# A Method of Transport Abstraction for 5G Radio Access Networks

Akiko Nagasawa, Yukio Hirano, Kenichi Nakura, Takeshi Suehiro, Hiromu Sato, Seiji Kozaki, Kazuyuki Ishida

Information Technology R&D Center

Mitsubishi Electric Corporation

5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501, Japan

{Nagasawa.Akiko@ak, Hirano.Yukio@dx, Nakura.Kenichi@aj, Suehiro.Takeshi@ea, Sato.Hiromu@ce, Kozaki.Seiji@ab, Ishida.Kazuyuki@dy}.MitsubishiElectric.co.jp

***Abstract*—In the fifth generation mobile technology era, networks which cover a variety of requirements will be constructed as virtual networks (slices) by giving them appropriate resources taking into account each service's requirements. To create a slice, the management and control of a range of physical resources within a heterogeneous network is a crucial issue. One key technique to enable this is a resource abstraction method. Also, it is important to adjust the resource abstraction level to meet the service requirement parameters set by the slice orchestrators or network operators. In this paper, we have proposed abstracted resources representing services' requirement parameters in a 5G Radio Access Network (RAN), and a method for abstracting resources from many types of physical network resource. In addition, a resource allocation procedure using the proposed abstracted resources has been introduced.**

***Keywords—5G, slice, resource abstraction method, resource allocation procedure***

## I. INTRODUCTION

For 5G operations, it is required to create networks flexibly and rapidly in order to meet the needs of a range of services such as enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC). The emerging concept of network slicing is being considered for 5G mobile networks by a variety of organizations (e.g., [1]). Also, the network architecture for 5G is being investigated in ITU-T SG13 [2]. With network softwarization, heterogeneous physical resources such as network equipment, computing and storage resources are abstracted. Elements of the architecture create slices which form multiple isolated networks by allocating abstracted resources to meet the performance requirements of each service. To facilitate the management and control of the resources, one important issue is the abstraction function including modeling of the abstracted resource, creating the resource and allocating slices to the resource in mobile fronthaul and backhaul systems.

The abstraction of heterogeneous physical resources is currently the subject of extensive research. For example, the Networking Service Interface (NSI) standardized by Open Grid Forum (OGF) has been proposed in order to provide interoperability for distributed computing. The network resources in a heterogeneous multi-domain environment are treated by NSI as a common interface [3]. Abstraction models for transport networks are also being investigated by standardization bodies such as the Internet Engineering Task Force (IETF) [4]. In addition, the work in [5] has proposed a Transport Software Defined Networking (SDN) model using the YANG data model for transport network abstraction: this model abstractly describes each network element (node, link and path) and their attributes (e.g., bandwidth). The work in [6] focuses specifically on Centralized Radio Access Network (C-RAN) with a dense wavelength division multiplexing (DWDM) transport network and proposes three transport abstraction models. One represents a single node with no information about the internal transport interconnections, another represents a set of virtual links interconnecting the transport I/O ports, and the third represents a notion of all the possible optical paths that can be used to interconnect the nodes.

However, there is a critical issue in that these abstracted models proposed by the related works do not cover the resources needing to be managed in order to realize 5G services such as eMBB, URLLC and mMTC. If for example URLLC for autonomous driving and telemedicine requires a network with low latency, a high data rate and high availability, the slice orchestrators or network operators need to manage the resources of latency, data rate and availability to create a slice suitable for such a service. Also, the method used to create this resource should be considered.

In order to represent the performance requirement parameters for 5G services needed by slice orchestrators or network operators, we propose the following three definitions and methods: 1) definition and modeling of abstracted resources based on the performance requirements needed by slice orchestrators or network operators, 2) a resource

---

* A part of this work was conducted under the R&D contract "Wired-and-Wireless Converged Radio Access Network for Massive IoT Traffic" with the Ministry of Internal Affairs and Communications, Japan, for radio resource enhancement.

978-3-903176-14-0

abstraction method to convert heterogeneous physical resources into abstracted resources, and 3) an abstracted resource allocation procedure to create slices based on the slice requirements needed by slice orchestrators or network operators.

In Section 2, we define abstracted resources based on the performance requirements needed by slice orchestrators or network operators. Section 3 shows how to create the proposed abstracted resources. In Section 4, we describe a resource allocation procedure to create the slices required by slice orchestrators or network operators.

## II. PROPOSED ABSTRACTED RESOURCE MODEL

### A. *Performance requirements needed by slice orchestrators or network operators*

The primary uses assumed for 5G are eMBB, URLLC and mMTC, as mentioned above. The enhancement of the key capabilities required by such uses is addressed in [7]. We extracted the performance requirements from the key capabilities described in [7] for wired networks as typified by mobile fronthaul and backhaul. The result of our extraction is that the primary performance requirements for wired networks are latency, data rate and availability. These performance requirements need to be defined as abstracted resources. Table 1 shows the relative importance of each of these performance requirements for each case. This is done using an indicative scaling with three steps: "high", "middle" and "low". URLLC for applications such as autonomous driving has low latency, a high data rate and high availability as very important requirements. For eMBB that supports various applications such as HD video, the data rate is the most important among the other capabilities. For mMTC that supports applications such as Smart Cities, latency, data rate and availability are less important capabilities than for the other uses.

Table 1. The relative importance of each performance requirement for each use

| | automotive (URLLC) | HD video (eMBB) | Smart City (mMTC) |
|---|---|---|---|
| latency | H | M | L |
| data rate | M | H | L |
| availability | H | L | L |

### B. *Proposed abstracted resource model*

The target network domains in this paper are radio access networks including mobile fronthaul (MFH) and mobile backhaul, as shown in Fig. 1. In order to represent the performance requirements such as latency, data rate and availability, we use the parameters described below.

First, we define a node located in the core network side as a "higher host" (HH), and one located in the access network side as a "lower host" (LH). For example, when applying this to MFH in Fig. 1, the candidate for HH is Central Unit1 (CU1), and the candidates for LH are Distributed Unit1 (DU1) and DU2. A pair of HH and LH nodes is called a "host pair". Next, we hide the physical resources such as nodes, links and topology between the host pair and define the result as an "abstracted path" (a-path). Each host pair has only one a-path. In the RAN described in Fig. 1, the two host pairs, CU1/DU1 and CU1/DU2, are represented as a-paths, and other nodes and links are well hidden. Thanks to this abstracted network configuration, it is not necessary to consider other nodes (OLT1, ONU1 and ONU2), and the links and topology between CU1 and DU1 or CU1 and DU1. In addition, each a-path possesses "a-path resources". We define three kinds of parameter as a-path resources, (a) lowest latency class, (b) maximum data rates for latency classes, and (c) availabilities for latency classes. The "latency classes" are defined by certain thresholds for latency value that reflect the service requirements of the networks. With the introduction of latency class, a-path resources for (b) maximum data rate and (c) availability are represented in each latency class.

As a result of the above definitions, the proposed abstracted resources could be represented as a group of a-paths that possesses multiple resource attributes. Representing the abstracted resources as above enables slice orchestrators or network operators to simplify the management and control of physical resources suitable for the performance requirements because they do not need to know the details of the physical resources. Therefore, the slice orchestrators or network operators can create slices quickly and flexibly.

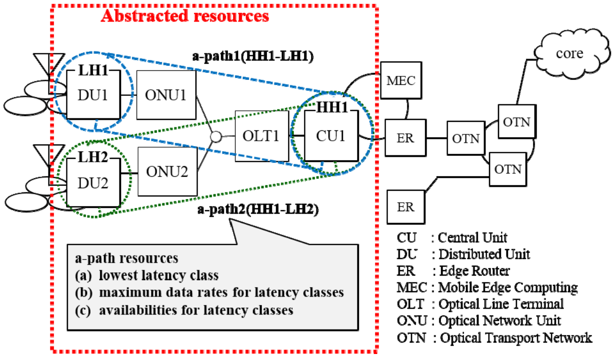

Fig. 1. Example of network and abstracted resources

## III. RESOURCE ABSTRACTION METHOD

In this section, we explain a method for creating the proposed abstracted resources and present an example for a MFH network in Fig. 2.

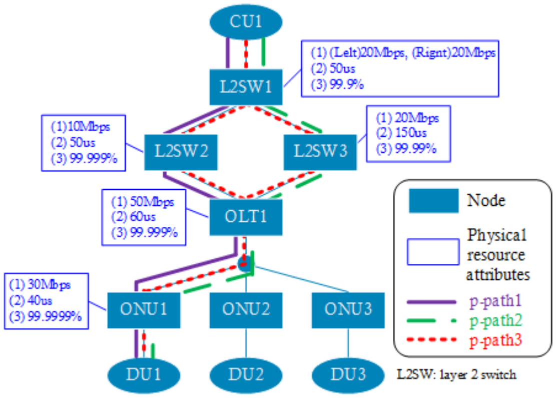

Fig.2. Example of physical resources in mobile fronthaul

Each node has as its physical resource attributes parameters such as (1) guaranteed rate per port, (2) processing latency and (3) availability (only the downlink parameters are shown). The abstracted resources are created in the following four steps.

Step1: Create a list of all host pairs

Create a list of all host pairs which are all combinations of HH and LH in the target network. In the example shown in Fig. 2, CU1 is assigned as HH, DU1, DU2 and DU3 are assigned as LH. There are three host pairs, CU1 and DU1 (host pair1), CU1 and DU2 (host pair2) and CU1 and DU3 (host pair3).

Step2: Search all physical paths

Search all physical paths (p-paths) between HH and LH for each host pair. If a host pair can take several paths, including multiple paths, we identify each path independently as a single path and a redundant path. A serial path is called a single path, while a parallel path is called a redundant path. Using redundant paths enables the system to maximize the data rate and availability. For example, host pair1 has three p-paths as shown in Fig. 2. The p-path3 includes the following nodes: CU1-L2SW1-(L2SW2 and L2SW3)-OLT1-ONU1-DU1.

Step3: Calculate the p-path resources

Calculate three parameters for each p-path as the p-path resources. The p-path resources are the aggregate of the physical resource attributes of the nodes comprising each p-path.

(i) data rate: the minimum guaranteed rate among all the nodes on the p-path.

(ii) latency: the total latency obtained by summing all the latencies of the nodes and links on the p-path.

(iii) availability: the value calculated by using formulae (1) and (2) for serial cases and parallel cases respectively.

$$availability = \prod_{node\ i\ \in\ p\text{-}path} availability\_node\ i \qquad (1).$$

$$availability = 1 - \prod_{node\ i\ \in\ p\text{-}path} (1 - availability\_node\ i) \qquad (2).$$

For p-path1, the data rate is 10 Mbps because the limiting rate is 10 Mbps, which is the guaranteed rate of L2SW2. The latency is 200 µs, which is the total latency of all the nodes in p-path1. The availability is 99.8979%. The results of the p-path resource calculations for host pair1 are shown in Table 2.

Table 2. The p-path resources for host pair1

| | data rate [Mbps] | latency [µs] | availability [%] |
|---|---|---|---|
| p-path1 | 10 | 200 | 99.8979 |
| p-path2 | 20 | 300 | 99.8889 |
| p-path3 | 30 | 300 | 99.8989 |

Step4: Calculate a-path resources

Calculate the a-path resources as follows.

(a) lowest latency class: the latency class to which the lowest latency p-path for the host pair belongs.

(b) maximum data rates for latency classes: the highest data rate among all the p-paths classified to each latency class for a host pair.

(c) availabilities for latency classes: the highest availability among all the p-paths classified to each latency class for a host pair .

Each latency class of a host pair contains all the p-paths whose latency is less than or equal to the corresponding threshold. Hereafter, we adopt 250 µs as the single threshold here because it is one of the most significant latency values corresponding to the lower split options in MFH [8], and call the resulting two latency classes the 250 µs class and the unconstrained class. In this example, the lowest latency class is the 250 µs class, to which the latency of p-path1 belongs. The maximum data rates of host pair1 for latency classes 250 µs and unconstrained are 10 Mbps and 30 Mbps, respectively. The availabilities of latency classes 250 µs and unconstrained are 99.8979% and 99.8989%, respectively. The results of the a-path resource calculations for host pair1 are shown in Fig. 3.

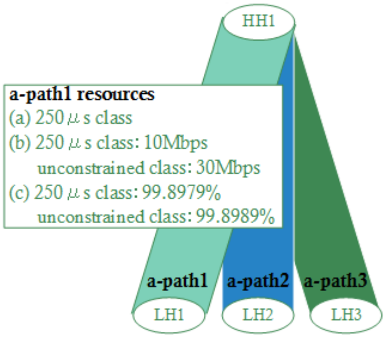

Fig. 3. Abstracted resources

By the above method of abstracting the various physical resources by representing those a-path resources which are characteristic p-path resources, it is possible to hide the network's complexity.

## IV. SLICE CREATION METHOD USING ABSTRACTED RESOURCE

We propose a method for creating slices by allocating the resources to each slice based on the slice requirement sets of the slice orchestrators or network operators using abstracted resources. Table 3 shows an example of a slice requirement set. We present an example of the allocation of the abstracted resources described in Fig. 3 to slice1.

Table 3 Example of a slice requirement set

| | host pair | latency class | data rate [Mbps] | availability [%] |
|---|---|---|---|---|
| slice1 | #1: CU1-DU1<br>#2: CU1-DU2 | 250 µs | 5 | 98.0 |

Step1: Select one host pair from the slice requirement set

Select one host pair from the slice requirement set. For the example shown in Table 3, assume that host pair 1 is selected first among the required host pairs.

Step2: Specify the a-path matching the selected host pair

Specify the a-path whose host pair is the selected host pair. For our example, a-path1 is specified because a-path1's host pair (CU1-DU1) matches host pair #1.

Step3: Judge using a-path resources

Judge whether or not the a-path resources of the specified a-path satisfy each performance requirement of the slice requirement set. The a-path resources represent the threshold values of the p-path resources. Therefore, an approximate judgment of whether a p-path satisfying the slice requirement exists or not is possible by making a judgment using the a-path resources. The judgment method using a-path resources is presented below.

In the first step, the required latency class is compared with the lowest latency class. If the required latency class is equal to or higher than the low latency class, go to the next step, otherwise the result of this judgment is that a p-path matching the slice requirement set does not exist (referred to as "failure"). In the second step, the required data rate is compared with the maximum data rate for the latency class corresponding to the required latency class. If the required data rate is equal to or less than the maximum data rate for the latency class, go to the next step, otherwise the judged result is failure. In the third step, the required availability is compared with the availability of the latency class corresponding to the required latency class. If the required availability is equal to or less than the availability of the latency class, it appears that the probability that a p-path matching the slice requirement set exists is high. Otherwise the judged result is failure. For our example, it is revealed that a p-path satisfying slice1's requirements probably exists.

Step4: Judge using p-path resources

Judge whether or not the p-path resources of p-paths which belong to the latency class satisfying the required latency satisfy each performance requirement of the slice requirement set. For our example, p-path1 is specified because p-path1 belongs to the required latency class and its data rate and availability match the slice requirement set of slice1.

Step5: Allocate the a-path resources

Allocate as many a-path resources as needed to match the slice requirement set of the slice. In this case, the a-path resources of a-path1, i.e. lowest latency class, maximum data rate for the 250 μs class and availability for the 250 μs class, are allocated to slice1 as 250μs, 5 Mbps and 98.0% respectively.

Step6: Allocate the p-path resources

Allocate p-path resources so as to satisfy the allocated abstracted resource. In this case, the p-path resources of p-path1, i.e. latency, data rate and availability, are allocated to slice1 as 250 μs, 5 Mbps and 98.0% respectively.

Step7: Recalculate the p-path resources and a-path resources

Recalculate the p-path resources and a-path resources using the resource abstraction method proposed in Section 3 because the available data rate of each node comprising the a-path is reduced due to the data rate of the a-path being allocated to the slice. Following this, if, among all the host pairs being assessed for the slice requirement set there remains a host pair which has not been allocated p-path resources, steps 1 to 7 shall be performed for this remaining host pair until all the required host pairs have been allocated p-path resources.

As a result of recalculating the abstracted resources, the maximum data rates for the 250 μs latency class and the unconstrained class are each reduced by 5 Mbps. Also, other a-path resources are affected by each a-path. Further, following this recalculation, steps 1 to 7 need to be performed for host pair2, because the slice requirement set needs to account for host pair 2.

If all required host pairs can be allocated p-path resources satisfying their slice requirements, the required slices can be created. In this way, allocating a-path resources to satisfy the performance requirements of the slices enables us to simplify the creation of slices.

Using this slice creation method, we evaluate the proposed pair of methods in terms of response time until completion of slice creation for a given number of p-paths, compared with an exhaustive p-path search by simulation. In this evaluation, we consider the number of judgment steps as representing the response time. As for the slice requirements, the number of slices in a slice requirement set is always one, and each slice has one host pair. The latency class is either of the two classes and the data rate takes any integer value between 1 and 5. As for the p-path parameters, each p-path has a random integer value for the physical resource attributes. The exhaustive p-path search judges whether or not p-paths satisfying the given slice requirements exist by checking each given p-path in order.

Fig.6 shows the average number of steps over one million trials for each case of the given number of p-paths. The result indicates that, as the number of p-paths increases, the exhaustive method takes longer to find an appropriate p-path, while the proposed pair of methods takes almost the same time regardless, due to the beneficial resource abstraction.

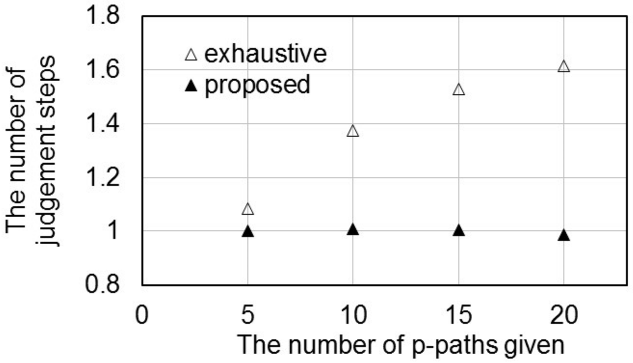

Fig. 6. Comparison of the number of steps to create a slice (normalized to the proposed algorithm for 5 p-paths)

## V. CONCLUSIONS

In this paper, we have proposed an abstraction method which includes the definitions of abstracted resources representing a group of a-paths, each of which has a set of resource attributes that takes into account the service requirement parameters of emerging services in a 5G radio access network. Also, we have presented a resource allocation procedure using abstracted resources and shown its effectiveness. These techniques will enable flexible and rapid 5G network slicing and service orchestration.

# REFERENCES

[1] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb and M. Bagaa. "End-to-End Network Slicing for 5G Mobile Networks," Journal of Information Processing, Information Processing Society of Japan, vol. 25, 153 – 163, Feb. 2017.

[2] ITU-T Y.3150, "High level technical characteristics of network softwarization for IMT-2020," Jan. 2018.

[3] G. Roberts, T. Kudoh, I. Monga, J. Sobieski, C. Guok , J. MacAuley, "Network Services Framework v2.0," Open Grid Forum (OGF), GFD–I-213, October, 2014.

[4] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of TE Networks (ACTN)," Internet Eng. Task Force (IETF), RFC8453, Aug. 2018.

[5] Chang-Gyu LIM, Soo-Myung PAHK, "Model of Transport SDN for Path Computation," in The 20th International Conference on Advanced Communication Technology, pp. 309 – 312, Feb. 2017.

[6] Matteo Fiorani, Ahmad Rostami, Lena Wosinska, and Paolo Monti, "Transport Abstraction Models for an SDN-Controlled Centralized RAN," IEEE Communicationts Letters, vol. 19, no. 8, pp. 1406 – 1409, Aug. 2015.

[7] ITU-R M.2083-0, "IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond," Sept. 2015.

[8] 3GPP TR 38.801 V14.0.0, "Technical Specification Group Radio Access Network; Study on new radio access technology: Radio access architecture and interfaces (Release 14)," Mar. 2017.