Apoptosis – the Programmed Death of Distributed Services

Christian Tschudin

Abstract. Active networks enable to deploy new services at run–time by using mobile code. While considerable effort is under way to build active network infrastructures and to understand how to create corresponding services, less is known about how to end them. A particular problem is the coordinated steering of mobile code based services, especially in the case of “strong” active networks where each data packet is replaced by a mobile program and where a distributed service can consists of a myriad of anonymous active packets. In this paper we introduce the concept of apoptosis for mobile code based services. This term is borrowed from cell biology and designates the programmed cell death. We discuss the need for a self–destruction mechanism inside a distributed mobile service and address the problem of securing such a mechanism against malicious activation, for which a simple solution is shown.

Keywords: Active networks, mobile code security, apoptosis.

1 From Mobile Servers to Highly Distributed Services

Most network services today are provided by stationary programs, either at the application level (e-mail) or at the network level (routing). Programmable networks enable to reconfigure the network’s nodes and to bind servers to new physical locations at run-time. “Network-aware services” may choose different server locations for optimizing the quality of service [5]. Similarly, application level gateways were proposed that can perform transcoding or downgrading of multimedia data [1]. Within such proxy architectures, the thin clients – typically mobile devices with wireless links to the fixed network – can program the gateway by uploading servlets. In both cases we have rather large servers and services with a limited amount of mobility.

By reducing the granularity of mobility, we can think of distributed network services that consist of many tiny mobile programs going forth and back. Maintaining routing tables, for example, does not require large server programs sitting somewhere in the network: Small active packets can collect and propagate topology changes to all nodes, updating the routing tables directly. A natural form of mobility of such services consists in having them look around for “unserved” nodes to which the service will attempt to extend. Such services will self-deploy and form a floating, gas-like service cloud that reaches into every niche of the network [12]. It is also possible that such a service cloud retracts from selected nodes because offering the service is not viable anymore for some places due to increased resource competition or lack of clients. For the rest of this paper we will focus on such fine–granular mobile code based network services and call them highly distributed mobile services (HDMS).
1.1 Steering Distributed Mobile Services

In current distribute systems it is the duty of human operators to steer a service. This also includes the update of software because of changes in the protocol standards or bug fixes in the implementation. While this does not happen very frequently with today's large and non-mobile servers, we expect this to be a fairly common case in active networks simply because it is so easy to deploy new functionality. The question then is how to steer such a mobile service in an active network. One of the central steering functionalities will be the termination of a distributed service.

While it is possible to track the heavy weight servers and to establish special control channels to them, such an approach is useless in the HDMS case. The sheer number of tiny elements executing in the network, and the network's topology changes leading to a never-ending adaption process of HDMS services, makes it impossible to locate and address the single elements in a centralized way. Terminating a service in this case will rather be a question of using another HDMS service that will "spread the word". Because the targeted HDMS service knows best how and in which niches it extends, it is useful to build a self-destruction capability into the service. Once triggered by some external event, a termination signal will propagate along the same line as the HDMS service deployed itself. This enables to shut down a HDMS service in an ordered way such that we can speak about the programmed death of a mobile service. The overall effect is a joint service suicide.

The concept of programmed death is known in biology as apoptosis. This word, pronounced either as APE-oh-TOE-sis or uh-POP-tuh-sis, is Greek and stands for "to fall off". In section 2 we first describe apoptosis from a cell biology's point of view. In section 3 we link these observations to the problems of protecting the termination of HDMS services and present a simple way to weld together a specific death signal with the self-destruct routine. We close this paper by a brief discussion of related work and with an outlook.

2 Apoptosis

Research in biology has revealed that cells have a limited capacity to divide (mitosis). This is not due to physical limitations – like for example exploiting some resource beyond usability – but is a predetermined, intrinsic behavior of the cell. Mechanisms at the molecular level are in place that can trigger the self-destruction of a cell. Several reasons have been identified why it makes sense that a cell commits suicide (two recent publications that give an overview of this field are [9, 2]):

**Cell death is as important as mitosis:**

During the growth of an organism and the specialization of cells it is necessary that some cells yield the place they occupy. Fingers, for example, are formed by apoptosis of the tissue between them. Another example is the formation of the connections between neurons in the brain that requires that surplus cells be eliminated by apoptosis.

**Combating cells infected with a virus:**

Cytotoxic T lymphocytes (CTL) can kill virus-infected cells by inducing apoptosis i.e., killing the cell and the virus.