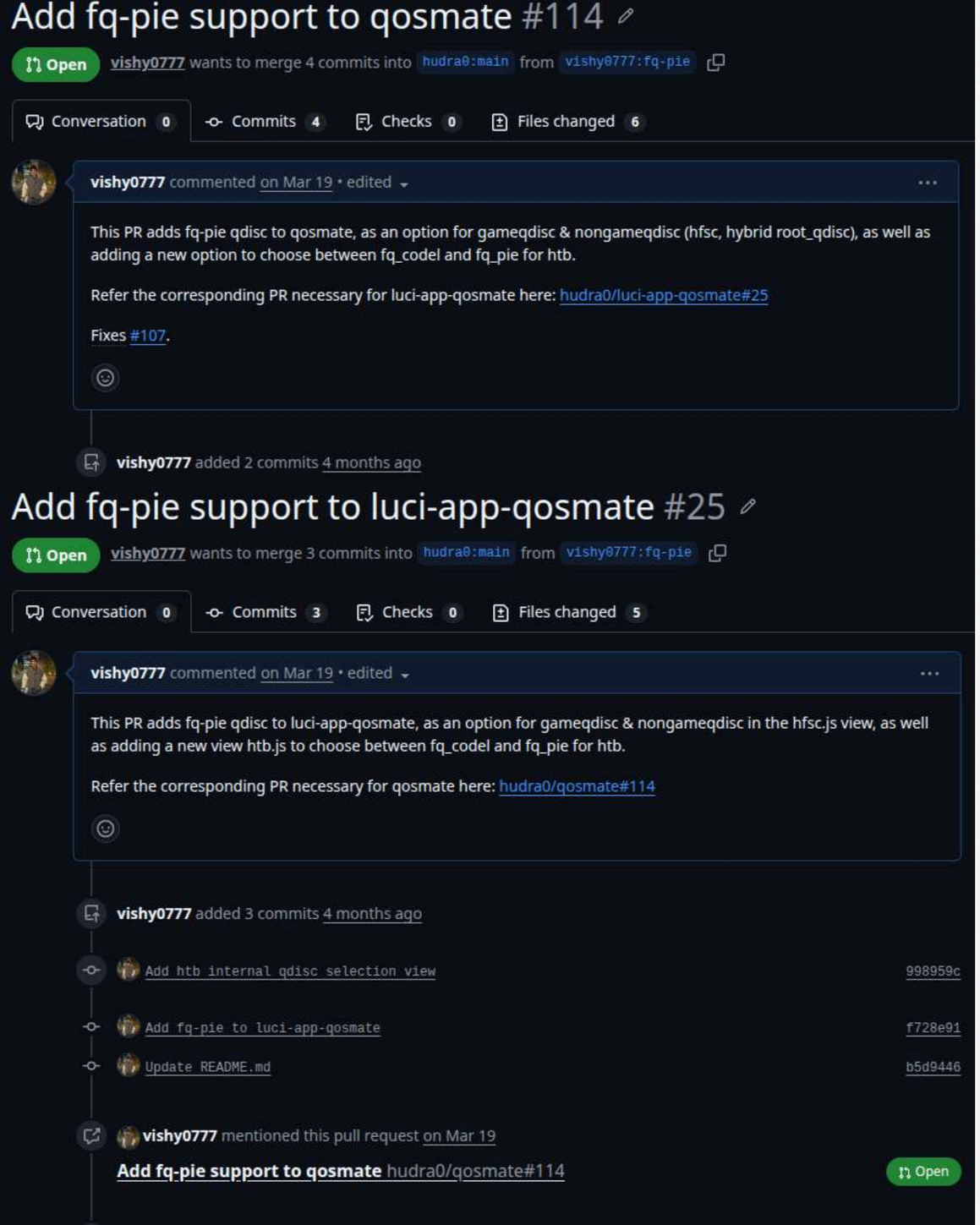
- FQ-PIE implementation was successfully merged!
- This was my 1st contribution to FOSS!
- Gave a talk at the December FOSS Activity @ FOSS Mangalore

FQ-PIE: QoSMate

- QoS tool for OpenWRT routers to optimize performance & traffic prioritization (gaming, real-time, etc.)
- QoSMate provides LuCI web interface for easy config
- NO FQ-PIE!

- Enable FQ-PIE on router, enable for all scenarios
- Configure FQ-PIE parameters / knobs for the all 3 scenarios
- Modify & implement new LuCI view for FQ-PIE selection (Javascript!)





Merge Request

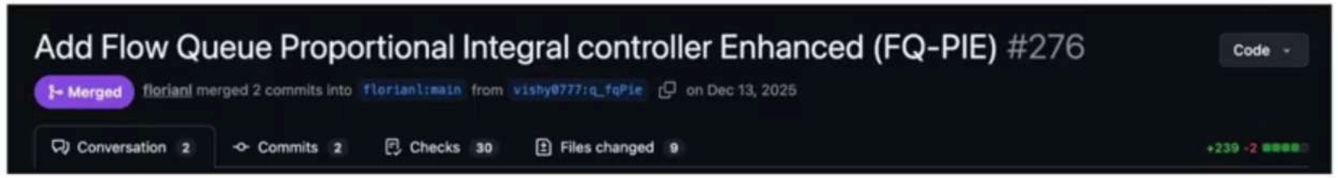
COSH
Centre for Open-source Software and Hardware

- PR opened in 2 repos:
  - For the bash scripts to configure FQ-PIE in QoSMate
  - For the LuCI web app
- New changes made to QoSMate => updates in the PRs!
- In progress to get merged!

IETF125
TSVWG

Enabling support of FQ-PIE in tools

- go-tc: traffic control in pure go - it allows to read and alter queues, filters and classes
  - Enabled the support for FQ-PIE: https://github.com/florianl/go-tc/pull/276



- QoSmate: Quality of Service tool for OpenWrt
  - Ongoing work to enable support for FQ-PIE:
    https://github.com/hudra0/qosmate/pull/114
    https://github.com/hudra0/luci-app-qosmate/pull/25

Credits: Vishal Kamath

5

Speaking: [Mohit Tahiliani]



COSH
Centre for Open-source Software and Hardware

IETF 125

FQ-PIE: Class stats in the Linux Kernel

- Per-bucket statistics:
  - FQ-CoDel ✅
  - FQ-PIE ❌
- Collaborated with **Hemendra Naik** to work on patches to Kernel, iproute2!
- *tc_fq_pie_xstats* userspace ABI gets *tc_fq_pie_cl_stats* appended
- Calculate per-flow stats in kernel
- Parse stats in userspace tool **tc**

```
+enum {
+       TCA_FQ_PIE_XSTATS_QDISC,
+       TCA_FQ_PIE_XSTATS_CLASS,
+};
+
+struct tc_fq_pie_cl_stats {
+       __u64 prob;                     /* current probability */
+       __u32 delay;                    /* current delay in microseconds */
+       __s32 deficit;          /* number of remaining byte credits */
+       __u32 avg_dq_rate;              /* current average dq_rate in
+                                        * bytes/second
+                                        */
+       __u32 dq_rate_estimating;       /* is avg_dq_rate being calculated? */
+};
+
+struct tc_fq_pie_xqd_stats {
+       /* placeholder for new qdisc-level stats */
+};
+
 struct tc_fq_pie_xstats {
        __u32 packets_in;       /* total number of packets enqueued */
        __u32 dropped;          /* packets dropped due to fq_pie_action */
@@ -963,6 +982,12 @@  struct tc_fq_pie_xstats {
        __u32 new_flows_len;    /* count of flows in new list */
        __u32 old_flows_len;    /* count of flows in old list */
        __u32 memory_usage;     /* total memory across all queues */
+       __u32 type;
+       union {
+                struct tc_fq_pie_cl_stats class_stats;
+                struct tc_fq_pie_xqd_stats xqdisc_stats;
+       };
+
 };
```

Patch Submission

COSH
Centre for Open-source Software and Hardware

Patchwork Netdev + BPF | Patches | Bundles | About this project | 0 vishy0777

Show patches with: Submitter = **Hemendra** | Archived = **No** | 34 patches

Change state | Delegate to | **Archived:** No change | Update | Bundle name | Create

Patch	A/R/T	S/W/F	Date	Submitter	Delegate	State
[iproute2-next,v7,2/2] tc: pie: print avg_dq_rate in human-readable format	- -	1 - -	2026-09-22	Hemendra M. Naik	dsa	New
[iproute2-next,v7,1/2] tc: fq_pie: add support for printing per-flow PIE statistics	- - -	1 - -	2026-09-22	Hemendra M. Naik	dsa	New
[net-next,v8,3/3] net/sched: pie: correct tc_pie_xstats field documentation	- - -	27 - -	2026-09-22	Hemendra M. Naik	netdev	Under Review
[net-next,v8,2/3] selftests: tc-testing: add fq_pie per-flow class stats test	- - -	27 - -	2026-09-22	Hemendra M. Naik	netdev	Under Review
[net-next,v8,1/3] net/sched: sch_fq_pie: add per-flow statistics via class ops	- -	27 -	2026-09-22	Hemendra M. Naik	netdev	Under Review
[iproute2-next,v6,3/3] tc: correct tc_pie_xstats field documentation	- - -	- - 1	2026-09-19	Hemendra M. Naik	dsa	Superseded
[iproute2-next,v6,2/3] tc: pie: print avg_dq_rate in human-readable format	- - -	- - 1	2026-09-19	Hemendra M. Naik	dsa	Superseded
[iproute2-next,v6,1/3] tc: fq_pie: add support for printing per-flow PIE statistics	- - -	- - 1	2026-09-19	Hemendra M. Naik	dsa	Superseded
[net-next,v7,3/3] net/sched: pie: correct tc_pie_xstats field documentation	- - -	27 1 -	2026-09-19	Hemendra M. Naik	netdev	Superseded
[net-next,v7,2/3] selftests: tc-testing: add fq_pie per-flow class stats test	- - -	27 1 -	2026-09-19	Hemendra M. Naik	netdev	Superseded
[net-next,v7,1/3] net/sched: sch_fq_pie: add per-flow statistics via class ops	- - -	27 1 -	2026-09-19	Hemendra M. Naik	netdev	Superseded

IETF126
TSVWG

net-next patch for per flow stats collection for FQ-PIE

```
FQ-PIE runs an independent PIE controller per flow but exposes no
per-flow statistics. This series wires up fq_pie_class_ops to expose
per-flow AQM state (prob, delay, deficit, avg_dq_rate)
via 'tc -s class show', following a similar pattern as FQ-CoDel.
---
Changelog:

v3:
 - No changes since v2.
 - Resending as the previous submission was deferred when the net-next tree closed during review.
 - Corresponding iproute2 patch updated in response to review comments; no changes required for this patch.

v2:
- Addressed ABI backward compatibility issue for tc_fq_pie_xstats. (https://lore.kernel.org/netdev/20260614125000.6058-1-hemendranaik@gmail.com/)

v1:
- https://lore.kernel.org/netdev/20260531125314.22492-1-hemendranaik@gmail.com/

Hemendra M. Naik (2):
  net/sched: sch_fq_pie: add per-flow statistics via class ops
  selftests: tc-testing: add fq_pie per-flow class stats test

 include/uapi/linux/pkt_sched.h               |  29 ++++-
 net/sched/sch_fq_pie.c                       | 118 +++++++++++++++++++
 tools/include/uapi/linux/pkt_sched.h         |   4 +-
 .../tc-testing/tc-tests/qdiscs/fq_pie.json   |  22 ++++
 4 files changed, 163 insertions(+), 10 deletions(-)

--
2.34.1
```

Credits: Hemendra Naik and Vishal Kamath

10





Speaking: []

COSH
Centre for Open-source Software and Hardware

IETF 126

Dave Taït

In Memoriam



Champion of **FQ-CoDel**, *CAKE*

Champion of Fighting The **Bufferbloat** & Reducing **LATENCY**

Continues to INSPIRE US ALL

COSH
Centre for Open-source Software and Hardware

Questions?

THANK YOU!

Collaborators:

Dr. Mohit P Tahiliani

Hemendra M Naik