Participatory Networking

Rodrigo Fonseca
Join work with Andrew Ferguson, Arjun Guha, Jordan Place
Networking in the Cloud
Networking in the Cloud

Can the cloud provider get **help** in configuring the network?
A few motivating examples
Large-scale processing
Defending from attacks
Untrusted VMs
Production Platform

Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Production Platform

Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Production Platform

Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Production Platform

Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)
Proposal
Participatory Networking
Participatory Networking
Participatory Networking
Participatory Networking

1. Requests
Participatory Networking

1. Requests
2. Hints
Participatory Networking

1. Requests
2. Hints
3. Queries
Participatory Networking
Participatory Networking

Safe?
Participatory Networking

Safe?  Secure?
Participatory Networking

Safe?  Secure?  Fair?
Participatory Networking

Safe?  Secure?  Fair?  Practical?
Participatory Networking

Participatory Networking
Participatory Networking

• End-user API for SDNs
Participatory Networking

- End-user API for SDNs
- Exposes existing mechanisms
Participatory Networking

• End-user API for SDNs
• Exposes existing mechanisms
• No effect on unmodified applications
Outline of PANE
1. Privilege Delegation Semantics
1. Privilege Delegation Semantics
2. Protocol Sketch
1. Privilege Delegation Semantics
2. Protocol Sketch
3. Dynamic Flow Processing
Outline of PANE

1. Privilege Delegation Semantics
2. Protocol Sketch
3. Dynamic Flow Processing
4. Current Status
Privilege Delegation Semantics
Shares
Shares
Flowgroup

Shares
<table>
<thead>
<tr>
<th>Flowgroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>src=128.12/16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shares</th>
<th>Shares</th>
<th>Shares</th>
<th>Shares</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Flowgroup</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>src=128.12/16 ∧ dst.port ≤1024</td>
</tr>
<tr>
<td>Flowgroup</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>src=128.12/16 ∧ dst.port ≤ 1024</td>
</tr>
<tr>
<td>Privileges</td>
</tr>
</tbody>
</table>

Shares
<table>
<thead>
<tr>
<th>Flowgroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>src=128.12/16 ∧ dst.port ≤1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>deny, allow</td>
</tr>
<tr>
<td>Flowgroup</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>src=128.12/16 ∧ dst.port ≤1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s</td>
</tr>
<tr>
<td>Flowgroup</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>src=128.12/16 \ dst.port \leq 1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>deny, allow</td>
</tr>
<tr>
<td>bandwidth: 5Mb/s</td>
</tr>
<tr>
<td>limit: 10Mb/s</td>
</tr>
<tr>
<td>hint</td>
</tr>
<tr>
<td>query</td>
</tr>
</tbody>
</table>

Shares
<table>
<thead>
<tr>
<th><strong>Flowgroup</strong></th>
<th>src=128.12/16 ∧ dst.port ≤1024</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speakers</strong></td>
<td><strong>Privileges</strong></td>
</tr>
<tr>
<td></td>
<td>deny, allow bandwidth: 5Mb/s</td>
</tr>
<tr>
<td></td>
<td>limit: 10Mb/s</td>
</tr>
<tr>
<td></td>
<td>hint query</td>
</tr>
<tr>
<td>Flowgroup</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td></td>
</tr>
<tr>
<td>src=128.12/16 $\land$ dst.port $\leq$ 1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>
Delegation
Delegation
Delegation
Delegation
Delegation
Delegation
### Delegation

<table>
<thead>
<tr>
<th>root</th>
<th>bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100Mbps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>root</th>
<th>bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>adf</td>
<td>50Mbps</td>
</tr>
</tbody>
</table>
Delegation

- **x**
  - root
  - bandwidth: 100Mbps

- **y**
  - root
  - bandwidth: 50Mbps

- **w**
  -

- **z**
  -
Delegation

Root share

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>bandwidth 50Mbps</td>
<td></td>
</tr>
<tr>
<td>root</td>
<td></td>
<td>bandwidth 100Mbps</td>
</tr>
</tbody>
</table>
### Flowgroup

| src=128.12/16 ∧ dst.port ≤1024 |

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>
### Flowgroup

\[ \text{src=128.12/16} \land \text{dst.port} \leq 1024 \]

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Flowgroup</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td></td>
</tr>
<tr>
<td>src=128.12/16 ∧ dst.port ≤1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>
Flowgroup
src=128.12/16 ∧ dst.port ≤ 1024

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>

Yes

Dynamic Context

PANE
**Flowgroup**

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice, Bob</td>
<td>deny, allow</td>
</tr>
<tr>
<td></td>
<td>bandwidth: 5Mb/s</td>
</tr>
<tr>
<td></td>
<td>limit: 10Mb/s</td>
</tr>
<tr>
<td></td>
<td>hint query</td>
</tr>
</tbody>
</table>

src=128.12/16 ∧ dst.port ≤1024

This traffic will be short and bursty.
**Flowgroup**

\[
\text{src}=128.12/16 \land \text{dst.port} \leq 1024
\]

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s hint query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>
**Flowgroup**

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s <em>hint</em> query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>

src=128.12/16 ∧ dst.port ≤1024

---

**How much web traffic in the last hour?**

**Dynamic Context**

**PANE**
# Flowgroup

\[ \text{src}=128.12/16 \land \text{dst.port} \leq 1024 \]

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>deny, allow bandwidth: 5Mb/s limit: 10Mb/s <em>hint</em> query</td>
</tr>
<tr>
<td>Bob</td>
<td></td>
</tr>
</tbody>
</table>

67,560 bytes
Current: 0 Mbps
Current: 0 Mbps

Root

share

Current: 0 Mbps

Current: 0 Mbps

x

y
Current: 0 Mbps

\[
\begin{array}{|c|c|}
\hline
\text{bandwidth} & 100\text{Mbps} \\
\hline
\end{array}
\]

\text{Root share}

\text{Current: 0 Mbps}

\text{Current: 0 Mbps}

\text{Reserve 80 Mbps?}

\text{Pane}
Current: **80 Mbps**

**Root share**

Current: **80 Mbps**

Current: **0 Mbps**

**Yes**

**PANE**
Current: **80 Mbps**

Current: **80 Mbps**

Current: 0 Mbps

Reserve 50 Mbps?

**Root share**

**x**

**y**

**PANE**
Current: **80 Mbps**

**Root share**

Current: **80 Mbps**

Current: **0 Mbps**

**OpenFlow**

**PANE**
Protocol Sketch
NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.
NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.

Root

Alice

OK

PANE
NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.

Grant A to Alice.
NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.

Grant A to Alice.
Root

NewShare A for
(user=Alice) [reserve <= 10Mb]
on rootShare.

Grant A to Alice.

OK

Alice

reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
Root

Alice

NewShare A for
(user=Alice) [reserve <= 10Mb]
on rootShare.

Grant A to Alice.

reserve(user=Alice,
dstPort=80) = 5Mb on A
from now to +10min.

OK

OK

PANE
Root

Grant A to Alice.

NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.

Alice

reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
NewShare A for (user=Alice) [reserve <= 10Mb] on rootShare.

Grant A to Alice.

reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
Alice

reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.
reserve(user=Alice, dstPort=80) = 5Mb on A from +20min to +30min.
reserve(user=Alice, dstPort=80) = 5Mb on A from +20min to +30min.
reserve(user=Alice, dstPort=80) = 5Mb on A from +20min to +30min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.

reserve(user=Alice, dstPort=80) = 5Mb on A from +20min to +30min.
reserve(user=Alice, dstPort=80) = 5Mb on A from now to +10min.

reserve(user=Alice, dstPort=80) = 5Mb on A from +20min to +30min.
Alice

PANE
Root

Alice

10.0.0.2

PANE
NewShare aAC for (dstHost=10.0.0.2) [deny = True] on rootShare.
NewShare aAC for (dstHost=10.0.0.2) [deny = True] on rootShare.

Root

Alice

10.0.0.2

OK

PANE
NewShare aAC for (dstHost=10.0.0.2) [deny = True] on rootShare.

Grant aAC to Alice.
NewShare aAC for (dstHost=10.0.0.2) [deny = True] on rootShare.

Grant aAC to Alice.
deny(dstHost=10.0.0.2, srcHost=10.0.0.3) on aAC from now to +5min.
deny(dstHost=10.0.0.2, srcHost=10.0.0.3) on aAC from now to +5min.
deny(dstHost=10.0.0.2, srcHost=10.0.0.3) on aAC from now to +5min.

OK

10.0.0.2

10.0.0.3

Eve

Alice

PANE
Dynamic Flow Processing
Hierarchy of Policies
Hierarchy of Policies

(dstPort = 22, Deny)

(dstIP=10.0.0.2, GMB=30)

(dstPort=80, GMB=10)

(srcIP=10.0.0.1, Allow)

Hierarchy of Policies
Hierarchy of Policies

Packet:
src 10.0.0.1
dst 10.0.0.2:80
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(dstPort = 22, Deny)
(dstIP=10.0.0.2, GMB=30)
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(dstPort = 22, Deny)
(dstIP=10.0.0.2, GMB=30)
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)

GMB=10
Hierarchical Flow Table

- (dstPort = 22, Deny)
- (dstIP=10.0.0.2, GMB=30)
- (dstPort=80, GMB=10)
- (srcIP=10.0.0.1, Allow)

Packet:
src 10.0.0.1
dst 10.0.0.2:80
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(dstIP=10.0.0.2, GMB=30)
(dstPort=22, Deny)
(dstPort=80, GMB=10)

(srcIP=10.0.0.1, Allow)

GMB=10
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(srcIP=10.0.0.1, Allow)

(dstPort=80, GMB=10)

GMB=10

(dstIP=10.0.0.2, GMB=30)

(dstPort = 22, Deny)
Packet:
src 10.0.0.1
dst 10.0.0.2:80

Hierarchical Flow Table

(dstPort = 22, Deny)

(dstIP=10.0.0.2, GMB=30)

(dstPort=80, GMB=10)

(srcIP=10.0.0.1, Allow)

Packet:
src 10.0.0.1
dst 10.0.0.2:80

GMB=10

Allow

0

+P

(dstIP=10.0.0.2, GMB=30)
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(dstPort = 22, Deny)
(dstPort = 80, GMB = 10)
(srcIP = 10.0.0.1, Allow)
(dstIP = 10.0.0.2, GMB = 30)
Hierarchical Flow Table

Packet:
src 10.0.0.1
dst 10.0.0.2:80

(srcIP=10.0.0.1, Allow)
(dstPort=80, GMB=10)
GMB=10

GMB=30

(dstIP=10.0.0.2, GMB=30)

(dstPort=22, Deny)
GMB=30

GMB=10

(dstPort=80, GMB=10)

(srcIP=10.0.0.1, Allow)
Requirements

Associative, 0-identity

Commutative

+D  In node

+P  Parent-Sibling

+S  Sibling

HFT operators
<table>
<thead>
<tr>
<th>HFT operators</th>
<th>Commutative</th>
</tr>
</thead>
<tbody>
<tr>
<td>+P Parent-Sibling</td>
<td>+S Sibling</td>
</tr>
<tr>
<td>+D In node</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements**

- In node: Deny overrides Allow.
- D and S identical.
- GMB combines as max.

- Child overrides Parent for Access Control.
There’s a slight problem...
Policy Tree

Share Tree

PANE user requests

Switches
Switches don’t grow trees
Switches don’t grow trees
Current Status
PANE user requests

Policy Tree

Share Tree

Linearization

Network Flow Table

Forwarding & Queue Configuration

Network Information Base (NIB)

Valid Configuration

OpenFlow Controller

OpenFlow messages

Switches
We’ve built all these components
Currently have Access Control and GMB
Toy Evaluation
Protecting Zookeeper
5 PANE-enabled Zookeeper servers
1 client
Connected via single OpenVSwitch (3.3Gbps)
iperf generating load on all links

Protecting Zookeeper
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Denial-of-service
Where to go?
Implement more operators!

Where to go?
Implement more operators!

Guaranteed latency

Where to go?
Implement more operators!

Guaranteed latency
Rate limiting

Where to go?
Implement more operators!

Guaranteed latency
Rate limiting
Path properties

Where to go?
Implement more operators!

Guaranteed latency
Rate limiting
Path properties
Hints

Where to go?
Implement more operators!

- Guaranteed latency
- Rate limiting
- Path properties
- Hints
- Queries

Where to go?
Implement more operators!

Guaranteed latency
Rate limiting
Path properties
Hints
Queries

Your application?

Where to go?
Implement more operators!
Guaranteed latency
Rate limiting
Path properties
Hints
Queries
Your application?
Build a market

Where to go?
Implement more operators!
Guaranteed latency
Rate limiting
Path properties
Hints
Queries
Your application?
Build a market
Scale

Where to go?
Conclusion
Application knowledge can help cloud provider PANE is our first step in realizing this
no ftp-server
write-enable
table
controller T1 3/0
framing sf
linecode ami
--- The outside
interface
--- Ethernet0/0
ip address 172.22.1.112
255.255.255.0
--- Apply the
access list to
permit SMTP/ESMTP
cable connections
--- to the mail
server. This also
allows Cisco IOS
Firewall to inspect
permitted traffic.
--- The static
translation for the
mail server.
--- The inside
source static
10.10.10.2
172.22.1.110
ip not inside
source static
10.10.10.5
172.22.1.111
--- Apply the
access list to permilt
SMTP/ESMTP to the
mail server.
--- Cisco IOS
Firewall inspects
permitted traffic.
--- Define the
rule that
applies to the
outside interface.
--- Apply the
inspection rule
OUT-IN inbound
on this interface.
--- The rule defines
SMTP/ESMTP
inspection.
--- Apply inspection
OUT-IN outbound
on this interface.
--- The rule defines
SMTP/ESMTP
inspection.
--- The inside
interface.
--- Apply inspection
OUT-IN outbound
on this interface.
--- The rule defines
SMTP/ESMTP
inspection.
--- Apply the
inspection rule
OUT-IN inbound
on this interface.
--- The rule defines
SMTP/ESMTP
inspection.
--- Apply the
inspection rule
OUT-IN outbound
on this interface.
--- The rule defines
SMTP/ESMTP
inspection.
--- The rule defines
SMTP/ESMTP
inspection.
--- The rule defines
SMTP/ESMTP
inspection.
--- The rule defines
SMTP/ESMTP
inspection.
--- The rule defines
SMTP/ESMTP
inspection.
--- The rule defines
SMTP/ESMTP
inspection.
OCCUPY EVERYTHING

#OCCUPYWALLST

We already know that we own everything—the task is to exclude the intrusions of capital and power.
Participatory Networking
Participatory Networking

1. management API
Participatory Networking

1. management API
2. network controller
Participatory Networking

1. management API
2. network controller
Participatory Networking

1. management API
2. network controller

Safe Secure
Participatory Networking

1. management API
2. network controller

Safe
Secure
Fair
Questions?

Rodrigo Fonseca
rfonseca@cs.brown.edu

Andrew Ferguson
adf@cs.brown.edu

Arjun Guha
arjun@cs.brown.edu

Shriram Krishnamurthi
sk@cs.brown.edu
A problem in the home
Buffering

89%
Network Working Group

Request for Comments: 2205

Category: Standards Track

B. Braden, Ed.

L. Elz

K. Fall

S. Floyd

T. Henderson

K. Rekhter

B. Want

Univ. of Michigan

September 1997

Resource ReSerVation Protocol (RSVP) --

Version 1 Functional Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community. Please refer to the current edition of the "Internet
Official Protocol Standards" (RFC 1700) for the standardization state
and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo describes version 1 of RSVP, a resource reservation setup
protocol designed for an integrated services Internet. RSVP provides
receiver-initiated setup of resource reservations for multicast or unicast
data flows, with good scalability and robustness properties.
TCP Nice: A Mechanism for Background Transfers

Anur Venkataramani
Ravi Kokik Mike Dulin
Laboratory of Advanced Systems Research
Department of Computer Sciences
University of Texas at Austin, Austin, TX 78712
{anur, mikes, dulin}@cs.utexas.edu

Abstract

Many distributed applications can make use of large background transfers — transfers of data that begin are not waiting for a time to improve availability, reliability, latency or consistency. However, given the rapid fluctuations of available network bandwidth and changing resource costs due to technology trends, handling the aggressiveness of background transfers (1) simplifying applications, (2) being too aggressive and putting the background transfers in the foreground, and (3) not putting the benefits of background transfers. Our goal is to offer the operating system a management resource in order to provide a simple abstraction of near non-background transfers. Our system, TCP Nice, can provably bound the interference inflicted by background flows on foreground flows in a shared network setup. And yet, microbenchmark and case study applications suggest that in practice it is often difficult to distinguish between foreground flows, maps a large fraction of share network bandwidth and amplify application execution and deployment. For example, in our preliminary case study applications, aggressive prefetching improves demand performance by a factor of three when Nice reduces resources, but the same prefetching has limited performance by a factor of two under standard network conditions.

1 Introduction

Many distributed applications can make use of large background transfers — transfers of data that begin are not waiting for a time to improve availability, reliability, latency or consistency. For example, a large range of applications and services such as data backup [19], prefetching [20], enterprise data distribution [22], Internet content distribution [2], and persistent storage [16, 41] can trade increased network bandwidth consumption and possibly disk space for improved service latency [15, 32] and improved availability [11, 13], increased scalability [2], stronger consistency [9], or support for mobility [29, 41, 45]. Many of these services have exponentially growing bandwidth demands where incrementally more bandwidth consumption provides incrementally better service. For example, a web prefetching system can improve the hit rate by fetching objects from a locally unlimited collection of objects that have non-zero probability of access [8, 40] or by opening cached copy more frequently as data change [31, 50, 60]. Technology trends suggest that in the future, per-system bandwidth costs and disk storage costs are likely to be increased in the (80-100% per day, (71, 94, 94) and network interface costs increase sharply, and long interfaces and buffer space become expensive.

Current operating systems and networking do not provide good support for aggressive background transfers. In particular, because background transfers compete with foreground requests, they can hurt overall performance and availability by increasing network competition. Applications must therefore carefully balance the benefit of background transfers against the risk of over-subscription of available network resources for the worst-case application performance. However, applications attempt to achieve this balance by using “magic numbers” such as the preference threshold in prefetching algorithms [22, 26] that have little obvious relationship to system goals (e.g., availability or latency) or constraints (e.g., network service levels).

One goal is to offer the operating system to manage network resources in order to provide a simple abstraction of near non-background transfers. A self-tuning background transfer layer will enable new classes of applications by (1) simplifying applications, (2) reducing the risk of being too aggressive, and (3) making...