QoS Challenges for Real Time Traffic

[tj]
tj@enoti.me

NEAT is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 644334.
# Internet Protocol Datagram

**Source**

**Destination**

## Type of Service
- [ ] high reliability
- [ ] high throughput
- [ ] low delay

## Protocol
- [ ] TCP
- [ ] UDP
- [ ] Other __________

## Precedence
- [ ] Routine
- [ ] Priority
- [ ] Immediate
- [ ] Flash
- [ ] Flash Override
- [ ] CRITIC/ECP
- [ ] Internetwork Control
- [ ] Network Control

## Fragmentation
- [ ] more to follow
- [ ] do not fragment
- [ ] this bit intentionally left blank

## Offset

## Identifier _______________________

## Length

## Header Length

## Data

*Print legibly and press hard. You are making up to 255 copies.*

## Time to Live

## Options

- [ ] Do not write in this space.

## Header Checksum

---

*For more info, check IPv4 specifications at [http://www.ietf.org/rfc/rfc0791.txt](http://www.ietf.org/rfc/rfc0791.txt)*
Packets
Packets
Packets
Middleboxes
Middleboxes
Middleboxes
Middleboxes
Per Hop Behaviours

• Default Forwarding
Differentiated Services
Differentiated Services

Me

You
Smart Queueing
Smart Queueing
Smart Queueing
Smart Queueing
Per Hop Behaviours

• Default Forwarding
  • The default PHB has best-effort (BE) forwarding characteristics

• Expedited Forwarding
  • The EF PHB has the characteristics of low delay, low loss and low jitter.

• Assured Forwarding
  • Assured forwarding allows the operator to provide assurance of delivery as long as the traffic does not exceed some subscribed rate.
The IP Type of Service (TOS) Field

7 6 5 4 3 | 2 1 0

Type of Service | Precedence
The IP Type of Service (TOS) Field

\[
\begin{array}{cccccc|cc}
7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\hline
\text{Type of Service} & | & \text{Precedence} \\
\hline
\text{Diff Serv Code Point} & | & \text{ECN} \\
\end{array}
\]
What Happens to Code Points?

- Generate a set of points to evaluate
  - RFC recommendations
- Measurement survey
  - Pass/Fail Test on Code Points
  - Verify Code Group Treatment
DSCP Treatment

- Pass
- Drop
- Bit Bleaching
- Remark
Code Points recommended by recent RFC’s

• Which Code Points should we use?
  • 80211
    • draft-ietf-tsvwg-ieee-802-11
  • MPLS
    • RFC5127
    • draft-ietf-tsvwg-diffserv-intercon-14
  • WebRTC
    • draft-ietf-tsvwg-rtcweb-qos
Code Points recommended by recent RFC’s

<table>
<thead>
<tr>
<th></th>
<th>CS0</th>
<th>AF11</th>
<th>AF31</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS1</td>
<td></td>
<td>AF12</td>
<td>AF32</td>
</tr>
<tr>
<td>CS2</td>
<td></td>
<td>AF13</td>
<td>AF33</td>
</tr>
<tr>
<td>CS3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS4</td>
<td></td>
<td>AF21</td>
<td>AF41</td>
</tr>
<tr>
<td>CS5</td>
<td></td>
<td>AF22</td>
<td>AF42</td>
</tr>
<tr>
<td>CS6</td>
<td></td>
<td>AF23</td>
<td>AF43</td>
</tr>
</tbody>
</table>

21 code points recommended
Graphs
Initial DSCP : 0 (BE)
Percentage of DSCP at the last hop of a path, for UDP and TCP

- 95%
- Number of flows : 600
- Average number of hops before first change : 5.4
- Average number of hops per path : 6.3

DSCP:
- 0 (BE)
- 8 (CS1)
- 10 (AF11)
- 18 (AF21)
Initial DSCP: 8 (CS1)
Percentage of DSCP at the last hop of a path, for UDP and TCP

Number of flows: 600
Average number of hops before first change: 3.6
Average number of hops per path: 6.3

- 45% for 0 (BE)
- 53% for 8 (CS1)
- 1% for 18 (AF21)
Initial DSCP : 46 (EF)
Percentage of DSCP at the last hop of a path, for UDP and TCP

Number of flows : 600
Average number of hops before first change : 3.7
Average number of hops per path : 6.3
Awesome! How do I use this?
Awesome! How do I use this?

```c
uint8_t dscp = 0x2e  //EF
```
Awesome! How do I use this?

```c
uint8_t dscp = 0x2e; // EF
uint8_t ecn = 0;
```
Awesome! How do I use this?

```c
uint8_t dscp = 0x2e; // EF
uint8_t ecn = 0;
uint8_t tos = dscp << 2 | ecn;
```
Awesome! How do I use this?

```c
uint8_t dscp = 0x2e //EF
uint8_t ecn = 0;
uint8_t tos = dscp << 2 | ecn;
if (setsockopt(flow->socket->fd, IPPROTO_IP, IP_TOS, &tos, sizeof(tos)) == -1) {
    return ERROR;
}
return OK;
```
The NEAT User Module

Diagnostics and Statistics
- NEAT Framework Components
- NEAT Selection Components
- NEAT Transport Components
- NEAT Policy Components
- NEAT Signalling and Handover Components

USER

KERNEL
- KPI
- Traditional Socket
- NEAT Socket

Policy Interface

NEAT User API
NEAT Application

APP Class 1

NEAT User API

Userspace Transport

NEAT User Module

TCP
UDP
SCTP
IP

Traditional Socket

USER
KERNEL
static struct neat_flow_operations ops;
static struct neat_ctx *ctx = NULL;
static struct neat_flow *flow = NULL;

ctx = neat_init_ctx()
flow = neat_new_flow(ctx)

prop = NEATPROPERTY_UDP_REQUIRED | NEATPROPERTY_IPV6_REQUIRED;
neat_set_property(ctx, flow, &prop)

ops.on_writable = on_writable;
ops.on_readable = on_readable;
ops.on_error = on_error;

neat_set_operations(ctx, flow, &ops)
neat_open(ctx, flow, argv[argc - 2], argv[argc - 1])
neat_start_event_loop(ctx, NEAT_RUN_DEFAULT);
static neat_error_code
on_writable(struct neat_flow_operations *opCB)
{
    neat_write(opCB->ctx, opCB->flow, buf)
    return NEAT_OK;
}

static neat_error_code
on_readable(struct neat_flow_operations *opCB)
{
    neat_read(opCB->ctx, opCB->flow, buf)
    return NEAT_OK;
}

https://github.com/NEAT-project/neat/blob/master/examples/client.c
NEAT QoS Setting
NEAT QoS Setting

```c
neat_set_qos(flow->ctx, flow, 0x2e);
```
NEAT QoS Setting

neat_set_qos(flow->ctx, flow, 0x2e);
neat_set_qos(flow->ctx, flow, NEAT_DSCP_EF);
NEAT QoS Setting

neat_set_qos(flow->ctx, flow, 0x2e);
neat_set_qos(flow->ctx, flow, NEAT_DSCP_EF);
neat_set_qos(flow->ctx, flow, NEAT_QOS_REALTIME_INTERACTIVE_DATA);
neat

https://www.neat-project.org

https://github.com/neat-project/neat

NEAT is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 644334.