Flow Utility Based Routing (FUBAR)

Nikola Gvozdiev, Brad Karp, Mark Handley
University College London (UCL)
Networks today

- Flow demands
- Path assignment (shortest path)
Networks today

Flow demands -> TCP (RTT-weighted fairness) -> Path assignment (shortest path)
Networks today

Flow demands → TCP (RTT-weighted fairness) → Some utility

TCP (RTT-weighted fairness) → Path assignment (shortest path)
Networks today

Flow demands

TCP (RTT-weighted fairness)

Path assignment (shortest path)

Fails to take advantage of path diversity

Some utility
Networks today

Flow demands

Path assignment (shortest path)

TCP (RTT-weighted fairness)

Higher utilization?

Lower utility
Networks today

Flow demands

Path assignment (shortest path)

TCP (RTT-weighted fairness)

Higher utilization?

Lower utility
Networks today

Flow demands

Path assignment (shortest path)

TCP (RTT-weighted fairness)

Higher utilization?

Better utility?

TE tweaks

~ hours

guess
Application-centric design

Flow demands → Measure demands → Path assignment (shortest path)

TCP (RTT-weighted fairness) → Best utility

Higher utilization? → ~ a minute
Why does this matter (today)?

• Throughput-hungry applications – would like to drive utilization up

• A lot of the flows are delay-critical
  – Bulk of TCP flows terminate very very quickly
  – VoIP and real-time traffic

• How can we achieve high utilization without impacting latency?
Why does this matter (today)?

• Throughput-hungry applications – would like to drive utilization up

• A lot of the flows are delay-critical
  – Bulk of TCP flows terminate very quickly
  – VoIP and real-time traffic

• How can we achieve high utilization without impacting latency?

Assuming traffic demands are stable, can we reconcile both throughput and delay requirements?
Status quo I: Minimize delay

All links are 10Gbps
Status quo I: Minimize delay

All links are 10Gbps
15Gbps A ➔ F and 10Gbps E ➔ B
Status quo I: Minimize delay

All links are 10Gbps
15Gbps A → F and 10Gbps E → B
Will fill the shortest path and exclude bottlenecks
Status quo I: Minimize delay

All links are 10Gbps

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B

A $\rightarrow$ F: 10Gbps on the shortest path
Status quo I: Minimize delay

All links are 10Gbps

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B

E $\rightarrow$ B: 10Gbps to the shortest path
Status quo I: Minimize delay

All links are 10Gbps

15Gbps $A \rightarrow F$ and 10Gbps $E \rightarrow B$

Exclude bottlenecks
Status quo I: Minimize delay

All links are 10Gbps
15Gbps A ➔ F and 10Gbps E ➔ B
E-B is gone, E-F and A-B are unidirectional
Status quo I: Minimize delay

All links are 10Gbps
15Gbps A → F and 10Gbps E → B
A → F: Still has 5Gbps, but no room to put it

Should not have used the shortest path for A → F to begin with ...
Status quo I: Minimize delay

(BG92, Danna INFOCOM ‘12, Jain SIGCOMM ‘13)

All links are 10Gbps

15Gbps A \(\rightarrow\) F and 10Gbps E \(\rightarrow\) B

A \(\rightarrow\) F: Still has 5Gbps, but no room to put it

Should not have used the shortest path for A \(\rightarrow\) F to begin with ...
Status quo I: Minimize delay

All links are 10Gbps
15Gbps A → F and 10Gbps E → B
A → F: Still has 5Gbps, but no room to put it

- Max-min fair lowest-delay allocation
- May be what we want if A→F flows can tolerate reduction in throughput

(BG92, Danna INFOCOM ‘12, Jain SIGCOMM ‘13)
Status quo II: Maximize throughput

All links are 10Gbps

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B
Status quo II: Maximize throughput

All links are 10Gbps
15Gbps A → F and 10Gbps E → B
Will tweak link metrics and do ECMP
Status quo II: Maximize throughput

All links are 10Gbps
15Gbps $A \rightarrow F$ and 10Gbps $E \rightarrow B$
Will tweak link metrics and do ECMP
Status quo II: Maximize throughput

All links are 10Gbps, cost is displayed
15Gbps $A \rightarrow F$ and 10Gbps $E \rightarrow B$
$A \rightarrow F$: Spread among $A-D-E-F$ and $A-B-C-F$ (cost 3)
Status quo II: Maximize throughput

All links are 10Gbps, cost is displayed
15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B
E $\rightarrow$ B: Send all on lowest-cost E-B (cost 2)
Status quo II: Maximize throughput

(Fortz INFOCOM ‘00, Ericsson ’12)

All links are 10Gbps, cost is displayed:

15Gbps $A \rightarrow F$ and 10Gbps $E \rightarrow B$

$E \rightarrow B$: Send all on lowest-cost E-B (cost 2)
Status quo II: Maximize throughput

(Fortz INFOCOM ‘00, Ericsson ’12)

All links are 10Gbps, cost is displayed.

- Common in networks today, but error-prone
- May be what we want if A→F flows can tolerate increased delay

10 ms (cost 2)

10 ms (1)

1 ms (2)
No simple answer

- Combine routing and TE
- Be topology-agnostic
- Take into account both throughput and delay
No simple answer

This is an optimization problem: need to define a utility function and optimize that

• Combine routing and TE
• Be topology-agnostic
• Take into account both throughput and delay
What is a flow’s utility?

- Two components: throughput and delay
- Multiplied together to produce a number [0-1]
- Captures the utility of a single flow
What is a flow’s utility?

**Delay utility**

- **Throughput utility**

Operator-provided
What is a flow’s utility?

**Delay utility**

- Operator-provided
- Knee is dynamically sensed and updated (see paper)

**Throughput utility**
What is a flow’s utility?

![Delay utility graph](image1)

![Throughput utility graph](image2)

Just an example – FUBAR works with arbitrary monotonic functions

(see paper)
Estimating global utility

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B, 10Gbps links
Estimating global utility

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B, 10Gbps links

Put everything on the shortest path
Estimating global utility

Know both throughput and delay (assuming small queues) for all aggregates, can get global utility.

Put everything on the shortest path.
Estimating global utility

15Gbps A → F and 10Gbps E → B, 10Gbps links A → F does not fit. Need to try an alternative path.
Estimating global utility

15Gbps A → F and 10Gbps E → B, 10Gbps links
How about A-D-E-B-C-F?
Estimating global utility

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B, 10Gbps links

What is the new utility?
Estimating global utility

15Gbps A $\rightarrow$ F and 10Gbps E $\rightarrow$ B, 10Gbps links
What is the new utility?
... depends on the throughputs
Estimating global utility

15Gbps A → F and 10Gbps E → B, 10Gbps links

What is the new utility?

... depends on the throughputs

... which depend on how the E-B link is being split
Estimating global utility

15Gbps A → F and 10Gbps E → B, 10Gbps links

What is the new utility?
... depends on the throughputs
... which depend on how the E-B link is being split

When different flows’ paths share the same bottleneck link, congestion control determines throughput allocation.
Estimating global utility

15Gbps A \(\rightarrow\) F and 10Gbps E \(\rightarrow\) B, 10Gbps links

What is the new utility?
... depends on the throughputs
... which depend on how the E-B link is being split

When different flows’ paths share the same bottleneck link, congestion control determines throughput allocation.
TCP-like \(1/\text{RTT}\) model (see paper)
Allocate to the lowest-delay path
Allocate to the lowest-delay path

Run model, get $U_{\text{init}}$
Allocate to the lowest-delay path

Run model, get $U_{\text{init}}$

Pick congested link (L)
Allocate to the lowest-delay path

For each aggregate (A) over L move fraction of flows to a new path

Run model, get $U_{init}$

Pick congested link (L)
Allocate to the lowest-delay path

For each aggregate (A) over L move fraction of flows to a new path

Run model, get $U_{init}$

Pick congested link (L)
Allocate to the lowest-delay path

For each aggregate (A) over L move fraction of flows to a new path

Run model, get $U_{init}$

Pick congested link (L)

Ideally try all possible paths, but that is not feasible ... will come up with a greedy offline heuristic
Choosing new paths
Choosing new paths

1. Throughput-first (avoid all congestion)
Choosing new paths

1. Throughput-first (avoid all congestion)
2. Delay-first (avoid single congested link)
Choosing new paths

1. Throughput-first (avoid all congestion)
2. Delay-first (avoid single congested link)
Choosing new paths

1. Throughput-first (avoid all congestion)
2. Delay-first (avoid single congested link)
3. Avoid self-congestion
Choosing new paths

1. Throughput-first (avoid all congestion)
2. Delay-first (avoid single congested link)
3. Avoid all congested links in the aggregate
Choosing new paths

1. Throughput-first (avoid all congestion)
2. Delay-first (avoid single congested link)
3. Avoid all congested links in the aggregate

FUBAR will try all three paths, and iterate until no utility improvement observed
Allocate to the lowest-delay path

Run model, get $U_{init}$

Pick congested link (L)
Allocate to the lowest-delay path

For each aggregate (A) over L move some flows to a path that avoids all congestion

Pick congested link (L)

Run model, get $U_{\text{init}}$

path that avoids L

path that avoids all congestion in A
Allocate to the lowest-delay path

For each aggregate (A) over L move some flows to a:
- path that avoids all congestion
- path that avoids L
- path that avoids all congestion in A

Run model, get $U_{\text{init}}$

Pick congested link (L)

Pick move that improves $U_{\text{init}}$ the most
FUBAR

Allocate to the lowest-delay path

For each aggregate (A) over L move some flows to a:

- path that avoids all congestion
- path that avoids L
- path that avoids all congestion in A

Run model, get $U_{init}$

Pick congested link (L)

Pick move that improves $U_{init}$ the most
Evaluation

FUBAR on Hurricane Electric’s backbone topology
All 961 aggregates (traffic from all to all endpoints)
Real-time or bulk-transfer utility functions
Evaluation

CDF of 100 runs of FUBAR
It always drives utility close to optimal
Runs in approximately 40 sec
Future work

• Incorporate queuing delay into the model
• Additional constraints to the optimization problem (e.g., granularity of splits)
• Pre-caching of results for fast failover in case of link failures
• Introduce path-based constraints as network policies
Conclusions

• FUBAR approaches routing in an application-centric fashion
• Looks at both throughput and delay, takes congestion control into account
• Improves overall utility of network for all participants
• Requires no modification of endpoints, network hardware or congestion control
• Runs quick enough for an offline system