Autonomic Transport Framework
for large family of communicating objects

Ernesto EXPOSITO
ernesto.exposito@laas.fr

LAAS/CNRS - OLC group, Toulouse
MCF - INSA de Toulouse
Outline

• Motivation
• The Autonomic Transport Protocol Framework
  - Design requirements
  - Communication patterns
  - Autonomic approach
  - Semantic for composition
• Conclusions and perspectives
Motivation: QoS and transport layer evolution

QoS requirements

- Reliability
- Order
- Bandwidth
- Time constraints (delay, jitter)
- Synchronization
- Partial reliability
- Partial order
- Others: security

Which transport service?
Which Transport/Network service composition?
How to incorporate new services or service specializations?
**Motivation: transport services space**

Traditional transport protocols:
- Order
- Reliability
- Congestion Control

TCP
- (total order, full reliability, CC)

SCTP
- (partial order, full reliability, CC)

DCCP
- (non order, non reliability but CC)

UDP
- (non order, non reliability)

TCP
- (total order, full reliability, CC)

POC: PO/I
- (partial order, partial reliability)

**Time**
- Order

**Reliability**
- and Congestion Control
- (network resources preservation)

**Internet of Things / Toulouse - Oct 2008**
Autonomic Transport Protocol Framework

Requirements
• Composite architecture
  – Existing mechanisms
    • Error control
    • Rate control
    • Congestion control
  – New mechanisms
    • Specialization of existing mechanisms
      – Adaptive Partial reliability
      – Adaptive congestion control
• Autonomic behavior:
  – Autonomic managers
  – Self-configuring
  – Self-optimizing
Composite communication pattern

active abstract class CommPattern

Data plane

Management plane
Example 1: Composite for TCP-Friendly Rate Control (TFRC)
Example 2: Composition/specialization of basic mechanisms reliability + rate/congestion control

Standard congestion control:
TFRC
(TCP-friendly Rate Control)

Congestion control and partial reliability
(TFRC & PR)

SendBufferRer

BufferReception
(order/reliability control)

ProcessFeedback
(TFRC & SACK)

LossDetection

Specialization of a management component

Specializations based on various rate control algorithms
(Best-effort and guaranteed network services)
Adaptive composite communication pattern
Example 3: Adaptive port for external reconfiguration (e.g. by the application)

Standard congestion control:
TFRC (TCP-friendly Rate Control)

Congestion control and partial reliability (TFRC & PR)

SendBufferRetr

BufferReception (order/reliability control)

ProcessFeedback
(TFRC & SACK)

LossDetection

Specialization of the management component for external configuration

Adaptive port for external reconfiguration

Internet of Things / Toulouse - Oct 2008
Example 4: Autonomic Manager and adaptive component

TimeControl: AutonomicManager

Monitored parameter: delay

Min: 0
Target: target
Max: max

Selective discarding (images I & P)

QoSParser

RateControl

QoS-aware & Time-constrained

ProcessFeedback
NoFeedback
CreateFeedback

Manag. plane

Data/control plane

output

input

H.263 VIDEO

Partial reliability (degradation)

Time constraints

Interactive: $(l, P) = \{(100, 100), (100, 50), (50, 0)\}$

$(T_{min}, T_{max}) = (25, 400)$
Example 5: More specializations/compositions

Standard congestion control: TFRC (TCP-friendly Rate Control)

Congestion control and partial reliability (TFRC & PR)
- SendBufferRetr
- BufferReception (order/reliability control)
- ProcessFeedback (TFRC & SACK)
- LossDetection

Congestion control and differentiated and time-constrained partial reliability TFRC & TD-PR
- QoSParser
- ProcessFeedback (TFRC & TD-SACK)

Congestion control (TD) et Partial reliability (TD) TD-TFRC & TD-PR
- RateControl
  - QoS-aware & Time-constrained

Internet of Things / Toulouse - Oct 2008
Composition of autonomic components

Internet of Things / Toulouse - Oct 2008
Example 6: Autonomic manager for VoD application based on the composition of two TD-TFRC instances

![Diagram of autonomic manager for VoD application](image-url)
Autonomic managers orchestration

- **intraflow**: com::AutonomicManager
  - Implicit Packet Meta Header
  - e.g. H.264 streams

- **intraApp**: com::AutonomicManager
  - Based on inter-flow priorities
  - Application reqs/preferences

- **intraSystem**: com::AutonomicManager
  - Based on inter-apps priorities
  - User reqs/preferences

- **intraGroup**: com::AutonomicManager
  - Based on inter-user priorities
  - Group of user reqs/preferences
  - QoS Semantic (reqs/preferences)
QoS provisioning model based

Actor policies
(i.e. user/application, service provider, network operator)

Objective function: maximize QoS

Provisioning constraints

Available communication services:
(i.e. network, transport, middleware)

Provisioning decision

Decision Model
QoS provisioning model based

Definitions

\[ F = \{ f_1, \ldots, f_n \} \] is the set of \( n \) multimedia flows competing for the available services

\[ S_j = \{ s_1, \ldots, s_m \} \]: end-to-end communication services (single/composition)

\( x_{ij} = \{0,1\} \) is the decision variable associated to the use of the service \( s_j \) to transmit the flow \( f_i \)

\( a_j \): Maximum availability of service \( s_j \)

\( p_i = \{1..p_{\text{max}}\} \) priority associated to the flow \( f_i \) (e.g. user or application preferences)

\( g_{ij} \): Gain in quality associating the use of the service \( s_j \) for the flow \( f_i \)

\( c_{ij} \): Cost of using the service \( s_j \) for the flow \( f_i \)

Decision model

Objective function:

\[
\max z = \sum_{j=1}^{m} \sum_{i=1}^{n} p_i \cdot g_{ij} \cdot x_{ij}
\]

Subject to:

\[
\sum_{j=1}^{m} c_{ij} \cdot x_{ij} \leq a_j
\]

\[
\sum_{j=1}^{m} x_{ij} \leq 1, \forall i
\]
Deployment: Service Component Architecture (SCA) approach

- Based on Service Component Architecture (SCA) [OASIS/OSOA]:
  - Provides a programming model for building systems based on SOA
- SCA definitions:
  - Service components: implement and use other services
  - Composites: assembly of components (including connections/bindings)
Semantic for Deployment

SCA approach based on Ontologies for characterization and composition

- QoS requirements
- Underlying network services

transport service specification

Service Characterization

Internal Service Composition (micro-protocols)
| Application profile | Stream  
|                     | e.g. VoD | Non-Stream  
|                     |          | e.g. file transfert |
| Network services    |          |                   |
| RT                  | ETP[]    | ETP[EC]           |
|                     | = UDP    |                   |
| NRT                 | ETP[gTFRC] | ETP[gTFRC+EC] |
| BE                  | ETP[TFRC+DT] | ETP[TFRC+DT+EC] |
Conclusions and perspectives

• **Communication Patterns specified** (available U2 model)
  – Basic transport services
  – Adaptation
  – Autonomic management
  – Composition and deployment

• **Semantic**
  – Requirements and preferences
  – Composition (components and services)

• **Projects/perspectives**
  – Feel@home project: unicast (intra-inter homes) + mcast
  – Studies for deployment in resource-constrained (e.g. sensor transport services)
Thanks,

Questions…?