J Internet Serv Appl (2011) 2:69–79
DOI 10.1007/s13174-011-0026-y

SI: FUTURE NET SERVICE MODELS & DESIGNS

Automated transport service management in the future Internet: concepts and operations

Franco Callegati · Aldo Campi · Walter Cerroni

Received: 1 February 2011 / Accepted: 18 May 2011 / Published online: 15 July 2011

Abstract This paper describes a possible approach to provide advanced communication services for the Future Internet in an automated way. This is achieved by introducing a new *Cognitive Transport Service* layer on top of the traditional transport layer, which is used by the network to automatically negotiate the required transport service with the applications. Experimental validation of the proposed solution is discussed with the help of two case studies: bandwidth reservation on a secure channel and dynamic bandwidth negotiation for variable-quality video streaming.

Keywords Cognitive networks · Transport service · SIP · NRDL · VPN · HD video

1 Introduction

The Internet offers an end-to-end transport service capable of interconnecting different and heterogeneous applications on top of a flat network layer that focuses on packet routing. Intelligence and trust in the use of the transport service are typically delegated to the end-user applications. Today it is widely agreed that the effectiveness of this approach is vanishing, due to the increased complexity of protocols and devices, which makes prohibitive for the average user to directly and effectively manage his/her communication services, and to the new network functions and sub-blocks (NAT, firewalls and application-level gateways, above all), which impact on the degree of "transparency" of the transport service [1].

The Future Internet should provide built-in functionalities capable of satisfying the final user communication requests well beyond the pure data transport. In the Next Generation Network (NGN) specifications this issue is addressed outlining a separation between the *service stratum* responsible for network service provisioning and the *transport stratum* responsible for IP-based packet forwarding [2].

In this work we intend to show that, exploiting technologies existing in the current Internet, it is possible to implement a new *Cognitive Transport Service* plane capable of self-configuring and self-organizing according to the application requirements. The final vision is to enable a new generation of Internet applications capable of triggering at once the search for the required resources and the setup of the necessary connectivity services across the transport networks, performing "on-demand" communication service provisioning. To enable such a scenario a proper signaling layer is necessary, responsible for:

– making the applications capable of exchanging semantically-rich messages with the network to issue service requests for the negotiation and reservation of the needed resources;
– defining a "technology-independent" methodology to provision network resources by mapping the service requests into network control plane directives.

This work was performed in the framework of the EU-funded ICT-BONE project [3], with the aim to demonstrate that it is already possible to implement service management capabilities on top of a general high-capacity transport network. This paper builds on concepts previously used to investigate with reference to Grid services on optical network,

F. Callegati (✉) · A. Campi · W. Cerroni
D.E.I.S., University of Bologna, Via Venezia 52, 47521 Cesena (FC), Italy
e-mail: franco.callegati@unibo.it

A. Campi
e-mail: aldo.campi@unibo.it

W. Cerroni
e-mail: walter.cerroni@unibo.it

using a fast optical switching technology to support short-lived consumer grid applications [4, 5] interacting with OBS control plane [6, 7], as well as with reference to the implementation of multimedia services with QoS guarantees in GMPLS-controlled networks [8, 9].

The paper is organized as follows. In Sect. 2 the general network model including the Cognitive Transport Service layer is introduced and discussed. In Sects. 3 and 4 the operations of the Cognitive Transport Service layer are described with reference to the service activation. In Sect. 5 the effectiveness of the current proposal is shown with reference to two case studies. Finally, some conclusions are drawn in Sect. 6.

2 Cognitive transport service layer

The proposal presented in this work is to add a *Cognitive Transport Service* (CTS) layer on top of the traditional TCP/IP-based transport layer. The CTS layer has the goal to support the applications to build complex communication services in a general way. The CTS layer will allow applications to manage communication services by integrating:

- *network resource virtualization*, which represents physical objects, equipment and services as logical resources;
- *service composition and orchestration*, where complex network services can be split up into simple network functions and operations.

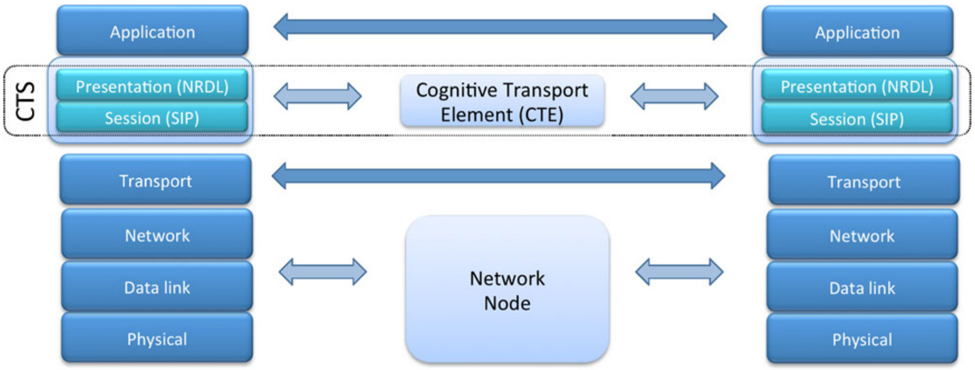
In the current Internet the packet routing service is always active and completely de-coupled from the transport layer; it is well known that TCP or UDP can exchange traffic as long as IP is on, with absolutely no control on what happens to the IP packets once in the network. The transport has no capability to interact with the network services unless there is an underlying pre-configured flow or path policy (like in static optical networks and/or in GMPLS networks). Enriching the transport with a CTS provides the means for orchestrating transport functions available in a given network section. As a consequence, the CTS achieves a number of cognitive operations to fulfill the user request in terms of communication service.

3 Implementation of the cognitive transport service layer

From our previous experience we believe that the CTS layer should be implemented by de-coupling the signaling protocol, which defines the syntax used for service negotiation, from the descriptive protocol providing the semantic to describe the service features, e.g., which resources must be use, which requirements must be satisfied, etc. As such, the CTS layer is divided into two sub-layers (see Fig. 1) which have been implemented as follows:

- The *Session Layer* has been implemented with the Session Initiation Protocol [10] which is a widely used and known signaling protocol for VoIP applications. SIP deals with session-oriented mechanisms, regardless of the scope(s) of the session. It specifies the message flows required to initiate, modify and terminate sessions. In other words, SIP does not provide services but provides primitives that can be used to implement services on top of sessions.
- The *Presentation Layer* is related to the network services to be implement. A few descriptive languages tailored to specific scenarios were proposed in the past [11–14]. We proposed a more general one called Network Resource Description Language (NRDL) [15], which is designed for a network service composition context based on an ontology model. NRDL is able to represent network resources as well as the attributes of a given communication and it is neutral to the session protocol used as driver. Therefore, it is not bound to SIP encapsulation. The goal of NRDL is to make the applications capable of describing and communicating general requirements for a given communication service, which the network will interpret and translate into signaling primitives for the infrastructure (e.g. create connection, reserve bandwidth, etc.).

Fig. 1 Cognitive transport service layer-enabled network model



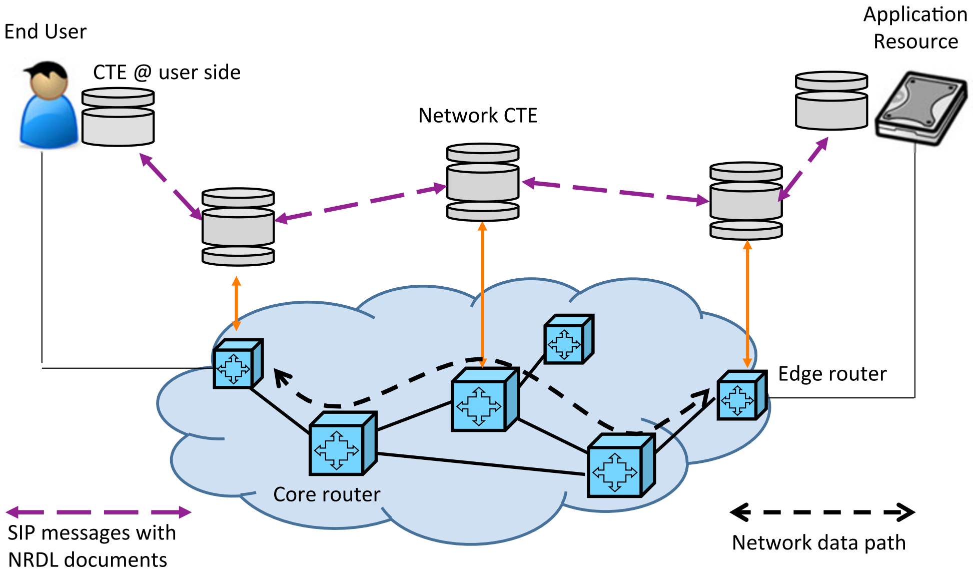


Fig. 2 Example implementation of the cognitive transport service layer by means of cognitive transport elements

3.1 The cognitive transport element

The CTS layer described above is implemented with a new network sub-block, called hereafter *Cognitive Transport Element* (CTE), located at the end-user equipment as well as at some (or all) network nodes as shown in Fig. 2. The CTE is equipped with:

- a SIP proxy with standard SIP communication facilities, responsible for managing sessions, finding the next hop and forwarding any SIP message to the destination;
- an NRDL parser, which reads and understands the presentation directives embedded in the SIP messages and relies on a relational database where it is possible to store information regarding application or network resources as well as service composition facilities;
- a network-dependent module to interact with the network control plane (or with the network node configuration interface) in order to properly configure the network according to the communication service requested.

The end-user CTEs request communication services whereas the CTEs along the network path are responsible for locating and activating the requested resources. From the application point of view the process to send and receive data is the same as in the current Internet but with the extended feature of requesting a predefined transport service. The request is managed by the end-user CTEs. The sender-side CTE prepares an NRDL document including the network service request according to the user/application needs. Then it initiates a SIP session with the receiver-side CTE via the SIP proxy embedded in the network CTE coupled with the ingress edge node and sends the NRDL document through the network carried by the SIP messages related to the session opening. The receiver-side CTE confirms the opening of the session and, by doing so, it activates the network configuration required by the requested communication service.

Owing to the native capability of SIP to intercept and read messages at any proxy along the path from source to destination, the service-related specifications in the NRDL documents are read along the network and allow to manage/activate the network transport functions (e.g. endpoint identification and congestion control). In the CTEs along the path the service requests are read, modified, if needed, and forwarded. By reading the service requests, the intermediate CTEs understand whether they have to interact with the network to support the incoming service request with a specific network configuration.

3.2 Resource management

Resource management, which is a crucial part of any cognitive system, can be implemented again by means of a session-oriented approach. SIP signaling supports resource management. Assuming network resources are divided into domains, the resource management operation follows in principles three different phases.

- Resource publication: each available resource must publish its own availability to the CTS. The publication phase allows the collection of network resource information by the CTEs.
- Resource discovery: it allows CTEs to find a set of available resources into the network including their location and current availability.
- Resource reservation: it allows the reservation of resources in the given domain.

Resources are managed by processors behaving as SIP User Agents, which are able to perform the typical signaling of

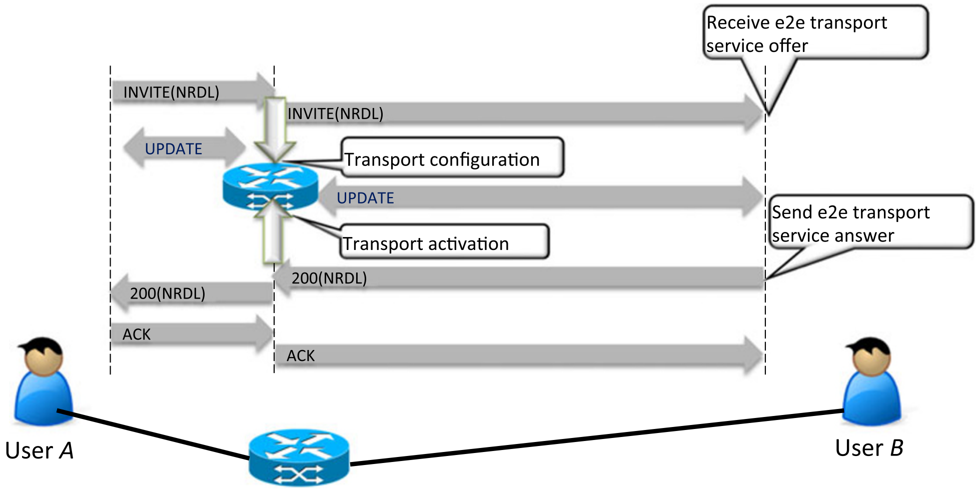


Fig. 3 Logical view of the network activation process

a standard SIP terminal. The SIP-oriented resource management approach was already experimented as reported in [16].

4 Cognitive transport service layer interaction

As described in the introduction, the goal of the CTS layer is to allow the automatic configuration of complex communication services that today require a series of actions by several different subjects. Owing to the NRDL multilayer descriptive characteristics, users may generate offers and provide answers regarding complex network services, such as tunneling, bandwidth reservation, segmentation, etc., while SIP, due to the session management capabilities, guarantees the communication consistency and delivery of such information.

The CTEs that implement the CTS layer process the information in the NRDL and use it to implement the end-to-end transport communication according to specific application requests. The functional schematic of such process is shown in Fig. 3, with reference to a simple configuration including a single network node for the sake of readability. The CTE at User *A* starts the session by sending a SIP INVITE message with a proper service request in the form of an NRDL document embedded in its body. The INVITE is processed by the intermediate CTE coupled with the network router. If some specific configuration is required in its network segment (e.g. bandwidth reservation, segmentation, etc.) the CTE interacts with the router and properly configures it. Then the INVITE with the related NRDL document is forwarded to the next hop (i.e. other network CTEs or the final receiver).

Once the NRDL document arrives at its final destination, the CTE at User *B* verifies the network services requested by the sender and accordingly prepares the answer again as an NRDL document. The answer works as an acknowledgment and describes the network services from the receiver point of view, adding or updating the service information (e.g. local port, certificate, etc.). The updated NRDL document is sent back allowing the network services activation in the opposite direction.

It is worth noting that several alternatives of interaction with the network are possible in this architecture. The intermediate CTE may

- start with full reservation of network resources, in case the network set-up time is considered a crucial parameter, knowing that those resources will have to be released if the communication request is not accepted by the callee;
- start with pre-reservation (or pre-configuration) of network resources (nodes) that will be confirmed only by the session establishment acknowledgment (i.e. 200 OK) on its way back to the INVITE source;
- exchange the current service status by means of SIP UPDATE messages. This is compliant with the standard SIP call flow where any modification to a non-confirmed dialog is notified by means of UPDATE messages.

Obviously, since network services can be composed of several steps, the information exchange may continue until the network services are completely established or finally canceled.

5 Experimental activities

An experimental test-bed has been set up to demonstrate the effectiveness of the proposed implementation of the CTS layer. In the test-bed, without lack of generality, three CTEs are put in place. The two CTEs co-located with the end users act either as a caller (at the client-side) or as a callee (at the server-side) and are linked to two routers (representing the access network), which are connected by a point-to-point link that can be assumed to represent the Internet "cloud."

The third CTE represents the CTS layer element within the "cloud" and may interact with the two edge routers to configure specific network directives.

The caller and callee CTEs are based on SIP User Agent (UA) modules implemented enhancing the PJSIP stack (http://www.pjsip.org) with the capability to send and receive messages with multi-part NRDL payloads. The network CTE is implemented with an OpenSIPS proxy (http://www.opensips.org) with two ad hoc external modules acting as NRDL parser and interface to network control plane, respectively. The NRDL parser is built using the open-source Raptor RDF parser library (http://librdf.org/raptor).

On this test-bed two case studies have been investigated and are reported here:

1. Two users requiring a secure communication channel with a predefined minimum bandwidth availability;
2. A user watching a standard-quality (SQ) video stream and interactively requesting the high-quality (HQ) version of the same content, triggering the required bandwidth re-negotiation.

In the view of the authors these case studies are rather general and may give a good understanding of the functionalities provided by the CTS layer.

In both examples the communication service requests from the users are mapped, due to the CTS signaling and actions, into a well-defined network configuration on the link between the two routers. What is peculiar of the examples is that both of them require communication services that cannot be straightforwardly achieved today without manual configuration, since they can be implemented only with the composition of specific setups addressing different aspects of the network transport:

- in the former case, a secure channel has to be set up between two Internet endpoints, e.g. by configuring a Virtual Private Network (VPN), while at the same time a given bandwidth reservation must be performed on the same network path;
- in the latter case, the bandwidth reserved on the path followed by the video stream has to be renegotiated and changed as soon as a change in the video quality is requested by the user.

The most important feature of the proposed architecture consists in its automatic capability of:

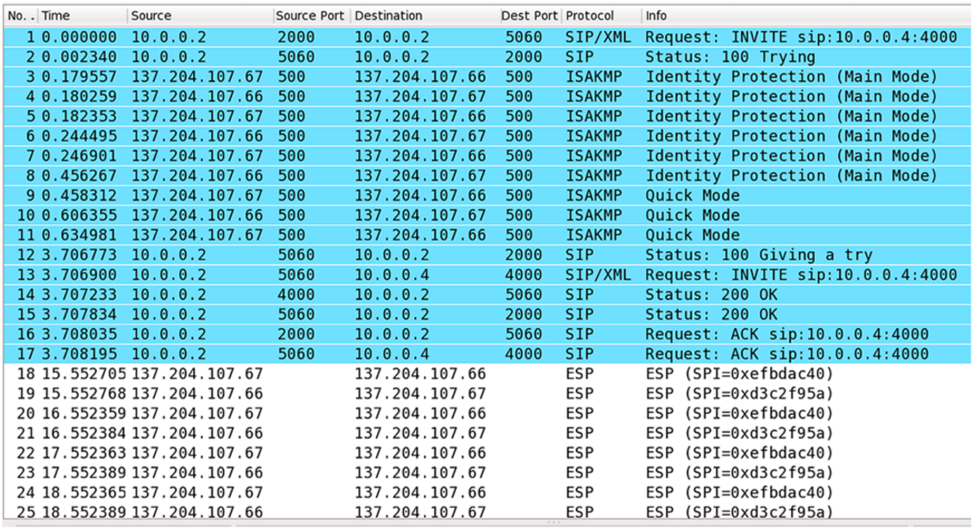
1. Receiving the service request in the CTS layer and understanding which network resources have to be orchestrated to satisfy the request;
2. Putting in place the resources making the required service available, by appropriately activating the pre-existing network resource management functions natively available in the network nodes;
3. Modifying the service instance according to changes in the network state and/or to specific user requests (as it happens in the HQ video request).

5.1 Case study 1

In this example we address the scenario of User A willing to set up a secure channel with a required minimum bandwidth with User B. The SIP UA of A sends the SIP INVITE message including the NRDL document with the information about the required communication service and negotiates the service setup both with the network and with User B.

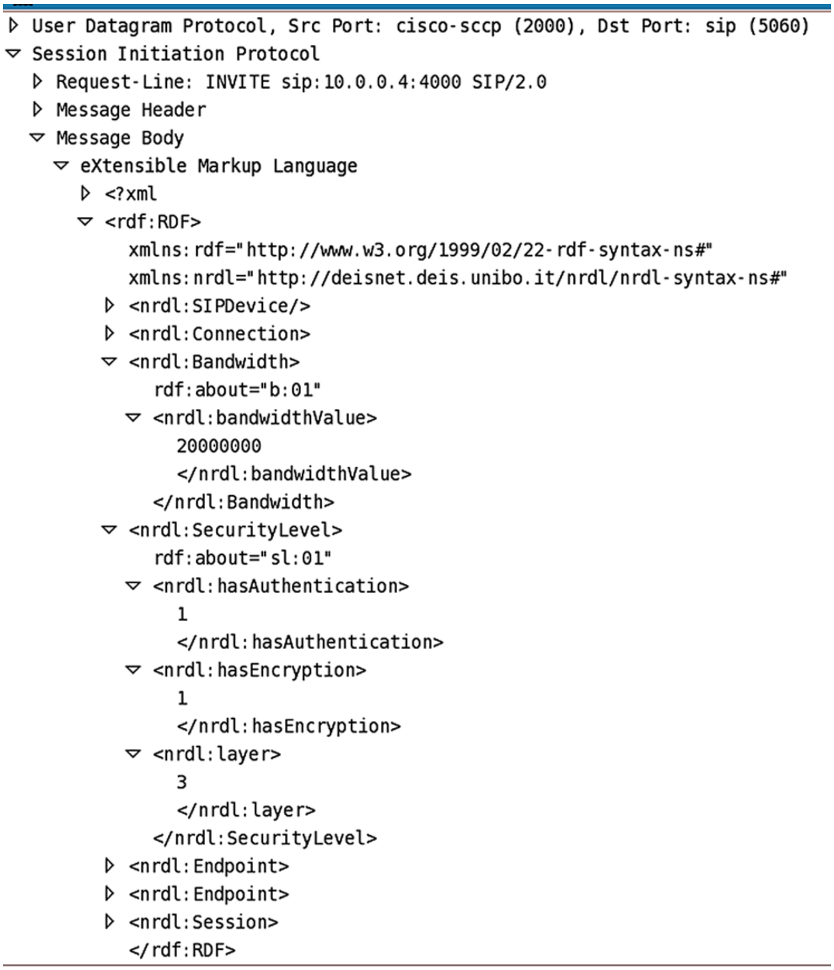
The successful negotiation carried out via SIP is shown in the capture of Fig. 4, where the initiator is identified by UDP port 2000, the receiver by UDP port 4000 and the network CTE by the default SIP UDP port 5060.

In this experiment a single SIP session has been used to negotiate network resources. The INVITE message sent by



```
No. . Time      Source         Source Port Destination    Dest Port Protocol Info
   1 0.000000  10.0.0.2       2000        10.0.0.2       5060      SIP/XML  Request: INVITE sip:10.0.0.4:4000
   2 0.002340  10.0.0.2       5060        10.0.0.2       2000      SIP      Status: 100 Trying
   3 0.179557  137.204.107.67 500         137.204.107.66 500       ISAKMP   Identity Protection (Main Mode)
   4 0.180259  137.204.107.66 500         137.204.107.67 500       ISAKMP   Identity Protection (Main Mode)
   5 0.182353  137.204.107.67 500         137.204.107.66 500       ISAKMP   Identity Protection (Main Mode)
   6 0.244495  137.204.107.66 500         137.204.107.67 500       ISAKMP   Identity Protection (Main Mode)
   7 0.246901  137.204.107.67 500         137.204.107.66 500       ISAKMP   Identity Protection (Main Mode)
   8 0.456267  137.204.107.66 500         137.204.107.67 500       ISAKMP   Identity Protection (Main Mode)
   9 0.458312  137.204.107.67 500         137.204.107.66 500       ISAKMP   Quick Mode
  10 0.606355  137.204.107.66 500         137.204.107.67 500       ISAKMP   Quick Mode
  11 0.634981  137.204.107.67 500         137.204.107.66 500       ISAKMP   Quick Mode
  12 3.706773  10.0.0.2       5060        10.0.0.2       2000      SIP      Status: 100 Giving a try
  13 3.706900  10.0.0.2       5060        10.0.0.4       4000      SIP/XML  Request: INVITE sip:10.0.0.4:4000
  14 3.707233  10.0.0.2       4000        10.0.0.2       5060      SIP      Status: 200 OK
  15 3.707834  10.0.0.2       5060        10.0.0.2       2000      SIP      Status: 200 OK
  16 3.708035  10.0.0.2       2000        10.0.0.2       5060      SIP      Request: ACK sip:10.0.0.4:4000
  17 3.708195  10.0.0.2       5060        10.0.0.4       4000      SIP      Request: ACK sip:10.0.0.4:4000
  18 15.552705 137.204.107.67             137.204.107.66           ESP      ESP (SPI=0xefbdac40)
  19 15.552768 137.204.107.66             137.204.107.67           ESP      ESP (SPI=0xd3c2f95a)
  20 16.552359 137.204.107.67             137.204.107.66           ESP      ESP (SPI=0xefbdac40)
  21 16.552384 137.204.107.66             137.204.107.67           ESP      ESP (SPI=0xd3c2f95a)
  22 17.552363 137.204.107.67             137.204.107.66           ESP      ESP (SPI=0xefbdac40)
  23 17.552389 137.204.107.66             137.204.107.67           ESP      ESP (SPI=0xd3c2f95a)
  24 18.552365 137.204.107.67             137.204.107.66           ESP      ESP (SPI=0xefbdac40)
  25 18.552389 137.204.107.66             137.204.107.67           ESP      ESP (SPI=0xd3c2f95a)
```

Fig. 4 Capture of the successful secure channel negotiation via SIP



```
▷ User Datagram Protocol, Src Port: cisco-sccp (2000), Dst Port: sip (5060)
▽ Session Initiation Protocol
  ▷ Request-Line: INVITE sip:10.0.0.4:4000 SIP/2.0
  ▷ Message Header
  ▽ Message Body
    ▽ eXtensible Markup Language
      ▷ <?xml
      ▽ <rdf:RDF>
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:nrdl="http://deisnet.deis.unibo.it/nrdl/nrdl-syntax-ns#"
        ▷ <nrdl:SIPDevice/>
        ▷ <nrdl:Connection>
        ▽ <nrdl:Bandwidth>
            rdf:about="b:01"
          ▽ <nrdl:bandwidthValue>
              20000000
              </nrdl:bandwidthValue>
            </nrdl:Bandwidth>
        ▽ <nrdl:SecurityLevel>
            rdf:about="sl:01"
          ▽ <nrdl:hasAuthentication>
              1
              </nrdl:hasAuthentication>
          ▽ <nrdl:hasEncryption>
              1
              </nrdl:hasEncryption>
          ▽ <nrdl:layer>
              3
              </nrdl:layer>
            </nrdl:SecurityLevel>
        ▷ <nrdl:Endpoint>
        ▷ <nrdl:Endpoint>
        ▷ <nrdl:Session>
          </rdf:RDF>
```

Fig. 5 SIP INVITE message encapsulating the NRDL resource request

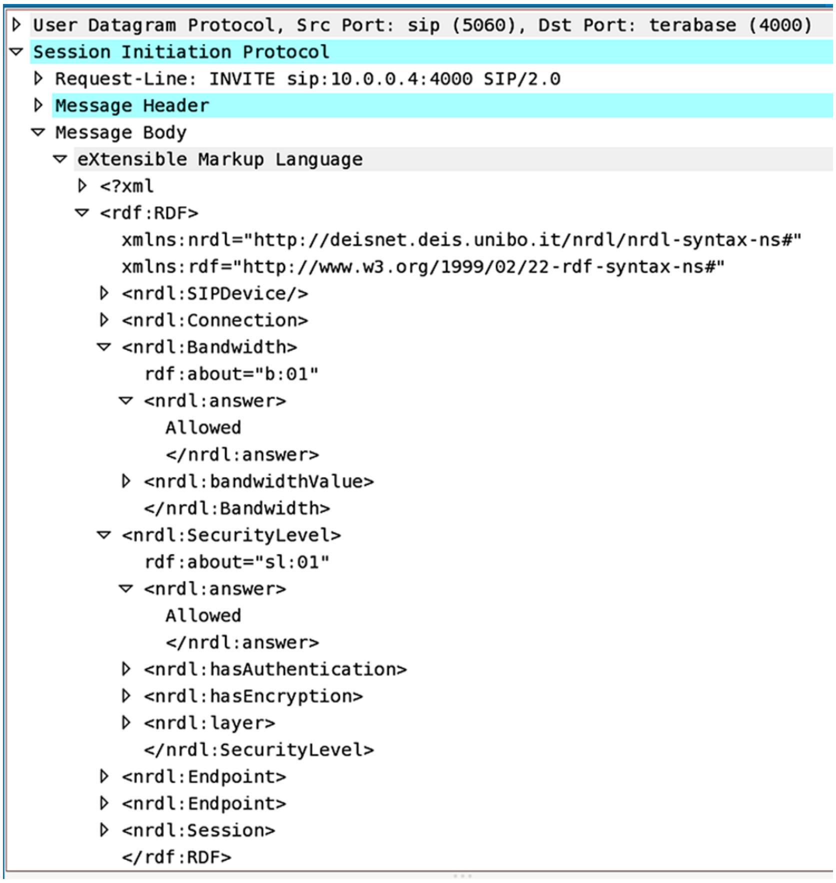
the initiator (packet no. 1) encapsulates the NRDL specification of the requested resources, including a minimum bandwidth of 20 Mbps and a secure channel with authentication and encryption enforced at layer 3, as shown in Fig. 5. The NRDL document used here is a rather simple example with just limited information for the sake of readability.

The requested features are processed by the network CTE and mapped into network-specific requirements. In particular, the bandwidth constraint is satisfied, since the two endpoints are connected through a 100 Mbps connection, whereas the secure channel specification is translated into the activation of an IPsec tunnel between the routers interconnecting the initiator and the receiver. The operation is successful, as demonstrated by the IPsec negotiation (packets nos. 3 to 11) and the resulting exchange of ESP encrypted packets when the actual data exchange begins (starting from packet no. 18).

The time difference between the two "100" messages shown in Fig. 4 (packets nos. 2 and 12) is due to the delay required by the network CTE to check for the availability of the requested resources and to allocate them. At this stage the network activation timing is not a crucial parameter to be tested. In fact, it is strictly technology-dependent.

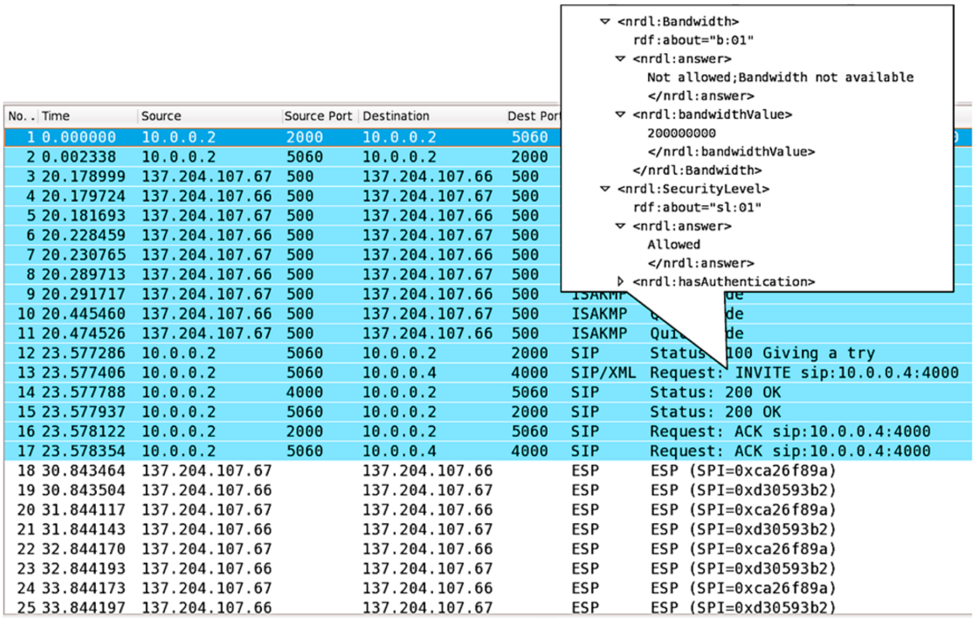
Once the resources have been successfully allocated, the network CTE updates the NRDL description accordingly, as shown in Fig. 6, where the "answer" field shows the "Allowed" value for both bandwidth and secure channel requirements. Finally, the network CTE forwards the modified INVITE message to the receiver, leading to the typical SIP conclusion of the session initiation phase (packets nos. 13 to 17).

The case of a partially successful negotiation due to lack of bandwidth has also been tested, as shown by the capture and the related INVITE message (packet no. 13) forwarded to the receiver displayed in Fig. 7. The minimum bandwidth required is 200 Mbps, which cannot be satisfied by the FastEthernet connection available between the two endpoints. However, the secure channel feature is still available, so this resource is provided. This implementation successfully proves the effectiveness of using a transport service activation process based on the CTS layer.



```
▷ User Datagram Protocol, Src Port: sip (5060), Dst Port: terabase (4000)
▽ Session Initiation Protocol
  ▷ Request-Line: INVITE sip:10.0.0.4:4000 SIP/2.0
  ▷ Message Header
  ▽ Message Body
    ▽ eXtensible Markup Language
      ▷ <?xml
      ▽ <rdf:RDF>
          xmlns:nrdl="http://deisnet.deis.unibo.it/nrdl/nrdl-syntax-ns#"
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        ▷ <nrdl:SIPDevice/>
        ▷ <nrdl:Connection>
        ▽ <nrdl:Bandwidth>
            rdf:about="b:01"
          ▽ <nrdl:answer>
              Allowed
              </nrdl:answer>
          ▷ <nrdl:bandwidthValue>
            </nrdl:Bandwidth>
        ▽ <nrdl:SecurityLevel>
            rdf:about="sl:01"
          ▽ <nrdl:answer>
              Allowed
              </nrdl:answer>
          ▷ <nrdl:hasAuthentication>
          ▷ <nrdl:hasEncryption>
          ▷ <nrdl:layer>
            </nrdl:SecurityLevel>
        ▷ <nrdl:Endpoint>
        ▷ <nrdl:Endpoint>
        ▷ <nrdl:Session>
          </rdf:RDF>
```

Fig. 6 Modified SIP INVITE message including the successful resource allocation



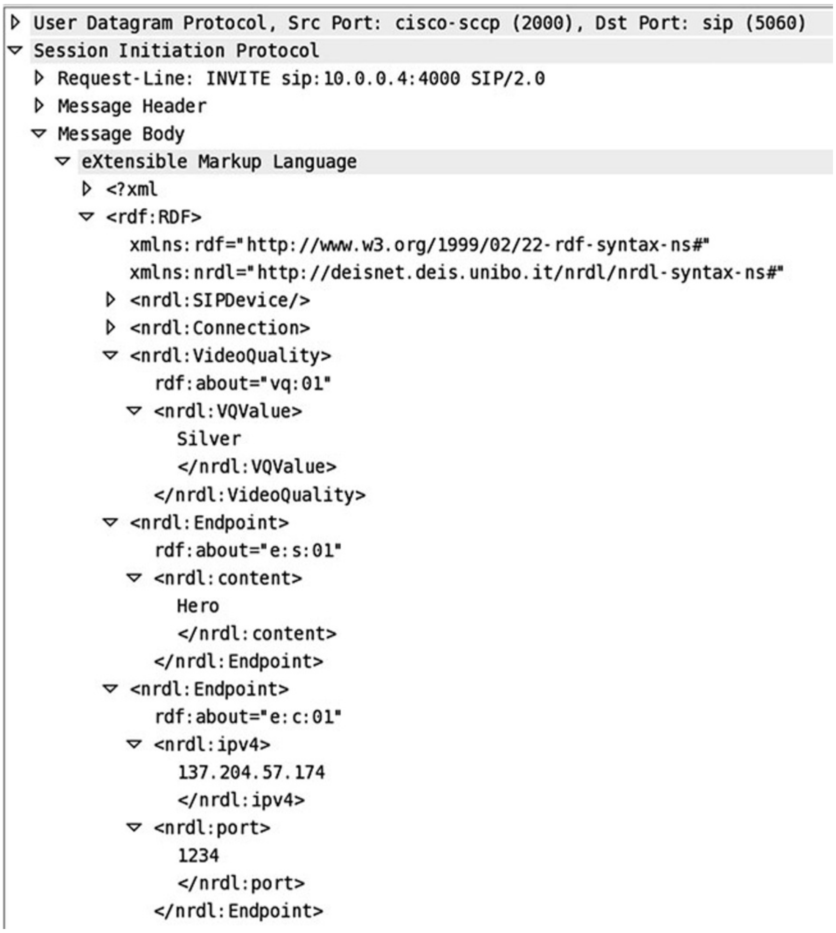
```
▽ <nrdl:Bandwidth>
    rdf:about="b:01"
  ▽ <nrdl:answer>
      Not allowed;Bandwidth not available
      </nrdl:answer>
  ▽ <nrdl:bandwidthValue>
      200000000
      </nrdl:bandwidthValue>
    </nrdl:Bandwidth>
▽ <nrdl:SecurityLevel>
    rdf:about="sl:01"
  ▽ <nrdl:answer>
      Allowed
      </nrdl:answer>
  ▷ <nrdl:hasAuthentication>
```

```
No. . Time      Source          Source Port Destination     Dest Port
 1 0.000000    10.0.0.2        2000        10.0.0.2        5060
 2 0.002338    10.0.0.2        5060        10.0.0.2        2000
 3 20.178999   137.204.107.67  500         137.204.107.66  500
 4 20.179724   137.204.107.66  500         137.204.107.67  500
 5 20.181693   137.204.107.67  500         137.204.107.66  500
 6 20.228459   137.204.107.66  500         137.204.107.67  500
 7 20.230765   137.204.107.67  500         137.204.107.66  500
 8 20.289713   137.204.107.66  500         137.204.107.67  500
 9 20.291717   137.204.107.67  500         137.204.107.66  500   ISAKMP      de
10 20.445460   137.204.107.66  500         137.204.107.67  500   ISAKMP   Q  de
11 20.474526   137.204.107.67  500         137.204.107.66  500   ISAKMP   Qui de
12 23.577286   10.0.0.2        5060        10.0.0.2        2000  SIP      Status 100 Giving a try
13 23.577406   10.0.0.2        5060        10.0.0.4        4000  SIP/XML  Request: INVITE sip:10.0.0.4:4000
14 23.577788   10.0.0.2        4000        10.0.0.2        5060  SIP      Status: 200 OK
15 23.577937   10.0.0.2        5060        10.0.0.2        2000  SIP      Status: 200 OK
16 23.578122   10.0.0.2        2000        10.0.0.2        5060  SIP      Request: ACK sip:10.0.0.4:4000
17 23.578354   10.0.0.2        5060        10.0.0.4        4000  SIP      Request: ACK sip:10.0.0.4:4000
18 30.843464   137.204.107.67              137.204.107.66        ESP      ESP (SPI=0xca26f89a)
19 30.843504   137.204.107.66              137.204.107.67        ESP      ESP (SPI=0xd30593b2)
20 31.844117   137.204.107.67              137.204.107.66        ESP      ESP (SPI=0xca26f89a)
21 31.844143   137.204.107.66              137.204.107.67        ESP      ESP (SPI=0xd30593b2)
22 32.844170   137.204.107.67              137.204.107.66        ESP      ESP (SPI=0xca26f89a)
23 32.844193   137.204.107.66              137.204.107.67        ESP      ESP (SPI=0xd30593b2)
24 33.844173   137.204.107.67              137.204.107.66        ESP      ESP (SPI=0xca26f89a)
25 33.844197   137.204.107.66              137.204.107.67        ESP      ESP (SPI=0xd30593b2)
```

Fig. 7 Capture of the partially successful negotiation via SIP and excerpt of the modified SIP INVITE message

5.2 Case study 2

In this second example a user wants to watch a video stream (based on UDP transport) with guaranteed quality. As a consequence, he/she demands for bandwidth reservation and, at some stage, decides that he/she wants to view the same video with a better definition. The request requires a change in the video source, but also a change in the bandwidth reservation within the network and these two actions must be automatically correlated.



```
▷ User Datagram Protocol, Src Port: cisco-sccp (2000), Dst Port: sip (5060)
▽ Session Initiation Protocol
  ▷ Request-Line: INVITE sip:10.0.0.4:4000 SIP/2.0
  ▷ Message Header
  ▽ Message Body
    ▽ eXtensible Markup Language
      ▷ <?xml
      ▽ <rdf:RDF>
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:nrdl="http://deisnet.deis.unibo.it/nrdl/nrdl-syntax-ns#"
        ▷ <nrdl:SIPDevice/>
        ▷ <nrdl:Connection>
        ▽ <nrdl:VideoQuality>
            rdf:about="vq:01"
          ▽ <nrdl:VQValue>
              Silver
              </nrdl:VQValue>
            </nrdl:VideoQuality>
        ▽ <nrdl:Endpoint>
            rdf:about="e:s:01"
          ▽ <nrdl:content>
              Hero
              </nrdl:content>
            </nrdl:Endpoint>
        ▽ <nrdl:Endpoint>
            rdf:about="e:c:01"
          ▽ <nrdl:ipv4>
              137.204.57.174
              </nrdl:ipv4>
          ▽ <nrdl:port>
              1234
              </nrdl:port>
            </nrdl:Endpoint>
```

Fig. 8 SIP INVITE message encapsulating the NRDL SQ video resource request

This case is representative of many application scenarios. For instance a user watching a program that is broadcast at different resolutions may decide to enhance size and quality of the video when something of particular interest is happening, or simply a change in the output device requires for a larger resolution (for instance a PC that is connected to a projector demands for a larger resolution when projecting the video at larger size than when watching it on the screen, etc.).

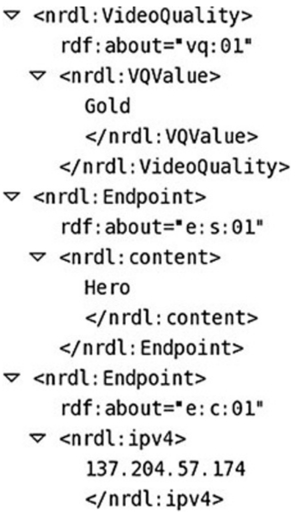
In this case User *A* starts the communication session similarly to the previous example. We do not report the complete negotiation capture here for space limitations. When the user selects the movie to watch, he/she starts asking for a “Silver” service level. The initiator then sends the INVITE message carrying the NRDL specification shown in Fig. 8 where the service level is expressed by the *VideoQuality* tag, the chosen content is represented by the movie title specified in the first *Endpoint* tag and the IP address and UDP port of the user’s video stream player are included in the second *Endpoint* tag.

It is worth to notice that the user does not need to specify to which video server he/she wants to connect, as it is assumed that the CTS layer itself implements a content distribution mechanism such that the network CTE automatically selects the best available server offering the required content and quality. The video quality (Silver) specified by the user is then used by the network CTE to properly configure the network equipment between source and destination, such that a minimum bandwidth is guaranteed. In the experiment shown here a standard-quality compressed VBR video is requested such that a 2 Mbps channel is sufficient. After the successful channel setup, the network CTE sends a command to the selected video server to start a UDP stream towards the user-specified address and port number. The video stream is correctly played by the user with the current channel setup also in case of network congestion, as shown by the screenshot of Fig. 9.

At a given time (after 130 seconds, in our experiment), the user realizes that the video he/she is currently watching could be much more enjoyable if watched at a higher quality. So the user decides to switch to a “Gold” service for the same content and commands the SIP UA to send a new INVITE message with the updated *VideoQuality* attribute, as shown in Fig. 10. The network CTE checks whether the required content is available on the current server or another one must be found (typical algorithms used in content-delivery networks could be applied in the latter case) and

Fig. 9 Screenshot of the SQ video stream playback when the 2 Mbps guaranteed bandwidth limit is enforced (Image courtesy of Beijing New Picture Film Co. and Elite Group Enterprises)

Fig. 10 Excerpt of the SIP INVITE message encapsulating the NRDL HQ video resource renegotiation



```
▽ <nrdl:VideoQuality>
    rdf:about="vq:01"
  ▽ <nrdl:VQValue>
      Gold
      </nrdl:VQValue>
    </nrdl:VideoQuality>
▽ <nrdl:Endpoint>
    rdf:about="e:s:01"
  ▽ <nrdl:content>
      Hero
      </nrdl:content>
    </nrdl:Endpoint>
▽ <nrdl:Endpoint>
    rdf:about="e:c:01"
  ▽ <nrdl:ipv4>
      137.204.57.174
      </nrdl:ipv4>
```

sends a command to stop the current UDP stream and start a new one with a HQ transmission.

The network CTE must also make sure that the available bandwidth between source and destination is capable of supporting the increased video resolution requested by the user, otherwise the user-perceived quality may be affected. In fact, if the video resolution change is not followed by a suitable renegotiation of the guaranteed bandwidth, in case of network congestion the routers will enforce the resource sharing policies, limiting the bandwidth to the current 2 Mbps value. The effect of this traffic policing behavior in terms of measured throughput and video playback quality is shown in Figs. 11 and 12, respectively.

To maintain the Gold service level requested by the user, the network CTE must reconfigure the routers along the stream path and set the guaranteed bandwidth to a suitable value, e.g. 5 Mbps in our experiment. This results in better measured throughput and improved user-perceived quality, as shown in Figs. 13 and 14, respectively.

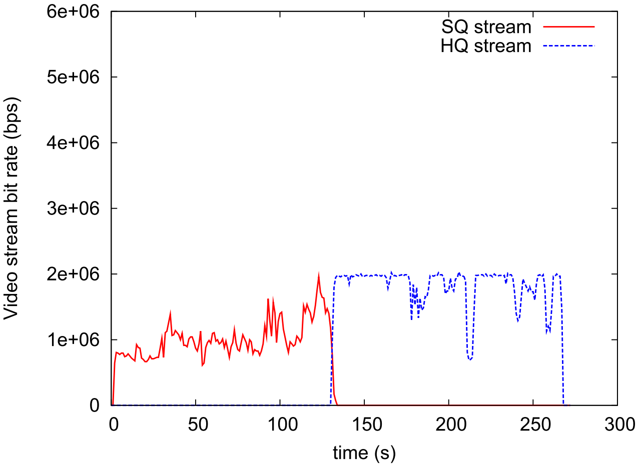


Fig. 11 Measured throughput of the SQ and HQ video streams and effect of the 2 Mbps guaranteed bandwidth limit in case of network congestion

5.3 Cognitive transport service layer performance

Indeed, the signaling due to the CTS layer and to the related exchange of information between CTEs requires some time and may be a source of delays in the service set-up time. Specific requirements on the maximum set-up delay in these specific scenarios do not exist yet, but the recommendations about the Call Set-up Delay (CSD) for VoIP applications issued by the ITU-T [17] can be considered as a general references. The ITU-T fixes a recommended upper bound for the CSD at 7.5 s.

In Fig. 4 it is easy to see that the delay due to the CTS layer signaling is in the range of 3 to 4 s, well below the ITU-T 7.5 s limit. Similar values have been obtained in case 2.

The delay measured here is almost entirely due to the time required to check for resource availability in the network CTE, a function that was not yet optimized for performance in the test-bed. On the other hand, the pure exchange of signaling between CTEs within the CTS layer just adds delays that are measured in the fractions of a second and are therefore negligible.

6 Conclusion

In this paper we showed that advanced service composition capabilities in the Internet can be achieved by using already existing building blocks in a novel and properly arranged way.

We have shown the implementation of a Cognitive Transport Service layer that allows users to request network services which require complex configurations usually achievable only by means of multiple manual operations. The proposed implementation combines a session management

Fig. 12 Screenshot of the HQ video stream playback when the 2 Mbps guaranteed bandwidth limit is enforced (Image courtesy of Beijing New Picture Film Co. and Elite Group Enterprises)

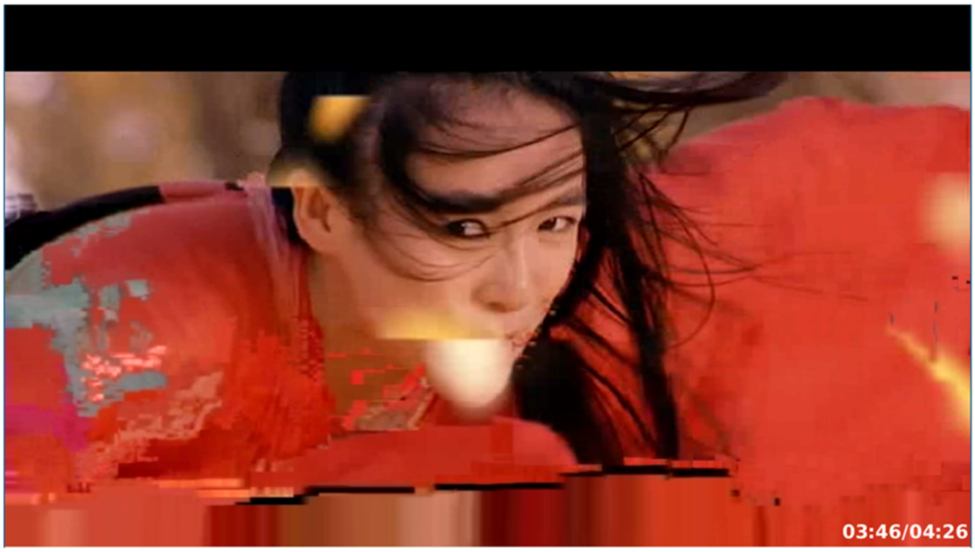


Fig. 13 Measured throughput of the SQ and HQ video streams and effect of the 5 Mbps guaranteed bandwidth limit in case of network congestion

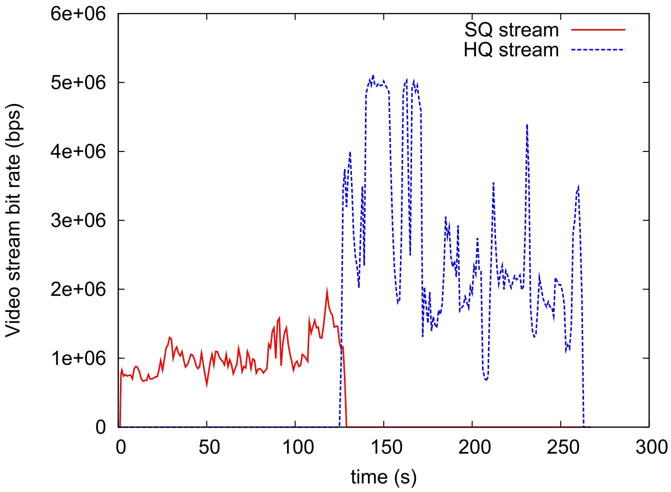
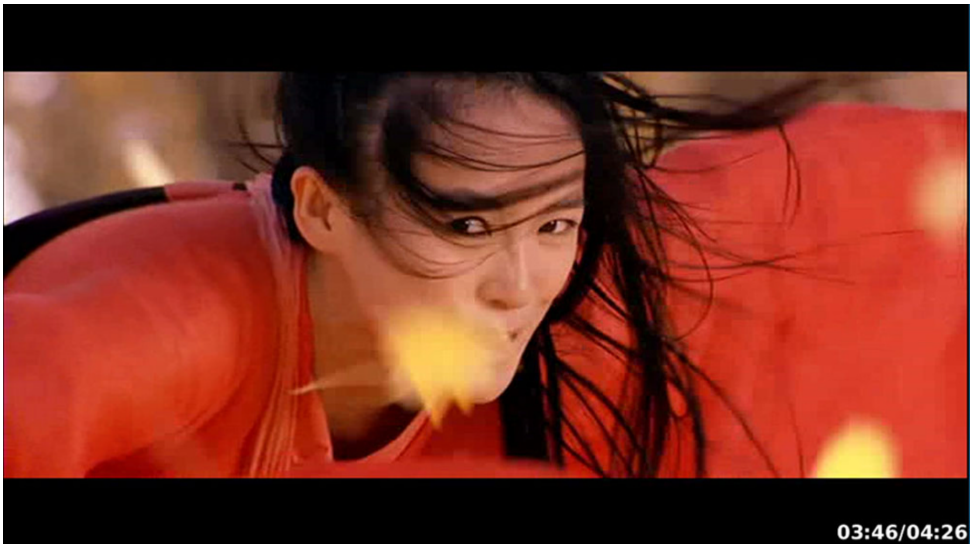


Fig. 14 Screenshot of the HQ video stream playback when the 5 Mbps guaranteed bandwidth limit is enforced (Image courtesy of Beijing New Picture Film Co. and Elite Group Enterprises)



protocol (SIP) with an ontology language for service and resource description (Network Resource Description Language or NRDL), resulting in a rather lightweight solution built upon a well-known and largely deployed protocol such as SIP.

The paper reports two experiments of automatic communication service composition, which proved successfully and demonstrated the effectiveness of the implementation. In both cases it was possible to achieve the automatic activation of complex communication services with a set-up delay bounded to a few seconds, which are considered to be acceptable for such applications.

Acknowledgements The work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe"), a Network of Excellence funded by the European Commission through the 7th ICT-Framework Programme. The images shown in Figs. 9, 12, 14 are from the 2002 movie "Hero" and appear courtesy of Beijing New Picture Film Co. and Elite Group Enterprises.

References

1. Rosenberg J (2008) New waist of the Internet hourglass. IETF draft, draft-rosenberg-internet-waist-hourglass-00
2. Knightson K et al (2005) NGN architecture: generic principles, functional architecture, and implementation. IEEE Commun Mag 43(10):49–56
3. BONE-project, Building the future optical network in Europe, www.ict-bone.eu
4. Callegati F et al (2009) SIP-empowered optical networks for future IT services and applications. IEEE Commun Mag 47(5):48–54
5. Ciulli N et al (2008) Architectural approaches for the integration of the service plane and control plane in optical networks. Opt Switch Netw 5(2–3):94–106
6. Zervas G et al (2007) Demonstration of a fully functional optical burst switched network with application layer resource reservation capability. In: Proc of ECOC 2007, Berlin, Germany
7. Campi A et al (2007) SIP-based OBS networks for grid computing. In: Proc of ONDM 2007, Athens, Greece
8. Martini B et al (2009) SIP-based service platform for on-demand optical network services. In: OFC 2009, San Diego, USA
9. Cerroni W et al (2010) Experimental validation of a SIP-based platform for service-oriented optical network. In: Proc of OFC 2010, San Diego, USA
10. Rosenberg J et al (2002) SIP: session initiation protocol. IETF RFC 3261, June
11. van der Ham JJ, Dijkstra F, Travostino F, Andree HM, de Laat CTAM (2006) Using RDF to describe networks. Future Gener Comput Syst 22(8):862–867
12. van der Ham JJ, Dijkstra F, Grosso P, van der Pol R, Toonk A, de Laat CTAM (2008) A distributed topology information system for optical networks based on the semantic web. Opt Switch Netw 5(2–3):85–93
13. Chien A (2004) The virtual grid description language: vgDL. University of Carlifonia, San Diego, Tech Rep CS2005-0817, August
14. Koslovski GP, Vicat-Blanc Primet P, Schwertner Charao A (2009) VXDL: virtual resources and interconnection networks description language. Lect Notes Inst Comput Sci Soc Inf Telecommun Eng 2:138–154. doi:10.1007/978-3-642-02080-3_15
15. Campi A, Callegati F (2009) Network resource description language. In: Proc of Globecom: Workshop on enabling the future service-oriented Internet, Hononulu, USA
16. Zervas G et al (2008) SIP-enabled optical burst switching architectures and protocols for application-aware optical networks. Comput Netw 52(10):2065–2076
17. ITU-T Rec Y.1531 (2007) SIP-based call processing performance, November