Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 50 (2006) 2519–2531
Computer Networks
www.elsevier.com/locate/comnet

Locating resources in a programmable networking environment

Paul Smith *, Steven Simpson, David Hutchison

Computing Department, InfoLab21, Lancaster University, Lancaster LA14WA, United Kingdom

Available online 5 June 2006

Abstract

Programmable networking is a technology that has been demonstrated to enable the rapid deployment of novel network services. This is achieved through the use of open interfaces that can be used to extend the functionality of a device by third-party service components. Some forms of programmable network allow such components to be deployed out-of-band on a suitable configuration of elements, but do not define mechanisms to determine the configuration.

We present a mechanism to resolve arbitrary service-specific deployment constraints into a suitable node configuration. To focus constraint resolution, we arrange programmable elements into an overlay, and use this to interpolate/extrapolate more favourable locations. Programmable service components are used to evaluate the suitability of individual nodes.

Keywords: Programmable networks; Peer-to-peer systems; Resource discovery

1. Introduction

Critical to the successful operation of a network service is the deployment of service components on appropriate network elements. The services supported by a conventional network element are intrinsic to that element and are static. However, services on a programmable element are dynamic and transient. The challenge in a programmable networking environment is to select from a set of *general purpose* programmable elements a subset that is suitable for deploying a set of service components.

When considering programmable service component deployment, two main approaches have been adopted to date: the in-band (or capsule) approach, where computational components are included with or referenced from within data traffic that is to be augmented [23,22]; or the out-of-band approach, where computational elements are loaded prior to the traffic's arrival by a third-party process [4,15]. The latter approach requires a process to select the appropriate set of programmable elements to deploy components on, unlike the former, where component deployment is integral to the manner in which the service is executed. In this paper, we are concerned with the out-of-band approach to service deployment.

Recently in particular, the peer-to-peer networking paradigm has proved to be a compelling means of implementing large-scale distributed systems. In a peer-to-peer system all entities should be capable of carrying out the same role as each of their peers, essentially conducting both client and server operations. This enables a system to be highly robust to failure and attack, as compromising a single entity only has local effect. Another benefit of such

* Corresponding author.
E-mail address: p.smith@comp.lancs.ac.uk (P. Smith).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2006.04.023

systems, and perhaps the catalyst for their prolific deployment, is the low organisational and financial burdens associated with introducing services. However, these properties typically come at the cost of a more challenging lookup problem in distributed resource sharing scenarios (than in client–server based systems, for example), and quite often inefficient network resource usage.

We present a critical part of a service-deployment architecture that enables the identification of programmable entities based upon a set of service-specific deployment constraints. We propose an architecture that is realised using the peer-to-peer networking paradigm in order to inherit the benefits associated with such systems. We present an algorithm, called GROUPNET, that can be used to construct a mesh between programmable elements that enables us to infer locations to search for superior elements. Programmable service components that encapsulate appropriate search algorithms and service-specific deployment constraints are used to determine suitable node configurations. To enable such components to determine element suitability, programmable elements expose open interfaces that enable the interrogation of local state.

In Section 2, we continue with a discussion of issues related to service-deployment constraints. Following that, in Section 3 an overview of the framework we advocate for programmable resource discovery is presented. An introduction to GROUPNET is given in Section 4, demonstrating our group membership algorithm and how to perform searches over it. Entities necessary for determining element suitability are presented in Section 5. In Section 6, we present an example service-deployment scenario and show how our approach can be used. An evaluation showing properties of searching over GROUPNET meshes is shown in Section 7. In Section 8, related work is discussed. Finally, in Section 9 we present the conclusions of our work.

2. Service-deployment constraints

Service-specific deployment constraints define suitable configurations of programmable devices on which to deploy service components. Deployment constraints can be described in terms of a number of *dimensions* plus acceptable *values* within those dimensions. For example, the network latency between a set of network end-points can be considered a constraint dimension. Acceptable values within such a dimension will prescribe a set of elements suitable for executing a service (for example, suitable hosts have an application-level round-trip delay below a given threshold).

In some cases, values within constraint dimensions can be traded against others in different dimensions. For example, a customer may be willing to pay a particular price in proportion to the Quality of Service (QoS) obtained. Values within some dimensions can change in relation to network topology. For example, a telephone call may cost less when the end-points are geographically nearer. Programmable services that demonstrate these properties are described in [5,17].

Entities that define constraint dimensions and acceptable values include the service user (an organisation or individual user), the programmable network (administrators), and the service itself. We believe that constraints can largely be organised into one of three categories based upon the entities they relate to. These categories can be used to suggest the manner in which acceptable values within dimensions can be resolved. A description of these categories follows:

1. *Policy-related constraints:* These constraints relate to organisational requirements defined by both the service user and the programmable network administrator. Service users may define policies that focus the set of suitable programmable devices, for example their choice of service provider. Likewise, network administrators may also define policies that relate to, for example, the sets of users that can execute services on their devices.
2. *Network-related constraints:* For network services to operate correctly, certain network-related characteristics must prevail. For example, for a real-time video conferencing application to function correctly, there must be appropriate network bandwidth available between participants. A discussion regarding these and other forms of network-related application requirements is presented in [9].
3. *Device-related constraints:* It is probable that a deployed programmable network infrastructure will be heterogeneous. This heterogeneity will relate to both the software and hardware systems in use. An element may intrinsically support a range of software and hardware services, such as available execution environments or network processors, for example. The availability of some resources will be highly transient, for example those of memory and processing.