Fluid Simulation of Large Scale Networks: Issue and Tradeoffs

Jim Kurose
Department of Computer Science
Univ. Massachusetts/Amherst
www.cs.umass.edu/~kurose

Joint work with D. Figueiredo, W. Gong, Y. Guo, B. Liu, D. Towsley
Overview:

- introduction
- understanding the computational workload of network simulation
  - packet versus fluid
  - effects of load, scheduling discipline, routing on network elements
- scenarios: event rate analysis
- decreasing a simulation’s event rate
- summary, future work
Motivation:

- workshop theme: the internet as a large-scale, complex system
- question: how to simulate its behavior?
  - answer: ns!

Question: how to simulate large scale networks in a computationally tractable way?
Speeding up a simulation

- Efficient event-list manipulation
- Higher abstraction level
- Simpler model
- Importance sampling
- Compute horsepower
- Faster CPU
- Parallel simulation
- Simulation technology
Discrete-event network simulation

Packet-based simulation:
- network traffic flow: model packets in flow
- # sources, data rates increase, so too does simulation workload

Fluid-based simulation:
- network traffic flow: continuous fluid
  - rate changes at discrete points in time
  - rate constant between changes
- can modulate rate at different time scales
  - single modeling paradigm for many time scales
  - abstract out fine-grained details: simulation efficiency
Packets and fluids

packet arrivals

fluid ~ packet

fluid ~ burst

fluid ~ session

Intuitively: fluid simulation can be more “efficient”
Question: efficiency of packet- versus fluid-based simulation?

- as a function of....
  - numbers of flows
  - topology, link speeds
  - level of modeling granularity
  - router scheduling discipline

- comparison with packet-based simulation

- important issue: accuracy (not today's topic)
Traffic source model: open loop

Markov Process

- packet rate: $\gamma$
- avg. packet size: $x$
- fluid rate: $\gamma x$

Key idea:
- Represent burst of packets with fluid chunk
- Less "work" to process one fluid chunk than many packets modeled by that chunk
Measuring simulation “work”

_event rate:_ number of event-list manipulations per unit time

- _packet:_ arrivals/departures at queues
- _simulation:_ rate changes at queues
- _we’ll see:_ analytic results for event rate correlate well with measured simulation run times
FIFO packet multiplexer event rate

simple: each arrival event generates one departure event

9 arrival events 9 departure events

arrivals to downstream router

rearranged in time

output multiplexer
**FIFO fluid multiplexer:**

- **Case:** $C > \sum \text{input fluid rates}$
  - no queueing
  - fluids “pass through” multiplexer with no change in event rates
FIFO fluid multiplexer: more interesting!

**Case:** $C < \Sigma \text{ input fluid rates}$
- output fluid rate affected by rate changes of other *input* flows!
- output event rate > input event rate: *ripple effect*
Ripple effect: bad news!

- ripples propagates to downstream routers
  - where they can be magnified
  - propagate further
- no ripple effect in packet simulation
WFQ fluid multiplexer: more interesting!

**Case:** \( C < \sum \text{input fluid rates} \)

- WFQ provides isolation among flows
- Queueing smooths out input rate variation within flow classes
Observations

- fluid event rates depend strongly on flow interactions, (service disciplines, link rates, propagation delays).
- Our efforts:
  - analytic characterization of fluid event rates in network setting: “calculus”
  - empirical investigation, comparison
  - techniques reducing fluid simulation event rates
Tandem queue example

Assume $\lambda=\mu=1$, $N = 10$, $\gamma = 3$

Event rates as $K$ (# queues) scales:

$$e^p = O(K) \quad e^f = O(K^2)$$

packet \quad fluid
Tandem: scaling # nodes

Model with 15 sources
Fluid simulation
Packet simulation $\gamma = 3$
Tandem: scaling \# sessions

\[ \lambda = \mu = 1, \ 20 = 10, \ \gamma = 3, \ \rho = 0.8 \]
Feedback networks

- Feedback: ripple effect can feedback upon itself!
- Propagation delays between nodes limits feedback cycle
Feedback networks
Event rate as measure of effort

Measure of computational effort: event rate
Event rate as measure of effort

Measure of computational effort: simulation run time
Speeding up a fluid simulation

**aggregation**: aggregate fluid flows when individual dynamics not of concern
**Aggregation: single queue example**

[Graph showing the relationship between the number of identical sources and queue departure event rate, with different lines representing aggregated and non-aggregated sources, and a packet simulation line with gamma equal to 20.]
Time-Stepped Fluid Simulation

So far:
- sources approximated by fluids
- exact simulation of fluid dynamics within network

Idea:
- smooth out fluid variations at queue outputs
  - over what time scale?
  - time-stepped fluid simulation
Time-Stepped Fluid Simulation
Simulation Experiment: Fluid Network

<table>
<thead>
<tr>
<th>Time-step(ms)</th>
<th>0.1</th>
<th>1</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgQ length(byte)</td>
<td>6787.0</td>
<td>6692.6</td>
<td>6454.1</td>
<td>6043.6</td>
</tr>
<tr>
<td>Thruput(byte/ms)</td>
<td>850.4</td>
<td>850.4</td>
<td>850.4</td>
<td>850.4</td>
</tr>
</tbody>
</table>
Fluid Network (2)
Big picture question:

- question: from packet- to session-level abstraction
- flow rate: determined by link capacity, sharing requirements
- networks of processor-sharing-like queues
Summary

- Packet versus fluid simulation: efficiency
  - propagation of rate change effects
- Current/future work:
  - study of WFQ
  - flow-level modeling
  - what’s the right abstraction level?
    - efficiency versus accuracy
Slides available at
http://gaia.cs.umass.edu/kurose/santa_fe_01.ppt

Papers available at
http://gaia.cs.umass.edu

- Y. Guo, W. Gong, D. Towsley, "Time-stepped Hybrid Simulation (TSHS) for Large Scale Networks, "Proc. IEEE Infocom'00 (Tel-Aviv, Israel, March, 2000).
Observations

- gaps:
  - between theory and theory
  - between theory and practice
  - in vocabulary

- interesting questions have hard-to-define (but important!) measures, e.g.: evolvability, complexity, maintainability, robustness
  - “the Internet is evolvable”
  - “Internet soft-state control is robust”

...we need to be precise and we need help with definitions
Observations

- many interesting questions
  - at application-level
  - in the control plane
  - in the legal plane
  - in how the infrastructure grows and is measured and modeled