# New Framework and Mechanisms of Context-Aware Service Composition in the Future Internet

Alberto J. Gonzalez, Ramon Martin de Pozuelo, Martin German, Jesus Alcober, and Francesc Pinyol

The ongoing proliferation of new services, applications, and contents is leading the Internet to an architectural crisis owing to its inability to provide efficient solutions to new requirements. Clean-slate architectures for the future Internet offer a new approach to tackle current and future challenges. This proposal introduces a novel clean-slate architecture in which the TCP/IP protocol stack is decoupled in basic functionalities, that is, atomic services (ASs). A negotiation protocol, which enables context-aware service discovery for providing adapted communications, is also specified. Then, we present how ASs can be discovered and composed according to requesters' requirements. In addition, a media service provisioning use case shows the benefits of our framework. Finally, a proof-of-concept implementation of the framework is described and analyzed. This paper describes the first clean-slate architecture aligned with the work done within the ISO/IEC Future Network working group.

Keywords: Future Internet, clean-slate architecture, service-oriented architecture, design, implementation, QoS/QoE provisioning, context-awareness.

## I. Introduction

In current-day society, there is no doubt that the Internet epitomizes a cornerstone tool for human communications, but, at the same time, it has triggered new challenging problems. Users expect higher levels of performance, security, and reliability. As a result of its gigantic growth, the Internet is reaching some technological and operational limits imposed by its architecture in its attempt to give full support to the new requirements introduced by increasing services, applications, and content. It still seems unclear how the current Internet architecture will handle these requirements, but the fast-growing need for strategies in the network that can guarantee a certain level of quality of service (QoS) and quality of experience (QoE) becomes paramount. In recent years, multiple initiatives have contemplated restructuring the current Internet architecture to cope with these limitations. Many of these solving strategies introduce novel clean-slate architectures that do not take TCP/IP as their groundwork. Following this trend, the principles on which service-oriented architecture (SOA) and service-oriented computing [1] are grounded offer some design guidelines that can pave the way to implement a new flexible architecture for the future Internet (FI) in which services could be easily deployed, discovered, and reused. SOA promotes the usage of services to support the development of rapid, interoperable, evolvable, and massively distributed applications. Furthermore, in [2], the authors presented a very disruptive idea on how to establish role-based architecture (RBA) network communications. Their proposal was to avoid the existing layered structure of the TCP/IP stack and in so doing extract the roles or network functionalities that as a result could be interconnected without current layer restrictions. Decomposition of the current protocol stack allows

a certain granularity of network functionalities and enables their selection only as required. This is the first step to provide inherent cross-layering functions to an FI architecture. It would allow avoiding numerous specific and complex cross-layer solutions, such as the improvements that are especially done in wireless networks [3]. Those solutions are too specific and cannot be reused for different purposes in varied situations. An architecture that does not have rigid layers (and thus is not handicapped by the hierarchical restrictions on their interconnection) can solve the same problems without this drawback as functionalities can be requested as required by a specific communication. To move service-oriented paradigms to the network level, this paper proposes and validates the decomposition (RBA-based) and composition (SOA-based) of network functionalities so as to enable a native cross-layering solution whilst avoiding functional duplicities. These functionalities can be seen as in-network services and as such should be discovered and combined to supply seamless communications. This approach aims at improving the satisfaction of users' expectations by matching offered service characteristics with requirements and preferences previously determined by them.

The rest of the paper is structured as follows. In section II, the related work is summarized. Section III presents the overview of the proposed architecture. In section IV, both the proposed scenario and testbed are introduced, including an analysis of some results obtained from a proof-of-concept implementation. Finally, conclusions and future works are presented in section V.

## II. Related Work

Currently, there are two major types of approach being utilized to redesign the Internet: evolutionary and disruptive. It is difficult to believe that evolutionary approaches can solve all the issues and challenges that the current Internet exposes (for example, multihoming, cross-layer interactions, middle-boxes, QoS, sublayer proliferation, mobility, security, and so on) in an efficient manner whilst providing enough flexibility to adopt future services and requirements that are yet unknown [4]. Most of the current Internet deficiencies derive from its original layered and monolithic architecture. Therefore, we advocate for a clean-slate redesign of the Internet architecture based on service composition approaches and SOA and RBA paradigms.

SOA [5] principles are very well known and widely deployed in Web services' environments. However, these services can be applied in a broader context, covering all levels of a communication. Some proposals seek to create services not only at the application level (for example, SOA4All [6]) but also at lower levels (network). Such relevant projects as

4WARD [7], RNA [8], SILO [9], RINA [10], and SONATE [11] composed functions that offer specific in-network operations. Although the projects defined the functions similarly, they referred to the functions using different terminology, that is, functionalities, services, protocol mechanisms, or building blocks. The main difference between them lies in how the projects defined modularization and composition. For example, RNA and RINA are based on the recursive composition of layers that are not prefixed, whereas SONATE is closer to the RBA modularization and SOA composition models. All of them tackle the lack of flexibility and the increasing complexity in the current protocol stack. However, they are focused on a single node and homogeneous composition of the communication stack and do not offer a complete global solution. Mechanisms to discover services and negotiate QoS and service-level agreements are critical for a future service-oriented and self-managed network.

Many service discovery strategies have been studied in literature in the Web service and ad hoc network fields. Some former strategies proposed solutions that use such repositories as UDDI, deployed in business environments. Others proposed different protocols, such as SLP, Salutation, Jini, and so on, which are too particular and present interoperability problems. Other P2P-based approaches, such as JXTA and Kazaa, operate only at application level. An extensive comparison can be found in [12]. In this work, we propose a generic service discovery protocol that works at network and application level and can also integrate context-aware capabilities.

In [13], the authors proposed a programmable network approach. The work introduced how over-the-top services could be allocated inside the network, but a complete framework in which users can discover, instantiate, and execute services remains undefined.

Closely related approaches were presented in the last few years in the software-defined networking field as well, with OpenFlow [14] protocol as the spearhead. OpenFlow defines an open protocol that enhances network flexibility. Although it is considered a very useful tool for network virtualization, it focuses on the management of a single domain network at link and network layers and does not offer a global solution. Dynamic allocation of resources and process delegation to network elements (for example, a video transcoding process) are uncovered issues. An SOA should indeed be required over the OpenFlow substrate.

Substantial prior work [2], [7]-[11] examined the benefits of new architectures. We view these proposals as complementary to ours, to complete a final definition of what FI should be at the network level, for the sake of a solid deployment of services without current restrictions. However, this work presents a novel approach for providing services adapted to

context conditions in heterogeneous networks, allowing services to be allocated depending on context parameters, such as link conditions, device capabilities, user preferences, and so on. This is achieved thanks to a generic context-aware service discovery protocol that integrates routing functions. There exists a parallelism between the context-aware service discovery protocol (SDP) described and the session initiation protocol/SDP for establishing voice/video communications over IP and describing the media sessions among the involved parties. Some other projects in the USA (CCNx/NDN, Geni), EU (ANA, PSIRP/PURSUIT, SAIL), and Asia (AKARI) have been issued to develop new network solutions from scratch as well. These clean-slate proposals share some common concepts, such as micro-modularization and virtualization as a means to support multiple architectures simultaneously, in their design and objective. Nevertheless, they also differ in scope or development (for example, protocol stack composition in a single node, focused in building a network of information).

In addition, the interest of the scientific community in proposing new solutions to current architecture has recently been driven by standardization bodies, such as ISO/IEC JTC1/SC6/WG7 (not discarding clean-slate) and ITU-T SG13 (IP-based). This work is completely aligned with the efforts done within ISO/IEC JTC1/SC6/WG7 "Future Network" Part 7 "Service Composition" [15].

## III. Architecture Overview

Services should not be fixed but dynamically composed where and when necessary, with respect to service requirements, network transfer capabilities, and surrounding context in the user and the network environments. This proposal presents a service-oriented framework able to deal with functionalities at all levels (connectivity, transport, application) by considering the provided service and not the technology behind the functionality. All these network functionalities can be seen as services by means of suitable service-oriented abstractions. Herein, existing functionalities and protocols can be included, as well as linked or enriched, and new functionalities can be easily introduced.

Subsets of services will be provided by nodes in the network and will be composed to create efficient end-to-end communications according to the requirements of the communication requesters. Hence, this task involves a context-aware service composition process. Depending on the type of requester, requirements may vary. Typically, the basic requirements of a communication are expressed in terms of QoS parameters. However, requirements can also be otherwise desired or even mandatory attributes in the communications, such as energy consumption, geographical location, and price.

Our work proposes a constraint-based routing (CBR) [16] performed hop by hop during the service discovery process. It establishes end-to-end virtual circuits between the requester and the provider of a demanded end service in a local domain.

Furthermore, this framework empowers requesters to choose, as it gives them the capacity to choose between different communications. Although it adds a certain level of complexity, it meets the socioeconomic requirements of the current commercial Internet, in which participating stakeholders have different and sometimes conflicting interests. For instance, network providers would like to minimize the consumption of their network resources, while service providers would prefer to maximize the quality and competitive pricing of their services over the network [17].

### 1. Service Framework

The proposed solution considers three basic components: atomic services (ASs), atomic mechanisms (AMs), and composed services (CSs). ASs are individual functions or roles commonly used in networking protocols, such as sequencing, acknowledgment, flow control, and so on. These are well-defined and self-contained functions, used to establish communications to create CSs. AMs are specific implementations for each AS, providing the desired AM functionality. An AS can be implemented by different AMs, as in the example shown later in section IV (Table 1). Finally, a CS is a combination of ASs that work together to provide a more complex service. CS logic needs to be specified in a workflow (WF) to describe the composition and execution process of functionalities or ASs that could be offered by different implementations (AMs).

The composition of ASs consists of discovering, selecting, combining, and allocating those services to be executed along the path from a requester node (RN) to the end service node (ESN) going through different intermediate nodes (INs). In this context, a composition process orchestrated by the RN is proposed with the aim to empower the requester's control over the communication establishment. The RN will therefore be able to decide which discovered services best meet its requirements and preferences by centralizing the process of service selection, composition, and allocation.

### 2. Service Discovery and Negotiation

Service discovery is the process of identifying the nodes that can provide the desired end service, as well as the ASs that may be required in the nodes of the communication path ranging from the RN to the ESN. This phase is divided into three steps as well. In the first step, requester requirements are mapped out to a service request. Secondly, the nodes that receive the query

evaluate whether they are able to provide the demanded service. Finally, context information is consulted to guarantee that the service can be provided under the required QoS parameters. Traditional Web service discovery mechanisms are focused on enterprise communications and use heavy formats based on XML. Pervasive computing SDPs do not provide a unique and integrated solution applicable to a global heterogeneous network. We hereby propose using an innovative negotiation protocol introduced in [4].

This negotiation protocol discovers a service in the network, taking into account specific requirements established by the requester. In addition, it is simple enough to work even in small and constrained environments, such as ad hoc sensor networks without infrastructure support. This negotiation protocol integrates service discovery and service allocation. The service discovery process consists of searching the network for services under certain conditions, by means of a communication request (Creq) message. To specify the criteria that will guide the search, a semantic negotiation protocol is proposed. This message should specify the requester's service requirements in terms of (a) network performance parameters, (b) additional constraints (for example, geographic requirements, domain restrictions, or attributes defined for certain services), and (c) required functionalities. Moreover, a service requirement is defined by either restrictive or nonrestrictive parameters. Restrictive parameters are those that are completely necessary to establish a communication, whereas nonrestrictive parameters are those that are not mandatory. Nonmandatory parameters allow for optimization of communication. Considering the inclusion of service requirements in the request, we propose the following generic definition of a Creq message:

$$Creq = session\_ID, End\_Service\_Name,$$
$$QoS\_Requirements[i](min, max),$$
$$Context[j](constraints, preferences),$$
$$AS[k](mandatory/optional), Effects, Resources, \quad (1)$$

where QoS_Requirements[i], Context[j], and AS[k] correspond to lists of QoS requirements, context parameters, and AS attributes, respectively. Additionally, effects can be specified as desired high-level features for the communication, such as security or reliability. A resource of a service, such as a film provided by a streaming service, can be specified as an extra parameter.

Using this type of request, information about the capabilities of nodes is discovered through the network. The default operation performed in a node when receiving a request is to evaluate whether it can provide the service. If it can provide the service, the node answers with a communication response (Cresp) message, which is transferred through the reverse path.

One important issue when propagating a service request is to limit the scope of the request so as not to flood the entire network or propagate it indefinitely. Depending on the kind of network, several approaches can be adopted. In a network without infrastructure support, such as an ad hoc sensor network, a time-to-live counter can be set. In the event, the node that is unable to provide the end service propagates the request to its neighbors until the requested service is found. A clear benefit of performing this operation by each node is that dynamic and frequent context changes can be faced. Moreover, neighbors' information can be exchanged by means of a context exchange protocol that notifies the nodes in a network of their presence and status and of the known nodes. This approach has the benefit of not requiring any infrastructure support, but scalability is limited to small environments due to the flooding methodology used in our experiments. On the other hand, if the network has infrastructure support, dedicated entities (for example, distributed directory nodes) can provide the required infrastructure/signaling services within a domain, considering a domain as a set of nodes interconnected according to any criteria, such as autonomous systems, administrative domain, geography, topology scope, and so on. Furthermore, to obtain high scalability in structured environments, the measures proposed in [18] could be applied.

This work focuses on the operation of the framework in a single domain. To verify the scalability of the proposed solution is a challenging issue that will be explored in future work. A possible implementation would be to use domain manager nodes in charge of relaying messages between different administrative domains and networks. These domain managers could be in charge of calculating the optimal path whilst considering different constraints. The main drawback is the complexity of this problem (NP complete) [19]. However, a near optimal path could be computed, taking into account the topology, capacity, and context constraints by means of specific heuristics. This approach is similar to compact routing approaches [20], such as stretch-3 [21].

In addition to this, it is remarkable that the proposed solution is agnostic to the underlying technology, including networks and devices, thanks to the methodology explained previously and a well-defined service abstraction (Fig. 1). Furthermore, the heterogeneity of the network can be addressed by checking the consulting nodes' capabilities, services, and resources, owing to the availability of context information.

Typically, QoS requirements specified in the request depend on dynamic conditions. Dynamism can be addressed by means of reservation messages before establishing the communication or thanks to admission control mechanisms. Then, what happens if a node fails? Resource reservation is a key feature of this approach when establishing the service, but it is not clear which model is the correct model for service adaptation. The
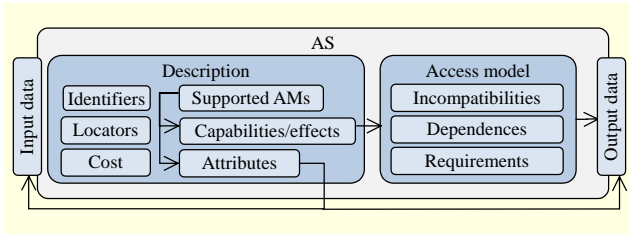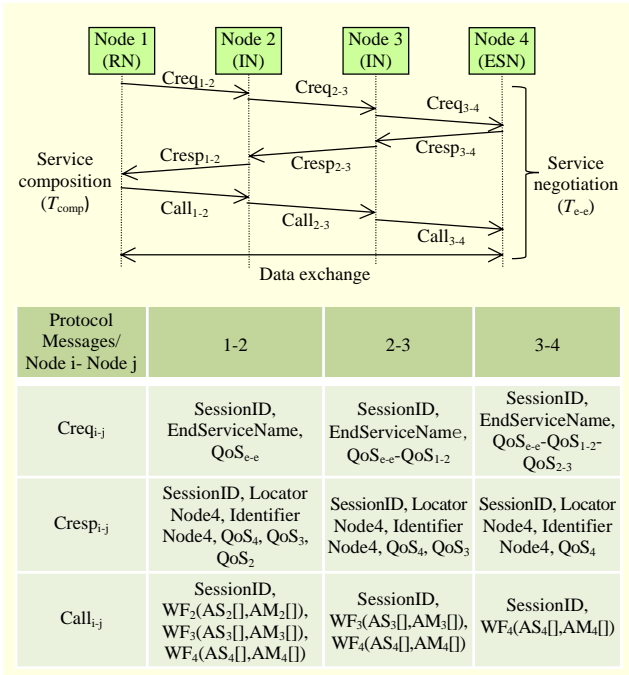
Fig. 1. Generic service interface.



Fig. 2. Negotiation process.

storage of alternative compositions can be another methodology to reduce the adaptation time. Further study is necessary to analyze where and when each solution is better suited. For an initial design, we consider four possible reactions, considering the type of adaptation required:

**Change an AM.** Identify the mechanism that provokes the failure and search for a mechanism that can fix the service behavior and solve the problem.

**Change the WF within a node.** When a node is unable to find a possible swap of AMs that fix the behavior of the service, it should find a change in its WF of ASs that fulfill the requirements once again. It must ensure that its outbound and inbound interfaces with other nodes will not change, to avoid the reconfiguration of other nodes.

**Delegate a WF to another node.** When a node cannot find a possible WF to provide the service achieving the demanded requirements, it should demand other nodes if they are capable of offering their WF or an equivalent one (a WF that provides the same functionalities but with better behavior).

**Recalculate the composition.** As a last option, if any one of

the previous solutions can fix the problem, the RN is informed, and it initiates a new composition from scratch, taking into account the updated context.

Following the service discovery process, we also propose a generic definition of a Cresp:

$$Cresq = Session\_ID, Node[m](node\_ID,$$
$$QoS\_Capabilities[j], AS[k], AS[l]). \qquad (2)$$

The requester will receive $N$ response messages that specify $N$ candidate paths. Each of them contains the identifiers of the $m$ nodes of the path, with the QoS capabilities, ASs, and AMs that each node can offer. Node[m], QoSRequirements[i], AS[k], and AM[l] are lists of the corresponding parameters.

Finally, the last message considered in the negotiation protocol is the communication allocation (Call) message:

$$Call = Session\_ID, Node[m](Node\_ID, WF), \qquad (3)$$

which is used to specify to each node which ASs and AMs must be executed. For this purpose, WF-based representations are used. Node[m] is a list of nodes in the end-to-end selected path. Figure 2 shows the whole negotiation process.

## 3. Service Composition

To empower the requester's control over the communication, the RN orchestrates the services. The requester will always choose among services that are discovered. The ASs that meet the requestor's requirements will be selected. To do that, it is important to have an expressive negotiation protocol, as described before, to allow the requested services to be matched with the services available in each node until the end service.

The information discovered in the network is organized in graph structures, in which the nodes of the graph are the ASs or AMs of each node. However, the Cresp obtained from the discovery process can be directly mapped into a tree of disjointed branches, such as the structure shown in Figure 3.

This work divides the composition process to create a CS made up of four phases. To solve the service composition process, we divide the process into the main subprocesses: filtering, AM scoring, AS composition, and path selection.

### A. Filtering

This phase consists of filtering all received Cresp messages according to the requirements specified by the RN. A range of possible costs acceptable to the user can be set up when specifying a constraint or a preference in the Creq. These filters are represented by specific rules, which can be solved by means of a constraint satisfaction problem method. The filtering process is inherent in service discovery. However, a secondary filtering phase is applied on the requester side, once
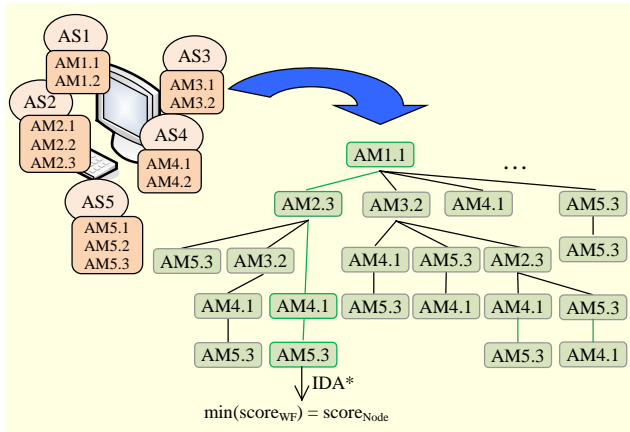
Fig. 3. AS composition process.

all the Cresp messages are received, to validate that all QoS requirements along the end-to-end path are fulfilled.

### B. AM Scoring

During this phase, the AM that implements each AS is selected according to specific scoring functions. This score considers different specific attributes related to the AS, such as the QoS parameters that they can provide and the priorities of the RN. For each AS, a set of possible AMs are scored and the best one is selected. In our preliminary implementation, the AM scoring considers each AM as an isolated process, regardless of the interconnected relationships of the AMs. We propose to use a generic weighting function (4) to score the AMs, wherein weights may vary depending on the preferences introduced by requesters.

$$Score_{AM} = A \cdot a + B \cdot b + C \cdot c + ... + Weight_n \, param_n. \qquad (4)$$

Herein, it is possible to define tradeoffs between different parameters, such as the quality provided by the network, requirements, and the price of paying for a service. However, scoring functions can be defined for each AS to consider specific requirements, as shown in [22], in which a score metric for audiovisual content was proposed.

### C. AS Composition

Usually, an operation can be offered by different combinations of ASs. For instance, a reliable service can be provided by means of acknowledgment, error detection, and retransmission functions or by applying forward error correction. Depending on the combinations, the provided QoS may vary. Thus, those best suited to satisfy the preferences of the requestor will be chosen. Note that the RN composes the services per each node in the path (RN, INs, and ESN) and generates the corresponding WFs. Considering the described architecture, once all services are discovered, the RN should be

able to create a tree graph (Fig. 3) with all discovered services at each node. Our solution evaluates first the different dependencies at each hop, as well as the input and output attributes among ASs, and concatenates those that could be executed within a node to satisfy a communication goal. Then, the best branch of services is selected, and the final CS is generated for each node. Depending on the level of granularity in service definition and the grade of accuracy when composing a service, service composition can be based on selecting predefined service compositions (for example, manually configured or calculated once and stored in specific repositories) or fully calculated considering all the possible combinations, thus obtaining the best possible solution available. Ideally, it would be necessary to find a tradeoff to find a feasible solution in a reasonable time. In completely predefined service compositions, the path should be known beforehand. Additionally, composition can be predefined only at some levels. For example, if a previously used link no longer exists but an equivalent link is found and its use does not change the characteristics of the communication, the same WF can be reused, especially in the case of edge nodes. Section IV describes the implementation of our initial proof of concept, for which we use predefined sets of compositions to enable fast service invocation. Then, we use service composition algorithms to observe the behavior of the system for more accurate compositions. This implementation of the service composer component applies A* algorithm [23] to search for the best solution, taking into account the nodes with the best score. Both in terms of computational time and memory space, A* has some scalability problems; thus, alternative algorithms implying tradeoffs will be analyzed in the future.

### D. Path Selection

We consider paths as sequences of nodes containing different ASs, which are capable of reaching the demanded end service from the RN. If a cost restriction specified by the RN is at stake, it is necessary to discard any path with a higher cost than the demanded restriction. When each node is scored, the path is selected using graph theory search strategies to determine the most cost-effective cost path or, in this case, branch. Depending on the preferences of the requester, the selected path can offer, for example, the best tradeoff between different parameters. For instance, selection could be based on the lowest delay path, the lowest cost per transmitted bit, or a tradeoff between both criteria by means of a weighted scoring function.

## IV. Enabling FI Service Provisioning

With the proliferation of multimedia devices, users demand

more audiovisual content. It is expected that by 2013, the sum of all forms of video will exceed 90% of the global consumer IP traffic [24]. Nowadays, Internet users are consistently demanding high-quality audiovisual services in a context in which network and service providers are still compelled to troubleshoot current technological limitations to provide optimized services and improve the QoE for users.

The main goal of the proposed solution is to enable FI service provisioning that meets QoS (for example, bandwidth, delay, and jitter) and QoE requirements. Thus, it takes into account requester needs and context information, including network, device, and user features. We show the main features and benefits of this approach by describing a challenging case of multimedia use and how to provide inherent adapted communications. Service and content adaptation is an extremely important issue for multimedia communications, especially when it comes to the distribution of audiovisual content in heterogeneous and dynamic networks, owing to the strong requirements they present in terms of bandwidth, delay, losses, device capabilities, and so on. To provide the best QoE to users, QoS needs to be guaranteed, while systems must react to dynamical changes in the network. However, this framework is designed not only to meet QoS and QoE (such as perceived video quality in a streaming communication) requirements in the provisioning of advanced multimedia services but to do so in an efficient and transparent manner.

Imagine a user ($U_a$) who wants to watch a film (F) online from the sofa. $U_a$ accesses the network using a tablet device supporting the following video codecs: MPEG4, MPEG2, and WMV. At home, $U_a$ uses WLAN 802.11g technology to access the Internet. Then, the user subscribes to an xDSL line (25 Mbps DL and 10 Mbps UL). This would be a basic specification of the context of $U_a$. In the network, there are four different streaming services available: service A ($S_a$), service B ($S_b$), service C ($S_c$), and service D ($S_d$). We assume that these services are placed in different ESNs, named $N_a$, $N_b$, $N_c$, and $N_d$, respectively. These are candidate service providers for $U_a$, as they can offer the service that the user is asking for. Table 1 summarizes the available ASs and AMs on these nodes. For the sake of clarity in this use case, we assume that each node (including INs) knows which services can provide. Thus, each node has a local repository with this information.

As the first proof of concept, a scenario without infrastructure support is considered. We consider this elementary scenario to test it without external support to store information. Note that we are composing services not only at application level but also at network level in a clean-slate manner, avoiding layered rigidities and using network-level services on demand. However, in future work, an approach with a distributed global directory will be undertaken. This

**Table 1.** ASs and corresponding AMs supported by each ESN providing streaming service.

| $N_a$, $N_b$, $N_c$, $N_d$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| AS | | | | AM | | | |
| Data_Tx | | | | tx_rx | | | |
| Data_Rx | | | | tx_rx | | | |
| Data_Fwd | | | | fifo, priority | | | |
| Seq | | | | Incremental, temporal | | | |
| ACK | | | | Ack, sack | | | |
| Rext | | | | retx | | | |
| Framing | | | | bit_oriented, byte_oriented | | | |
| MAC | | | | csma/cd | | | |
| Error_Detection & Handling | | | | Parity, crc | | | |
| $N_a$ | | $N_b$ | | $N_c$ | | $N_d$ | |
| AS | AM | AS | AM | AS | AM | AS | AM |
| Video coding | MPEG-1 MPEG-2 MPEG-4 VCx WMV FLV | Video coding | WMV 3GP MOV FLV | Video coding | MPEG-4 FLV | Video coding | MPEG-2 MPEG-4 |
| Audio coding | AAC MP3 | Audio coding | AAC MP3 | Audio coding | WAV AAC MP3 | Audio coding | WMA MP3 |

approach will require the support of specific nodes to improve service and context information searches but will be compatible with the negotiation protocol to assure interconnectivity between heterogeneous networks.

To get the film, $U_a$ sends a Creq to its neighbors, which is propagated hop by hop. Each node evaluates whether it can provide the services requested under the desired conditions, that is, in this case, the QoS parameters that the user requires. At this point, each node applies the logic presented in Fig. 3. $S_a$ cannot be reached because the INs in path 1 (P1) make the path unsuitable for the communication, as they introduce too much delay. $S_b$, $S_c$, and $S_d$ can be reached through path 2 (P2), path 3 (P3), and path 4 (P4), respectively. Nodes from P2, P3, and P4 build a Cresp that goes back to the requester through the reverse path from $S_b$, $S_c$, and $S_d$, respectively. Once this stage has been accomplished, the RN evaluates each received Cresp. This is done by applying a service composition algorithm, such as the one proposed in subsection III.3. Remember that this algorithm allows us to play with all the possible combinations of the available services.

In a practical scenario, a more scalable algorithm achieving a tradeoff between response time and accuracy of the resulting
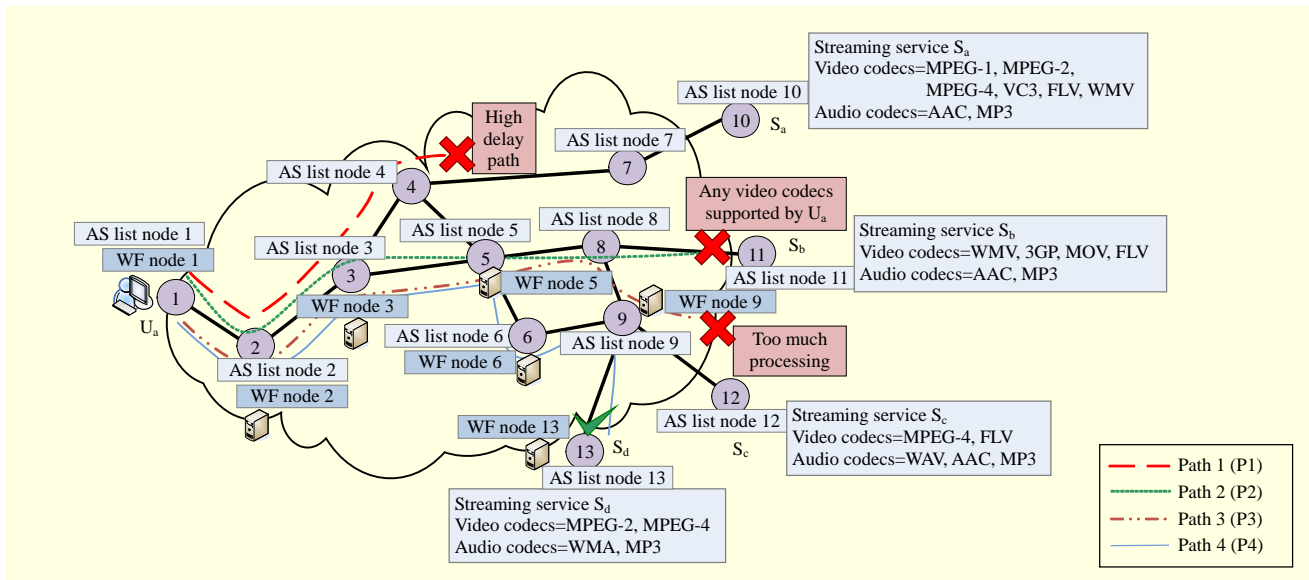
Fig. 4. Adapted multimedia communication use case.

solution would be used. In this case, $U_a$ can select $S_b$ if $U_a$ wants the offered service with the lowest delay. Regarding coding compression, $S_c$ is the best service. However, $S_d$ ranks the best if considering a tradeoff between energy consumption (it uses a less demanding codec) and audiovisual quality (measured using objective [peak signal-to-noise ratio] and subjective [mean opinion score] metrics). Finally, $U_a$ opts for $S_d$ because it meets the visualization preferences of $U_a$ and makes better use of the life of the battery in comparison with the previously tracked down services. As an example, MPEG-2, which is the video codec (AM) available in $S_d$, requires eight times less the processing power for encoding and three times less the processing power for decoding in comparison with H264/AVC [25] (Fig. 4).

Once services are selected for each node and WFs are created, $U_a$ sends a Call message through the selected path. This message is the last message defined in the basic negotiation protocol, and its main goal is to allocate the services. The total time to consume a service, that is, the time spanning from the moment a service is demanded to the moment data is received, can be expressed as follows:

$$T_{e\text{-}e} = T_{comp} + (N-1) \cdot (3 \cdot T_{prop} + T_{fwd} + T_{resp}), \quad (5)$$

where $T_{e\text{-}e}$ represents the total end-to-end time required. $T_{comp}$ is the time required for AS selection, composition, path selection, and, consequently, per node WF creation. $T_{prop}$ is the propagation delay between two consecutive nodes (it is multiplied by 3 because of the three-way handshake used). In the depicted scenario, we assume a constant time for each link. $T_{fwd}$ is the time required to decide whether a node can provide the demanded end service, $T_{resp}$ is the time needed to write the

Cresp parameters offered by each node, and $N$ represents the total number of nodes in the scenario.

Using the proposed protocol, services can be discovered whilst evaluating context conditions hop by hop to guarantee the required QoS. To achieve this, the discovery process includes routing to the ESN, which is done on a per-hop CBR basis during the establishment of the end-to-end path. Thus, the routing is undertaken considering the context of the network and available services.

This use case shows a network with homogeneous INs that perform the same operations. However, the network could be composed of different network nodes with different capabilities and different services. Service composition and allocation specify which services should be placed and executed at each node to obtain the best possible communication. Consequently, a node with Wi-Fi and wired (for example, copper providing xDSL access) interfaces can use different ASs, depending on the context. An example would be to use congestion control functionalities in the wired interface whilst avoiding them in the Wi-Fi interface. This is possible thanks to the RBA-based decomposition and the SOA-based composition of functionalities, which allow the modulation of segments of the network and the placement of services when and where needed.

### 1. Testbed

This subsection describes the proof-of-concept implementation of the proposed solution within the TARIFA project [26] and some preliminary results obtained when a context-aware service search into a local network is performed.

Regarding the generated code, it is migrated from the

Table 2. Detail of the size of generated code.

| Name | Storage size (bytes) | RAM (bytes) |
|---|---|---|
| Base code | 1,194,204 | 116,180 |
| Service composition & allocation | 217,341 | 61,896 |
| Search service engine | 58,838 | 18,913 |
| Constraint-based routing | 75,681 | 32,697 |
| Total | 1,546,064 | 229,686 |

Table 3. Scenario specification (S1, S2, S3, S4).

| Goals | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| Data_transmission | RN | RN | RN | RN |
| Data_forwarding | IN | IN | IN | IN |
| Data_reception | ESN | ESN | ESN | ESN |
| Decoding | RN | RN | RN | RN |
| Encoding | ESN | ESN | ESN | ESN |
| Security | | RN, ESN | | RN, ESN |
| Reliability | | | RN, ESN | RN, ESN |



Fig. 5. End-to-end time and composition time.

original modules specified in [4], developed in a system on chip (SoC) CC2430 [27] from a Texas Instruments platform. Moreover, code is adapted to run in a Linux-based desktop computer to test the proposed solution in a more complex network. Finally, we extend it with new modules (Table 2). The whole development requires 1.5 MB of total memory space and 229 KB of RAM, which is very low for the core architecture and allows us to run it in very small devices (for example, sensors).

In this testbed (Fig. 4), a total number of 13 nodes are used (1 RN, 4 ESN, 8 IN). All of them are Intel Pentium 4 540 (32 GHz, 1,024 KB L2 Cache) with 512 MB of RAM and the 32-bit OS Ubuntu 11.04. All are connected using several network interfaces configured in full-duplex 100-Mb/s Ethernet mode.

## 2. Results

The time required to establish an end-to-end communication ($T_{e-e}$) and the resource consumption of the process are measured. Concretely, different Creq messages, asking for different requirements and network functionalities, are tested. Table 3 specifies the high-level communication goals for each test. In practice, these goals are associated with different combinations of ASs, which can in turn be implemented by different AMs. As an example, imagine we have an encoding goal that can be achieved through *video_coding* and *audio_coding* ASs. Each AS can then be provided by such an AM as MPEG-1, MPEG-2, MP3, or WMA.
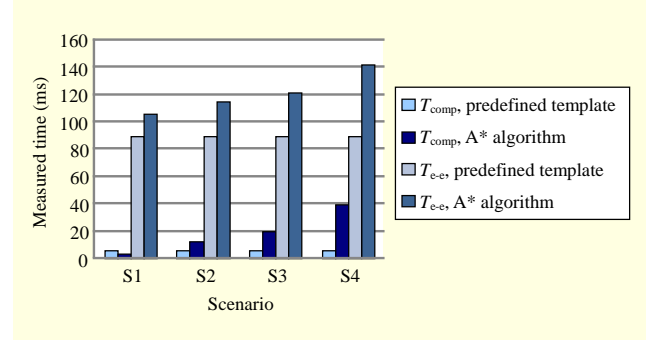
Regarding performance parameters, the average total consumption during the process of negotiation is RN 13% CPU and 224.3 KB RAM, INs 5% CPU and 113.4 KB RAM and, finally, ESN 9% CPU and 192.3 KB RAM.

The time required to negotiate the end-to-end communication ($T_{e-e}$) is shown in Fig. 5. Concretely, we show the results for the longest path (P4) of our testbed. We specify the time required to start a communication using two different approaches for service composition: (a) using A* as an exhaustive service composition algorithm and (b) using predefined templates specifying the services offered by a node.

Note that in our tests, $T_{prop}$ is almost negligible, as we use dedicated links in a local testbed. $T_{fwd}$ is constant, as each involved node performs a lookup in its local database of ASs, and each AS is the same size.

Regarding $T_{resp}$, it is slightly different in the ESN than in the IN because it must insert more data into the Cresp (each node inserts information about its ASs, AMs, and QoS capabilities).

The gathered results are preliminary for the different scenarios introduced in Table 3. The most representative value is the time required to negotiate the services that will be used ($T_{e-e}$), and the most influential parameter is the composition time needed to decide which services to use ($T_{comp}$). Composition can be a very demanding process if full flexibility and the best possible solutions are required. Mostly, its value depends on the number of goals to successfully target and the number of ASs and AMs supported by a node. As specified in subsection III.3, the proposed composition algorithm must calculate all the possible combinations between ASs to select the best one. The more services there are, the more combinations must be calculated. For example, S3 is higher than S2 because the complexity of the rules that we use to calculate the reliability goal is higher than for security. In future work, some techniques that improve this process will be studied. Once a composition is performed, the resulting WF of services could be stored for future reuse so as to avoid calculating all the combinations of services again. Finally, note that in the presented prototype, monitoring functions are not

implemented. An efficient monitoring system that provides context information is especially important for the development of this solution. In the future, we expect to use specific monitoring mechanisms to obtain real context data.

## V. Conclusion and Future Work

This work proposed a clean-slate and service-oriented framework that focuses on service combination and adaptation to context conditions by means of service discovery and service composition. These processes are necessary to enable FI service provisioning in an adaptive manner, satisfying the specific QoS/QoE requirements demanded by users. Additionally, it makes efficient use of network resources. To achieve this, routing functions are integrated into a service discovery protocol that evaluates context conditions hop-by-hop when a communication is requested. We also proposed a service negotiation protocol that enables us to find and to compose services that meet requesters' requirements efficiently.

In addition, we provided the main details of a first implementation of the proposed solution and discussed the preliminary results. Moreover, the adoption of a service-oriented approach will allow us to introduce the proposed architecture gradually over current infrastructures (such as SaaS or IaaS approaches) while trying to extend the paradigm to lower layers in the future to achieve a real disruptive approach. The composition of basic network-level services calls for a clean-slate approach to the Internet, while the composition of higher level (transport and application) services prompts for an evolutionary approach. Moreover, this architecture would be the first clean-slate deployment completely aligned with the current work being done by the ISO/IEC JTC1/SC6/WG7 Future Network working group.

In future work, the authors will focus on analyzing and testing scalability issues from two major perspectives. We will aim, firstly, to provide realistic and larger scenarios and, secondly, to find composition mechanisms that present satisfactory tradeoffs between performance and response time in different environments. Message format and semantics are also open issues that need to be explored to reduce the overhead of the discovery protocol. Additionally, the study of reservation mechanisms and their implications when services are unavailable during the allocation phase will be required in future work.

## References

[1] M.P. Papazoglou et al., "Service-Oriented Computing: State of the Art and Research Challenges," *Computer*, vol. 40, issue 11, Nov. 2007, pp. 38-45.

[2] R. Braden, T. Faber, and M. Handley, "From Protocol Stack to Protocol Heap: Role-Based Architecture," *ACM SIGCOMM Computer Commun. Rev.*, vol. 33, issue 1, Jan. 2003, pp. 17-22.

[3] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead," *IEEE Commun. Mag.*, vol. 43, issue 12, Dec. 2005, pp. 112-119.

[4] X. Sanchez-Loro et al., "Can Future Internet Be Based on Constrained Networks Design Principles?" *Computer Netw.* vol. 55, issue 4, Mar. 2010, pp. 893-909.

[5] M. Bell, *Service-Oriented Modeling: Service Analysis*, *Design*, *and Architecture*, Hoboken, NJ: John Wiley & Sons, Inc., 2008.

[6] J. Domingue, D. Fensel, and R. González-Cabero, "SOA4All, Enabling the SOA Revolution on a World Wide Scale," *IEEE Int. Conf. Semantic Computing*, Aug. 2008, pp. 530-537.

[7] A. Karouia et al., "SOA-Based Approach for the Design of the Future Internet," *CNSR*, Montreal, Canada, May 2010, pp. 361-368.

[8] J.D. Touch, Y.S. Wang, and V. Pingali, "A Recursive Network Architecture," *ISI*, *Tech. Rep.*, No. 2006-626, Oct. 2006.

[9] R. Dutta et al., "The SILO Architecture for Services Integration, Control and Optimization for the Future Internet," *IEEE Int. Conf. Commun. ICC*, Aug. 2007, pp. 1899-1904.

[10] RINA project, "Recursive InterNetwork Architecture," 2011. Available: http://csr.bu.edu/rina/index.html

[11] B. Reuther and P. Müller, "Future Internet Architecture – A Service Oriented Approach," *Inf. Technol.*, vol. 50, no. 6, Apr. 2008, pp. 383-389.

[12] R. Ahmed et al., "Resource and Service Discovery in Large-Scale Multi-domain Networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 4, 2007, Feb. 2008, pp. 2-30.

[13] S. Shanbhag and T. Wolf, "Automated Composition of Data-Path Functionality in the Future Internet," *IEEE Netw.*, vol. 25, no. 6, Nov. 2011, pp. 8-14.

[14] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Commun. Rev.*, vol. 38, no. 2, Apr. 2008, pp. 69-74.

[15] ISO/IEC JTC1/SC6/WG7, "ISO/IEC TR 29181," 2012. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57480

[16] F. Kuipers et al., "An Overview of Constraint-Based Path Selection Algorithms for QoS Routing," *IEEE Commun. Mag.* vol. 40, issue 12, Dec. 2002, pp. 50-55.

[17] D.D. Clark et al., "Tussle in Cyberspace: Defining Tomorrow's Internet," *IEEE/ACM Trans. Netw.*, vol. 13, issue 3, June 2005, pp. 462-475.

[18] X. Sanchez-Loro, A.J. Gonzalez, R. Martin de Pozuelo, "A Semantic Context-Aware Network Architecture," *Future Netw. Mobile Summit*, June 2010, pp. 1-9.

[19] J. Song, H.K. Pung, and L. Jacob, "A Multi-constrained Distributed QoS Routing Algorithm," *IEEE Int. Conf. Netw.*, Sept. 2000, pp. 165-171.

[20] G. Mooney, "Evaluating Compact Routing Algorithms on Real World Networks," project dissertation, University of Glasgow, Glasgow, Scotland, UK, June 2010.

[21] L.J. Cowen, "Compact Routing with Minimum Stretch," *J. Algorithms*, vol. 38, no. 1, Apr. 2001, pp. 170-183.

[22] A.J. Gonzalez et al., "Context-Aware Multimedia Service Composition Using Quality Assessment," *IEEE Int. Conf. Multimedia Expo*, July 2011, pp. 1-6.

[23] P.E. Hart, N.J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.* vol. 4, issue 2, July 1968, pp. 100-107.

[24] Cisco Systems, "Entering the Zettabyte Era," *Cisco Visual Netw. Index (VNI)*, May 2012. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyper connectivity_WP.pdf

[25] Haivision, "MPEG-4 AVC (H.264) and Why, Only Now, It Can Save 60% of the Network Video Bandwidth and Storage Requirements," white paper.

[26] TARIFA project, "The Atomic Redisign of the Internet Future Architecture," 2011. Available: http://www.i2cat.net/en/projecte/tarifa-1

[27] Texas Instruments, "True System-on-Chip Solution for 2.4 GHz IEEE 802.15. 4/ZigBee," 2011. Available: http://www.ti.com/lit/ds/symlink/cc2430.pdf

**Alberto J. Gonzalez** received his BSc in telecommunication engineering, his MSc (hons.) in telecommunication engineering & management, his MSc in telematics, and his PhD (cum laude) in telecommunications in 2006, 2007, 2009, and 2012, respectively, from the Universitat Politècnica de Catalunya (UPC – BarcelonaTech), Barcelona, Spain. He joined the Department of Telematics Engineering in 2006 as a research fellow. Since 2004, he has been collaborating with the i2CAT Foundation, Barcelona, Spain, in research and project management tasks. He has been a participant in Euro-NF since 2008. Since 2010, he has been participating in the standardization group ISO/IEC JTC 1/SC 6, "Future Network," and, since 2011, has been the editor of part 7, "Service Composition." His research interests include multimedia content adaptation, heterogeneous environments, peer-to-peer, video transmission, error resilience, video coding, service composition, and future networks.
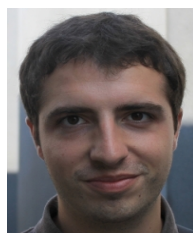
**Ramon Martin de Pozuelo** received his BSc and MSc in telecommunications engineering from Universitat Ramon Llull (URL), La Salle campus, Barcelona, Spain, in 2006 and 2009, respectively. He is currently pursuing his PhD at the Center of Technologies for Media (CTMedia) at URL, where he has been participating in the area of digital television. He participated in some national and European publically funded research projects mainly related to scalable video streaming. Since 2010, he has been participating in the standardization group ISO/IEC JTC 1 SC 6, "Future Network," and, since 2011, has been the co-editor of part 7, "Service Composition." His research interests include context-aware service composition and media provisioning, especially over heterogeneous environments and future Internet architectures.

**Martin German** obtained his BSc in computer science in 2007 from the University of the Republic (Uruguay), Montevideo, Uruguay. In 2011, he received his MSc in computer architecture, networks, and systems from Universitat Politècnica de Catalunya (UPC – BarcelonaTech), Barcelona, Spain. Since 2010, he has been working for the i2CAT Foundation as a research fellow, and, since the beginning of 2011, he has been a PhD student in the Architecture Department at Barcelona Tech (UPC). He is actively involved in the IETF community, particularly with the Path Computation Element WG, for which he has contributed to the improvement of RFCs and their implementation. His research interests lie in the design of scalable routing paradigms for the Internet, novel routing/switching paradigms, network performance, and traffic engineering.

**Jesus Alcober** is an associate professor at the Universitat Politècnica de Catalunya (UPC – BarcelonaTech), Barcelona, Spain. He received his MS in telecommunications in 1993 and his PhD in telecommunications from the UPC in 1997. He is currently the director of the Audiovisual Unit in the i2CAT Foundation and the president of the Spanish subcommitte of ISO/IEC JTC 1/SC 6. His research interests focus on future network architectures to provide advanced media and social services.

**Francesc Pinyol** received his MSc in telecommunications engineering from Universitat Politècnica de Catalunya (UPC – BarcelonaTech), Barcelona, Spain, in 1993 and his PhD from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1998. Currently, he is working as the scientific director at the Center of Technologies for Media (CTMedia), Universitat Ramon Llull (URL), La Salle campus, Barcelona, Spain. He is actively involved in publically funded European and national research projects. His current research interests include media coding and media transmission.