Bundler: A New Middlebox for
Site-to-Site Internet Traffic Control

Frank Cangialosi, Akshay Narayan, Prateesh Goyal,
Radhika Mittal, Mohammad Alizadeh, Hari Balakrishnan
“site”: a single physical location with many endpoints sharing an access link to the internet
“site”: a **single physical location** with many endpoints sharing an **access link** to the internet

(Heterogenous traffic sources and network requirements)
Site-to-Site Internet Traffic Control
Site-to-Site Internet Traffic Control

Company >

AWS US-West

Public Internet

Bottleneck queue

Latency-sensitive apps stuck waiting
Site-to-Site Internet Traffic Control
Site-to-Site Internet Traffic Control
Site-to-Site Internet Traffic Control

Company

>_

scheduler?

Public Internet

queue

AWS US-West
Site-to-Site Internet Traffic Control

Company >_

Public Internet

AWS US-West

scheduler?

bottleneck

queue
Site-to-Site Internet Traffic Control

Company >

scheduler?

Public Internet

bottleneck

queue

queue

AWS US-West
Site-to-Site Internet Traffic Control

Company

Public Internet

AWS US-West

scheduler?

bottleneck

queue

queue
Site-to-Site Internet Traffic Control
Site-to-Site Internet Traffic Control

Company

AWS US-West

Public Internet

bottleneck

queue

queue
Problem: want to enforce scheduling policy for your traffic, but often don’t control the bottleneck where packets queue.
Key Idea: “shifting” the queues
Key Idea: “shifting” the queues
Key Idea: “shifting” the queues

Rate limiting can “shift” the queue to our site!

Question: how do we pick the right rate?
Key Idea: “shifting” the queues

Company > _ > 10

Public Internet bottleneck

10 Mbps

AWS US-West

queue

<10

queue
Key Idea: “shifting” the queues
Key Idea: “shifting” the queues

Company

Public Internet

AWS US-West

queue

=10

bottleneck

10 Mbps

queue

5 Mbps
cross traffic
Key Idea: “shifting” the queues
Key Idea: “shifting” the queues

Congestion Control algorithms aim to calculate exactly the rate we need!
Bundler's Architecture
Bundler’s Architecture
Bundler’s Architecture

**Bundle:** set of flows with endpoints in same pair of sites

Company

AWS US-West

Public Internet
Bundler's Architecture

**Bundle:** set of flows with endpoints in same pair of sites
Bundler's Architecture

Bundle: set of flows with endpoints in same pair of sites
Bundler's Architecture

**Bundle:** set of flows with endpoints in same pair of sites

Separate queue and rate limit per bundle
Bundler's Architecture

**Bundle:** set of flows with endpoints in same pair of sites

Company

AWS US-West

Public Internet

Bundler

sendbox

rcvbox
Bundler's Architecture
Bundler's Architecture

sendbox

Scheduler → Measurements → Congestion Control → Rate Limiter

Recvbox Feedback
Transparent Measurement Scheme

• Leave connections **intact**
  — Don’t modify packets
  — Don’t disrupt end-to-end connections
• Out-of-band feedback **per RTT**
• Sample the same packets at both boxes **without communication**
Transparent Measurement Scheme

- Leave connections **intact**
  - Don’t modify packets
  - Don’t disrupt end-to-end connections
- Out-of-band feedback **per RTT**
- Sample the same packets at both boxes **without communication**

Compared to alternatives (e.g., TCP proxy)...

- Low overhead and complexity
- Simple datapath
Handling Unfavorable Conditions

1. Flows in a bundle don’t share the same bottleneck

2. Bundle competing with long-lasting buffer-filling cross traffic

But… in our experience, unfavorable conditions are rare.
Bundler (using FQ) improves FCTs

Traffic: 1 million TCP cubic flows, sizes sampled from CAIDA internet backbone distribution

Site A: senders (200) to (forward) Site B: receivers

emulated bottleneck: 96Mbps, 50ms

Small
(<10KB, 97.6%)

Medium
(10KB-1MB, 2.3%)

Large
(1MB-1GB, 0.1%)

Flow Completion Time
Flow Size

Slowdown

4
3
2
1

4
16
12
8
4

16
12
8
4
**Bundler (using FQ) improves FCTs**

**Traffic:** 1 million TCP cubic flows, sizes sampled from CAIDA internet backbone distribution.

**Graph:**
- **Flow Completion Time (FCT):** Avg. FCTs for flows of different sizes.
- **Flow Size:** Small (<10KB, 97.6%), Medium (10KB-1MB, 2.3%), Large (>1MB, 0.1%).
- **Emulated bottleneck:** Increased latency affects FCTs.
- **Box plots:** Show distribution of FCTs under Status Quo (forward) and with FQ (forward).
- **Site A** to **Site B:** Flows pass through an emulated bottleneck, affecting FCTs.

**Flow Completion Time (FCT) distribution**
- **Small** (97.6%): Avg. FCT is lower with FQ than Status Quo.
- **Medium** (2.3%): Avg. FCT is significantly increased with FQ.
- **Large** (0.1%): Avg. FCT is increased but less pronounced with FQ.
Bundler (using FQ) improves FCTs

**Traffic:** 1 million TCP cubic flows, sizes sampled from CAIDA internet backbone distribution

---

**Small**
- Size: <10KB, 97.6%
- Flow Completion Time
- **Status Quo**

**Medium**
- Size: 10KB-1MB, 2.3%
- Flow Completion Time
- **Status Quo**, **In-Network (Optimal)**

**Large**
- Size: >1MB, 0.1%
- Flow Completion Time
- **Status Quo**, **In-Network (Optimal)**
Bundler (using FQ) improves FCTs

**Traffic:** 1 million TCP cubic flows, sizes sampled from CAIDA internet backbone distribution

<table>
<thead>
<tr>
<th>Flow Size</th>
<th>Slowdown Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small (&lt;10KB, 97.6%)</td>
<td>Status Quo: 4, In-Network (Optimal): 3</td>
</tr>
<tr>
<td>Medium (10KB-1MB, 2.3%)</td>
<td>Status Quo: 16, In-Network (Optimal): 12</td>
</tr>
<tr>
<td>Large (&gt;1MB, 0.1%)</td>
<td>Status Quo: 16, In-Network (Optimal): 12</td>
</tr>
</tbody>
</table>

**Site A**
- senders (200)

**Bundler (FQ)**

**Site B**
- bundler
- receivers

**Emulated bottleneck:** 96Mbps, 50ms

Flow Completion Time
Flow Size
Bundler (using FQ) improves FCTs

Traffic: 1 million TCP cubic flows, sizes sampled from CAIDA internet backbone distribution

Site A senders (200) bundler (FQ) Site B bundler receivers

emulated bottleneck: 96Mbps, 50ms

Small (<10KB, 97.6%)

Medium (10KB-1MB, 2.3%)

Large (>1MB, 0.1%)

Flow Completion Time

Flow Size
Summary

**Bundler** is a new middlebox that enables scheduling regardless of where congestion occurs in the network

Source code and evaluation scripts available at:

[github.com/bundler-project](https://github.com/bundler-project)
**Bundler** dynamically adjusts to cross-traffic

![Graph showing Bundler's throughput and delay with and without cross-traffic](image-url)
**Bundler** dynamically adjusts to cross-traffic

No Cross Traffic

Throughput (Mbps) vs Time (seconds)

- Bundler Throughput
- Cross Traffic Throughput
- In-Network Queueing Delay

Queue Delay

Buffer Filling Cross Traffic

Non-Buffer Filling Cross Traffic

Slowdown

Bundler Status Quo

Bundler dynamically adjusts to cross-traffic.
**Bundler** dynamically adjusts to cross-traffic

[Graph showing Bundler throughputs and delays under different traffic conditions]
**Bundler** dynamically adjusts to cross-traffic
**Bundler** dynamically adjusts to cross-traffic

![Graph showing throughputs and delays under different traffic conditions](image)

- **No Cross Traffic**
- **Buffer-Filling Cross Traffic**
- **Non-Buffer-Filling Cross Traffic**

**Throughput (Mbps)**

**Delay (ms)**

**Slowdown**

Bundler dynamically adjusts to cross-traffic.
**Bundler** can shift queues on real internet paths
**Bundler** can shift queues on real internet paths

With Bundler:
Throughput stays the same as status quo,
But latency is reduced back to base RTT