One More Bit Is Enough

Yong Xia, RPI
Lakshmi Subramanian, UCB
Ion Stoica, UCB
Shiv Kalyanaraman, RPI

SIGCOMM’05, Philadelphia, PA
08 / 23 / 2005
Motivation #1: TCP doesn’t work well in high b/w or delay

![Graph showing bottleneck utilization vs. bandwidth (Mbps) and round-trip delay (ms)]

**bottleneck utilization**

- **TCP**
  - 100% utilization at high bandwidth (Mbps)
  - 70% utilization at moderate bandwidth (Mbps)

- **VCP**
  - 2% utilization

**round-trip delay (ms)**

- **TCP**
  - 100% delay at high round-trip delay (ms)
  - 50% delay at moderate round-trip delay (ms)

VCP
Motivation #1: Why TCP does not scale?

- TCP uses binary congestion signals, such as loss or one-bit Explicit Congestion Notification (ECN).

- AI with a fixed step-size can be very slow for large bandwidth.
Motivation #2: XCP scales

- XCP decouples efficiency control and fairness control

- But, XCP needs multiple bits (128 bits in its current IETF draft) to carry the congestion-related information from/to network
**Goal**

Design a TCP-like scheme that:

- requires a small amount of congestion information (e.g., 2 bits)
- scales across a wide range of network scenarios
Key Observation

Fairness is not critical in low-utilization region

- Use Multiplicative Increase (MI) for fast convergence onto efficiency in this region
- Handle fairness in high-utilization region
Variable-structure congestion Control Protocol (VCP)

- Routers signal the level of congestion
- End-hosts adapt the control algorithm accordingly
VCP vs. ECN

- ECN doesn’t differentiate between low-load and high-load regions

TCP

- Multiplicative Decrease (MD)
- Additive Increase (AI)

AQM/ECN

- load factor
- region
- code
- overload (1)
- underload (0)
- 1-bit ECN

sender

router

ACK

receiver
An illustration example

- MI tracks available bandwidth exponentially fast
- After high utilization is attained, AIMD provides fairness

![Graph showing MI and AIMD performance with 9 flows joining at 100s and leaving at 200s, link utilization, and flow cwnd (pkt) over time (sec).]
VCP vs. ECN

VCP

TCP+RED/ECN

utilization
cwnd
time (sec)
time (sec)
VCP key ideas and properties

- Use network link load factor as the congestion signal
- Decouple efficiency and fairness controls in different load regions
- Achieve high efficiency, low loss, and small queue
- Fairness model is similar to TCP:
  - Long flows get lower bandwidth than in XCP (proportional vs. max-min fairness)
  - Fairness convergence much slower than XCP (solvable with even more, e.g., 8 bits)
**Major design issues**

- **At the router**
  - How to measure and encode the load factor?

- **At the end-host**
  - When to switch from MI to AI?
  - What MI/Al/MD parameters to use?
  - How to handle heterogeneous RTTs?
Design issue #1: measuring and encoding load factor

- Calculate the link load factor $\rho$

$$load\_factor = \frac{demand}{capacity}$$

$$= \frac{arrival\_traffic + queue\_size}{link\_bandwidth \times t_\rho}$$

- The load factor is quantized and encoded into the two ECN bits

$t_\rho = 200ms \gg most\ rtt$
Design issue #2: setting MI/AI/MD parameters \((\xi, \alpha, \beta)\)

\[
\begin{align*}
\text{MI} : \quad cwnd(t + rtt) &= cwnd(t) \times (1 + \xi) \\
\text{AI} : \quad cwnd(t + rtt) &= cwnd(t) + \alpha \\
\text{MD} : \quad cwnd(t + \delta t) &= cwnd(t) \times \beta
\end{align*}
\]
Design issue #2: setting MI/Al/MD parameters ($\xi$, $\alpha$, $\beta$)

- Q: load factor transition point $\rho^*$ for MI $\rightarrow$ AI?

\[ \xi = k \left(1 - \rho^*\right) / \rho^* \]

where $k = 0.25$ (for stability)

- TCP: $\xi = 1.0$
- VCP: $\xi = 0.06$
- STCP: $\xi = 0.01$
Design issue #3: Handling RTT heterogeneity for MI/AI

- Scale $\xi$ to prevent MI from overshooting capacity when RTT is small

\[ 1 + \xi = (1 + \xi_s) \frac{t_\rho}{rtt} \]
VCP scales across b/w, rtt, num flows

- Evaluation using extensive ns2 simulations

- 150Mbps, 80ms, 50 forward flows and 50 reverse flows
VCP achieves high efficiency

VCP

bottleneck utilization

bandwidth (Mbps)

round-trip delay (ms)
VCP minimizes packet loss rate

- **Bottleneck utilization**
  - TCP
  - VCP
  - XCP

- **Packet loss rate**
  - TCP
  - VCP
  - XCP

- **Bandwidth (Mbps)**

- **Round-trip delay (ms)**
VCP comparisons

- Compared to TCP+AQM/ECN
  - Same architecture (end-hosts control, routers signal)
  - Router congestion detection: queue-based → load-based
  - Router congestion signaling: 1-bit → 2-bit ECN
  - End-host adapts (MI/AI/MD) according to the ECN feedback
  - End-host scales its MI/AI parameters with its RTT

- Compared to XCP
  - Decouple efficiency/fairness control across load regions
  - Functionality primarily placed at end-hosts, not in routers
Theoretical results

- **Assumptions:**
  - One bottleneck of infinite buffer space is shared by synchronous flows that have identical RTTs;
  - The exact value of load factor is echoed back.

- **Theorem for the VCP fluid model:**
  - It is **globally stable** with a unique and fair equilibrium, if $k \leq 0.5$;
  - The equilibrium is **max-min fair** for general topologies;
  - The equilibrium is **optimal** by achieving all the design goals.

- **VCP protocol differs from the model in fairness.**
Conclusions

- With a few minor changes over TCP + AQM/ECN, VCP is able to approximate the performance of XCP
  - High efficiency
  - Low persistent bottleneck queue
  - Negligible congestion-caused packet loss
  - Reasonable (i.e., TCP-like) fairness
Future work

- How do we get there, incrementally?
  - End-to-end VCP
  - TCP-friendliness
  - Incentive

- Extensions
  - Applications: short-lived data traffic, real-time traffic
  - Environment: wireless channel

- Security
  - Robust signaling, e.g., ECN nonce
The end

Thanks!
Design issue #3: Handling RTT heterogeneity for MI/AI

- TCP throughput is biased against flows with large RTT

\[
rate \approx \frac{packet\_size}{rtt \cdot \sqrt{loss\_rate}}
\]

- VCP scales \( \alpha \) for fair rate sharing (regardless of RTT)

For AI: \( \alpha_{rate} \leftarrow \alpha \cdot \frac{rtt}{t_\rho} \cdot \frac{rtt}{t_\rho} \)

\[
rate = \frac{cwnd}{rtt}
\]
**VCP keeps small bottleneck queue**

queue length in % buffer size

<table>
<thead>
<tr>
<th>Bandwidth (Mbps)</th>
<th>TCP</th>
<th>XCP</th>
<th>VCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

per-flow b/w-delay-product ≤ 1 packet

round-trip delay (ms)

<table>
<thead>
<tr>
<th>Round-trip Delay (ms)</th>
<th>VCP</th>
<th>XCP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Vary the number of flows

- **Bottleneck utilization**
  - VCP
  - XCP
  - TCP

- **Queue length in % buffer size**
  - VCP
  - XCP
  - TCP

**Number of flows**
Influence of RTT on fairness

- To some extent, VCP distributes reasonably fairly.
Influence of RTT on fairness (cont’d)
VCP converges onto fairness
VCP converges onto fairness faster with 8 bits
Responsiveness

![Diagram showing responsiveness with different scenarios: 50 flows, 50 flows + 150 flows, 50 flows - 150 flows.](image)

RTT: 60ms — 158ms

50 flows  + 150 flows  - 150 flows