Context-Aware Network Stack Optimization

Olivier Mehani <olivier.mehani@inria.fr>

Inria Rocquencourt, Imara project-team

INRIA ROCQUENCOURT

IMARA

August 5, 2009

Outline

1. Context and Objectives
2. Target Architecture
3. Constraint-based Decision Algorithm
4. Example Scenarios
5. Current Goals

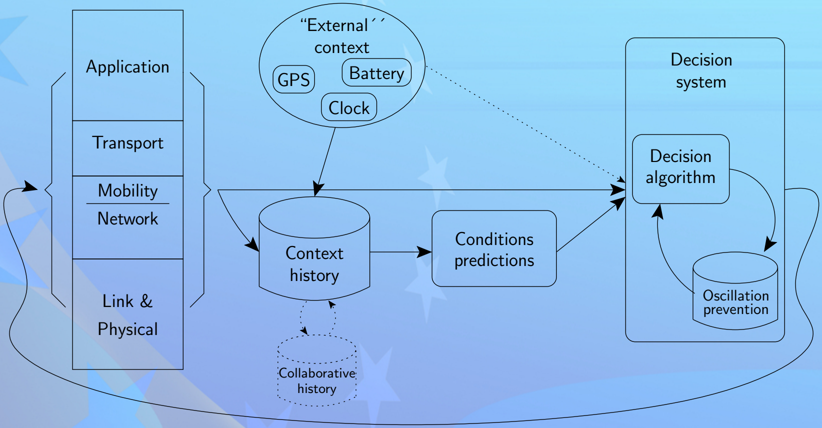
Context and Objectives

- Mobile environment
  - Rapidly changing conditions
  - Mobility patterns
- Network mobility and Multihoming
  - Several interfaces, networks and routes available at once to choose from
- Ad-hoc environment
  - few or no structural organisation
  - possibility of insider attacks
- Information and control scattered between layers
  - Consideration of generic cross-layer approaches

$\Rightarrow$ Need (and possibility) to use the network resources as soon as possible and as much as supported

Target Architecture

Towards a global optimization system



Target Architecture

Combining observation, prediction and decision

- Abstracted metrics from the stack
  - 802.21, achievable datarate, delay
  - routing metrics (incl. ETX/ETT?) and costs
  - RTTs, throughputs, loss rates (if available) along the current path (cf. routing)
  - application requirements and "satisfaction"
  - timescale $\simeq$ 100 ms (RFC 4907)
  - ▸ More on metrics
- Context observations and history
  - monitor current "context" (*e.g.* available networks, localization, battery levels, time of the day)
  - human timescale: days or weeks
  - predict forthcoming conditions based on previous observations
  - possible use of remote collaborative databases (cf. MobiSys 2009 pp. 29 and 123)
- Decision engine

  internal history to avoid short-period oscillations

  hints to the stack to finally optimize the network usage

INRIA ROCQUENCOURT

Target Architecture

Optimization decision based on global knowledge

- Globally aggregated information from the stack layers
  - Application: $I_{\mathrm{app}}(s,t) = \{c = \text{codec}(t), \ldots\}$
  - Transport: $I_{\mathrm{trp}}(s,t) = \{thr = \text{throughput}(s,t), \text{rtt}(s,t), \ldots\}$
  - Network: $I_{\mathrm{rt}}(t) = \{nh_B = \text{nextHop}(B,t), \ldots\}$
- Relative impact of the combination on current performance
  - "For socket s, a throughput of thr, needed by codec c, is achievable towards node B along a path starting with nh_B."
  - Identification of other communications with common characteristics (*e.g.* same destination) but different performance (*e.g.* higher throughput)
- Hint the network stack layer to adapt to the possible conditions *e.g.*,
  - change interface modulation or power
  - switch route for an address (or range), perform a handover
  - update transport state parameters
  - modify application parameters (*e.g.* encoding or rate)

Constraint-based Decision Algorithm

Motivation and basic idea

- Large configuration space → Combinatorial search techniques
- Modelled as a Constraint Programming problem
- The constraints solver unifies parameters to derives hints for the stack
  - Application quality: Qual(*Quality*, *Throughput*, *Jitter*, *RTT*, *PER*)
  - Application socket: SocketDestination(*Socket*, *Destination*)
  - Transport and routing conditions: Routing(*Destination*, *Route*, *Interface*, *Throughput*, *Jitter*, *RTT*, *PER*)
- Variables to unify are
  - observed conditions on the links/networks/paths (offer)
  - possible configurations of the layers (demand)

Constraint-based Decision Algorithm

Example relations for a simplified model

Observed network performances

Destination	Route	Interface	Throughput	Jitter	RTT	PER
Addr1	NH2	eth0	2 Mbps	1×10^{-4} s	10×10^{-3} s	0%
Addr1	NH1	wlan0	900 kbps	1×10^{-3} s	100×10^{-3} s	10%
Addr2	NH1	wlan0	450 kbps	1×10^{-3} s	250×10^{-3} s	30%
			...			

Socket between applications and destinations

Socket	Application	Destination
1	App1	Addr1
	...	

Interface costs (switching + usage)

Interface	Cost
eth0	10
wlan0	100
ppp0	250
...	

Application App1 parameters and requirements

Quality	Throughput	Jitter	RTT	PER
1	$\geq$ 1.5 Mbps	$\leq 10^{-3}$	$\leq 10 \times 10^{-4}$ s	$\leq 10 \times 10^{-3}$
2	$\geq$ 1 Mbps	idem	idem	idem
3	$\geq$ 500 kbps	$\leq 10^{-2}$	$\leq 10 \times 10^{-3}$ s	idem

Example Scenarios

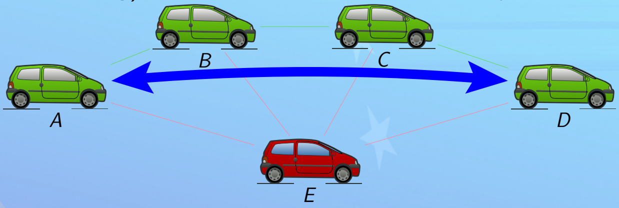
Transport adaptation on link characteristics change

- Several possible changes in link configuration
  - increase or reduction in the number of hops in a mesh network
  - switch from NEMO routes to MANET routes (or opposite)
- Transport protocol not directly aware of such changes
  - Slow feedback-based adaptation to new link and path characteristics
  - → Provide hints about the link performance (*e.g.* datarate to next hop) and path (*e.g.* if information available from another connection)
- Similar issues for a new connection
  - Hints about the expected path throughput as observed on other sockets along the same path
  - Update *e.g.* the slow-start threshold (for TCP)

Example Scenarios

Rogue VANET Node

- E advertises its presence but doesn't forward (or stops forwarding) other vehicles' traffic properly



- communication between A and D is not possible along the shortest path
- switching to another route would be desirable
- may benefit from quicker adaptation at various layer (*e.g.* transport layer parameters to adapt to larger RTT and smaller throughput)

Example Scenarios

Other examples

- Need to demonstrate use of contextual information
  - relative position or current node (vehicle) with neighbors and CN to help select route/next hop
  - choose wireless networks depending on previous history (*e.g.* been there for two hours the last three days) [CAMP]

Current Goals

- Build an experiment for transport adaptation scenario
  - three (or more) cars
  - OLSR/NEMO + MCoA (Manabu's work)
  - hint passing to transport
  - incorporation of CP solver
- Populate the CSP's tables in order to evaluate the solver's (offline) accuracy based on real data
  - data acquisition (*e.g.* throughputs, applications needs or RTTs)
- Identify more expressive scenarios combining context- and stack- provided information

Backup

Target Architecture

Metrics, statuses and parameters

- Physical/Link

  Raw characteristics rate, status, bytes, lost segments (RR), lost fragments (FER), link, noise (RSSI, SNIR) ; Notifications: link_up, link_down, link_quality_changed [ULL], transmission power

  - for each MAC neighbour (*e.g.* AP/Cell Tower in infrastructure mode; all neighbours for ad-hoc modes)

  Events Link Up/Down/Parameters Change/Going Down; Load Balancing; Operator Preferences [802.21]

  Contextual information VLAN [802.1q], SSID [802.11], CellID,...

- Network/Mobility
- Transport
- Application
- See also [RFC4907]

Target Architecture

Metrics, statuses and parameters

- Physical/Link
- Network/Mobility
  - abstract route metrics
  - MTUs
  - possible next hops to an address/range,
  - route addition/removal/change
  - ARP/NDP: mapping from next hop to MAC address
  - single-interface handoff decisions
- Transport
- Application
- See also [RFC4907]

Architecture description

Target Architecture

Metrics, statuses and parameters

- Physical/Link
- Network/Mobility
- Transport
  - throughput (for TCP: cwnd, sstthr, wnd)
  - RTT
  - loss rate
  - congestion information about paths
  - path MSS
  - peers' network address(es)
- Application
- See also [RFC4907]

Architecture description

Target Architecture

Metrics, statuses and parameters

- Physical/Link
- Network/Mobility
- Transport
- Application
    - requirements in end-to-end bandwidth
    - end-to-end delay limits
    - need for packet reliability (implicitely stated when chooosing the transport)
    - configurable modes of operation (*e.g.* codecs), and all of the above for each
    - "satisfaction" (completely ad-hoc to the application *e.g.*, peer feedback on validity of data)
- See also [RFC4907]

◂ Architecture description