Embedding end-user concerns into the network: from QoS to adaptive sensor networks

Jim Kurose
Department of Computer Science
University of Massachusetts
Amherst MA USA

ICNP 2008, Orlando, FL
Overview

- QOS retrospective
- what have we learned?
- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization
- conclusions
Overview

- QOS retrospective
- what have we learned?
- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization
- conclusions
Overview

- QOS retrospective
  - resource reservation
  - service differentiation
  - over-provisioning, application-level support
  - isolation: VPNs

- what have we learned?

- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization

- conclusions
Back in the day: resource reservation
(early-mid 1990’s)

1. characterize session input
2. queue scheduling (given)
3. characterize session output
IETF Intserv QoS: [rfc2211, rfc 2212]

Guaranteed service:
- worst case traffic (rspec): leaky-bucket-poled source
- simple provable) bound on delay [Parekh 1992, Cruz 1988]
  - min-plus algebra (Chang98, Cruz99, LeBoudec 2001)
  - bounds on delay distribution (Kurose92, Sidi93, Kesidis00, Chang01)

Related:
- ATM (CBR, VBR)
- effective bandwidth
- call admission

Q: does it scale?
Next “big thing”: differentiated service
(mid-late 1990’s)

edge routers:
- packet marking

“stateless” core routers:
- no per-session state
- forwarding: based on packet marking
Back to the Future: overprovisioning, application infrastructure  (late 1990’s – today)

- throw resources at problem: enough bandwidth to “ensure” no contention for resources
  - inefficient use of resources, but …
  - research question: how much over-provisioning is enough (e.g., Guerin 05, Roberts …)

- enhance service using sophisticated application-level infrastructure (caches, relays, peers)
  - Skype supports 10M VoIP callers
  - PPlive, BBC, IPTV
  - Push2Peer video distribution service
Meanwhile, in industry  (1990’s to today)

- **Q:** What do our customers want?
- **A:** Their own network!

---

**VLANS, VPNs**

Networks *perceived* as being private networks by customers using them, but built over shared infrastructure owned by service provider.
VPN reference architecture

VPN1
VPN2
service provider network

customer edge router
provider edge router

VPN1
VPN2
VPN: logical view

- logical isolation from other VPNs
  - multiplexing/performance issues within VPN

Virtualization:
- Ethane (in a LAN)
- Planetlab, GENI “slice”
- VINI (with lots of programmability)
<table>
<thead>
<tr>
<th>Unit of allocation</th>
<th>guarantee</th>
<th>deployment</th>
<th>complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per flow</td>
<td>Hard or soft</td>
<td>minimal high</td>
<td>aggregate priority more medium hope for the best massive low</td>
</tr>
<tr>
<td>Per network/application</td>
<td>hard or soft</td>
<td>some medium</td>
<td></td>
</tr>
</tbody>
</table>
QoS retrospective: summary

- **Research**: per-call resource reservation (Intserv, ATM)
- **Practice**: aggregates (difserv), VLANs, MPLS
- **Application-layer overlays (including p2p)**: MPLS, VPNs

Time:
- 1990
- 2000
- Today

Multiplexing granularity:
- Fine
- Coarse
Overview

- QOS retrospective
- what have we learned?
- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization
- conclusions
The right level of complexity

Q: What process determines the “right” level of complexity?

[adapted from Hluchyj 2001]
Middle-aged Internet: losing the hourglass?

middle age: a narrowing mind, a widening waist

Youthful, IP “hourglass”

Internet at middle-age: lovehandles? ?
Middle-aged Internet: *keeping* the hourglass!

middle age: a expanding mind, a slim waist

Applications

TCP UDP

IP

Eth token

PPP 802.11

radio, copper, fiber

IP “hourglass”
Overview

- QOS retrospective
- what have we learned?
- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization
- conclusions
QoS meets sensor networking

Scenario:
- *multiple users* access remote sensors
- *sense-and-response*: *actuated* sensors, *controlled* by user requirements
  - competing, conflicting user resource demands
    (sensor targeting, computation, bandwidth)

Challenge:
- how to allocate resources when and where user needs (utilities) are greatest?
  - multi-resource, end-end QoS
CASA:** collaborative adaptive sensing of the atmosphere

CASA: dense network of low power radars:
- sense lower 3 km of earth’s atmosphere
- *collaborating* radars:
  - improved sensing
  - improved detection, prediction
- finer spatial resolution
- responsive to multiple *end-user needs*

“Sample atmosphere when and where end-user needs are greatest”
What’s needed to solve this problem?

**CASA**
Collaborative
Adaptive
Sensing
of the Atmosphere

Remote sensing
Data-intensive systems
Networking
Real-time systems
Numerical prediction
Emergency management
Radar meteorology
Quantitative inversion
Climate studies
Social impact
Antenna design

**core partners**

**NSF**
Engineering Research Center
Sept. 2003

**expertise**
Sense-and-response control loop

**CASA:** Collaborative Adaptive Sensing of the Atmosphere

1 Mbps (moment)  
100 Mbps (raw)

30 sec. “heartbeat”

**Challenge:** how to allocate resources (sensing, computing, bandwidth) when and where user needs (utilities) are greatest?

- data storage
- feature detection
- resource allocation

users: NWS, emergency response

policy

utility function
Oklahoma 4-Node Test Bed

End users: national weather service, emergency response managers, researchers

(Centralized or decentralized)
Off-the-Grid Test Bed

- no reliance on infrastructure
- solar/battery-operated nodes
- multi-antenna multi-hop 802.11 directional antenna
2007 storm season in Oklahoma

FLUS74 KOUN 102343
AWUOUI

AREA WEATHER UPDATE
NATIONAL WEATHER SERVICE
NORMAN OK
742 PM CDT TUE APR 10 2007

..WARNING DECISION UPDATE

THIS WARNING DECISION UPDATE CONCERNS COMANCHE AND GRADY COUNTIES.

MESOCYCLONE NEAR STERLING CONTINUES TO STRENGTHEN PER TWO RADAR VIEWS. CASA NETWORK ALSO SHOWING PRONOUNCED HOOK. STORM WILL ENCOUNTER WARM FRONT...

BURKE
Embedding user preferences: utility functions

Find configuration that optimizes utility at time step $k$:

$$J = \max_{\text{configurations}, C} \sum_{\text{tasks}, t} U(t, k)Q(t, C)$$

Utility – “how important” is task $t$ to the users at time $k$?

$$U(t, k) = \sum_{\text{groups}, g} w_g U_g(t, k)$$

Quality – “how good” is scanning configuration $C$ (distance, coverage, # radars) for task $t$?
How to define “how important”: $U_g(t,k)$

- user values for detected weather features

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Prior Information available</th>
<th>NWS utility Wt=0.4</th>
<th>EM utility Wt=0.3</th>
<th>Researcher utility Wt=0.2</th>
<th>Vieux utility Wt=0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TVS detection</td>
<td>AOP</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Remote</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Mesocyclone</td>
<td>AOP</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Remote</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Storm cell</td>
<td>AOP</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
How to define “how important”: $U_g(t,k)$

- “naturally”: group-sensitive utility for each feature (tornado, wind shear, hail core) scanned
- … and the survey says…..

User feedback:

- NWS: want “mental movie” scanning “areas of interest” at regular intervals
- need context: scan areas around features (storm cell)
- want to joystick system (want their own network)
User Utility Rules (revised)

- *interval-based preferences*: “do X every Y time units”
- utility considers both objects, time

<table>
<thead>
<tr>
<th>Rules</th>
<th>Rule trigger</th>
<th>Sector Selection</th>
<th>Elevations</th>
<th># radars</th>
<th>Contig.</th>
<th>Sampling interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N1</td>
<td>time</td>
<td>360</td>
<td>lowest</td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>N2</td>
<td>storm</td>
<td>task size</td>
<td>lowest</td>
<td>1</td>
<td>Yes</td>
<td>1 / 2.5 min</td>
</tr>
<tr>
<td>EMs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>time</td>
<td>360</td>
<td></td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>E2</td>
<td>reflectivity over AOI</td>
<td>task size</td>
<td>lowest</td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>E3</td>
<td>velocity over AOI</td>
<td>task size</td>
<td>lowest</td>
<td>2+</td>
<td>Yes</td>
<td>1 / 2.5 min</td>
</tr>
</tbody>
</table>
How to define “how important”: $U_g(t,k)$

- “naturally”: group-sensitive utility for each feature (tornado, wind shear, hail core) scanned
- … and the survey says…..

User feedback:
- NWS: want “mental movie” scanning “areas of interest” at regular intervals
- need context: scan areas around features (storm cell)
- want to joystick system (want their own network)
want to joystick system
(want their own network)

Virtualization: enabling the end user

- virtualization of computing, communication, and sensing resources
- each user:
  - sees “standalone” sensor network
  - can modify, download, execute, experiment with own code
  - implements user-specific service outside (architecturally above) infrastructure provider
Virtualization: making end users happy

instead of this….

….this system view
Virtualization: enabling end user

instead of this….

logical user-view: dedicated system!
Why virtualization?

- users want programmability/resources at *in-network* nodes: computing over local data, storage
  - good application: avoid active networking redux

- challenges: virtualizing sensing resources:
  - *sharing*: sensed data from one user usable by another (unlike bandwidth, computing)
  - *admission control*: mediating among different users with different priorities
  - partially satisfiable user requests? (negotiate?)
  - time-vary allocation of resources?
  - priorities among users (policy?)

research challenges
Sensor Virtualization Architecture
Overview

- QOS retrospective
- what have we learned?
- meeting “end-user” needs: sensor net scenario
  - embedding end-user needs
  - virtues of virtualization
- conclusions
The really big picture

- importance of user requirements

“It’s the user, stupid”

“It’s the application, stupid”

“It’s the network, stupid”

of course, not everyone agrees ....

Verizon product, 2008
The *really* big picture

- importance of user requirements
  - “It’s the end-user, stupid”
  - “It’s the application, stupid”
  - “It’s the network, stupid”

- architecture (as opposed to stovepipe) for embedding user requirements into network?
  - sensor networks
  - content distribution
  - special-purpose overlays
The end
thanks!

?? || /* */