Transport Evolution on top of the BSD's

[tj]
tj@enoti.me

NEAT is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 644334.
Cameras up mountains
TCP Congestion Control

- Slow Start Phase
- Steady State
- Multiplicative Reduction to Loss
HTTP Video Workloads

- Bursty
- Terrible Reaction to loss
  - ‘Confused, Timid and Unstable’
- Very long connection life time
- Lots of packets in flight
NewCWV

• TCP Adaptation for Rate Limited Traffic
• Improved window validation
• FreeBSD Implementation here:
  • https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=191520
TCP Congestion Window

TCP Congestion Window (Bytes)

Time

5 Minutes
pipeACK sample (Bytes)

<table>
<thead>
<tr>
<th>Sample A</th>
<th>Sample B</th>
<th>No Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sample</td>
</tr>
<tr>
<td>\ 5</td>
<td></td>
<td>\</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\</td>
</tr>
<tr>
<td>\ 3</td>
<td></td>
<td>\</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\</td>
</tr>
<tr>
<td>\ ---</td>
<td></td>
<td>\</td>
</tr>
<tr>
<td>/ \ -----</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

+---+-+-+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+---+-+

+----------+----------+ +----------+----------+

+----------+----------+

Figure 1: Example of Measuring pipeACK Samples
NewCWV Status

<table>
<thead>
<tr>
<th>Hiren Panchasara</th>
<th>2016-12-22 20:22:46 UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>This work is supposed to land from Netflix in coming months.</td>
<td>Comment 5</td>
</tr>
</tbody>
</table>
Internet Protocol Datagram

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
</table>

**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**
- __________

**Length**
- __________

**Header Length**
- __________

**Data**

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

for more info, check IPv4 specifications at http://www.ietf.org/rfc/rfc0791.txt
The Naive View of the Network
The Reality of the Network

Middleboxes drop non http traffic
Firewalls Block protocols
Security devices inject ‘bad’ data
Proxies break TLS
Networks vary a ton

Middleboxes drop non http traffic
Firewalls Block protocols
Security devices inject ‘bad’ data
Proxies break TLS

2.5G is slow
3G has unpredictable latency
WiFi maintains really high bandwidth
4G has high, but variability in bandwidth
The socket API has ossified

`getaddrinfo();` // Look up host
`socket();` // Create a socket
`setsockopt();` // Configure the socket
`getsockopt();` // Check parameters
`connect();` // Start connection
`send();`
`recv();`
Do it in userspace

- usersctp
- QUIC
  - draft-ietf-quic-transport-01
- GUE
  - draft-ietf-nvo3-gue-05
- TCP over UDP
The NEAT System

- APP Class 0
- APP Class 1
- APP Class 2
- APP Class 3
- APP Class 4

Middleware

NEAT APP Support API
NEAT APP Support Module

NEAT User API
NEAT Framework

Transport Components
Selection Components
H and S Components

Policy Interface
NEAT Policy Manager

Policy Information Base
Characteristic Information Base

Userspace Transport
SCTP/UDP
...
SPUD/UDP
Exp Mech

Traditional Socket
NEAT Socket

NEAT Kernel Module

PCAP
RAW IP
Experimental Mechanisms
TCP
UDP
SCTP
SCTP/UDP
TCP Minion
Experimental Mechanisms
NEAT Application

APP Class 1

NEAT User API

NEAT User Module

Userspace Transport

SCTP/UDP  ...  Exp Mech

TCP  UDP  SCTP

IP

KERNEL

USER
The NEAT User Module
The NEAT User Module

NEAT Flow Endpoint

<table>
<thead>
<tr>
<th>NEAT User API</th>
</tr>
</thead>
</table>
The NEAT User Module

5 Groups of components:
The NEAT User Module

5 Groups of components:
- NEAT Framework Component: API, Logic
The NEAT User Module

5 Groups of components:
- NEAT Framework Component: API, Logic
- NEAT Selection Components: Choose candidates
The NEAT User Module

5 Groups of components:
- NEAT Framework Component: API, Logic
- NEAT Selection Components: Choose candidates
- NEAT Policy Components: Policy and Characteristics
The NEAT User Module

5 Groups of components:
▶ NEAT Framework Component: API, Logic
▶ NEAT Selection Components: Choose candidates
▶ NEAT Policy Components: Policy and Characteristics
▶ NEAT Transport Components: Instantiate transports
The NEAT User Module

5 Groups of components:
- NEAT Framework Component: API, Logic
- NEAT Selection Components: Choose candidates
- NEAT Policy Components: Policy and Characteristics
- NEAT Transport Components: Instantiate transports
- NEAT Handover and Signalling Components
The NEAT User Module

5 Groups of components:

- NEAT Framework Component: API, Logic
- NEAT Selection Components: Choose candidates
- NEAT Policy Components: Policy and Characteristics
- NEAT Transport Components: Instantiate transports
- NEAT Handover and Signalling Components
NEAT Application

```c
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 = NEAT_PROPERTY_UDP_REQUIRED | NEAT_PROPERTY_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
Porting Apps

• Firefox
• rsync
**Waterfall**

<table>
<thead>
<tr>
<th>last build</th>
<th>freebsd-arm</th>
<th>freebsd-head</th>
<th>freebsd-stable</th>
<th>netbsd</th>
<th>osx</th>
<th>ubuntu</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle</td>
<td>build successful</td>
<td>build successful</td>
<td>build successful</td>
<td>build successful</td>
<td>idle</td>
<td>failed compile</td>
</tr>
<tr>
<td>current activity</td>
<td>idle</td>
<td>idle</td>
<td>idle</td>
<td>idle</td>
<td>idle</td>
<td>idle</td>
</tr>
<tr>
<td>changes</td>
<td>freesbsd-arm</td>
<td>freesbsd-head</td>
<td>freesbsd-stable</td>
<td>netbsd</td>
<td>osx</td>
<td>ubuntu</td>
</tr>
<tr>
<td>CET</td>
<td>Fri 03 Feb 2017 17:05:02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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