Influence of Software and Hardware Failures with Imperfect Fault Coverage on PONs OPEX

Álvaro Fernández, Norvald Stol

Abstract—**Passive Optical Networks (PONs) are one of the preferred technologies to deploy broadband access networks. As time passes, end users presuppose network connectivity to be always available, and expect PONs to be highly dependable. Yet operators, from an economic view, are interested in the costs related to failures. Thus, PONs dependability and associated costs have been extensively studied, but only focusing on hardware failures. Contrarily, this paper performs a thorough analysis of the impact of software failures in failure-related costs. Based on real empirical data, software failures are thoroughly characterized and classified in four different categories according to their severity. Also, the effect of software failures on the behavior of PON's fiber protection and recovery mechanisms is detailed. Software failures are included into a Markov cost model, implementing a comprehensive cost framework. This way, the dependability-related costs of PONs are analyzed, accounting for hardware and software failures, as well as for the consequences of software failures on well-known PON protection mechanisms. Moreover, how the testing phase duration and user profile (residential or business) impact these costs is pinpointed.**

Index Terms—**Failure coverage; Operational Expenditures; Passive Optical Networks; protection; software failures**

I. Introduction

Due to the rapidly increasing bandwidth demands of new services, network operators are being pushed towards the deployment of broadband access networks. Certainly, Passive Optical Networks (PONs) are widely recognized as the best suited solution for supporting such demands [1]. Amid other features, PONs offer high bandwidth on a per-user basis, as well as scalability and flexibility. PONs also present low energy consumption and are cost-effective as costs are shared among several customers. Hence, PONs and Next-Generation PONs (NG-PONs) are regarded as the most promising solution for future fiber-based access networks [2].

Yet, as end users are starting to take network connectivity for granted, dependable service delivery is also expected from PONs. Consequently, to satisfy the need of reliable access, dependability of access networks has become an important case of interest nowadays. In fact, several protection schemes and dependability analyses for different PONs and NG-PONs flavors can be found in current literature [3], [4], [5].

Commonly, a system's dependability is assessed by its availability. Still, from a financial point of view, an operator is typically more interested in the failure-related costs, known to be part of the operational expenditures (OPEX). Notably, this interest arises as a proper understanding of failure-related OPEX can be used in cost optimization or risk management analyses. Usually, failure-related OPEX cover the cost of repair and extra equipment, payment of penalties and loss of reputation if a large number of users are affected by failures.

However, most of the published PON dependability studies are focused only on hardware, physical faults. Few papers address software dependability or its consequences with respect to OPEX, even though software faults account for an important part of service failures in many systems [6]. Furthermore, software failures also represent impairments to the correct behavior of protection schemes. This is more important as PONs/NG-PONs evolve in complexity, serve more users or are used in e.g. data centers.

Chiefly, this paper provides a comprehensive analysis of the effects of software failures in Time Division Multiplexed (TDM) PONs' failure-related OPEX. Extending the work in [7], a thorough characterization of Gigabit-capable TDM PONs (GPONs) software failures is performed, based on empirical data [8], [9]. How software failures hinder the performance of fiber protection schemes (i.e. fault coverage) in TDM PONs is also deeply detailed, based on real data [10]. Applying Duane model for software reliability growth [11], the software failure intensity is estimated as a function of the testing time and included in a Markov cost model. Hence, the impact of hardware and software failures, as well as of imperfect fiber protection recovery (due to software) in PON's failure-related OPEX is analyzed, accounting for the length of the testing phase and the user profile (residential or business).

This paper is organized as follows. First, Sect. II presents the PON architecture and fiber protection scheme. Section III describes the software dependability and failure coverage modelling. Section IV details the Markov cost model used to assess the failure-related OPEX, while results are presented in Sect. V. Finally, Sect. VI gives the conclusions of this work.

II. TDM PON Architecture and Protection

Plainly, the basic TDM PON architecture is shown in Fig. 1 (a). At the operator's Central Office (CO), the Optical Line Terminal (OLT) is located, consisting of two elements: OLT ports where fibers are connected and the OLT chassis housing them. Besides, the OLT chassis also hosts the OLT software in charge of the PON correct behavior. At the user's side, an

A. Fernández and N. Stol are with the Department of Telematics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: alvarof@item.ntnu.no).

Optical Network Unit (ONU) is deployed. Between the CO and the ONUs, the Remote Node (RN) acts as a splitting point, placed in a street cabinet. The RN consists of a RN chassis housing a set of the passive elements for signal splitting. Following the GPON ITU-T standard [12], splitters with a split ratio of 1:32 are assumed as passive elements. Finally, two sections of fiber can be identified (maximum reach of 20 km [12]): Feeder Fibers (FF), between the OLT and the RN; and Distribution Fibers (DF), between the RN and the ONUs. Feeder fibers span several kilometers and serve all end users. Contrarily, the length of distribution fibers is usually smaller.

Concerning protection in PONs, feeder fiber protection has been shown to be one of the most cost-efficient protection mechanisms by several authors [3], [4], [5]. Decidedly, this is mainly due to the large number of end users affected in case of feeder fiber failure and the relatively large probability of feeder fiber cut (as feeder fibers cover several kilometers). Succinctly, this mechanism implies the deployment of both a protection feeder fiber between the OLT and the RN, and an optical switch at the CO, as shown in Fig. 1 (b). Certainly, working and protection feeder fibers must span over disjoint paths/trenches to avoid common failures. In case of fiber cut (digging), loss of signal will be detected at the CO, and the optical switch will flip to the protection feeder fiber in order to keep connectivity between the OLT and the ONUs. Necessarily, the OLT software should be prepared to perform the switching and preserve service provision after it.

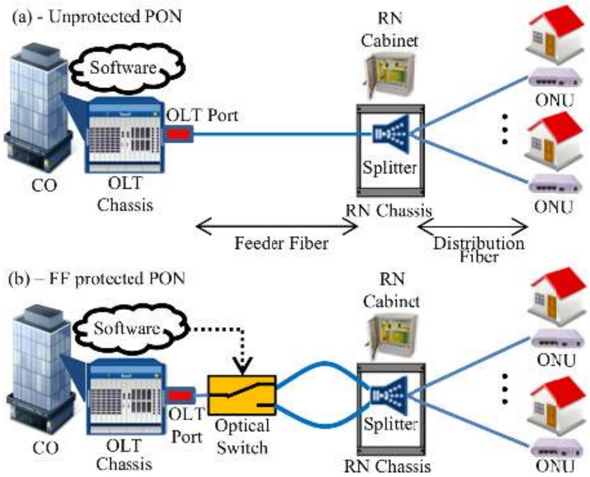


Fig. 1. Basic unprotected (a) and feeder fiber protected (b) PON architecture.

III. OLT Software Dependability Modelling

Essentially, the dependability of the OLT software is evaluated by means of the Duane model for software reliability growth. Reliability growth models allow for predicting the software failure intensity taking into account the test and debugging processes. In the Duane model, software failures occur according to an inhomogeneous Poisson process whose failure intensity decreases with the testing time (see e.g. [11] for a full description of this model).

Notably, the OLT software modelling in this work builds on the analysis presented in [7], where the Duane model was shown to fit empirical data from a GPON OLT software (taken from [8]). Thus, the software failure intensity as a function of the testing time (t) was found in [7] to follow

$$z(t) = 0.237133 * t^{-0.238842} . \quad (1)$$

Yet, the work in [7] adopted a minimal software failure classification and description. Consequently, this paper aims at performing a more thorough failure characterization, with a clearer indication of failure consequences on service delivery. In the empirical data reported in [8], software failures were classified into four categories; namely "low priority bugs", "average critical bugs", "highly critical bugs", and "very highly critical bugs". Although this bug description is not precise, it markedly matches the software defects taxonomy (regarding severity) from [9]. Purposely, the software bugs reported in [8] will be modelled following this taxonomy (based on real projects), which assumes four severity levels.

Severity 4 (S4) defects refer to "low priority bugs" in [8] (10% of total bugs – 0.1 probability of occurrence), not causing service disruption. For example, cosmetic faults or trivial errors not leading to failure fall in this category.

Severity 3 (S3) defects map into "average critical bugs" in [8] (57% of total bugs – 0.57 probability of occurrence). Whereas services are not interrupted, performance is hindered. Consequently, S3 defects affect a subset of business users served by the OLT (due to bandwidth and delay demands). Residential users (without stringent demands) are not affected.

Severity 2 (S2) defects relate to "highly critical bugs" in [8] (31% of total bugs – 0.31 probability of occurrence). Part of the services are stopped and performance is heavily hampered. Still, no total stoppage occurs, thus only a number of users (business and residential) are regarded as down.

Severity 1 (S1) defects refer to "very highly critical bugs" (2% of total bugs – 0.02 probability of occurrence). As these defects cause total stoppage, all business and residential users served by the OLT cease being served.

Additionally, the exact mapping of software failures into the Markov model is detailed in Sect. IV.

A. *Feeder Fiber Protection Fault Coverage*

Further extending the work in [7], the concept of fault coverage is now introduced. In dependability, this term is defined as the probability of the failure recovery mechanism succeeding upon failure occurrence [13]. If the recovery mechanism does not succeed, an uncovered fault occurs, which requires additional recovery actions.

In order to estimate the fault coverage regarding feeder fiber protection, the results reported in [10] will be employed. Chiefly, [10] describes a fault injection campaign carried in a GPON deployment, where feeder fiber cut and protection switching were emulated. After restoring connectivity, several software failures caused services not to be correctly recovered.

Based on these results, the fault coverage of the feeder fiber protection mechanism can be estimated as follows. Surprisingly, only 20% of the emulated cases resulted in correct service restoration. Consequently, the feeder fiber protection fault coverage is fixed as 0.2. Besides, in 71% of the cases medium criticality failures (mapped as S3 defects – business users are not correctly restored) were reported.

Hence, the probability of an uncovered S3 failure upon feeder fiber failure is 0.71. Finally, 9% of the cases caused high criticality failures (corresponding to S2 defects – a number of the affected users are not recovered). Correspondingly, the probability of a S2 uncovered feeder fiber failure is 0.09. Finally, these probabilities (which apply only to uncovered feeder fiber failures) will vary with the testing time in the same proportion as the software failure intensity (1).

We would like to remark that to our appreciation, a fault coverage of 0.2 is unexpectedly low. Due to the lack of other results, it was not possible to verify it. Subsequently, this fault coverage should be taken with care. Thus, results in this work are better seen as a worst case scenario.

IV. Markov Cost Model for Failure-Related OPEX

In order to analyze the TDM PON's OPEX, a Markov cost model has been employed [14]. As in [7], both hardware and software failures have been included. Thus, the scenario with no software failures is used as baseline. Also, a Markov cost model is especially well suited for this work. This is because the software-hardware interaction and imperfect recovery (fault coverage) in Sect. III cannot be modelled with static models (reliability blocks) due to independence assumptions.

Briefly, two types of failure-related costs are considered in this study, namely cost impulses and cost rates. Cost impulses are associated to transitions in the Markov model, i.e. a transition from state *i* to state *j* has an associated impulse cost C_{ij} (in \$). More precisely, these costs cover the extra equipment that must be bought to replace a faulty component. Consequently, impulse costs only apply to hardware failures.

Additionally, cost rates (c_i – cost per unit time in \$/h) are associated to each state *i*. Cost rates consist of two terms: payment of the repairmen (denoted Repair Cost Rate – RCR) and payment of penalties (referred as Penalty Cost Rate – PCR). Simply, the RCR in a state *i* is proportional to the salary (S_H and S_S for hardware and software repairmen, in \$/h) and the number of repairmen in state *i* (OC_i).

As for the PCR, it includes the cost of penalties and the cost of reputation due to failures. Yet the latter can be seen as loss of revenue instead of cost, it is included in the PCR for simplicity. Hence, the PCR depends on the penalty rates (PR_R and PR_B in \$/h, subscripts R and B denote residential or business users), the failed clients in state *i* ($FC_{R,i}$ and $FC_{B,i}$) and a reputation rate gauging reputation cost (RR_R and RR_B in \$/h). To account for an increased cost of reputation if large outages occur (e.g. negative press releases), an impact factor χ is introduced (χ_R and χ_B). Thus, the PCR in state *i* follows:

$$PCR_i = \sum_{K=R,B} (FC_{K,i}^{\chi_K} * RR_K + FC_{K,i} * PR_K). \qquad (2)$$

Let us consider the Markov cost model with only hardware failures. State definition depends on the type of failed element, with the PON elements described in Sect. II. Due to the 1:32 splitting ratio, 32 ONUs are assumed. Failure and repair rates are taken from [3] and [15]. Notably, the fiber failure rate depends on the fiber length, allowing for different scenarios. Besides, it is assumed that there is only one repair crew. If there are two or more failed elements, the element leading to the highest reduction in cost in a shorter time is repaired first.

Concerning the number of failed elements in each state, it also depends on the type and number of failed elements. According to previous studies [5], it is fixed as follows. OLT chassis affect 1600 clients, while RN chassis affect 100 clients. OLT ports and splitters affect 32 clients and ONUs affect only 1 client. As for feeder fiber failures, the number of failed clients is modelled as a uniform variable between 1000 and 5000 clients. Intentionally, this models the fact that a digging close to the CO will cut several fibers, thus disconnecting a large number of clients. Yet if a digging occurs far from the CO, the number of failed clients is smaller. Likewise, the uniform variable is defined between 1 and 100 clients for a distribution fiber cut. Business clients (if present) are uniformly distributed among the total served clients.

A. *Software Failures Modelling*

Concisely, software failures are included in the analysis by extending the model as shown in Fig. 2 (all hardware-software combinations are not depicted for clarity). Because S4 failures do not lead to failed clients, they are not considered. If the OLT software is working properly, any severity failures may occur. Yet, low priority failures cannot appear if a higher severity software failure has already occurred. As the OLT software runs on the OLT chassis, there cannot be software failures if the OLT chassis has failed. Moreover, hardware repair of the OLT chassis assumes to fix software failures.

In Fig. 2, λ_{soft} denotes z(t) in (1), which depends on the testing time *t*. Failure and repair rates of the OLT chassis are denoted λ_{OLT_C} and μ_{OLT_C}. Finally p_1, p_2 and p_3 relate to S1, S2 and S3 failures probability (0.02, 0.31 and 0.57 respectively).

As for the number of failed clients due to software failures, S1 failures cause total stoppage, thus affecting 1600 clients. As mentioned in Sect. III, S2 failures affect a subset of served users (residential and business), modelled as a uniform variable from 1 and 400. The same uniform variable applies for S3 failures, but only affecting business users in this range.

Finally, software repair rates are denoted γ_1, γ_2, and γ_3 with respect to S1, S2 and S3 failures. As S3 failures do not lead to service stoppage, a quick restart (5 min., $\gamma_3 = 12\ h^{-1}$) where the

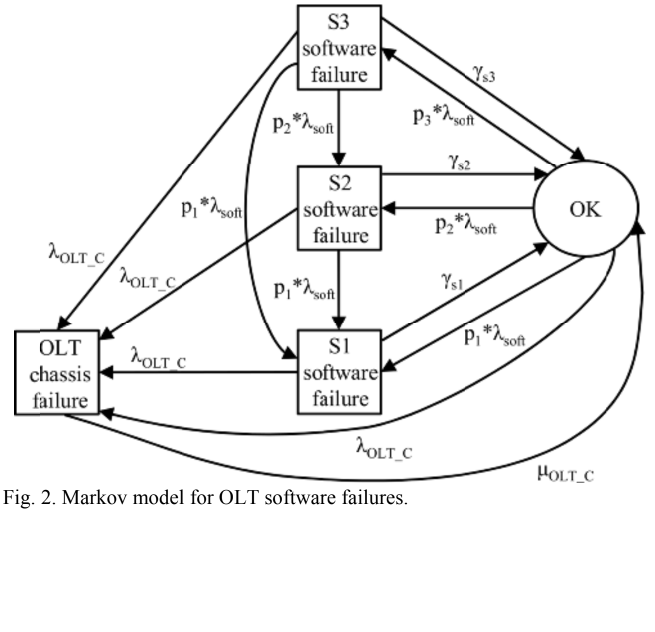


Fig. 2. Markov model for OLT software failures.

failed process is reset is assumed to fix the failure. S1 and S2 failures require a more complex repair action where the processor is reset and the software and data reloaded. This reload is assumed to take 30 min. on average ($\gamma_1 = \gamma_2 = 2$ h^{-1}).

B. Feeder Fiber Protection and Fault Coverage Modelling

To model the fiber protection and fault coverage, the states depicted in Fig. 3 are added. For clarity, combinations with previous states are not shown. From a fail-free state, the switch, the protection or the working fiber may fail. Working fiber failures may be covered or uncovered with the appropriate probabilities. If both working and protection fibers fail, the system is regarded as down.

Briefly, λ_{FF} (μ_{FF}) and λ_{Sw} (μ_{Sw}) denote the failure (repair) rates of the feeder fiber and optical switch respectively. Also, c_{S3} and c_{S2} are the probabilities of S3 and S2 uncovered failures, (0.71 and 0.09 respectively). Also, 1- c_{S3}- c_{S2} = 0.2 is the fault coverage (correct recovery). Uncovered failures repair rates are the same as in the previous section, i.e. $\mu_{Unc\_S3} = \gamma_3 = 12$ h^{-1} and $\mu_{Unc\_S2} = \gamma_2 = 2$ h^{-1}. Upon fiber failure, a uniformly distributed random variable between 1000 and 5000 is drawn. If both fibers fail, this is the number of failed clients. If the failure corresponds to an uncovered working fiber failure, a second uniform variable, between 1000 and the previous result, is drawn. In case of S2 uncovered failure, these are the failed clients; whereas S3 uncovered failures affect only business clients in this range. Recall that covered working fiber failures and protection fiber failures do not lead to failed users. Finally, switch failures cause 32 failed clients.

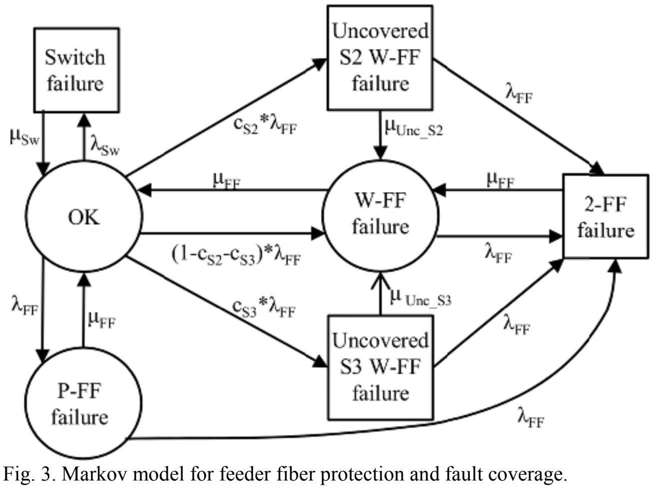


Fig. 3. Markov model for feeder fiber protection and fault coverage.

V. Dependability-Related OPEX Evaluation

In this section, after solving the Markov cost models by simulation, the failure-related OPEX results over a time span of 1 year are presented as expected costs (in $). In steady state, the expected cost over a given time span is calculated by multiplying the Expected Cost Rate (ECR) by the time span. The ECR is assessed from the cost rate of each state (c_i), their probabilities (p_i), and the impulse costs and rates of each transition from state *i* to state *j* (C_{ij} and λ_{ij} respectively) as:

$$ECR = \sum_{\forall i} (c_i + \sum_{\forall j} C_{ij} * \lambda_{ij}) * p_i . \qquad (3)$$

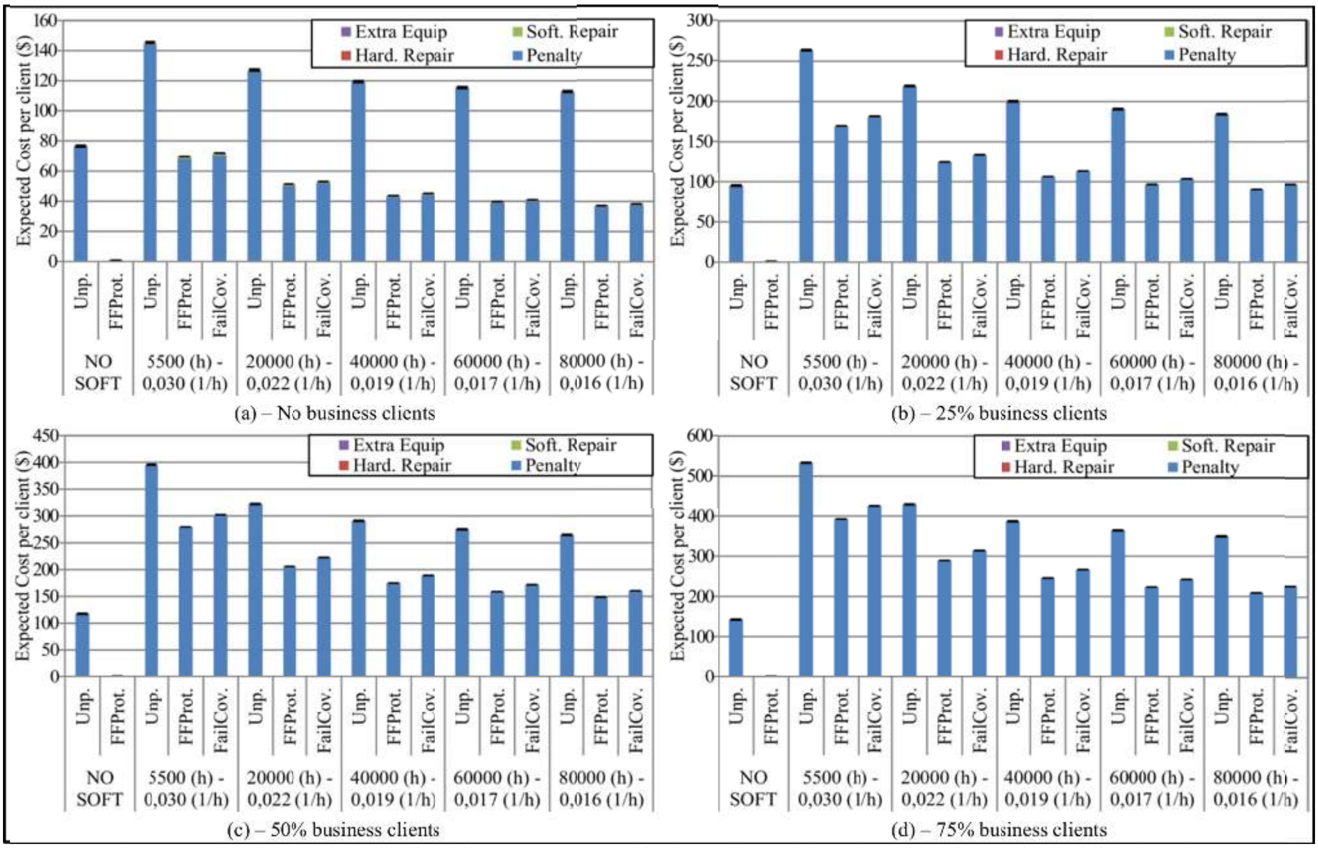
Intentionally, the presented results are broken down into extra equipment (regarding impulse costs), hardware repair (RCR because of hardware failures), software repair (RCR due to software failures) and penalties (relating to the PCR). Besides, results are presented with 95% confidence intervals.

As for the parameters, they are defined as follows. For the impulse costs, related only to hardware failures (purchasing of extra equipment), the prices of the components are taken from [3]. The salary of hardware repairmen (S_H) is fixed to 190 $/h, whereas that of software repairmen (technicians – S_S) is 80 $/h. Because residential users are not willing to pay extra for protection, they are no subjected to penalties. Thus, no penalty is assigned to residential users ($PR_R = 0$), but the reputation rate (discomfort with the operator) is fixed to 30 $/h ($RR_R$). Yet for business users a penalty of 100 $/h is assumed ($PR_B$), while the reputation rate is 50 $/h ($RR_B$). The impact factors are also different for both types, fixed to 1.1 for residential (χ_R) and 1.2 for business (χ_B). Finally, different length for the fibers allow for modelling two types of scenarios. Dense scenarios model densely populated areas, with the lengths for feeder and distribution fibers are 3.75 and 0.375 Km. respectively. In sparse scenarios (suburban or rural areas), the lengths are fixed to 18.2 and 1.8 Km. respectively.

Results are presented for different percentages of business clients. As baselines, the cases with no software (only hardware failures) are shown, with or without ("Unp.") feeder fiber protection. If there is no software, protection works as intended (making c_{S3} and c_{S2} in Fig. 3 equal to 0). The cases with software failures are presented for different testing times (h) and the corresponding software failure intensity (h^{-1}) from (1). The case labeled as "FFProt." assumes the protection scheme always works as intended (for reference purposes).

In Fig. 4, the results for dense scenarios are shown. It can be seen that penalties account for most of the OPEX, even with no business clients. This is explained because a large percentage of failures affect several clients, dominating the OPEX. Expectedly, increasing the testing time (i.e. reducing the software failure rate) reduces the costs when software failures are present. Although this reduction is noticeable for small testing times, is less important for large testing times (above 40000 hours). Particularly, this is because at the beginning of the testing phase, software failures are easily spotted and fixed. After several hours of testing, fewer faults are present and are more difficult to identify and fix. Yet, the reduction in OPEX due to increased testing time is more important the larger the percentage of business users.

Let us focus now on the importance of software failures in the OPEX. When business clients are present, the impact of software failures in OPEX (i.e. comparing unprotected cases with/without software) is bigger than the impact of feeder fiber failures (i.e. comparing unprotected and FF protected cases). Also, the higher the percentage of business, the higher the impact of software failures (the higher the difference with respect to the impact of feeder fiber failures). Certainly, the fact that business users are easily affected by software failures (due to bandwidth and delay demands) justifies this. Still, with a 25% of business users, the impact of software failures and



(a) – No business clients

(b) – 25% business clients

(c) – 50% business clients

(d) – 75% business clients

Fig. 4. Expected cost per client in dense scenarios for different percentages of business clients with varying testing time (software failure intensity).

feeder fiber failures becomes comparable for very long testing phases. Also interesting is the fact that, with no business users, feeder fiber failures contribute more to the OPEX than software failures, independently of the testing time. Yet in this case, both effects are similar for short testing times.

Moreover, Fig. 4 shows that in dense scenarios, uncovered feeder fiber failures do not contribute excessively to the OPEX (especially in business-free areas). Mainly, this is because with short fibers, associated failures are uncommon, thus uncovered failures become even rarer. Let us now remark the importance of including software failures in dependability and OPEX analysis. An idealized analysis, e.g. feeder fiber protection without software, may lead to almost negligible OPEX, even with a high percentage of business users. Still, a real analysis shows that this conclusion may be misleading.

Finally, Fig. 5 depicts the results in sparse scenarios, where trends identified before can be also seen. First, penalties also dominate the OPEX in sparse scenarios. Second, increasing the testing time above 40000 hours does not reduce the OPEX substantially. Especially without business users, the OPEX reduction due to increasing the testing time is almost trivial.

Necessarily, the OPEX results are higher due to the larger fibers, making this type of failures more common. Thus, costs associated to feeder fiber failures now dominate the OPEX (for any percentage of business), whereas software failures are less relevant. Although the impact of feeder fiber failures is always higher than that of software failures, the latter gains in relevance as the percentage of business users increases.

Decidedly, as feeder fibers are more prone to failures, uncovered failures in dense scenarios gain in importance. Still not very relevant in a business-free scenario, the impact of these failures in the OPEX is significant when business users are included. This result is especially important because Long-Reach PONs (LG-PONs, extending the fiber reach up to 100 Km.) have gained interest lately as a possible economic and profitable evolution of PONs. With this fiber reach, uncovered failures will become more likely to occur, hindering the dependability and OPEX of future PONs even with fiber protection schemes. Thus, uncovered failures because of software must be taken into account and considered.

VI. Conclusions

A detailed failure-related OPEX analysis of PONs has been presented in this paper. Software failures have been the main object of the study, with a twofold focus. Namely, not only the direct impact of these failures in OPEX has been assessed, but also how they affect the performance of fiber protection schemes in PONs. Also, hardware failures have been included for completeness. Based on real data, the contributions of this work include a detailed characterization and classification of software failures in PONs, which proves useful in further research due to the lack of this information in current literature. Besides, a comprehensive method and framework for analyzing failure-related OPEX in PONs, based on Markov cost models theory has been proposed. This method is applicable to any PON technology by correct tailoring of parameters, capturing dynamic interactions and imperfect recovery that cannot be handled by static models.

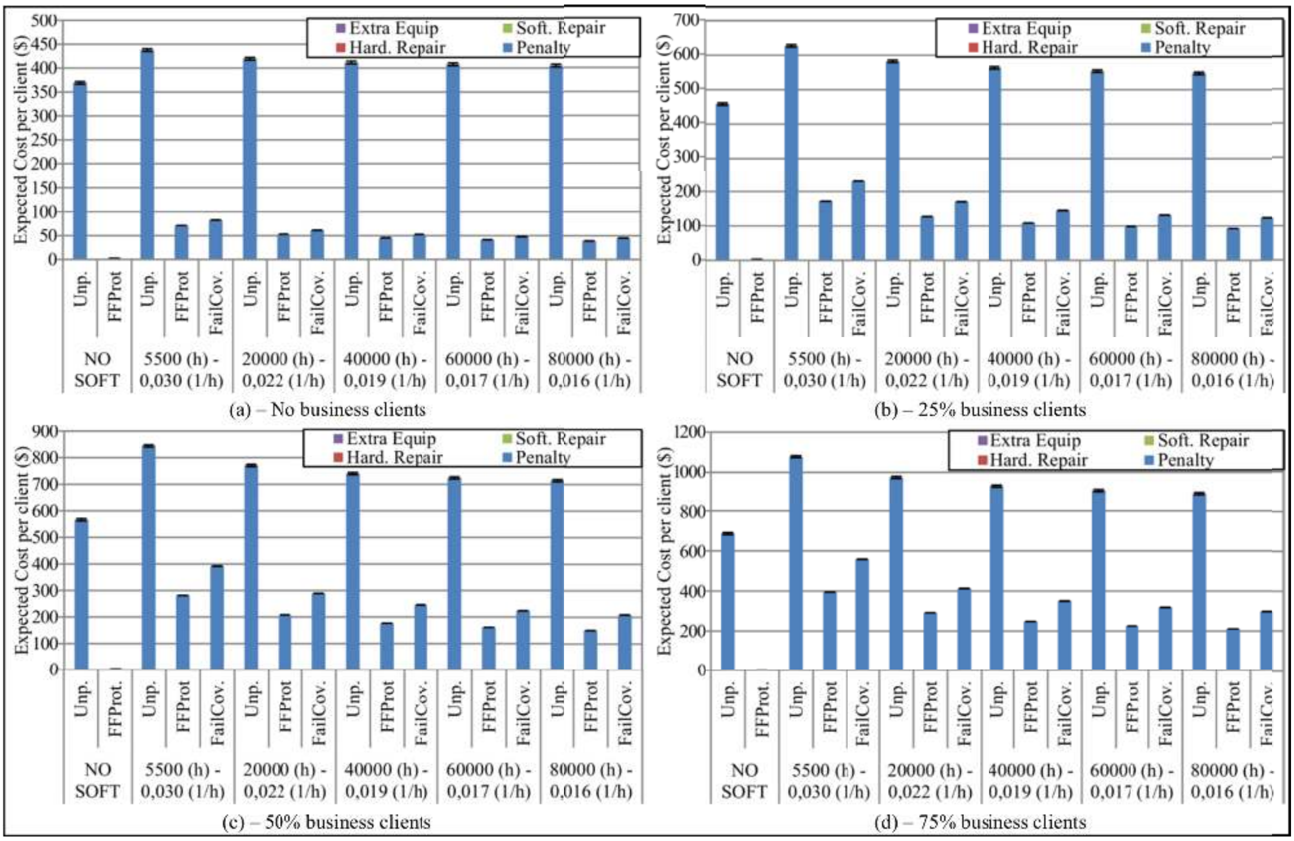


Fig. 5. Expected cost per client in sparse scenarios for different percentages of business clients with varying testing time (software failure intensity).

Decidedly, software failures heavily increase the failure-related OPEX of PONs, especially in dense scenarios and if business users are present. Yet in sparse scenarios feeder fiber failures dominate the OPEX, and it can be reduced with protection, software failures still account for an important part of the OPEX, increasing with the percentage of business users. Regarding imperfect recovery due to software failures, it does not increase the OPEX remarkably in dense scenarios. Yet in sparse deployments, as fibers become more prone to failures, it does contribute to the OPEX, albeit in a lesser way than other failures. Hence, this work also shows that software failures should not be neglected when assessing the OPEX of PONs. Also, increasing the duration of the testing phase reduces the effect of software failures. Yet this reduction is limited for testing times greater than 40000 hours.

Finally, this results call for extending this work to other PON flavors and protection schemes. In Long-Reach PONs, the imperfect recovery impairment must be further analyzed. Similarly, duplicating the OLT is also seen as a cost-effective protection. Yet as software failures tend to propagate, they may markedly hinder this protection scheme. Also, it would be interesting to assess for which failure intensity (testing time) software failures become negligible, yet this has to be accompanied by a feasibility/economic analysis.

References

[1] F. Effenberger *et* al., “An introduction to PON technologies,” *IEEE Commun. Mag.,* Vol. 45, Issue 3, pp. S15 – S25, Mar. 2007.

[2] G. Kramer, M. De Andrade, R. Roy and P. Chowdhury, “Evolution of optical access networks: architectures and capacity upgrades,” *Proc. of the IEEE,* Vol. 100, pp. 1188 – 1196, May 2012.

[3] J. Chen, L. Wosinska, C. M. Machuca and M. Jaeger, “Cost vs. reliability performance study of fiber access network architectures,” *IEEE Commun. Mag.*, Vol. 48, Issue 2, pp. 56 – 65, Feb. 2010.

[4] E. Wong, “Survivable architectures for time and wavelength division multiplexed passive optical networks,” *Journal of Optics Communications*, Vol. 325, pp. 152 – 159, Apr. 2014.

[5] A. Fernandez and N. Stol, “OPEX simulation study of PONs based on a network geometric and Markov cost models,” in *Proc. of ONDM*, Stockholm, Sweden, 2014, pp. 252 - 257.

[6] L. A. Barroso, J. Clidaras and U. Hölzle, “Dealing with Failures and Repairs,” in *The datacenter as a computer: an introduction to the design of warehouse-scale machines*, 2nd ed. Morgan & Claypool Publishers, 2013, ch. 3, sec. 7.2.2, pp. 105 – 107.

[7] A. Fernandez and N. Stol, “On the Impact of Software Failures on Time Division Multiplexed Passive Optical Networks Dependability,” in *Proc. of RNDM*, Barcelona, Spain, 2014, pp. 225 – 231.

[8] A. C. Fadel, R. Moraes and E. Martins, “Automated validation of embedded optical network software,” in *Proc. of LADC 2011*, Sao Jose dos Campos, Brazil, 2011.

[9] C. Jones and O. Bonsignour, *The Economics of Software Quality,* 1st ed., Addison Wesley Professional, 2011.

[10] A. C. Fadel, R. Moraes, P. Martins and E. Martins, “Automating Software Validation of a GPON Network,” in *Proc. of ISSREW*, Pasadena, USA, 2013, pp. 59.

[11] M. R Lyu (ed.), “Software Reliability Modelling Survey,” in *Handbook of Software Reliability Engineering*, 1st ed. Mcgraw-Hill, 1996, ch. 3, sect. 3.5.1, pp. 98 – 99.

[12] *ITU-T Rec. G984.1*, “Gigabit-capable passive optical networks (GPON): general characteristics,” 2008, pp. 7.

[13] B. E. Helvik, “Fault tolerance,” in *Dependable Computing Systems and Communication Networks Design and Evaluation*, 5th ed., Norway: Department of Telematics, NTNU, 2009, ch. 2, sect. 2.3.2, pp. 97 – 102.

[14] G. J. Anders and A. M. Leite da Silva, “Cost related reliability measures for power system equipment,” *IEEE Trans. Power Syst.*, Vol. 15, No. 2, pp. 654 – 660, May 2000.

[15] *D4.2.1: Technical Assessment and comparison of next-generation optical access systems concepts*, OASE Project, 2011, pp. 37 - 38.