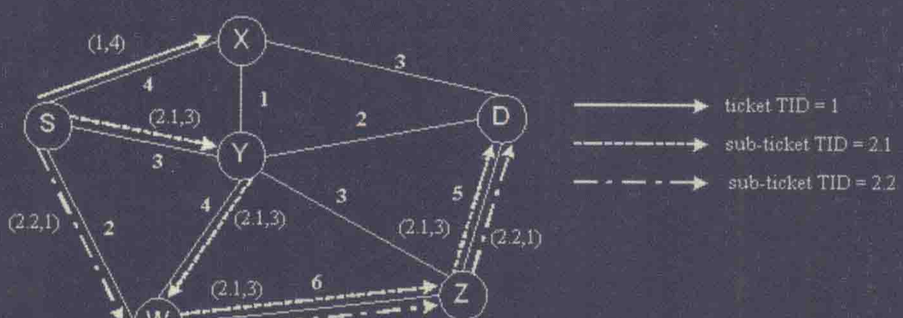
Maria Carla Calzarossa
Erol Gelenbe (Eds.)

Tutorial

LNCS 2965

Performance Tools and Applications to Networked Systems

Revised Tutorial Lectures



Springer

Maria Carla Calzarossa   Erol Gelenbe (Eds.)

Performance Tools and Applications to Networked Systems

Revised Tutorial Lectures

江苏工业学院图书馆
藏书章

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Maria Carla Calzarossa
Università di Pavia
Dipartimento di Informatica e Sistemistica
via Ferrata 1, 27100 Pavia, Italy
E-mail: mcc@alice.unipv.it

Erol Gelenbe
Imperial College
Department of Electrical and Electronic Engineering
London SW7 2BT, UK
E-mail: e.gelenbe@imperial.ac.uk

Library of Congress Control Number: 2004104315

CR Subject Classification (1998): C.2, C.4, E.1, D.2

ISSN 0302-9743
ISBN 3-540-21945-5 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 10998946 06/3142 5 4 3 2 1 0

Preface

This volume is dedicated largely to the performance-oriented design of modern computer networks, both wired and wireless. It is the consequence of the Tutorial Session which was held on 12th October 2003, preceeding the IEEE Computer Society's Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, held in Orlando, Florida. In addition to the core tutorial presentations, which covered both advances in network quality of service (QoS) and in performance evaluation methodology, we felt that it would be useful to assemble a volume in the Lecture Notes in Computer Science series which would also include specific application areas of performance modeling and measurements. Thus the current volume includes three parts:

- a first part that specifically addresses performance and QoS of modern wired and wireless networks;
- a second one that discusses current advances in performance modeling and simulation; and
- a final part that addresses other specific applications of these methodologies.

The network-oriented portion of the volume itself comprises three complementary topics:

- a first group of chapters deals with novel designs and new issues related to broad-based network performance and QoS, without limitations concerning the connection technologies that are being used;
- a second group addresses the wireless context; and
- a final grouping of contributions considers the topic of wireless ad hoc networks more specifically.

The part of the volume dealing with methodologies discusses the software specification of models, the use of certain formalisms such as Petri nets, and some recent advances in simulation. Finally, the third part of the volume discusses other performance applications related to scheduling and to specific architectures.

The first paper in the part on general networking issues provides a useful survey of Content Delivery Networks which is of great current interest. The second paper also addresses a question of major importance concerning the modeling of computer virus propagation. Electronic mail still remains central to the role of networks, and the performance of mail systems is addressed in a third paper. The research represented in these three contributions was supported via the national Italian research project FIRB–PERF on computer and network performance. These papers are followed by a novel packet network protocol design, the Cognitive Packet Network (CPN), which provides QoS-driven routing in wired and wireless networks; this paper also gives detailed measurement results concerning user performance and QoS obtained in an experimental CPN testbed. Issues of network reliability and path restoration in mesh networks are discussed in the fifth paper.

The integration of wireless and wired networks and the use of novel wireless systems as they become available is discussed first in a contribution on the wireless internet and then also in a contribution on the performance of systems that exploit 2.5/3G wireless. The role of the IP protocol in the wireless context is also examined in a separate paper.

Ad hoc wireless networks are discussed in a set of three papers, the first of which examines the role of peer-to-peer computing in this context, followed by another contribution that studies the role of multipath routing. A review of ad hoc wireless algorithm designs is presented in the third paper covering this area.

The section on performance evaluation methodologies begins with a paper on performance management which discusses certain general issues related to QoS in systems. The combination of UML and Petri nets is discussed in a second paper, while a novel extension to the performance evaluation tool PEPA so as to include the possibility of evaluating computer networks is discussed in the third paper of this group. A contribution on the use of simulation within a visually "true" augmented reality environment completes the set of methodologically oriented papers.

This volume closes with two papers on significant scheduling problems, the first in the area of distributed systems, and the second on mass memories and disk arrays.

We believe that this volume constitutes a very useful tool for both the practitioner and the researcher. To the practitioner we offer a set of comprehensive pointers to performance and QoS issues and we feel that the didactic style and the numerous references provided in each paper can be of great use in understanding how the field can help solve practical problems. For the researcher, we have consciously selected a set of contributions not only for their research value but also for their novelty and use in identifying areas of active research where much further work can be done.

February 2004

Maria Carla Calzarossa
Erol Gelenbe

此为试读，需要完整PDF请访问：www.ertongbook.com

Table of Contents

A Walk through Content Delivery Networks

Novella Bartolini[1]*, Emiliano Casalicchio[2], and Salvatore Tucci[2]

[1] Università di Roma "La Sapienza", Via Salaria 113 - 00198 Roma, Italy,
novella@dsi.uniroma1.it
[2] Università di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy,
casalicchio@ing.uniroma2.it, tucci@uniroma2.it

Abstract. Content Delivery Networks (CDN) aim at overcoming the inherent limitations of the Internet. The main concept at the basis of this technology is the delivery at edge points of the network, in proximity to the request areas, to improve the user's perceived performance while limiting the costs. This paper focuses on the main research areas in the field of CDN, pointing out the motivations, and analyzing the existing strategies for replica placement and management, server measurement, best fit replica selection and request redirection.

1 Introduction

The commercial success of the Internet and e-services, together with the exploding use of complex media content online has paved the way for the birth and growing interest in Content Delivery Networks (CDN). Internet traffic often encounters performance difficulties characteristic of a non dedicated, best effort environment. The user's urgent request for guarantees on quality of service have brought about the need to study and develop new network architectures and technologies to improve the user's perceived performance while limiting the costs paid by providers. Many solutions have been proposed to alleviate the bottleneck problems and the most promising are based on the awareness of the content that has to be delivered. The traditional *"content-blind"* network infrastructures are not sufficient to ensure quality of service to all users in a dynamic and ever increasing traffic situation. New protocols and integrated solutions must be in place both on the network and on the server side to distribute, locate and download contents through the Internet.

The enhancement of computer networks by means of a content aware overlay creates the new architectural paradigm of the CDN. Today's CDN act upon the traditional network protocol stack at various levels, relying on dynamic and proactive content caching and on automatic application deployment and migration at the edge of the network, in proximity to the final users. Content replicas in a CDN are geographically distributed, to enable fast and reliable delivery to any end-user location: through CDN services, up-to-date content, can be retrieved by end-users locally rather than remotely.

* The work of Novella Bartolini has been funded by the WEB-MINDS project supported by the Italian MIUR under the FIRB program

M.C. Calzarossa and E. Gelenbe (Eds.): MASCOTS 2003, LNCS 2965, pp. 1–25, 2004.
© Springer-Verlag Berlin Heidelberg 2004

CDNs were born to distribute heavily requested contents from popular web servers, most of all image files. Nowadays, a CDN supports the delivery of any type of dynamic content, including various forms of interactive media streaming. CDN providers are companies devoted to hosting in their servers the content of third-party content providers, to mirroring or replicating such contents on several servers spread over the world, and to transparently redirecting the customers requests to the 'best replica' (e.g. the closest replica, or the one from which the customer would access content at the lowest latency). Designing a complete solution for CDN therefore requires addressing a number of technical issues: which kind of content should be hosted (if any) at a given CDN server (replica placement), how the content must be kept updated, which is the 'best replica' for a given customer, which mechanisms must be in place to transparently redirect the user to such replica. A proper placement of replica servers shortens the path from servers to clients thus lowering the risk of encountering bottlenecks in the non-dedicated environment of the Internet. A request redirection mechanism is provided at the access routers level to ensure that the best suited replica is selected to answer any given request of possibly different types of services with different quality of service agreements. The CDN architecture also relies on a measurement activity that is performed by cooperative access routers to evaluate the traffic conditions and the computational capacity and availability of each replica capable of serving the given request. Successfully implemented, a CDN can accelerate end user access to content, reduce network traffic, and reduce content provider hardware requirements.

This paper explores architectures, technologies and research issues in content delivery networks [53]. In section 2 we describe the core features of a CDN, discussing the motivations and how content delivery can alleviate internet performance problems. In section 3 we examine types of content and services that can beneficiate from content delivery techniques. Section 4 describes the architecture and working principles of a CDN. A detailed discussion on replica placement and management is provided in section 5. Section 6 introduces the problem of how measures can be taken to select the replica that can better fulfil an incoming request, while request redirection mechanisms are described and compared in section 7. Section 8 concludes the paper.

Let us point out that this paper is related to other papers contained in this volume. Issues related to QoS are discussed in several papers in this volume, including [8] [31] [42], while content delivery is related to peer-to-peer networking which is discussed in [39]. Content is often of multimedia nature, such as augmented reality [30] which will have high bandwidth and significant QoS needs, and the type of tools described in [32] can contribute simpler evaluation tools which are applicable to the systems we discuss.

2 Motivations for Content Delivery

Internet users commonly get frustrated by low performances and may decide to abandon a web site or to disconnect a multimedia session when experiencing performance difficulties, causing revenue to be lost. Though centralized models are still in place in the Internet today, these architectures are poor in terms of adaptivity and scalability.

If a provider establishes a content server in a single physical location from which it disseminates data, services, and information to all its users, the single server is likely to become overloaded and its links can easily be saturated. The speed at which users can access the site could become unpredictably higher than the maximum request rate the server and its links can tolerate. Since it is impossible, with this approach, to adapt to the exponential growth of the Internet traffic, the centralized model of content serving is inherently unscalable, incapable of adaptivity and produces performance losses when traffic bursts occur. Though this leads to the conclusion that a certain amount of servers must be adopted, a cluster of servers (also known as *server farm*, that is a multi-server localized architecture, is not necessarily a solution yet.

The server computational and link capacity is only the first source of performance difficulties that may be encountered while downloading content over the Internet. There are many other possible congestion causes that may lead to unacceptable user perceived quality. The non dedicated, best effort nature of the Internet is the inborn limit to the possibility of having any sort of performance guarantee while delivering content over it.

The Internet is a network of heterogeneous networks composed of thousands of different autonomous systems ranging from large backbone providers to small local ISPs. The autonomous systems connect to each other creating the global Internet. The communication between two networks is achieved through the connection of border routers in a peering session. Two peer routers periodically exchange routing information and forward the received packets to carry each packet to its correct destination. This structure of the Internet as an interconnection of individual networks is the key to its scalability but is not sufficient to guarantee that a quickly growing number of users, services and traffic do not create bottlenecks that, if left unaddressed, can slow down performance. Bottlenecks may occur at many points in the core Internet and most of all in correspondence to peering points and backbones.

The network capacity is determined by the capacity of its cables and routers and although cable capacity is not an issue, the strongest limit to the backbone capacity comes from the packet-forwarding hardware and software of the routers. Once a peering point has been installed, traffic may have grown beyond expectations, resulting in a saturated link, typically because a network provider purchases just enough capacity to handle current traffic levels, to maximize the link utilization. The practice of running links at full capacity is one of the major causes of traffic bottlenecks showing very high utilization but also high rates of packet loss and high latency. Further the capacity of long backbones cannot always be adapted to the sudden and fast increases of the Internet traffic.

2.1 Move the Content to the Edges: An Approach to Improve the Internet Performance

The current centralized or partially distributed model of Internet content distribution requires that all user requests and responses travel several subnetworks and, therefore, traverse many possibly congested links. The first solution adopted to distribute the content trough the Internet consisted in mirroring. This technique statically replicates the web content in many locations across the Internet. Users manually select, from a list of servers, the best suited replica. The replica selection mechanism was automated and became transparent to the end-users with the introduction of the distributed web server systems [11][10].

With the introduction of proxy caching techniques to disseminate the content across the Internet, the bottlenecks at the server level and at the peering points were considerably reduced, though not ensuring a complete controllability of those systems by the content provider due to the absence of an intelligent and automated layer to perform server measures and request redirection. Proxy caching is only a semi-transparent mechanisms: the users, aware of the presence of a proxy server in their network, can/must configure their browser to use it, while the ISPs transparently manage their proxy caches. Large ISP proxy caches may also transparently cooperate with each other in a semi hierarchical structure. Proxy caches may experiment performance losses becoming themselves a bottleneck if there are frequent cache misses or cache inconsistencies. Besides this, proxy caches serve all requests independently of the required content, and do not prioritize users and QoS requirements.

In a CDN, by moving the content from multiple servers located at the edge of the Internet, a much more scalable model of distributing information and services to end-users is obtained, that is the so called edge delivery. In other words, a user would be able to find all requested content on a server within its home network. In this solution the requested content doesn't cross all the network before reaching its final destination, but only traverses the network part between the edge and the end-user. Further, cooperative access routers can be endowed with measure and server selection capabilities to perform a tradeoff solution between load balancing among the available servers and choosing the best suited replica to fulfil the agreements on quality of service.

2.2 The Features of a CDN

The design of a CDN requires, together with the distribution of replica servers at the edge of the network, a set of supporting services and capabilities. In order to be efficient for a significant number of users and for a considerably wide area, the edge servers must be deployed in thousands of networks, at different geographically spread locations. Optimal performance and reliability depend on the granularity of the distribution of the edge servers. The establishment of a CDN requires the design of some important features.

- **Replica placement mechanisms** are needed to decide the replica server locations and to adaptively fill them with the proper content prior to the

此为试读，需要完整PDF请访问： www.ertongbook.com

request arrival (pre-fetching). Thus servers are not filled upon request like in traditional proxy caching, but are pro-actively updated, causing a one time offloading overhead that is not repeated for every access to the origin server. Adaptivity in replica placement is required to cope with changing traffic condition and is not related to a pull behavior like in traditional caching.
- **Content update mechanisms** must be provided to automatically check the host site for changes and retrieve updated content for delivery to the edges of the network, thus ensuring content freshness. Standard mechanisms adopted in proxy caching do not guarantee content freshness since content stored on standard cache servers does not change as the source content changes.
- **Active measurement mechanisms** must be added to cooperative access routers to have immediate access to a real-time picture of the Internet traffic, in order to recognize the fastest route from the requesting users to the replica servers in any type of traffic situations, especially in presence of "flash crowds", that is sudden heavy demand, expected or not, for a single site. A measurement activity is at the basis of the replica selection mechanism.
- **Replica selection mechanisms** must be added to cooperative access routers to accurately locate the closest and most available edge server from which the end users can retrieve the required content. A robust service must also keep its servers from getting overloaded by means of access control and load balancing.
- **Re-routing mechanisms** must be able to quickly re-route content requests in response to traffic bursts and congestion as revealed by the measurement activity.

Also, the CDN infrastructure, must allow the service providers to access directly the caches and control their consistency and to get the statistics information about the accesses to the site, available from the cooperative access routers.

3 Types of Content and Services in a CDN

CDN providers host third party contents to fasten the delivery of any type of digital content, e.g. audio/video streaming media, html pages, images, formatted documents or applications. The content sources could be media companies, large enterprises, broadcasters, web/Internet service provider. Due to the heterogeneous nature of the content to be delivered, various architectures and technologies can be adopted to design and develop a CDN. We now analyze the characteristics of the content and of the applications that most likely take advantages of a CDN architecture.

- **Static web based services.** Used to access static content (static html pages, images, document, software patches, audio and/or video files) or content that change with low frequency or timely (volatile web pages, stock quote exchange). All CDN provider (Akamai Inc., Speedera Inc., AT&T inc., Globix Inc. just to mention some) support this type of content delivery. This type of content can easily be cached and its freshness maintained at the edge using traditional content caching technologies.

- **Web storage services.** Essentially, this application can be based on the same techniques used for static content delivery. Additional features to manage logging and secure file transfer should be added. This type of application can require processing at the origin site or at the edge.
- **File transfer services.** World wide software distribution (patch, virus definition, etc.), e-learning material from an enterprise to all their global employees, movies-on-demand from a large media company, highly detailed medical images that are shared between doctors and hospitals, etc. All these content types are essentially static and can be maintained using the same techniques adopted for static web services.
- **E-commerce services.** The semantic of the query used in browsing a product catalogue is not complex, so frequent query results can be successfully cached using traditional DB query caching techniques[29][33]. Shopping charts can be stored and maintained at the replica server and also orders and credit card transactions can be processed at the edge: this requires trusted transaction-enabled replica servers. In [9] the authors propose a framework for enabling dynamic content caching for e-commerce site.
- **Web application.** Web transactions, data processing, database access, calendars, work schedules, all these services are typically characterized by an application logic that elaborates the client requests producing as results a dynamic web page. A partial solution to the employment of a CDN infrastructure in presence of dynamic pages is to fill the replica servers with the content that most frequently composes the dynamically generated web pages, and maintaining the application and its processing activity that produces the dynamic pages at the origin server. Another approach is to replicate both the application (or a portion of it) and the content at the edge server. In this way all the content generation process (application logic and content retrieval) are handled by the replica server thus offloading the origin server.
- **Directory services.** Used for access to database servers. For example, in the case of a LDAP server, frequent query results or a subsets of directories can be cached at the edge. Traditional DB query caching techniques [29] may be adopted.
- **Live or on-demand streaming.** In this case the edge server must have streaming capability. See section 3.1 for details.

Streaming media and application delivery are a challenge in CDN. A more detailed description of the solutions adopted for media streaming and dynamic contents can be found in the following subsections.

3.1 Streaming Media Content

Streaming media can be *live* and *on-demand*, thus a CDN needs to be able to deliver media in both these two modes. *Live* means that the content is delivered "instantly" from the encoder to the media server, and then onto the media client.

This is typically used for live events such as concerts or broadcasts. The end-to-end delay is at a minimum 20 seconds with today's technologies, so "live mode" is effectively "semi real-time". In *on-demand*, the content is encoded and then stored as streaming media files on media servers. The content is then available for request by media clients. This is typically used for content such as video or audio clips for later replay, e.g., video-on-demand, music clips, etc. A specialized server, called a media server, usually serves the digitalized and encoded content. The media server generally consists of media server software that runs on a general-purpose server. When a media client wishes to request a certain content, the media server responds to the query with the specific video or audio clip. The current product implementations of streaming servers are generally proprietary and demand that the encoder, server, and player all belong to the same vendor. Streaming servers also use specialized protocols (such as RTSP, RTP and MMS) for delivery of the content across the IP network. In [55] a typical streaming media CDN architecture is described. In streaming media CDNs a replica server must have, at least, the additional functionalities listed below.

- The ability to serve *live content* such as newscasts, concerts, or meetings etc. either in Multicast or Unicast mode.
- Support for delivery of stored or *on-demand content* such as training, archived meetings, news clips, etc.
- *Caching capability* of streaming media. Caching a large media file is unproductive, so typically media files are split in segment. Neighbor replica must be capable to share and exchange segment to minimize the network load and cache occupancy.
- *Peering capability* to exchange and retrieve content from the neighbor streaming cache in case of cache miss. Streaming cache node can be organized in a hierarchy.
- *Media transcoding functionality*, to adapt media streams for different client capabilities, e.g., low quality/bandwidth content to dial-up users, high quality/bandwidth to xDSL users.
- *Streaming session handoff* capability. The typically long life of a streaming session, in presence of user's mobility, causes the need for midstream handovers of streaming session between replica servers [5,49].

3.2 Web Application

Accessing dynamic content and other computer applications is one of the major challenges in CDN. CDN supporting this kind of content and services are also called Application Content Delivery Networks (ACDN). Some providers like AppStream Inc. and PIVIA Inc., implement ACDN using the so called "fat client" solution: the application is partitioned in "streamlets" or special applets and sent to the client. The client receives enough code to start the application and execute it, the other parts of the application are sent on demand. These solution use patented and proprietary technologies. Another approach is to migrate the application to the edge server using general utility such as Ajasent[1] and vMatrix[4]. However application replication may be expensive especially if performed on demand. A completely different solution is to automatically deploy the application at the replica server. In [47] the authors define an ACDN architecture relying on standard technologies such as HTTP protocol, web servers, CGI/FastCGI scripts or servlets. Rabinovich et al. define the additional capabilities of an ACDN in terms of: an *application distribution framework* capable to dynamically deploy the application at the edge and to keep the replica consistent, a *content placement mechanism* to decide where and when to deploy the application, a *request distribution mechanism* aware of the location of the involved applications.

4 Content Delivery Networks Architecture

The main goal of server replication in a CDN is to avoid large amounts of data repeatedly traversing possibly congested links on the Internet. As Figure 1 shows, there are a variety of ways and scale (local area or wide area networks) in which content networks may be implemented. Local solutions are web clusters, that typically hosts single site, and web farms, typically used to host multiple sites. Wide area solutions include: distributed web server systems, used to host single or multiple sites; cooperative proxy cache networks (a service infrastructure to reduce latency in downloading web objects) and content delivery networks [53] that are the focus of this paper.

A typical server farm is a group of servers, ranging from two to thousands, that makes use of a so-called cooperative dispatcher, working at OSI layers 4 and/or 7, to hide the distributed nature of the system, thus appearing as a single origin site. A layer 4 web switch dispatches the requests, among a group of servers, on the basis of network layer information such as IP address and TCP port. A content switch, working at the application layer, examines the content of requests and dispatches them among a group of servers. The goals of a server cluster/farm include: load-balancing of requests across all servers in the group; automatic routing of requests away from servers that fail; routing all requests

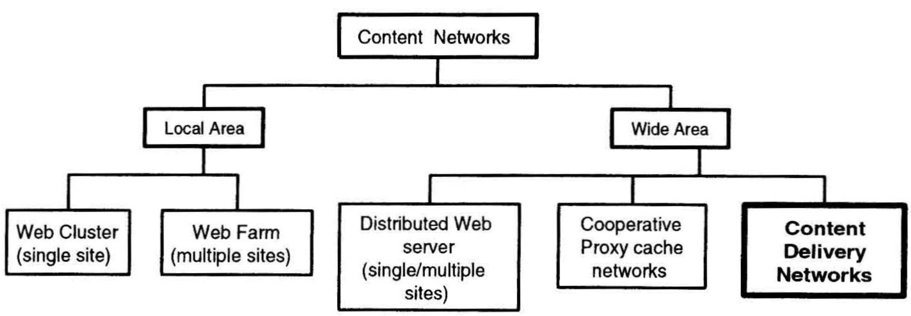


Fig. 1. Taxonomy of Content Networks

for a particular user agent's session to the same server, if necessary to preserve session state.

A type of content network that has been in use for several years is a caching proxy deployment. Such a network might typically be employed by an ISP for the benefit of narrow bandwidth users accessing the Internet. In order to improve performance and reduce bandwidth utilization, caching proxies are deployed close to the users. These users are encouraged to send their web requests through the caches rather than directly to origin servers, by configuring their browsers to do so. When this configuration is properly done, the user's entire browsing session goes through a specific caching proxy. This way the proxy cache would contain the hot portion of content that is being viewed by all the users of that caching proxy. A provider that deploys caches in many geographically locations may also deploy regional parent caches to further aggregate user requests thus creating an architecture known as hierarchical caching. This may provide additional performance improvements and bandwidth savings. Using rich parenting protocols, redundant parents may be deployed such that a failure in a primary parent is detected and a backup is used instead. Using similar parenting protocols, requests may be partitioned such that requests for certain content domains are sent to a specific primary parent. This can help to maximize the efficient use of caching proxy resources. Clients may also be able to communicate directly with multiple caching proxies.

Though certainly showing better scalability than a single origin server, both hierarchical caching and server farms have their limits. In these architectures, the replica servers are typically deployed in proximity to the origin server, therefore they do not introduce a significant improvement to the performance difficulties that are due to the network congestion. Caching proxies can improve performance difficulties due to congestion (since they are located in proximity to the final users) but they cache objects reactively to the client demand. Reactive caching based on client demand performs poorly if the requests for a given object, while numerous in aggregate, are spread among many different caching proxies.

To address these limitations, CDNs employ a solution based on proactive rather than on reactive caching, where the content is prefetched from the origin server and not cached on demand. In a CDN, multiple replicas host the same content. A request from a browser for a single content item is directed to the replica that is considered the best suited at the moment of the request arrival, and the item is served to the client in a shorter time than the one it would have taken to fetch it from its origin server. Since static information about geographic locations and network connectivity are not sufficient to choose the best replica, a CDN typically incorporates dynamic information about network conditions and load on the replicas, to redirect requests and balance the load among the servers. Operating a CDN is therefore a complex and expensive activity. For this reason a CDN is typically built and operated by a network/service provider that offers a content distribution service to several content providers.

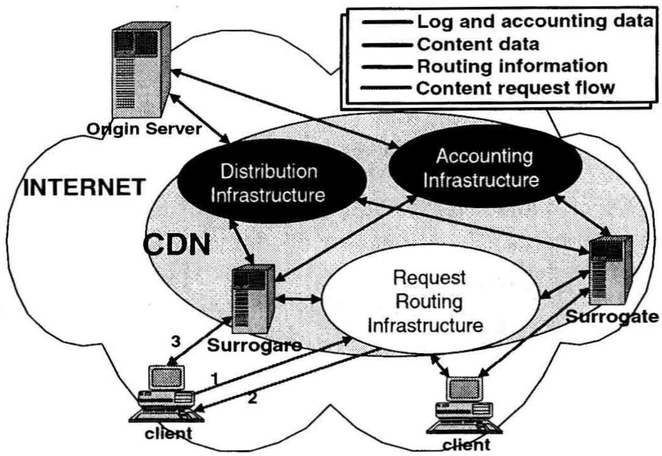


Fig. 2. Infrastructure components of a Content Delivery Network

A content delivery architecture consists of a set of **surrogate servers** that deliver copies of content to the users while combining different activities (see figure 2).

- the **request-routing** infrastructure consists of mechanisms to redirect content requests from a client to a suitable surrogate.
- the **distribution** infrastructure consists of mechanisms to move contents from the origin server to the surrogates.
- the **accounting** infrastructure tracks and collects data on request-routing, distribution, and delivery functions within the CDN creating logs and reports of distribution and delivery activities.

The **origin server** (hosting the content to be delivered) interacts with the CDN in two ways (see figure 2):

- it pushes new content to the replica servers, (the replica themselves request content updates from the origin server through the distribution infrastructure);
- it requests logs and other accounting data from the CDN or the CDN itself provides this data to the origin server through the accounting infrastructure.

The **clients** interact with the CDN through the request routing infrastructure and surrogate servers. Figure 2 shows one of the possible scenarios of interaction between the clients, the access routers, the replica servers and the origin server.

The user agent sends (1) a content request to the routing infrastructure, that redirects (2) the client request to a surrogate server, to which the client subsequently asks (3) the desired content.

此为试读，需要完整PDF请访问：www.ertongbook.com