Integrity: An Interaction Case Study

Mehdi Khorasani†, Dr. Lionel Sacks‡

† Lucent Technologies, ‡ UCL, Departement of E&EE

Abstract: Integrity has always been a major concern in the telecommunications network, which is one of the most complex real time systems in the globe. This system is becoming increasingly complex. A new level of sophisticated inter-connectivity of GSM, PSTN and data networks is being introduced in the 3rd Generation Mobile Networks.

This paper discusses some of the integrity aspects of such a global system. We will focus on the contributing elements to the integrity problems and complexity issues, using a case study. The case considers a collapse in a traditional telephone system.

1. Introduction

Integrity as a part of computing would never be a major issue if David Hilbert, the leading mathematician of the 20's and 30's, could prove his proposed programme, i.e.: 1- mathematics is complete, 2- mathematics is consistent, and 3- mathematics is decidable, that is there exists an algorithm, which given a mathematical statement S, returns true if S is true and false if S is false. S could be any system. In 1931 Kurt Gödel showed that mathematics is incomplete. In 1936, Alan Turing (and Alonzo Church) showed that the predicate calculus is unpredictable. However Gödel showed that there exists an algorithm, which returns true if S is true, but which may loop forever if S is false. In this way, mechanical computing machines, computers, were born.

Since then many researches have been done and numerous technologies and tools have been developed to address the integrity problems. Specially, studying the behaviour of complex systems on their maximum boundaries has been one of the main focuses for the research communities. Some studies suggest increasing the level of automation to predict and avoid integrity problems. The lessons from AT&T tragedy in January 1990 [1] recommend that *over-automation* caused the big crash. One AT&T 4ESS toll switch in New York City goes down. The rest of the network automatically adjusts itself. The crashed node also recovers itself, but when it comes back to the network, the network goes to a non-recover state.

The study of big crashes show that the major elements, which contribute to an integrity problem, include: the system is under continuous heavy load and/or spiky traffic (e.g. Richard & Judy, 'Guess the Weight', Feb 97, two BT trunk switches crashed), the lack of efficient overload control mechanisms, miss-configuration, human mistakes, components' incompatibilities, wrong level of automation, software 'bugs' and intrusion. To focus on the study, we only mention the main factors.

This case study will focus on the architectural elements and the chain of events, which led to a big 'brown out' in a fixed telephone network around Easter of 2000. The crash was not due to the system not working to specification, or a failure due to poor design, implementation or execution. Rather it was a massive failure, which resulted from convulsion of circumstance.

2. Scenario

To understand the failure, the reader needs to have an appreciation of the basic architecture of the system. This is illustrated in Figure 1. The overall system consists of a number of components. These components can be divided into four major domains: the core network, the Intelligent Network node (SCP node), Other Licensed Operators (OLOs) or the core network gateways to any other networks, and finally end users. To concentrate on the failure, the architecture is hugely simplified.

The core network, almost in all cases, is designed hierarchically with a high level of multiple interconnectivity between the nodes. At the bottom of the pyramid, there are Digital Local Exchanges. The next layer in the hierarchy is Digital Trunk Switch layer, which switch calls between Digital Local Exchanges.

The second major domain is the IN node, which controls the call. In an IN architecture, the core network is referred to as Service Switching Point (SSP) and the IN node is referred to as Service

Control Point (SCP). This architecture allows the implementation of telephony services like pre-paid, post-paid, Free*f*one (0800, 0500), Lo-call (0845, 0345), POPulator Internet, Nationalcall (0990, 0870), etc. An SCP node, by itself, can be a very complex system keeping complicated data base records and providing sophisticated call control. The SCP node in this case study is a complex and vastly distributed platform.

Other components in the architecture concern the core network gateways to other networks. The OLOs interfaces are usually implemented at trunk level. A call maybe originated in another network, being terminated in the target network and vice versa.

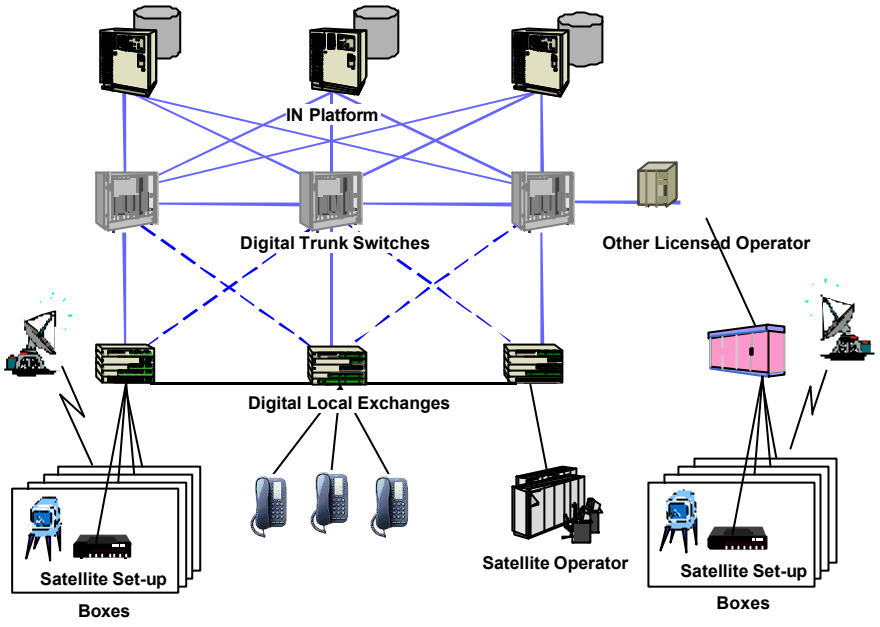

Figure 1 - Typical architecture of a Plain Old Telephony Service (POTS)

The final domain, which plays a major role in this study, is 'end user', which in this case is a Satellite Television Service Provider (we call it STSP). STSP distributes Set-up Boxes (SuBs) to its subscribers. These boxes are attached to homes' fixed telephone lines and are programmed to automatically phone back STSP to report on usage statistics. They used to make calls early mornings. These phone calls, known as telemetry, are done using free phones because STSP is thinking to take the most advantage of the latest technologies to reduce the cost on its subscribers. STSP's subscribers are using telephone lines from different telephone operators. All telemetry calls are terminating in a very small range of free phone numbers in the target operator.

The last missing ring in the chain of fully understanding the scenario is how the overload control mechanism works in the system. For many years, telephone network operators have studied the characteristics of incoming call streams to create a mathematical model for controlling the overall load on the network. This study has found two interesting characteristics of incoming patterns; spiky traffic and continuous heavy load. The study has been focused on televoting as it has both characteristics in it. In 1996, 2.5 millions people phoned two telephone numbers to say 'Yes' or 'No' for ITV's television program 'Monarchy, Yes or No'. It happened only in two hours. The recognized pattern here is 'human repeated attempt', where people attempt to call a number. The number is engaged, they try repeatedly until either they get through or they give up. Based on Statistics and Probability Theory, a model for inter-arrival calls has been designed. This model has been used as a basis for an overload control mechanism, feedback system. To understand feedback systems, we need to know the Call Gap functionality of switches.

Call Gap is a mechanism to restrict the overall number of calls being placed. Call Gap has three basic parameters; interval, duration and (sub-)string destination telephone number.

Call Gap can be applied to a full destination number or a sub-string of a destination number, for instance 0800xxxxxx or 0800 454 35x. IN compliant switches allow an SCP to apply call gaps through their IN interfaces. This type of call gaps is known as automatic call gapping as oppose to manual call gapping, which is set by operational people at switches level.

The idea of feedback system is that SCPs keep track of incoming traffic, calculate a set of gap parameters and ask SSPs to apply the gap. SSPs feed back on these parameters by applying the gap hence changing the rate of incoming traffic to the network and IN node. Based on the new traffic, SCPs calculate a new set of gap parameters and send them down to SSPs. This phenomenon repeats continuously 24 hours a day, seven days a week (forever loop).

The feedback system has been implemented using the leaky bucket algorithm, which is widely used in ATM networks for traffic shaping and policing The SCP node has a throttle system that it uses to manage the number of requests it receives. Each incidence contributes a certain amount to the bucket (a splash) and the bucket leaks at a configurable constant rate. As the bucket 'fills up', various thresholds are passed and these trigger appropriate actions. The design of the leak rate and fill rates (splash values) and thresholds constitutes an implicit model of the external world which is used – for our purposes – to decide when and how to apply call gapping from the SCP node.

3. The Incident

The following discussion identifies a number of Factors, which contributed in various ways to the incident under study.

(**Factor A**) Set-up Boxes of the satellite operator start every day at early morning. to push calls to the network. These boxes have been configured to try to repeat calls to engaged destination numbers for a fixed period, then sleep for a while and try again. This pattern repeats for a day. If not successful, the boxes try the same pattern next day. The danger of computer repeated attempts as oppose to human repeated attempts is that it never gives up until it crashes the whole system! This is clearly an error and most distributed systems theories conclude that a degree of randomisation at this point avoids pathological situations arising.

(**Factor B**) Manual call gapping can be applied to each layer, local exchanges and/or trunk level. It had already been decided to apply call gapping to the satellite operator's free-phone numbers. This had originally been applied at trunk level. At some point, it was decided to move the gapping of the numbers to the local exchange level (a human mistake!). The result of this was that, although all the calls originating within the operator were controlled, those originating from other operators were not; as these enter into the system at the trunk level. If it did not happen, we might have faced a much bigger incident under totally another circumstance with more complex contributors. Who knows … .

The situation at this stage is that at some time in an early morning, the SCP node started to get flooded with 0800 calls. The throttle control part of the SCP node spotted that it was getting a lot of 0800 calls (those originating from other operators) and triggered the call gapping. So far so good. However, at this point one basic error occurred. There is a flag in the IN signalling system which indicates whether a number should be interpreted with or without leading digits such as international part, area code etc. The flag is used to encode and/or decode called party and calling party numbers. (**Factor C**) This flag was set differently on the SCP node than in the core network as a result of an early upgrade (miss-configuration). So, 0800 numbers were coming in, causing the leaky buckets to fill up and a gap request for 800 to be sent to the switches. This had no affect on the incoming calls rate since no '800 xxx' numbers were being generated. However it did have the affect of filling the trunk switch resources up. (**Factor D**) The throttle system of the SCP node never spotted that its model of the world was wrong, i.e. its attempt to control the outside was having no affect.

At this point the trunk switches began to run out of 'IN service' resources and started to send 'resource limitation' signals to the SCP node. (**Factor E**) The SCP node should reject all incoming requests and abort all ongoing activities upon the receiving of 'resource limitation' messages from the switches, but it did not. This, in its turn, uncovered a latent problem with the SCP node causing the SS7 signalling links, connecting the switches to the SCP node, to gradually run out of resources, pushing the SCP node into a congested state. Although the SCP node and SS7 links were under extremely high pressure

they survived but the trunk switches did not. (**Factor F**) Due to high inter-connectivity between network's nodes in order to have backup systems and redundancy, when the first trunk switch fell over – or, indeed started not to be able to handle the offered traffic – load was re-directed to a second trunk switch. Equally, at this time two other SCP nodes, which are offering other types of IN services, also fell over, possibly because of the strange call flows being seen. These various re-directions passed on the problem to other nodes and a big number of trunk switches fell over in less than 5 hours. In general, and in many parts of the telecommunications systems, the degree of interconnection is a critical measure for its integrity and robustness.

There are some unanswered questions like why it happened only in one instance of the SCP node, while many instances of this platform has been deployed into the network? Maybe spikier traffic has come through this node. Figure 2 shows the incoming traffic pattern to one of the free phone numbers of the Satellite operator for a period of 1 minute.

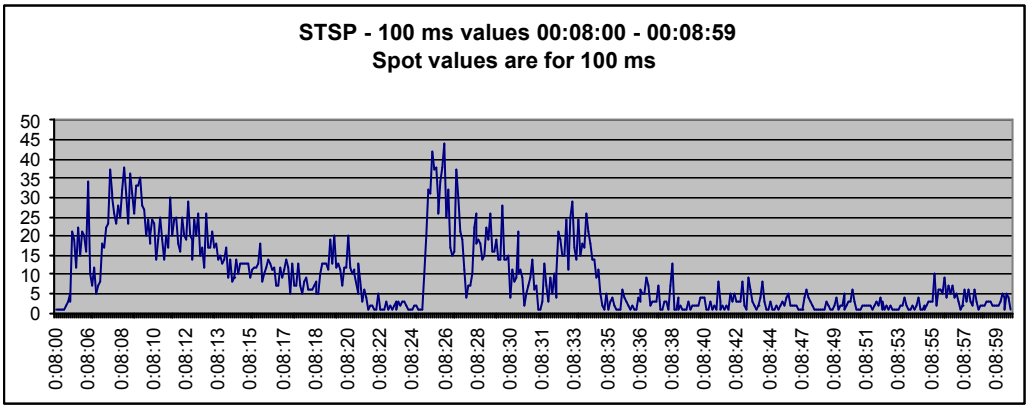

Figure 2 – STSP Incoming Call Pattern

4. Analysis

This incident illustrates a number of important factors. Independently many of these are well understood. Analysis with a complexity point of view provides an important and unifying perspective in the situation. The point here is, in a sense, to turn the emergence question on its head; i.e. to understand failure as an emergent property.

The first important point is that the system had a continuous spiky background pressure or driving force; much as placing sand on a sand pile is the background driving force for Self Organised Criticality in the case of the sand piles of Per-Bac. Indeed, it is clear that things can 'relax' into failure – i.e. when the minimum energy level is not the proper place of operations; and the objective of a lot of control engineering is to ensure that the minimum energy is exactly the proper operational point. In this case, we have an away from equilibrium operational context. The system is always 'driven' by the call rate. This driving force is both an intrinsic part of the operational situation – people make phone calls after all! – and a contributor into the pathology of exceptional operational situations.

Secondly very few of the Factors identified above are actual errors or 'bugs', except that they can be seen as such in hindsight. In fact the only clear ones are that the IN signalling system was not coherently or integrally configured (Factor C) and the SCP node did not act accordingly upon the reception of 'resource limitation' signals (Factor E). Overall this failure can be explained as a linear sequence of events (this happened then that happened), but this mode of explanation is neither correct nor satisfactory. In practice it is necessary to see both the specific events (one evening, the SuBs made calls etc.) and the environmental factors (things where designed thus and so).

One conclusion from this case study is to add a reasonable level of intelligence to the future overload control systems to enable them to protect incessantly and dynamically how close they are modelling the world.

References

[1] ACM, *The Risks Digest*, Volume 9, Issue 63, January 1990.