

Network Working Group  
Internet Draft  
Intended status: Informational

Dimitri Papadimitriou  
Martin Vigoureux  
Alcatel-Lucent

Expires: April 14, 2013

Wouter Tavernier  
Didier Colle  
UGent  
October 15 2012

IRS Architectural Framework  
draft-dimitri-irs-arch-frame-00.txt

## Abstract

This document details the possible architectural and design choices in order to determine i) the spatial location of the so-called "IRS interface" and the functional entities it directly and/or indirectly involves, ii) the number of levels of redirection between routing modules and application (and associated identifier space), and iii) the various constraints that can be anticipated when dealing with the coupling of functional planes including the prevention of oscillations between two or more routing system states, coupling effects (leading to amplification chains) and, concurrency (leading to antagonistic action-reaction).

Disclaimer: this document doesn't intend to promote any specific architecture; its purpose is to understand (some of) the architectural challenges when designing interaction between routing system and applicative levels/systems.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>



The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 14 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction.....	3
Conventions used in this document.....	4
2. Terminology, Notation and Acronyms.....	5
2.1. Terminology and Notation.....	5
2.2. Acronyms.....	6
3. Use Cases.....	7
4. Architectures.....	9
4.1. Dual architectures.....	9
4.1.1. Boundary on external interface.....	9
4.1.2. Boundary on internal interface.....	13
4.2. Star architecture.....	15
4.2.1. Co-located Dispatch Function.....	15
4.2.2. Non Co-located Dispatch Function.....	16
4.3. Meshed architecture.....	18
5. Comparative Analysis.....	19
6. Security Considerations.....	19
7. IANA Considerations.....	19
8. Conclusions.....	19
9. References.....	20
9.1. Normative References.....	20
9.2. Informative References.....	20
10. Acknowledgments.....	20

## 1. Introduction

Nowadays, applicative layers don't only rely on the shared infrastructure for information communication purposes but also for information storage (e.g., content delivery/caching, large data sets) and processing (e.g., grid computing, virtual machines). The expansion of the functional role of the shared infrastructure has in turn induced the rise of intermediate systems/application level hubs under the control / supervision of providers (usually ranging from application providers to mediators).

With the current Internet model where the routing system is a closed system, applications have to run their own measurement end-to-end to determine forwarding path characteristics through traffic monitoring but also have no means by which some network path can be enforced (by network path we refer here to the path as computed on-line by routing algorithms or those made available by off-line means). The best example is the triangular inequality violation where the shortest path between two end points (shortcut) may not be the best bandwidth x delay path between these two points. Indeed, as there is no deployed mean (e.g., source routing) by which applicative layers can tell the network which path IP datagrams shall follow, measurements don't allow any subsequent decision tuning on which network path to follow to reach a given destination. The alternative is to drive the routing table entries by applicative-triggers exchanged by means of an API (referred to as IRS since one end of this interface terminates in the "routing system") assuming applications share with the routing system a common understanding of distance and resource usage metrics. In other terms, compared to the decoupling between traffic engineering decisions concerning network path and applicative layers needs, such model would allow cooperation in the routing decision process.

In this context, the actual objectives of the present document are the following:

- i) Enumerate possible architectural and design choices to determine the spatial location of the so-called "IRS interface" and the functional entities it directly and/or indirectly involves. In turn, this enables determining internal vs. external interfaces (those that MAY be standardized and those that MUST be standardized, and those that require specific architectural focus when dealing with security aspects).

- ii) Determine the number of levels of redirection between routing modules and application (and the associated identifier space).

iii) Identify pro's and con's of possible architectures depending on use cases (bottom-up) that in general will combine one or more of these elements: distance/path, topology ((abstract) nodes, (abstract) links), location/mobility (end-point, data, etc.); note that the inclusion of traffic properties leads de facto to the introduction of monitoring points.

iv) Anticipate the constraints (from base distributed control and decision theory) when dealing with the coupling of functional planes and some architectural invariants identified (top-down). These includes the prevention of i) oscillations between two or more routing system states, ii) coupling effects (leading to amplification chains) and, iii) concurrency (leading to antagonistic action-reaction).

v) Even if the initial architectural scope would be limited to single autonomous system as starting point, propose a generic enough description to enable inter-domain use cases in the future.

It is anticipated that the time dimension will limit applicability of such information exchange to the control part of the routing system (i.e. not the forwarding component). Indeed, the closer to the forwarding component the smaller the time scale to ensure proper functionality. The full Shortest Path Tree (SPT) computation takes of the order of 10microsec per IGP destination prefix, optimizations can be obtained using incremental Shortest Path First (iSPF) algorithms. Updating the central node Routing Table (RT) ranges in the same order of magnitude per destination prefix. The consecutive time the routing engine CPU is spending to update the RT entries or their distribution to the Forwarding Table (FT) is determined by the quantum of the swapping process. The quantum time can typically be configured between an order of 10 to 100ms. If we would consider an upper level components and interactions not directly associated to the routing system, it would logically operate in the order of the order 100ms or more generally of the order of seconds and above. The routing system is thus expected to remain the entity in charge of short-term adaptations to network "topological" or "reachability" changes.

#### Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [RFC2234].

## 2. Terminology, Notation and Acronyms

### 2.1. Terminology and Notation

- Agent: in general terms equipping a given function (in present context an exchange function) with memory property leads to the notion of object. Providing these objects with the capacity to adapt and to modify their behavior with respect to changing/variable conditions leads to the notion of adaptive agent.
- Application: for the sake of clarity, application refers here to computer software (organized collections of computer data and instructions) performing tasks such as data processing and/or storage, running on top of TCP/IP and accessible directly (or indirectly) to the its end-users on terminals/hosts/servers to accomplish these specific tasks.

Note: in practice, it is expected that the IRS will be initially used in the context of "network applications", i.e., programs receiving access to the routing modules and routing table in order to associate specific routing protocol decision/execution and routing table entries to specific needs as explained in Section 3.

- APP<sub>i</sub>: agent associated to application i (where index i = 1,...,L) that enables information exchanges with the dispatch function d.
- Boundary: determines/indicates the logical intersecting edge/area between the routing system and the application system. The existence as well as to the proper operation and management of this edge/area is key in developing and maintaining coherence across the intersecting systems.
- Co-located: refers to the placement of entities in close physical/logical proximity so that remotely/distantly they appear as occupying a single position/location.
- Dispatch function (D and d): generic term describing the functional block redirecting information. Redirection of information is defined either between dispatch functions or between dispatch function and agents.

Additionally the following functionality may be associated to this functional block: syntax and semantic information processing, aggregation/abstraction of information, orchestration, and scheduling.

Equipping this function with (at least some of) these additional functionality is also part of the architectural discussion and specification.

- External interface: interface that is addressable by objects/components outside of the objects/components they interconnect/put in relation and which usually belong to different entities; each object/component defining an entry point to the entity to which they individually belong thus resulting in higher security requirements.
- Internal interface: interface that is not addressable by objects/components outside of the objects/components it interconnects/puts in relation or the single entity that comprises them. The objects/components interacting through an internal interface are co-located.
- M<sub>m</sub>: traffic monitoring module/point m (where index m = 1,..P). Monitoring modules capture traffic and interface either with the IRS or with other routing modules via the dispatch function.
- RTR<sub>j</sub> (Router j, where index j = 1,..,M): entity hosting multiple routing modules (RM).
- RM<sub>k</sub> (Routing module k, where index k = 1,..,N): functional entity such as Interior Gateway Protocols (IGP), Exterior Gateway Protocol (EGP), etc. including the node global Routing Table (database structure and controller) as well as shared routing path computation functions (which for RSVP-TE is referred to as Path Computation Element or PCE). An agent indexed in the same way is associated to each routing module interacting with a dispatch function.

## 2.2. Acronyms

EGP	Exterior Gateway Protocol
FT	Forwarding Table
IGP	Interior Gateway Protocol
IRS	Interface to the RS
RM	Routing Module
RS	Routing System
RT	Routing Table

RTR      Router

### 3. Use Cases

The general purpose of the IRS is to enable the augmentation or the enhancement of the IGP (and even the EGP) based routing system with i) functionality it is not able to perform when operating on a pure IGP prefix basis with or without additional state information associated to their link (e.g. spatial traffic engineering information), ii) functionality involving time varying information at a higher rate than what IGP (and even EGP) can sustain by design, and/or iii) functionality it has not been initially designed for but that strongly influence the working of the routing system (this category includes anomaly detection, intrusion detection, etc.). Cases belonging to the sets i) and ii) would increase the memory cost and adaptation cost if the IGP would be directly involved in the corresponding routing information exchange process.

From this perspective, use cases can be further classified as follows:

- Isolation of source (some of which may be included in IGP prefixes and/or prefixes included in the routing table) and/or destination addresses subset of the prefixes stored by the routing table (some of which being derived from the IGP routing information base).
- Tuning of the routing entries parameters associated to destination prefix(es) (with as particular case, host-based prefixes) being subset of IGP destination prefixes from which routing table entries are derived.
- Sequencing/ordering of the computation and activation of routing table entries (where the sequence if left to the IGP alone would for instance delay convergence).

Note that the cases outlined here below are not claimed to be genuine but expectedly illustrative of the possible usage of the "IRS interface". The proposed classes are expected to be generic enough in order to "unify" its design.

Isolation refers to cases dealing with distributed intrusion detection and distributed traffic anomaly detection (e.g., dDoS) where the objective is to detect and identify the intrusion/anomaly and possibly prevent this intrusion/anomaly to further propagate. The latter typically involves determining through which router the corresponding traffic enters and possibly leaves the routing domain as well as determining the best router from where and to which the



incriminated traffic should be deflected. The term distributed means that several routers monitors collect traffic traces/records from the routing domain and the IRS is used as interface to a network application module capable of processing traffic records taken out of various monitors. Techniques enabling detection of never-seen-before patterns are now available (see e.g., [ECODE]) that limit the amount of configuration and maintenance required on each node (as long as the information captured on multiple monitors can be cross-processed). In case of isolation, the propagation of the information by means of the IGP would increase the number of entries to be stored at each router while only some of them require the corresponding information.

Tuning refers to use cases dealing with traffic engineering path computation and decision on a finer granularity than the one enforced by the destination prefixes distributed by means of the IGP. Parameter setting includes the possibility to associate a given incoming traffic flow to a specific routing entry (instead of using the "default" entry as determined by the destination address). Such parameters include in particular the bandwidth x delay product that can be computed for some of the (edge-to-edge) segments crossing the routing domain. The IRS can also be used to tune the parameter of the active monitors used to compute the available and residual capacity together with the edge-to-edge delay. On the other hand, passive monitors would not need to be activated on edge routers but placed so as to monitor specific spatial patterns of traffic. Moreover, instead of running at a default rate, the rate of packet / flow capture could be individually adapted depending on the applicative needs and the topological location of the router following that pattern.

Sequencing typically involves cases related to distributed decisions that need to be taken more rapidly and/or executions that need to be performed more rapidly than the convergence time IGP would impose (with all associated detrimental effects, e.g., routing loops, loss-of-traffic, etc.). This class includes fast re-routing (activation of a loop-free alternate routing/forwarding entry) but also configuration and maintenance operations involving for instance link metric changes, activation/deactivation of interfaces, etc.

As stated above, enabling several of these use cases would increase the memory and adaptation cost if the IGP would be directly involved in the corresponding routing information exchange process. Instead by assuming that i) not all additional entries are needed during the same period of time and ii) the number of active entries  $\ll$  total number of routing entries, one should be able to ensure higher "scaling" (or equivalently lower cost of scale) compared to the default situation IGP imposes (resulting from flooding of routing information).

#### 4. Architectures

The below section/sub-sections detail the architectures as far as our understanding goes in providing an interaction channel between the routing system (and its individual routers) and the applicative level (and its numerous applications).

##### 4.1. Dual architectures

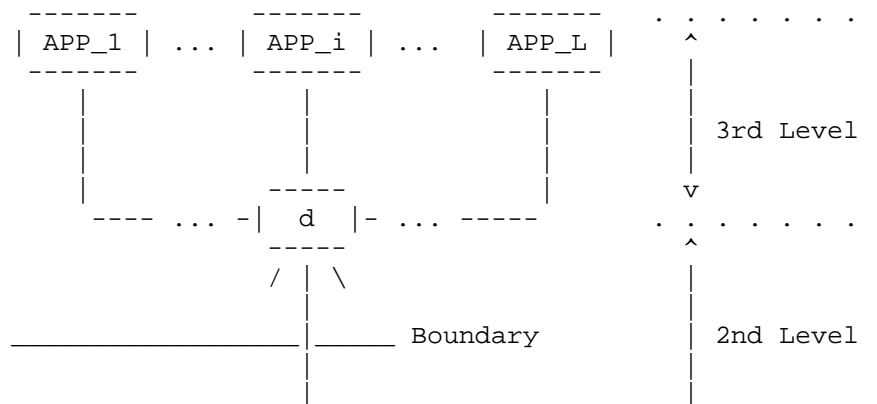
These architectures are characterized by three levels (or tiers) of redirection and a logical boundary between the routing system and the application system defined by the interface between their respective dispatch function.

###### 4.1.1. Boundary on external interface

This architecture is similar to the one exposed in the [RTG\_Area] presentation, where each routing entity or router (RTR) owns its individual dispatch function D and each APP agent is associated to a dispatch function d, distinct from the function D. This architecture involves three levels of redirection as depicted in Fig.1a. The logical boundary between the routing system and the application system is determined by the external interface between the dispatch function d and D (see below).

From this figure, the following relationships can be derived:

- Relationship APP to d: n:m (particular case: 1:1)
- Relationship d to D: m:n (particular case: 1:1)
- Relationship D to RM (same RTR): 1:n (particular case: 1:1)



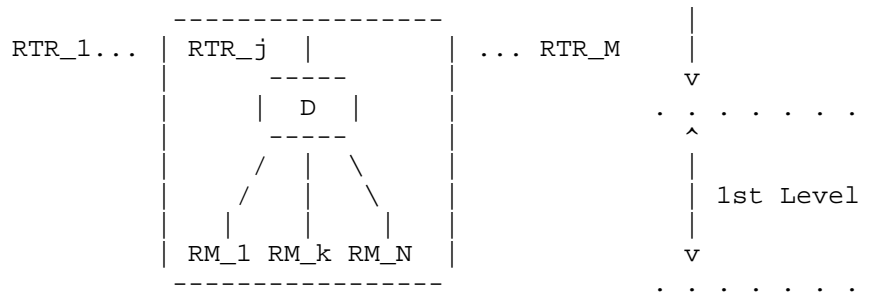


Fig.1a: Boundary on external interface d-D (co-located dispatch function D)

From Fig.1a, one can identify the following:

- \* Dual dispatch function:
  - . One dispatch function D is associated to each RTR entity.  
This dispatch function D is co-located with the RTR to which it is associated.
  - . One dispatch function d is associated to a set of one or more APP agents.  
This dispatch function d is not co-located with the APP agents to which it is associated.
- \* RTR level:
  - . Includes co-located dispatch function D
  - . Agents running on each routing module RM\_k communicate with the dispatch function D via an internal interface (the associated routing modules may be known to the local dispatch function D by means of a registration process).
- \* Interfaces:
  - . External interface between APP agents and d
  - . External interface between dispatch functions d and D
  - . Internal interface between D and RM\_k agent of RTR\_j
- . In terms of IRS:
  - The IRS would include the interface defined between the dispatch functions D and d (which is an external interface)
  - The question which remains is: does the IRS span the interfaces defined between i) RM\_k and the dispatch function D and/or ii) APP\_i and the dispatch function d. If not then the dispatch function will have to provide the syntax and semantic functionality to process messages receives from various agents.

Note that one can extend this architecture and assume that the dispatch function D is not co-located to its associated RTR as depicted in Fig.1b. This architecture also involves three levels of redirection as depicted in Fig.1b. In that case, the following relationships can be derived:

- Relationship APP to d: n:m (particular case: 1:1)
- Relationship d to D: m:n (particular case: 1:1)
- Relationship D to RTR: m:n (particular case: 1:1), meaning that a given dispatch function D can be shared among multiple RTRs.

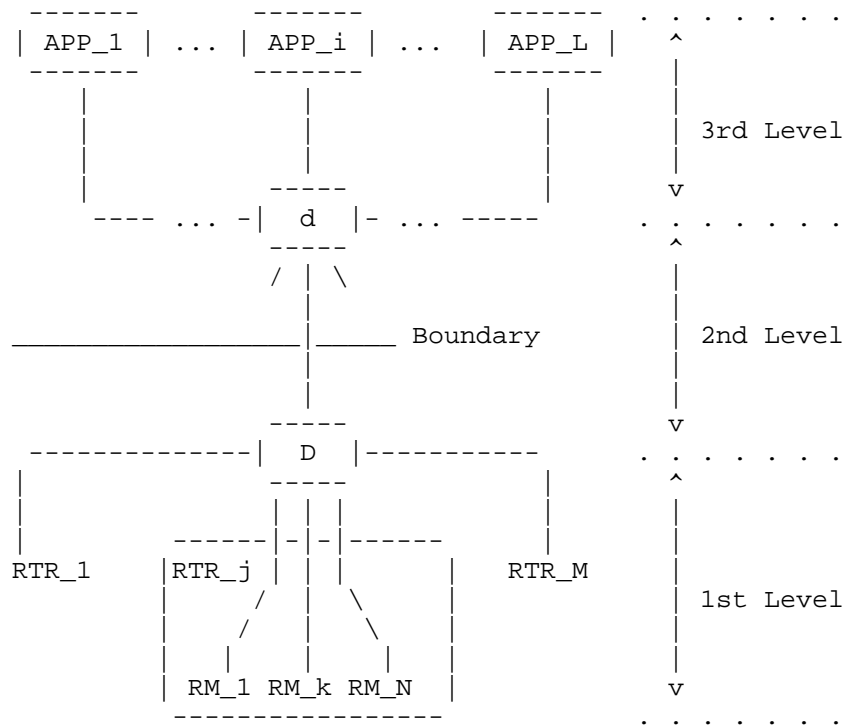


Fig.1b: Boundary on external interface d-D (non co-located dispatch function D)

From Fig.1b, one can identify the following:

- \* Dual dispatch function:

- . One dispatch function D is associated to a set of one or more RTR.  
This dispatch function D is not co-located with either of the RTR to which it is associated.
  - . One dispatch function d is associated to a set of one or more APP  
This dispatch function d is not co-located with either of the APP to which it is associated.
  - . The dispatch functions d and D are themselves not co-located, i.e., exchanges are performed by means of an external interface.
- \* RTR level:
- . No co-located dispatch function D
  - . Agents running on each routing module RM\_k communicate with the dispatch function D via an external interface (the associated routing module may be known to the local function D by means of a registration process).
- \* Interfaces:
- . External interface between APP agents and d
  - . External interface between dispatch functions d and D
  - . External interface between D and RM\_k agents of RTR\_j
- . In terms of IRS:
- The IRS would include the interface defined between the dispatch function D and d (which is an external interface)
  - Here too, the question which remains is: does the IRS span the interfaces defined between i) RM\_k and the dispatch function D and/or ii) APP\_i and the dispatch function d. If not then the dispatch function will have to provide the syntax and semantic functionality to process messages receives from various agents.

Remark: this architecture mandates/imposes to standardize all interfaces involved in the exchange process between the routing system and the applicative level.

#### 4.1.2. Boundary on internal interface

This architecture involves three levels of redirection as depicted in Fig.1c. The main differences with the architecture depicted in Section 4.1.1/Fig.1a are i) non co-located dispatch function D (with its associated RTR) and ii) an internal interface between the dispatch function D and d (instead of an external interface). The main differences with the architecture depicted in Section 4.1.1/Fig.1b is the internal interface between dispatch functions D and d (instead of an external interface). The logical boundary between the routing system and the application system is determined by the internal interface between the dispatch function d and D.

The following relationships can be derived:

- Relationship APP to d: n:m (particular case: 1:1)
- Relationship d to D: m:n (particular case: 1:1)
- Relationship D to RTR: m:n (particular case: 1:1)

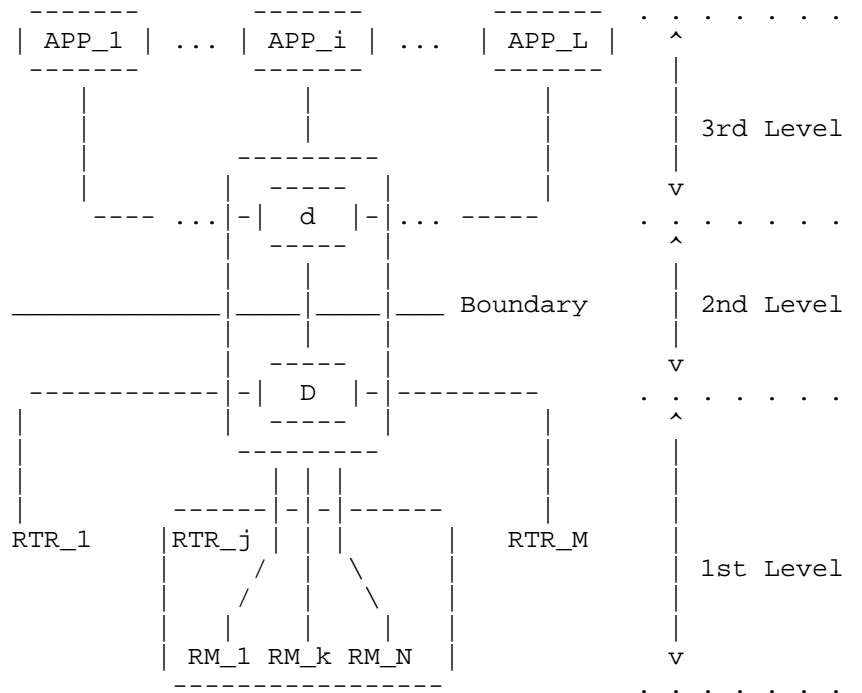


Fig.1c: Boundary on internal interface d-D (non co-located dispatch function D)

From Fig.1c, one can identify the following:

- \* Dual dispatch function:
  - . One dispatch function D is associated to a set of one or more RTR.
  - . One dispatch function d is associated to a set of one or more APP.
  - . The dispatch functions d and D are themselves co-located, i.e., exchanges are performed by means of an internal interface.
- \* RTR level:
  - . No co-located dispatch function D
  - . Agents running on each routing module RM\_k communicate with the dispatch function D via external interface (associated routing module may be known to the local function D by means of a registration process).
- \* Interfaces:
  - . External interface between APP agents and d
  - . Internal interface between dispatch functions d and D
  - . External interface between D and RM\_k agents of RTR\_j
- . In terms of IRS:
  - The IRS would include the interface defined between the dispatch function D and d (which is an internal interface)
  - Here too, the question which remains is: does the IRS span the interfaces defined between i) RM\_k and the dispatch function D and/or ii) APP\_i and the dispatch function d. If not then the dispatch function will have to provide the syntax and semantic functionality to process messages receives from various agents.

Remark: this architecture doesn't require standardizing the interface between dispatch functions (only procedures performed by the dispatch functions would need specification).

## 4.2. Star architecture

These architectures are characterized by i) two levels of redirection, and ii) common dispatch function (dD) between APP and RTR. The boundary in these architectures can be seen as determined by the intersecting/common function dD. When this function is supplied with memory property, it is referred to as a boundary object.

### 4.2.1. Co-located Dispatch Function

In this case, the following relationships can be derived:

- Relationship APP to dD: n:m (particular case: 1:1)
- Relationship dD to RM (same RTR): 1:n (particular case: 1:1)

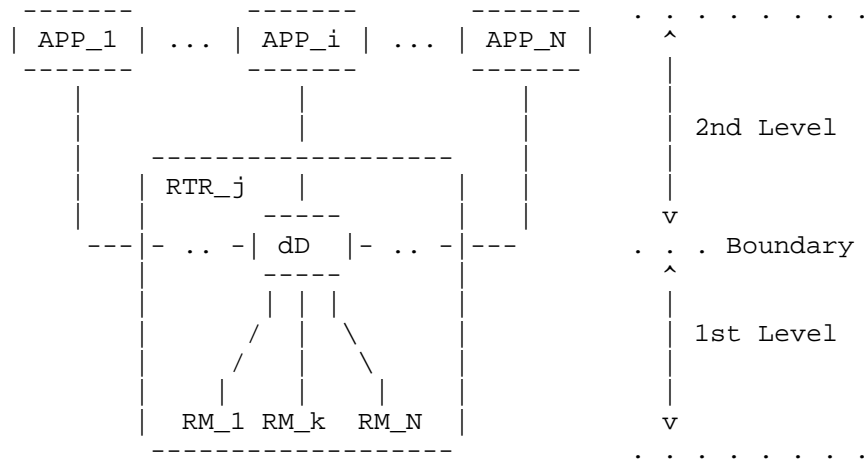


Fig.2a: Star architecture - Co-located common dispatch function

From Fig.2a, one can identify:

- \* Common dispatch function:
  - . A dispatch function dD is associated to each RTR. The same dispatch function dD is associated to a set of one or more APP. One can thus refer the function dD to as a common dispatch function
  - . The dispatch function dD associated to the RTR\_j is co-located with the RTR\_j
- \* RTR level:
  - . Includes co-located dispatch function dD



. Agents running on each routing module RM\_k communicate with the dispatch function dD via internal interface (associated routing module may be known to the local function D by means of a registration process).

\* Interfaces:

. External interface between APP agents and dispatch function dD  
 . Internal interface between dispatch function dD and RM\_k agents of RTR\_j

. In terms of IRS:

- In this case, the IRS would span the interfaces defined between APP\_i and the dispatch function dD. The question
- The question which remains is: does the IRS span the interfaces defined between RM\_k and the dispatch function D. If not then the dispatch function will have to provide the syntax and semantic functionality to process messages receives from various agents.

#### 4.2.2. Non Co-located Dispatch Function

In this case, the following relationships can be derived:

- Relationship APP to dD: n:m (particular case: 1:1)
- Relationship dD to RTR: m:n (particular case: 1:1)

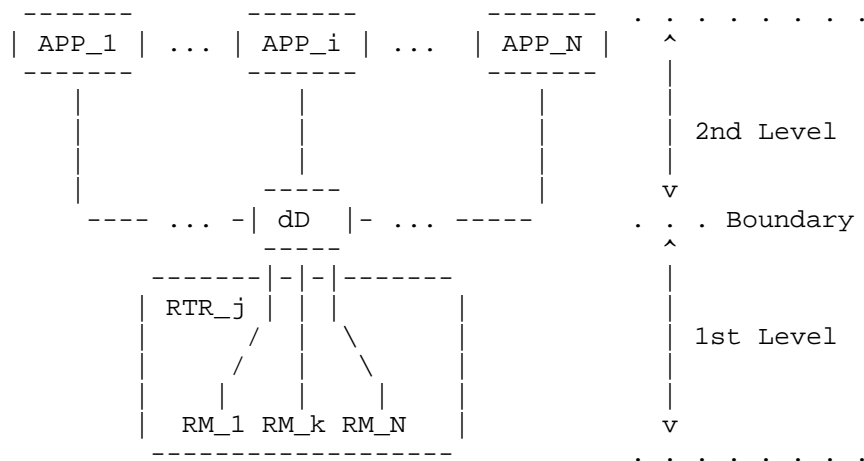


Fig.2b: Star Architecture - Non co-located common dispatch function

From Fig.2b, one can identify:

- \* Common dispatch function:
  - . One common dispatch function dD is associated to a set of one or more RTR and to a set of one or more APP.
  - . This dispatch function is neither co-located with the router (RTR) (or its routing modules) nor with the APP agents. In this case, one can refer to a commonly shared dispatch function dD.
- \* RTR level:
  - . No co-located dispatch function dD
  - . Agents running on each routing module RM\_k communicate with the dispatch function dD via external interface (associated routing module agents may be known to the local function D by means of a registration process).
- \* Interfaces:
  - . External interface between APP agents and dispatch function dD
  - . External interface between dispatch function dD and RM\_k agents of RTR\_j
- . In terms of IRS:
  - In this case the IRS would span i) the interfaces defined between APP\_i and the dispatch function dD and ii) the interfaces defined between RM\_k and the dispatch function dD.

#### 4.3. Meshed architecture

Alternatively, one may consider that each agent executes its own dispatch function and the relationship between APP\_i and RM\_k agents is n:m. In this architecture, the boundary is thus determined by the set of APP agent to RM agent relationships.

This architecture however implies that i) APP and RM agents are knowledgeable of each other, and ii) the individual routers must provide means to ensure consistency and coherence of decisions (by their routing modules) but also to prevent concurrency between decisions (leading to antagonistic action-reaction). For this purpose, a decision control function (hD), performing decision verification, consistency-check, etc. is to be considered that is co-located with individual RTR (see Fig.3). The interface between each routing module RM\_k and the function hD (indicated with asterisks in Fig.3) falls beyond the scope of IRS.

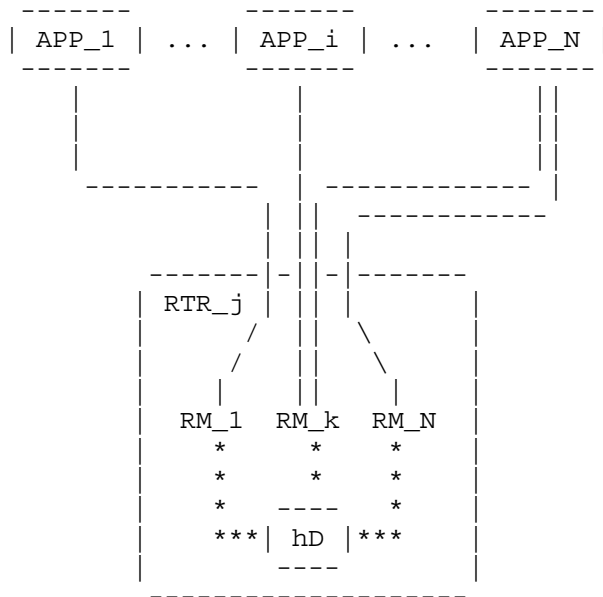


Fig.3: Meshed architecture

In a sense, the architectures presented in Section 3.1 and 3.2 perform inbound decision control whereas the architecture depicted in Fig.3 performs outbound decision control. Inbound (to RM) means that the function D (in the Fig.1 and Fig.2) processes the incoming information to ensure that the proper decision left subsequently to

individual RM but the function D is not part of the decision control process (verification, consistency-check, etc.). This process is performed by the RM themselves. On the other hand, outbound (to RM) means that the function hD (in Fig.3) performs verification, consistency-check, etc. of the (individual) decisions taken by the RMs.

## 5. Comparative Analysis

TBD.

## 6. Security Considerations

There are multiple levels at which security considerations shall be embedded and from the start as part of the architecture. Indeed, application and routing systems may (and often will) be under the control of different parties/administrative unit and communication between them (using the various external interfaces outlined in this document) may be established across the Internet.

- Accessibility and communication: prevent denial of system/service and overloads, enforce authentication of individual communicating entities and prevent unauthorized access/masquerades (authorization).

- Information exchange: enforce information integrity and validity (verification process) as well as prevent information/data spoofing; information confidentiality may be critical for certain environments, not necessarily for others (e.g., NREN).

- Semantic processing: assuming that routing modules would receive information to derive decisions, individual information can be valid but its interpretation and resulting decisions may lead to executions weakening the routing system.

## 7. IANA Considerations

None

## 8. Conclusions

TBD

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

### 9.2. Informative References

- [RTG\_Area] Atlas, A., et al. "Interface to the Internet Routing System (IRS)", Routing Area Meeting, Vancouver (BC), Canada, July 2012.
- [ECODE] Project website: <http://www.ecode-project.eu>
- [OFELIA] Project website: <http://www.fp7-ofelia.eu>

## 10. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Part of the input that led to this document has been derived from the outcomes of the [ECODE] project (Grant No.223936) that is funded by the Framework Program 7 (FP7) of the European Commission and from the [OFELIA] FP7 project; both projects are part of the Future Internet Research and Experimentation Initiative (FIRE) of the European Commission.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Authors' Addresses

Dimitri Papadimitriou  
Alcatel-Lucent Bell N.V.  
Dep.: Bell Labs Benelux  
Copernicuslaan 50  
2018 Antwerpen  
Belgium  
Email: dimitri.papadimitriou@alcatel-lucent.com  
Tel: +32 3 2408491

Martin Vigoureux  
Alcatel-Lucent Bell Labs France  
Dep.: Bell Labs France  
Route de Villejust  
Nozay 91620  
France  
Email: martin.vigoureux@alcatel-lucent.com

Didier Colle  
Ghent University  
Dep.: INTEC  
Gaston Crommenlaan 8 bus 201  
9050 Ledeberg - Gent  
Belgium  
Email: didier.colle@intec.ugent.be  
Tel: +32 9 3314900

Wouter Tavernier  
Ghent University  
Dep.: INTEC  
Gaston Crommenlaan 8 bus 201  
9050 Ledeberg - Gent  
Belgium  
Email: wouter.tavernier@intec.ugent.be  
Tel: +32 9 3314900