Squat-based Resource Management Strategy for Enabling Shared Infrastructures over Optical Networks

Joan A. Garcia-Espin, Xavier Hesselbach*, *Member, IEEE*
i2CAT Foundation, C/Gran Capità 2-4, 08034 Barcelona, Catalonia, Spain
** Technical University of Catalonia, C/Jordi Girona 1-3, 08034 Barcelona, Catalonia, Spain*
Tel: (+34) 93 553 2518, Fax: (+34) 93 553 2520, e-mail: jage@i2cat.net
Tel: (+34) 93 401 5987, email: xavier.hesselbach@entel.upc.edu

ABSTRACT

This paper proposes a new paradigm and strategy for resource management in optical networks considering that resources allocated to a user or service can be *squatted* in by third parties temporarily in order to provide support for emergency situations or allocate traffic from different priorities that experience a sudden increase of demanded resources in the network. We introduce the strategies of Soft Squatting and Hard Squatting. Moreover, we propose a model and a preliminary evaluation for fibre infrastructure sharing in all-optical networks using the proposed squatting strategies.

Keywords: optical network sharing, squat, lambdas, auto-provision, shared resource usage.

1. INTRODUCTION

The new emerging high-performance and high-capacity applications are greedy network and IT resources requesters. Future Internet is envisioned to rely on new, dynamic, premium network services, which go hand in hand with the optical network infrastructures. Together with these new requirements, several challenges in optical networking are to be faced in the coming years. Several architectures have been presented in the literature in the last years [1-3]. At the same time, logical abstraction of network resources has achieved enough maturity in order to be considered a robust and stable tool to be applied at any network layer. Special consideration is given in this article to L0/L1 optical resource virtualisation, such as fibre partitioning between Optical Cross Connects (OXCs) for parallel transmission of demanding data flows along a WDM links. Also during the last years, network virtualisation has been determined to be one of the main actors in the Future Internet context, where it is said to be one of the leading mechanisms towards the new Internet design [4]. Consequently, we believe building co-existing virtual infrastructures on top of the optical substrate is a matter of adequately abstracting and partitioning optical resources, such as fibres and OXCs. As a result, the provision of sufficient flexibility and manageability to present optical networks will definitely impact on their overall efficiency. Indeed, it yields to considerable higher usage rates of the transport network due to smart sharing of the optical resources by using similar strategies such as presented in this article.

This paper is structured as follows: we first overview some virtual optical network infrastructure challenges. After that, section 3 presents current virtualisation paradigms as enablers for smart sharing optical networks. The squatting strategy is presented in section 4. Finally, we close this manuscript with the conclusions.

2. CHALLENGES IN VIRTUALIZATION FOR OPTICAL INFRASTRUCTURES

Virtualisation is known and widely used in Grid and Cloud computing, since it provides techniques that can be (and have been) commercially exploited by many IT service providers such as Amazon. Optical network infrastructures can be equally partitioned and virtualised in multiple, parallel, tailored networks over the same physical substrate. This concept is broadly known as IaaS or *Infrastructure as a Service* [5, 6] in the IT world, and it is incipiently being ported to networking. In the last few years, several initiatives have worked in this direction, in to bring the concept of IaaS into optical network infrastructure by providing a mechanism for sharing network-partitioned resources. Example of these initiatives is Argia/UCLP [7] addressing IaaS at L1.

3. VIRTUALISATION PARADIGMS FOR OPTICAL NETWORKS

The concept of multiple co-existing logical networks running over the same optical substrate is not new in the literature: it has been there for a long period adopting forms such as L1VPNs [8], active and programmable networks, and overlay networks [9], among many others. In the optical networking area, virtualisation paradigm is considered by its proponent as the key driver for fulfilling new generation requirements commented above.

Different mechanisms can be followed for creating virtualised optical networks: aggregation/composition (*N*:1), where several optical resources at the physical substrate can be seen as a simple node at the logical level; optical resource partitioning (1:*N*), a well-known mechanism inherent to network virtualisation; and finally simple abstraction (1:1) mechanism. An example of the aggregation mechanism can be easily found when considering a WDM ring. In this case, the whole topology can be abstracted to a single, aggregated node with as much ports as user interfaces can be found in the nodes of the ring. Consequently, the technology and complexity of the optical substrate are hidden. In this article, we only consider the 1:*N* virtualisation paradigm, where an optical resource (a fibre between two WDM devices) is partitioned into *N* virtual instances, gaining a higher granularity for smart, per-flow service management. Each of the *N* virtual fibres is emulated by using a lambda (λ). The total resource amount (fibre capacity) is computed as the concatenation or sum of all the lambdas that are multiplexed in it. Moreover, for simplicity, we assume the total number of traffic flows that cross the fibre is also *N*, having a one-to-one mapping between traffic flow identifier and lambda identifier. Traffic flows are given a prioritisation, in order to emulate differentiated services behaviour among them, which directly maps to lambda priorities as follows: a sub-script character is included that indicates the numerical priority of the flow/lambda: 1 for the highest priority entity and N for the lowest priority one.

4. THE SQUATTING STRATEGY

4.1 Preliminary considerations and notation

In this section we define a *Squatting* strategy devoted to smartly managing the aforementioned *virtual fibres* (lambdas) as partitions of an optical resource (fibre). We denote the usage a traffic flow makes of a given lambda as x_i. The maximum lambda capacity is denoted as X_i. It must be noted that a traffic flow can request more resources than the ones available in its associated lambda. For modelling this situation, we define:

- A_i are the reserved resources for flow i, that is, the reserved resources in λ_i.
- S_i are the allocated resources for flow i, that is, the allocated resources in λ_i.
- G_i are the demanded resources by the flow i, before being effectively reserved and allocated.
- S_{ij} are the allocated resources for flow i that were originally reserved for flow *j*. It is also named *cross-allocation*.

4.2 Concept

The Squatting strategy is based on the idea that a given traffic flow can make use of the resources in the lambda that were originally reserved for a flow of different priority, in case of huge resource demand.

Squatting is a pacific fibre resource sharing strategy based on the “use it or lend it” idea for the idle part of the lambdas, that is, the resources per lambda that each flow is not using. Thus, this strategy can be implemented in two different ways, depending on which flow is applied the Squatting procedure, in terms of priority. These implementations are namely:

- Soft Squatting (SS)
- Hard Squatting (HS)

Soft Squatting implementation initiates squatting process against the lambda resources belonging to higher priority flows, starting from the present one and upwards. The way idle lambda resources are requested is sequential. That is, being the i-th flow the initiator of the squatting process, it will firstly occupy idle lambda resources from flow $i-1$, then from $i-2$, after that from $i-3$, and so on, up to the highest priority flow.

On the other hand, Hard Squatting implementation initiates squatting in the opposite direction, that is, taking idle lambda resources from the adjacent lower priority flow ($i+1$), after that a lower priority one ($i+2$), and will iterate this way up to the lowest priority one.

In either SS or HS case, the squatting strategy follows two primary rules:

- A given class will never expel another class from its allocated resources, under no circumstances.
- If the target flow does not have (enough) idle lambda resources to satisfy the demand of the squatting initiator, the next adjacent priority flow will be selected instead. This process can be iterated if needed.

Other rules have been considered at the time of writing this article, but have been reserved for future studies in the field. An example of these other rules is the definition of the “squatting threshold”, that is, the parameter that will define the maximum amount of idle resources a given traffic flow can squat from another flow.

As it has been commented before, Soft and Hard Squatting have different operating moods. Whereas Soft Squatting only allows a traffic flow to consider higher priority resources to be consumed in case of need, the Hard Squatting operates in the opposite direction. This has a direct impact on the overall quality of service the traffic flows will be granted.

Consequently, the way a class behaves when forced to initiate one of the previous processes is:

1st Perform a Soft Squatting (upgrading with unused, higher priority lambda resources)
2nd Perform a Hard Squatting (downgrading to lower priority lambda resources)

4.3 Simple models for Soft and Hard Lambda Squatting

As introduced in the previous sections, in the following model definitions we consider *N* traffic flows/lambdas, where 1 is the maximum priority and *N* the minimum priority. The maximum amount of resources (fibre capacity) is normalised to the unity.

Therefore, the initial resource reservation per flow is:

$$A_i = 1/N \tag{1}$$

This equation assumes the whole fibre is equally reserved for all flows. It is not a realistic approach and must be considered merely for illustrative purposes.

At this point, we define the *Potentially Usable Resource (PUR)*, as the resource amount a given flow can effectively use, and thus be allocated to, under certain circumstances. Therefore, the PUR is an indicator of how much resource a given flow can use as a result of implementing soft or hard squatting strategies when facing a huge demand. It must be noted that PUR is obviously driven by the resource demand of the flow, G_i, and becomes the allocated resource for the same flow, S_i, when the squatting process successfully finishes.

For simplicity, it is assumed that all the flows increase their demanded resource linearly with a slope s, until a maximum of:

$$S_i^{max} = s \cdot X_i \tag{2}$$

4.3.1 Soft Squatting (SS)

In this scenario, the allocable resource variable for all flows but the lowest priority one is limited as follows:

$$S_i = PUR_i = s \cdot x_i \leq \frac{1}{N} \quad (i \neq N) \tag{3}$$

This constraint considers the equation in (1) and applies to all flows except for the *N*-th, which is allowed to increase its resource allocation without limitation. This way, the need for occupying extra resources (more than a lambda) appears in the lowest priority flow when having a huge demand. PUR_N is calculated in the next paragraphs.

Analytically, we can define the behaviour as:

$$PUR_i = \begin{cases} s \cdot x_i & for \quad x_i \leq X_i \\ s \cdot X_i & for \quad x_i > X_i \end{cases} \quad (\forall i \neq N) \tag{4}$$

At a given point in time, G_N increases over the initially reserved value (A_N), which means the *N*-th flow needs extra resources from other lambdas in order to raise S_N up to the demanded rate, G_N. At this point, the *N*-th flow could start occupying idle resources available in other lambdas, thanks to the soft squatting strategy.

From eq. (4) we can easily find that the aggregated, normalized resource available for squatting by flow *N* is:

$$\sum_{i \neq N} \left(\frac{1}{N} - s \cdot X_i\right) \tag{5}$$

If we consider all flows have the same, homogeneous lambda capacity limit, that is $X_i = X \ (\forall i)$, (5) can be re-written as follows:

$$\left(\frac{1}{N} - s \cdot X\right) \cdot (N-1) \tag{6}$$

Therefore, the total potentially usable resource for the *N*-th flow corresponds to the self-owned one plus the squatted. When considering the homogeneous maximum resource value simplification for eq. (5) and using eq. (6), the PUR for the lowest priority flow can be written as follows:

$$PUR_N = \frac{1}{N} + \left(\frac{1}{N} - s \cdot X\right) \cdot (N-1) \tag{7}$$

This trivially leads to the simplified expression:

$$PUR_N = 1 - s \cdot X \cdot (N-1) \tag{8}$$

And consequently we can specify the total allocated resource for the *N*-th flow, after squatting, as:

$$S_N = PUR_N = \begin{cases} s \cdot x_N & for \ \ x_N \leq \frac{1}{s} - X \cdot (N-1) \\ 1 - s \cdot X \cdot (N-1) & for \ \ x_N > \frac{1}{s} - X \cdot (N-1) \end{cases} \tag{9}$$

When comparing both PUR_i from eq. (4) and $S_N = PUR_N$ from eq. (9), we can appreciate the effect of the squatting technique over the PUR for the *N*-th flow, which pushes up its allocable lambda resource much higher than the maximum lambda capacity for all other flows.

Figure 1a shows the *N*-th flow can potentially have an effective fibre resource usage much greater than the resources allocated in the regular behaviour (limited to lambda capacity). In general, this result can be extended to any flow that has higher priority flows, by easily re-factoring the indexes in eq. (9).

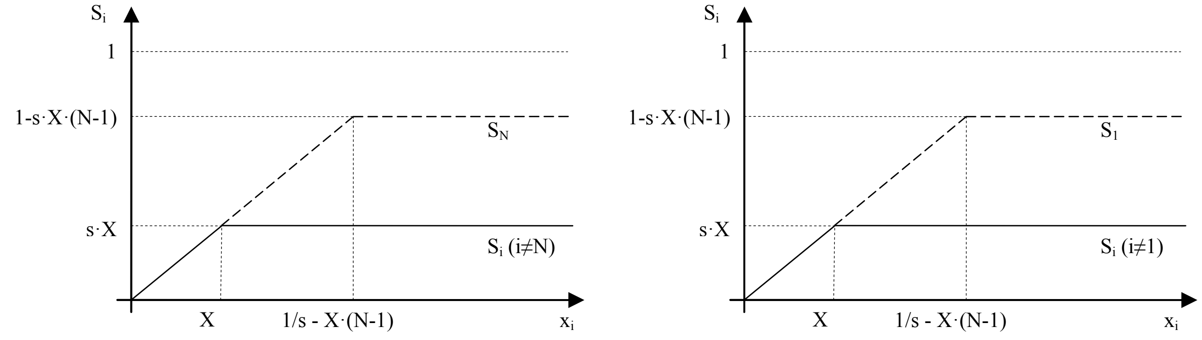
4.3.2 Hard Squatting (HS)

Analogously to Soft Squatting presented before, the Hard Squatting implements the squatting technique but considering now that the higher priority flow suffers a dramatic increase on its resource demand and starts occupying idle lambda resources from lower priority flows.

Assuming all flows increase their lambda resource usage in the same way, that is, linearly up to a given value X under the same context conditions as in the Soft Squatting scenario, we can conclude that the potentially usable fibre resource for class 1 is:

$$S_1 = PUR_1 = \begin{cases} s \cdot x_1 & for\ x_1 \leq \frac{1}{s} - X \cdot (N-1) \\ 1 - s \cdot X \cdot (N-1) & for\ x_1 > \frac{1}{s} - X \cdot (N-1) \end{cases} \tag{10}$$

And it has an identical graphical representation as eq. (9), but considering the new indexes. Figure 1b shows this Hard Squatting scenario, where the 1-st flow squats idle resources from any other flows.



(a) (b)

Figure 1. Per-flow normalised, allocable resource using: (a) the Soft Squatting strategy (S_N), compared to non-squatting flows (S_i , i≠N); and (b) the Hard Squatting strategy (S_1), compared to non-squatting flows (S_i , i≠1)

5. CONCLUSIONS

In this paper we have presented our vision in the optical network infrastructure challenges for the Future Internet and what current virtualisation paradigms are enablers for smartly sharing optical networks. We have introduced a model for a simple, auto-provisioned lambda management strategy, namely *Squatting* strategy, with two variants: the *Soft* and *Hard Squatting*. Our analytical results show how traffic flows implementing the Squatting strategy can achieve a much higher Potentially Usable Resource rate, referred to lambda metrics, compared to the ones implementing legacy limited lambda capacity. In any case, overall increase of the fibre capacity usage is observed, compared to traditional, lambda-isolated, WDM transmission.

ACKNOWLEDGEMENTS

This work has been partially supported by the national Spanish project CICYT TSI2007-66637-C02, the EU-FP7-IST GEYSERS project (contract no. 248657); the Spanish FIT projects Enigma Enhanced (Enigma 2) and Enigma 3 Management System (E3MS), and the European Network of Excellence Euro-NF FP7-ICT-2007-1.

REFERENCES

[1] D.H.C. Du: Clean Slate Design of Internet for Supporting Service-Oriented Computing and Applications, *in Proc. IEEE Int. Conf. On Service-Oriented Computing (SOCA)*, p. 3, 2007.
[2] R. Dutta, *et al.*: The SILO Architecture for Services Integration, control, and Optimization for the Future Internet, *in Proc. IEEE Int. Conf. on Communications*, pp. 24-28, June 2007.
[3] S. van der Meer, *et al.*: The Design of a Novel Autonomic Management Methodology and Architecture for Next Generation and Future Internet Systems, *in Proc. 6th IEEE Conf. EASe*, pp. 3-12, April 2009.
[4] T. Raptis, *et al.*: Technological Enablers for Self-Manageable Future Internet Elements, *Future Computation World*, November 2009.
[5] The Infrastructure as a Service (IaaS) Framework [Online]. Available at: http://iaasframework.com
[6] E. Grasa, *et al.*: UCLPv2, a Network Virtualization Framework built on Web Services, *IEEE Comm. Magazine, Feature Topic on Web Services in Telecommunications*, vol. 46; num. 3, pp. 126-135, 2008.
[7] E. Grasa, *et al.*: Extending the Argia Software with a Dynamic Optical Multicast Service to support High Performance Digital Media, *Elsevier J. of Optical Switching and Networking*, vol. 6, pp. 120-128, 2009.
[8] T. Takeda: Framework and Requirements for Layer 1 Virtual Private Networks (RFC 4847), *IETF Network Working Group*, April 2007. Available at: http://tools.ietf.org/html/rfc4847
[9] K. Mosharaf, *et al.*: Network Virtualization: State of the Art and Research Challenges, *IEEE Comm. Magazine*, vol. 47, issue 7, pp. 20-26, July 2009.