

# A survey into CDNs Security

Lucas Begnini Costa<sup>1</sup>, Carlos A. Maziero<sup>1</sup>

<sup>1</sup>LARSIS - Departamento de Informatica – Universidade Federal do Paraná (UFPR)  
Curitiba – PR – Brazil

lucasbegnini@gmail.com,

**Abstract.** *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

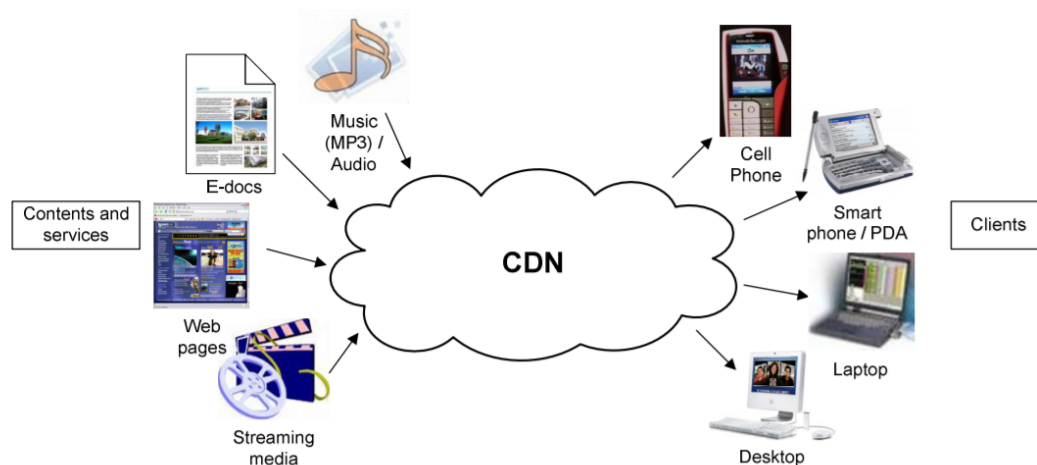
**Resumo.** *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

## 1. Introdução

Vivemos em um mundo rodeado de tecnologia, onde a cada dia somos surpreendidos com uma coisa totalmente inovadora, disruptiva. Uma era onde a internet foi responsável por atravessar mares, superar distâncias e até mesmo idiomas. Hoje se pode comunicar em tempo real com pessoas que estão em lados completamente opostos ao seu.

Hoje é possível que empresas estrangeiras muito distantes fisicamente, como China, Índia, EUA e etc, forneçam serviços para regiões mais remotas do mundo. Isso inclui serviços de multimídia como armazenamento de fotos, de vídeos e até conteúdos de consumo instantâneo.

Esse encurtamento de distância pode parecer simples, mas vem de um sistema complexo que visa fornecer ao usuário final uma experiência agradável com conteúdos entregues de maneira satisfatória mas sem, necessariamente, replicá-lo por todo o globo. O que nos leva a dizer que uma CDN (*Content Delivery Network*) é uma rede de distribuição de conteúdo que tem como objetivo fornecer ao usuário de aplicações globais uma experiência satisfatória na utilização de serviços, principalmente sob demanda.



Temos vários serviços que se utiliza no dia a dia onde essa noção de CDN é completamente abstrata ao usuário final. Como serviços de Video-On-Demand de empresas de TV, Spotify, Amazon Prime Video e até Netflix, como é mostrado no artigo do [Adhikari et al. 2012].

Existem hoje diversas empresas que fornecem esse serviço ao redor do globo. Como:

- Akamai;
- Limelight;
- Level 3;
- e etc.

Essas três redes são hoje, as principais fornecedoras de serviço de CDN da Netflix. Cada uma com uma característica e voltado para um público.

### 1.1. AKAMAI

**Origem** - Massachusetts Institute of Technology (MIT), em 1995, com Tim Berners-Lee.

**Figura 1. Rede de distribuição Akamai**



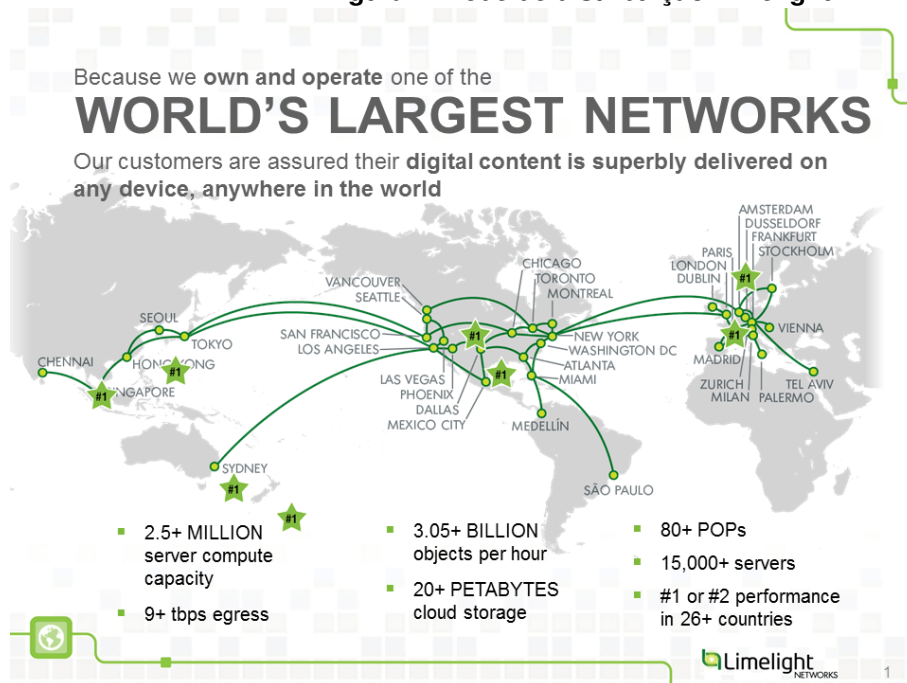
## Pontos de atuação

**Principais clientes** - Adobe, Airbnb, American Idol, Audi, Autodesk, EMC2, e muitas outras.

## 1.2. Limelight

**Origem** - Tempe, Arizona, EUA.

**Figura 2. Rede de distribuição Limelight**



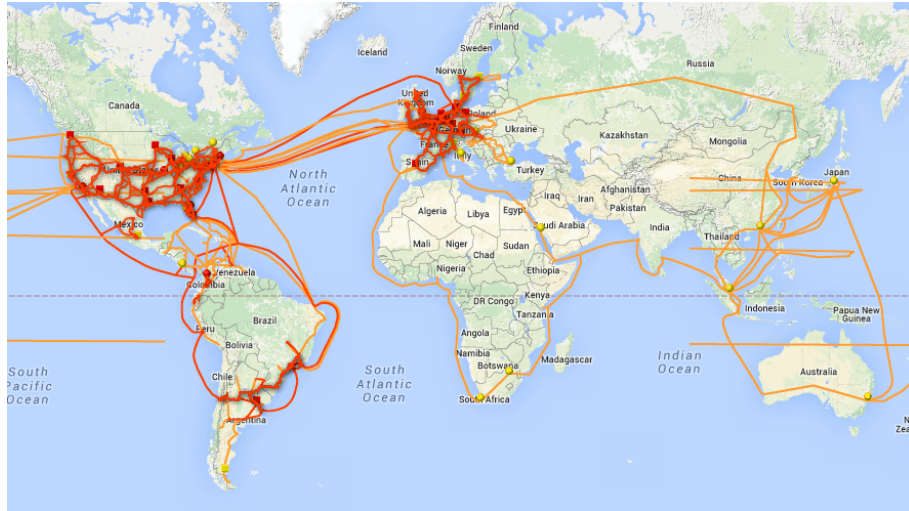
## Pontos de atuação

**Principais clientes** - Não informado.

## 1.3. Level 3

**Origem** - Monroe, Luisiana, EUA.

**Figura 3. Rede de distribuição Level 3**



## **Pontos de atuação**

**Principais clientes** - Não informado.

## **1.4. Visão geral**

Todas essas redes se concentram praticamente no mesmo ponto. É claro, movida por uma questão financeira, todas elas estão basicamente concentradas nos Estados Unidos, Europa e algumas adentram até o mercado asiático.

Nesse artigo vamos tentar entender um pouco mais a respeito de CDNs e também um pouco do seu sistema de segurança. Sendo assim, poderemos dividir o restante desse artigo em duas partes: A primeira será relacionada somente a composição da CDN, ou seja, tudo aquilo que é necessário para se constituir uma CDN. Na segunda etapa falaremos primeiro um pouco sobre segurança em um âmbito geral e depois aprofundaremos em seus conceitos voltados para CDN.

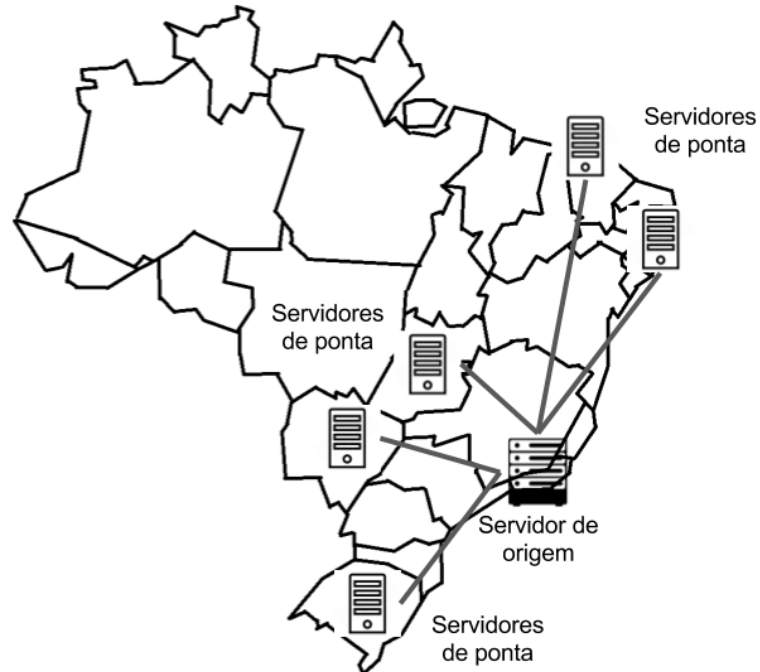
## **2. Composição de uma CDN**

Uma CDN pode ser definida segundo os seguintes pontos:

- Organização;
- Servidores;
- Relacionamentos;
- Protocolos de interações;
- E tipos de conteúdo.

## 2.1. Tipos de servidores

**Figura 4. Tipos de servidores**



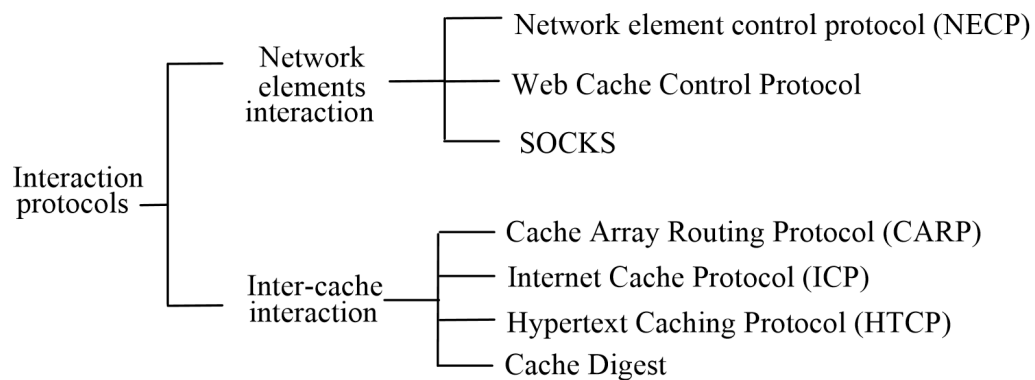
Os servidores são definidos dos seguintes modos:

- Servidor de origem
- Servidor de ponta

Podemos ilustrá-los conforme a figura 4.

## 2.2. Protocolos de interações

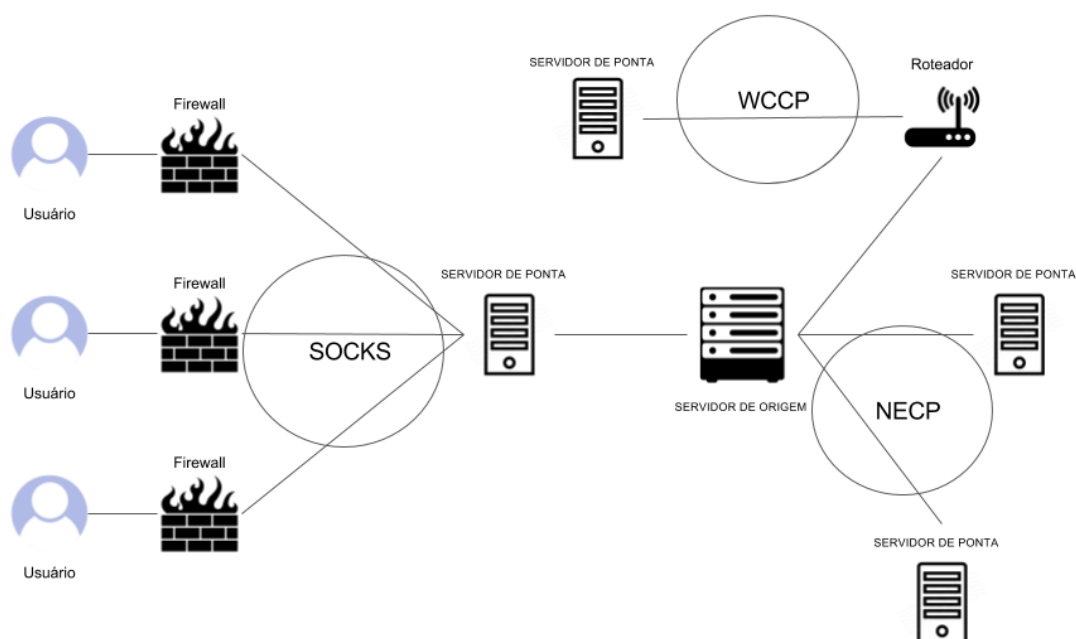
**Figura 5. Tipos de relacionamentos**



Os protocolos de interações podem ser divididos em duas partes: Protocolos de interações de elementos da rede e Protocolos de interações entre os servidores de cache da CDN.

### 2.2.1. Interações dos elementos da rede

**Figura 6. Tipos de protocolos de interações**



Dentro dos protocolos de interações dos elementos de rede podemos verificar que cada um possui sua especificidade e funcionalidade bem definida, como podemos ver na figura 6 , tentando proteger o não só a rede mas também o usuário, o servidor, os roteadores e a comunicação entre os mesmos.

### 2.2.2. Interações de cache

Os protocolos de interações de cache são protocolos que organizam as trocas de informações entre os servidores, ou seja, é ele que dita como irá funcionar a distribuição da informação dentro da rede.

Conforme vimos na figura 5, e segundo [Pathan and Buyya 2007], existem 4 tipos de protocolos aplicados nessa circunstância, que são:

- HTCP - Hypertext Caching Protocol
- ICP - Internet Cache Protocol

Ambos são concorrentes entre si e tem como funcionalidade controlar o fluxo de informação entre os caches. Sendo através deles que se controla o que irá para um determinado servidor de ponta, por exemplo. Falaremos mais sobre ambos em 3.2.3 e 3.2.4 respectivamente.

Existe também os protocolos:

- CARP - Cache Array Routing Protocol
- Cache Digest

Esses dois protocolos, também concorrentes entre si, servem para controlar o conteúdo existente dentro de cada servidor e saber onde estão os outros conteúdos. Falaremos mais sobre ambos em 3.2.6 e 3.2.7 respectivamente.

### **2.2.3. HTCP**

Como dito anteriormente o HTCP, Hypertext Caching Protocol, é um protocolo de interação entre os caches, suas principais características são:

- Protocolo para descobri Caches HTTP;
- Suporte ao HTTP 1.0;
- Permite incluir cabeçalhos nas respostas;
- Podem ser enviados via TCP/UDP;
- Devem ser resilientes à falhas.

### **2.2.4. ICP**

Já o ICP, Internet Cache Protocol, é um protocolo muito mais leve que possui as seguintes características:

- Protocolo de mensagem leve;
- Utilizado para comunicação de Caches;
- Utiliza consultas para determinar localização mais apropriada;
- Suporte ao HTTP 0.9;
- Comunica-se com caches vizinhos;
- recebe MISS ou HIT como resposta;
- Enviado via UDP;
- Falha por timeout indica caminho quebrado;
- Fornece informações para balanceamento através das medidas de perda.

### **2.2.5. HTCP x ICP**

Analisando os dois protocolos, HTCP e ICP, podemos fazer um quadro comparativo entre os e colocá-los da seguinte maneira (figura 7):

**Figura 7. HTCP x ICP**

| Serviços                        | HTCP | ICP |
|---------------------------------|------|-----|
| Envio TCP                       | ✓    | ✓   |
| Envio UDP                       | ✓    |     |
| Suporte HTTP 1.0                | ✓    |     |
| Permite enviar apenas cabeçalho | ✓    |     |
| Monitora caches remotos         | ✓    |     |
| Permite monitoramento de falhas |      | ✓   |

### **2.2.6. CARP**

CARP - Cache Array Routing Protocol

Protocolo de armazenamento distribuído baseado em uma lista conhecida de proxies suavemente acoplada e uma função hash para dividir o espaço URL entre esses proxies.

- Cliente HTTP pode enviar requisição à qualquer proxy da lista.

### **2.2.7. Cache Digest**

Cache Digest

Protocolo de intercâmbio e formato de dados entre caches.

- Fornecem um resumo dos conteúdos na resposta;
- Soluciona os problemas de congestionamento e timeout;
- Torna possível determinar se um servidor possui em cache um conteúdo;
- Executado via HTTP ou FTP;
- Contém tempo de expiração na resposta;
- Podem ser utilizados para eliminar redundância.



## **2.3. Seleção e entrega de conteúdo**

Dentro de uma CDN temos que nos preocupar com a forma como esse conteúdo vai catalogado, armazenado e distribuído dentro da rede, o que vimos no item 3.2, como também temos que nos preocupar como esse conteúdo vai chegar até o cliente (usuário) da forma mais otimizada possível, ou seja, o servidor o qual vai fornecer as informações para ele será o mais perto ou mais rápido.

Temos que destacar também a importância da otimização do fluxo de informação pela rede. Visto que quanto maior o tráfego de informação pela rede significa que a informação está mais distante do usuário e também que vai ter um custo maior pela troca intensa de informação.

Segundo [Krishnamurthy et al. 2001], na tentativa de otimizar o redirecionamentos de URL para o usuário se sacramentou dois tipos de técnicas de redirecionamentos:

- Full - site
- Partial - site

### **2.3.1. Full - site**

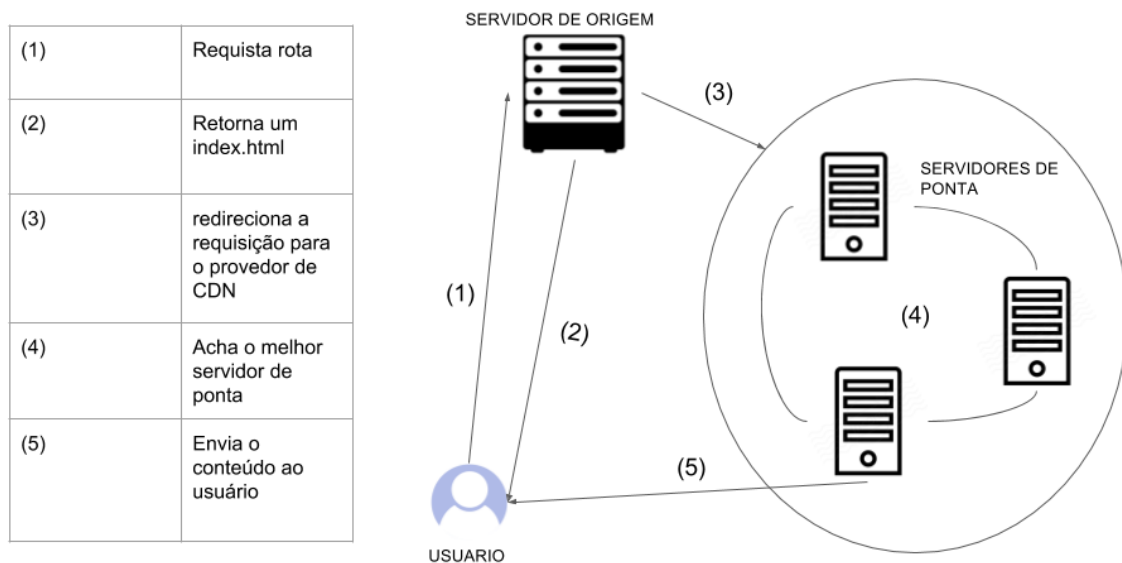
Na técnica de Full-site todo o conteúdo é entregue ao usuário de um servidor ponta único. Ou seja, o usuário faz uma requisição ao servidor principal, onde o mesmo processa um algoritmo de roteamento para encontrar o servidor ponta que melhor se enquadra como resposta, e então retorna ao usuário o endereço onde então será consumido por fim todas as informações requisitadas.

É importante salientar que essa técnica é amplamente utilizada por serviços que fazem pouco uso de dados da rede. Uma página estática da web, por exemplo, se encaixaria perfeitamente nesse contexto. Visto que possui baixo grau de modificações e seu tamanho é pequeno perto de outros tipos de mídias que circulam na web.

### **2.3.2. Partial - site**

Já redirecionamentos do tipo Partial-sites os servidores principais retornam para o usuário uma parte do conteúdo e disparam, automaticamente, um algoritmo de roteamento para encontrar o restante da informação e retornar ao usuário. Conforme podemos ver na figura 8

**Figura 8. Entrega de conteúdo**



Nela podemos ver que todo o processo acontece em, basicamente, 5 etapas. (1) o usuário faz uma requisição ao servidor principal, depois, em (2) o servidor principal retornar um html com as principais informações e dispara automaticamente (3) um processo de roteamento (4) para buscar o melhor servidor e retornar (5) para o usuário os conteúdos.

Entretanto há em (4) diversas formas de fazer esse roteamento quanto a distribuição do conteúdo pela rede e quanto a aglomeração desse conteúdo dentro do servidores de cache.

Tipo de distribuição nada mais é do que a forma como o conteúdo vai ser disperso na rede, como esse conteúdo vai se aproximar do nó que está mais perto do usuário.

Os tipos de distribuição mais frequentemente utilizados são:

- Empírico
- Popularidade

Empírico trata, como o próprio nome diz, de uma forma de distribuição sem nenhum conhecimento específico a respeito, utilizando-se apenas um conhecimento experimental de onde seria melhor posicionado o conteúdo.

Em um esquema baseado em popularidade a distribuição é feita conforme a notoriedade. Ou seja, quanto mais a requisição do conteúdo mais ele vai ficar nos servidores

de ponta perto do usuário.

Ambos esquemas não são necessariamente excludentes, pode-se inicialmente aplicar a forma empírica para gerar dados a respeito da distribuição e depois utilizar esses dados para aplicar o modo de popularidade.

Também existe as formas de aglomerações de conteúdo. Isso existe porque os conteúdos podem ser conjuntos de objetos ou objetos independentes. As formas de aglomerações são:

- Objeto
- Conjunto de objetos

Aglomeração por objeto vai juntar os objetos mais selecionados e distribui-los individualmente entre os servidores. Já por conjunto de objetos ele vai separa-los em grupos e distribui-los em conjuntos.

Podemos exemplificar da seguinte forma: Em um site temos vários elementos, temos o HTML, temos o CSS e temos a media de um vídeo qualquer. Podemos separar da seguinte forma temos 3 elementos onde 2 deles são altamente dependentes(HTML e CSS) e temos u que pode ser diferente em cada região do país que é o vídeo. Então podemos misturar as formas de aglomerações, para o HTML e para o CSS agrupamos em conjuntos de objetos, e para o vídeo fazemos a aglomeração por objeto, já que será distribuído de maneira independente nos servidores.

Analisando os tipos se perceber que nenhuma das opções são excludentes entre si. Pode-se combinar quaisquer tipo de distribuição e aglomeração e também. As escolhas vão depender das necessidades de cada aplicação.É necessário uma análise minuciosa de cada aplicação para chegar em um veredito da melhor abordagem.

### 2.3.3. VOD

**VOD - Video on Demand** - Segundo [Garfinkle 1996], é um sistema de que proporciona uma interface de comunicação com o usuário de produtos disponíveis de uma estação central remota.

É um sistema muito utilizado por operadoras de conteúdo onde é o cliente que decide o horário que irá consumir determinado conteúdo.

Isso permite que pessoas que antes tinham que esperar horário certo para consumir determinado conteúdo, agora podem a qualquer momento e em qualquer lugar fazer uso do mesmo. Visto que está disponível online.

**Microsoft Smooth Streaming** É um sistema de consumo de vídeo onde a qualidade do vídeo transmitida vai ser definida conforme a qualidade da banda disponível. Clientes que

possuem alta disponibilidade de banda terão maior qualidade do vídeo.

Para conseguir tocar um vídeo nesse formato o player do usuário tem que ser capaz de interpretar um manifesto que contém dentro de outras coisas, o caminho de onde está localizado a mídia desejada.

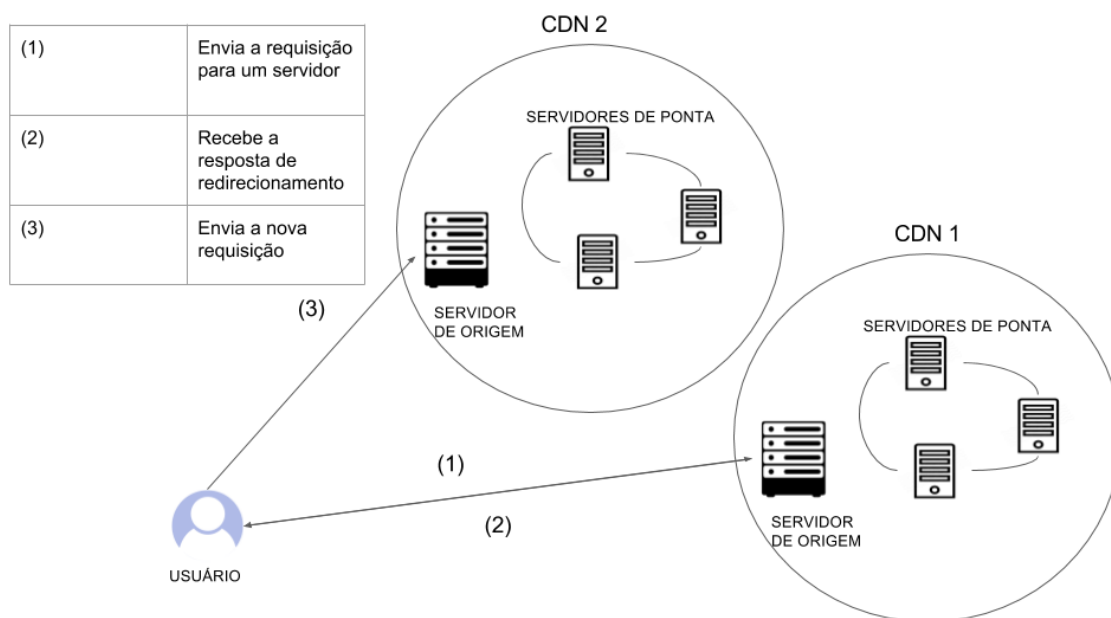
Já para conseguir fazer a transição de qualidade o vídeo é quebrado em pequenos pedaços chamados de "chunks". Então, conforme é verificada um aumento ou decréscimo da banda disponível é só o player começar a consumir chunks de qualidade diferente da atual e assim o usuário já passará a ver a diferença significativa na tela.

No artigo [Zambelli 2009] podemos ver todas as aplicações e implicações dessa forma de consumo de mídia.

#### 2.3.4. Exemplo

Agora vamos ilustrar com um exemplo mais prático. Utilizaremos uma requisição de uma URL de um VOD(Video On Demand) onde a parte de roteamento é feita no usuário. A aplicação do usuário faz uma requisição à CDN e enquanto o cabeçalho da resposta não for 200 ele vai ler um campo dentro do cabeçalho de resposta e fazer uma nova requisição. A figura 9 ilustra a situação.

**Figura 9. Redirecionamento de CDN**



Como podemos observar na figura 10 a aplicação manda para o player a url associada ao VOD que é responsável por responder com o caminho oficial, ou intermediário, do manifesto. Na figura 10 ele está localizado no campo "play\_info" da requisição.

Figura 10. exemplo VOD

| No.   | Time     | Source       | Destination  | Protocol | Length | Info  |
|-------|----------|--------------|--------------|----------|--------|---|
| 12278 | 2.081222 | 192.168.15.8 | 192.168.15.7 | HTTP     | 288    | PUT /player/18cd39/stop HTTP/1.1  |
| 12328 | 2.084946 | 192.168.15.8 | 192.168.15.7 | HTTP     | 339    | PUT /player/18cd39/play HTTP/1.1 (application/json)   |
| 29501 | 4.105385 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1  |
| 29521 | 4.107044 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found  |
| 29815 | 4.131296 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |
| 30697 | 4.201873 | 201.0.52.15  | 192.168.15.7 | HTTP     | 635    | HTTP/1.1 200 OK (application/manifest)  |
| 31441 | 4.287982 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1  |
| 31456 | 4.289669 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found  |
| 31590 | 4.300898 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |
| 36377 | 4.844038 | 201.0.52.15  | 192.168.15.7 | MP4      | 724    |   |
| 36782 | 4.869234 | 201.0.52.15  | 192.168.15.7 | MP4      | 1542   |   |
| 37006 | 4.954702 | 192.168.15.7 | 201.0.52.15  | HTTP     | 322    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |

> Frame 12328: 339 bytes on wire (2712 bits), 339 bytes captured (2712 bits) on interface 0  
 > Ethernet II, Src: Digibras\_87:db:dd (64:1c:67:87:db:dd), Dst: ArrisGro\_b8:39:dc (20:f1:9e:b8:39:dc)  
 > Internet Protocol Version 4, Src: 192.168.15.8, Dst: 192.168.15.7  
 > Transmission Control Protocol, Src Port: 42096, Dst Port: 80, Seq: 225, Ack: 1, Len: 273  
 > [2 Reassembled TCP Segments (497 bytes): #12294(224), #12328(273)]  
 > Hypertext Transfer Protocol  
 > JavaScript Object Notation: application/json  
 > Object  
 > Member Key: play\_info  
 > String value: http://o2.b38489.cdn.telefonica.com/38489/00/00/89/895872\_BA471EA6FDED1B47/BRA\_HD\_US\_169\_\_c6b6ea465fa14f84.ism/manifest?ifmt=mss  
 > Key: play\_info  
 > Member Key: timeshift\_enabled  
 > Member Key: metadata

O player, por sua vez, como responsável por enviar as requisições, envia ao servidor principal a requisição do manifest do VOD que ele quer tocar. Figura 11.

Figura 11. exemplo VOD

| No.   | Time     | Source       | Destination  | Protocol | Length | Info  |
|-------|----------|--------------|--------------|----------|--------|---|
| 12278 | 2.081222 | 192.168.15.8 | 192.168.15.7 | HTTP     | 288    | PUT /player/18cd39/stop HTTP/1.1  |
| 12328 | 2.084946 | 192.168.15.8 | 192.168.15.7 | HTTP     | 339    | PUT /player/18cd39/play HTTP/1.1 (application/json)   |
| 29501 | 4.105385 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1  |
| 29521 | 4.107044 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found  |
| 29815 | 4.131296 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |
| 30697 | 4.201873 | 201.0.52.15  | 192.168.15.7 | HTTP     | 635    | HTTP/1.1 200 OK (application/manifest)  |
| 31441 | 4.287982 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1  |
| 31456 | 4.289669 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found  |
| 31590 | 4.300898 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |
| 36377 | 4.844038 | 201.0.52.15  | 192.168.15.7 | MP4      | 724    |   |
| 36782 | 4.869234 | 201.0.52.15  | 192.168.15.7 | MP4      | 1542   |   |
| 37006 | 4.954702 | 192.168.15.7 | 201.0.52.15  | HTTP     | 322    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169__c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1 |

> Frame 29501: 311 bytes on wire (2488 bits), 311 bytes captured (2488 bits) on interface 0  
 > Ethernet II, Src: ArrisGro\_b8:39:dc (20:f1:9e:b8:39:dc), Dst: Tellesco\_aa:9b:f9 (10:72:23:aa:9b:f9)  
 > Internet Protocol Version 4, Src: 192.168.15.7, Dst: 201.0.52.116  
 > Transmission Control Protocol, Src Port: 49166, Dst Port: 80, Seq: 1, Ack: 1, Len: 245  
 > Hypertext Transfer Protocol  
 > GET /38489/00/00/89/895872\_BA471EA6FDED1B47/BRA\_HD\_US\_169\_\_c6b6ea465fa14f84.ism/manifest?ifmt=mss HTTP/1.1\r\n  
 Host: o2.b38489.cdn.telefonica.com\r\n  
 User-Agent: libcurl/7.43.0 OpenSSL/1.0.1t zlib/1.2.8\r\n  
 Accept: \*/\*\r\n  
 Accept-Encoding: deflate, gzip\r\n  
 \r\n  
 [Full request URI: http://o2.b38489.cdn.telefonica.com/38489/00/00/89/895872\_BA471EA6FDED1B47/BRA\_HD\_US\_169\_\_c6b6ea465fa14f84.ism/manifest?ifmt=mss]\r\n  
 [HTTP request 1/1]

Na figura 12 vemos a CDN, que já fez um roteamento para uma outra rede(ou servidor), que retornou com um código 302 indicando que o retorno da requisição não é a

resposta final. Dentro do cabeçalho da resposta tem um campo chamado "location" o qual é responsável por retornar o endereço final (ou intermediário) para nova requisição. Esse processo será repetido até o código de retorno da requisição for 200.

Figura 12. exemplo VOD

| http   |          |              |              |          |        |  |
|--|----------|--------------|--------------|----------|--------|--|
| No.  | Time     | Source       | Destination  | Protocol | Length | Info   |
| 12278  | 2.081222 | 192.168.15.8 | 192.168.15.7 | HTTP     | 288    | PUT /player/18cd39/stop HTTP/1.1                             |
| 12328  | 2.084946 | 192.168.15.8 | 192.168.15.7 | HTTP     | 339    | PUT /player/18cd39/play HTTP/1.1 (application/json)          |
| 29501  | 4.105385 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c  |
| 29521  | 4.107044 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found   |
| 29815  | 4.131296 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c |
| 30697  | 4.201873 | 201.0.52.15  | 192.168.15.7 | HTTP     | 635    | HTTP/1.1 200 OK (application/manifest)                       |
| 31441  | 4.287982 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c  |
| 31456  | 4.289669 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found   |
| 31590  | 4.300898 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c |
| 36377  | 4.844038 | 201.0.52.15  | 192.168.15.7 | MP4      | 724    |  |
| 36782  | 4.869234 | 201.0.52.15  | 192.168.15.7 | MP4      | 1542   |  |
| 37006  | 4.954702 | 192.168.15.7 | 201.0.52.15  | HTTP     | 322    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c |
| > Frame 29521: 459 bytes on wire (3672 bits), 459 bytes captured (3672 bits)   |          |              |              |          |        |  |
| > Ethernet II, Src: Tellesco_aa:9b:f9 (10:72:23:aa:9b:f9), Dst: ArrisGro_b8:39:dc (20:f1:9e:b8:39:dc)                                      |          |              |              |          |        |  |
| > Internet Protocol Version 4, Src: 201.0.52.116, Dst: 192.168.15.7  |          |              |              |          |        |  |
| > Transmission Control Protocol, Src Port: 80, Dst Port: 49166, Seq: 1, Ack: 246, Len: 393   |          |              |              |          |        |  |
| v Hypertext Transfer Protocol  |          |              |              |          |        |  |
| > HTTP/1.1 302 Found\r\n   |          |              |              |          |        |  |
| Server: TelCdn/0.1\r\n   |          |              |              |          |        |  |
| Date: Fri, 01 Dec 2017 17:08:53 GMT\r\n  |          |              |              |          |        |  |
| Connection: Close\r\n  |          |              |              |          |        |  |
| > Content-Length: 0\r\n  |          |              |              |          |        |  |
| Location: http://o2.b38489-p0-h62.6.cdn.telefonica.com/_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/manifest |          |              |              |          |        |  |
| Access-Control-Allow-Headers: X-TCDN\r\n   |          |              |              |          |        |  |
| Access-Control-Expose-Headers: X-TCDN\r\n  |          |              |              |          |        |  |

Ainda na figura 12 podemos observar que ele recebe um 200 OK da requisição em seguida do 302 Found. O que simboliza que aquela URL é a URL final e o player pode utiliza-la como base do manifesto.

Figura 13. exemplo VOD

| http  |          |              |              |          |        |  |
|-------|----------|--------------|--------------|----------|--------|--|
| No.   | Time     | Source       | Destination  | Protocol | Length | Info   |
| 29501 | 4.105385 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1  |
| 29521 | 4.107044 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found   |
| 29815 | 4.131296 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1 |
| 30697 | 4.201873 | 201.0.52.15  | 192.168.15.7 | HTTP     | 635    | HTTP/1.1 200 OK (application/manifest)   |
| 31441 | 4.287982 | 192.168.15.7 | 201.0.52.116 | HTTP     | 311    | GET /38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1  |
| 31456 | 4.289669 | 201.0.52.116 | 192.168.15.7 | HTTP     | 459    | HTTP/1.1 302 Found   |
| 31590 | 4.300898 | 192.168.15.7 | 201.0.52.15  | HTTP     | 321    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1 |
| 36377 | 4.844038 | 201.0.52.15  | 192.168.15.7 | MP4      | 724    |  |
| 36782 | 4.869234 | 201.0.52.15  | 192.168.15.7 | MP4      | 1542   |  |
| 37006 | 4.954702 | 192.168.15.7 | 201.0.52.15  | HTTP     | 322    | GET /_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1 |
| 40391 | 5.304082 | 192.168.15.7 | 201.0.52.15  | HTTP     | 325    | [TCP ACKed unseen segment] [TCP Previous segment not captured]   |
| 48581 | 6.256095 | 201.0.52.15  | 192.168.15.7 | HTTP     | 1506   | [TCP ACKed unseen segment] [TCP Previous segment not captured]   |

> Frame 37006: 322 bytes on wire (2576 bits), 322 bytes captured (2576 bits)

> Ethernet II, Src: ArrisGro\_b8:39:dc (20:f1:9e:b8:39:dc), Dst: Tellesco\_aa:9b:f9 (10:72:23:aa:9b:f9)

> Internet Protocol Version 4, Src: 192.168.15.7, Dst: 201.0.52.15

> Transmission Control Protocol, Src Port: 40969, Dst Port: 80, Seq: 193, Ack: 84000, Len: 256

▼ Hypertext Transfer Protocol

> GET /\_38489/00/00/89/895872\_BA471EA6FDED1B47/BRA\_HD\_US\_169\_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000) HTTP/1.1

Host: o2.b38489-p0-h62.6.cdn.telefonica.com\r\n

User-Agent: Mozilla/5.0-OpenSTB-2017.11.27.00.01.59-MSS\r\n

Accept: \*/\*\r\n

\r\n

[Full request URI: [http://o2.b38489-p0-h62.6.cdn.telefonica.com/\\_38489/00/00/89/895872\\_BA471EA6FDED1B47/BRA\\_HD\\_US\\_169\\_c6b6ea465fa14f84.ism/QualityLevels\(400000\)/Fragments\(video=20020000\)](http://o2.b38489-p0-h62.6.cdn.telefonica.com/_38489/00/00/89/895872_BA471EA6FDED1B47/BRA_HD_US_169_c6b6ea465fa14f84.ism/QualityLevels(400000)/Fragments(video=20020000)) HTTP/1.1]

[HTTP request 2/5]

[Next request in frame: 52827]

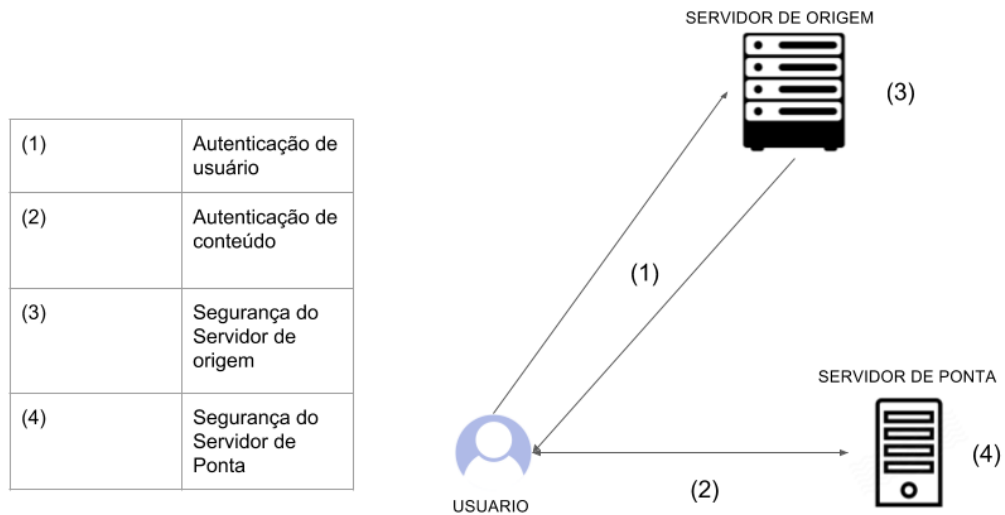
Em 13 vemos o player começando a consumir os "chunks" e por consequência tocando o vídeo.

### 3. Segurança de uma CDN

Segundo [Ferreira 2004] segurança é definido como um conjunto de ações que e dos recursos utilizados para proteger algo ou alguém, ou, o que serve para diminuir os riscos ou perigo. Portanto, segurança é um tema extramente abrangente e complexo. Podemos falar de tópicos como segurança física, social, de um sistema ou até mesmo de um grupo de servidores interconectados, que é o caso da CDN, e ainda coloca-los todos dentro do mesmo grupo. Ou seja, podemos tratá-los de forma separada ou analisando o conjunto todo.

Ainda dentro de Segurança de CDN podemos tratar de pontos principais, onde o vazamento é ponto crucial e mais lógico de ser atacado.

**Figura 14. Segurança**



Como vemos na figura 14 podemos observar se tem quatro principais pontos a serem garantidos em uma rede. O primeiro trata-se da autenticação de usuário que é capacidade do sistema de garantir que aquele usuário que está acessando o conteúdo tem mesmo os requisitos para acessá-lo. Ou seja, é mesmo um usuário do sistema. Por ser um tema bem abrangente será tratado com mais profundidade em 4.3.

Em segundo lugar temos a autenticação de conteúdo. Que é a garantia que o conteúdo acessado é o mesmo buscado pelo usuário, que o usuário tem acesso a ele e também é um conteúdo que faz parte da rede de disponibilidade da CDN(que não é um conteúdo inserido por terceiro sem autorização prévia). Tudo isso tem que ser minuciosamente tratado e averiguado antes de retornar ao usuário. Tudo isso será abordado em 4.4.

Nos dois outros, (3) e (4), trata-se da segurança dos servidores. Aí podemos falar de segurança física e lógica. Tem-se que levar em conta a forma de acesso de cada um na hora de mensurar os aspectos de segurança de cada um deles.

No servidor de origem se tem um acesso via DNS, com um endereço "legível". Esse acesso se dará muitas vezes por várias parte do globo, tendo que deixá-lo disponível, portanto, vulnerável à diversos tipos de ataque, o quê o torna extremamente complexo na hora de definir suas regras de segurança, não basta apenas subir um *firewall* bloqueando múltiplos acessos, é preciso saber exatamente os tipos de aplicações que serão tratadas para assim começar a desenhar as regras de *firewall* que serão aplicadas, e também, na maioria dos casos, será necessário a implementação de protocolos de proteção dentro da própria aplicação que rodará no servidor.

Já no servidor de ponta a preocupação com o acesso via DNS já é uma coisa a menos, visto que muitas vezes o acesso funcionará via redirecionamento do servidor de origem, sendo o controle de endereço feito pelo o mesmo. Mas há diversos outros aspectos que tem que serem levados em conta, como parte da autenticação de conteúdo para o usuário,



que nada mais é que garantir que o usuário para o qual está sendo enviado o conteúdo tem acesso ao mesmo. Outra grande preocupação é quanto a sua disponibilidade, pois é necessário que o mesmo esteja "de pé" quando for requisitado conteúdo e que durante o processo de transferência, caso haja alguma interrupção a mesma seja informada ao servidor de origem e o usuário transferido para outro servidor sem que haja muitos danos à experiência. Isso sem contar nas inúmeras preocupações que se deve ter com servidores. No livro [Stallings 1995] se pode aprofundar um pouco mais nessas questões de segurança na *web*.

Mas para se aprofundar um pouco mais dentro dos conceitos de segurança de usuário e conteúdo de uma CDN é necessário primeiro conhecer um pouco sobre protocolos AAA e ter as definições de segurança muito bem esclarecidas, pois assim esses conteúdos o serão mais digeríveis. Ambos temas serão tratados em 4.1 e 4.2 respectivamente.

### **3.1. protocolos AAA - Authentication, Authorization and Accounting**

A associação dos protocolos de autenticação, autorização e auditoria em um termo, protocolos AAA, se deu porque na maioria dos sistemas seguros esses três protocolos são extremamente necessários e amplamente utilizados. em [Metz 1999] se pode aprofundar melhor. Mas vamos há uma rápida síntese dos mesmos.

**Autenticação** - Verifica a identidade digital do usuário de um sistema, que segundo [Metz 1999], envolve a validação da identidade do usuário final afim de permitir seu acesso à rede.

Esse procedimento é baseado na apresentação de uma identidade junto com uma ou mais credenciais, que podem ser senhas, certificados digitais e etc.

**Autorização** - É a concessão de uso para determinados serviços. Essa etapa só é acionada mediante a autenticação prévia de um usuário. Ela leva em conta a identidade, o serviço requerido e o estado atual do sistema. Muitas vezes ela trabalha com a utilização de filtros para retornar que tipo de protocolos ou aplicações são suportadas.

Na maioria dos casos os protocolos de autenticação e autorização trabalham juntos. Onde um depende diretamente do funcionamento do primeiro. Em ambientes de VOD, como se foi apresentado em 3.3.4, essa etapa normalmente é verificada nos servidores de ponta, os quais são responsáveis por fornecer o conteúdo ao usuário.

**Auditoria** - A última etapa do processo é relacionada a coleta de informações de tráfego utilizado pelo usuário.

Há dois tipos de coletas: coletas em tempo real e coleta *batch*. A primeira se refere a informações captadas em tempo de uso, ou seja, enquanto o usuário faz uso da rede essas informações são enviadas. Já a segunda se refere a uma coleta que é primeiro armazenada e enviada posteriormente.

São essas informações que são utilizadas pelas operadoras para cobrança e taxação do serviços de VOD.

### **3.2. Definições de segurança**

Como [O’Gorman 2003] fala em seu artigo as definições relativas aos sistemas e método em segurança são definidas como forte ou fraca. Isso acontece pois quando usamos termos

relativos, segundo o mesmo, somos claros na mensagem que queremos passar a respeito da informação. Por exemplo, quando dizemos que uma porta com uma trava é mais segura, mais forte, que uma porta que não possui trava alguma, relativamos a segurança da trava tornando claro a mensagem que por mais que seja mais segura não é impossível de ser quebrada.

Por mais que a relativização muitas vezes seja vista com maus olhos no campo da informática a garantia de um sistema inviolável é quase impossível e quando possível altamente custoso.

Como podemos ver no livro do [Stuttard and Pinto 2011] há diversas maneiras de se burlar um sistema web por exemplo, mesmo sendo um sistema altamente visado e onde pessoas se preocupam o tempo todo em protegê-lo. Mas um ponto comum de quase todo ataque é o que conhecemos como engenharia social. A entrada sempre acontece por meio de pessoas, ou fragilidade dessas pessoas.

É muito difícil mensurar segurança em termos absolutos. Uma forma de mensurar é através da força e da fraqueza de um sistema. Um sistema forte é aquele que o custo de atacar é muito maior que o ganho o atacando. Ou seja, o trabalho para conseguir a informação vai ser tão grande que não compensará. Um exemplo é em [Biryukov et al. 2010] a alta complexidade e tempo para se quebrar uma chave AES-256. Paralelamente à isso temos a fraqueza de um sistema. Essa é o oposto da força. Ou seja, o custo de ataque é menor do o ganho obtido com os resultados. E de que custo podemos entender não só dinheiro como também tempo, potencial para punição criminal e etc.

### **3.3. Autenticação de usuário**

[Hsiang and Shih 2009]

### **3.4. Autenticação de conteúdo**

conforme [Pomelo 2009]. Exemplos [Wein et al. 2012] e [Leighton et al. 2007]

### **3.5. Modelos de ataques à uma CDN**

## 4. References

### Referências

- Adhikari, V. K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., and Zhang, Z.-L. (2012). Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628. IEEE.
- Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., and Shamir, A. (2010). Key recovery attacks of practical complexity on aes-256 variants with up to 10 rounds. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 299–319. Springer.
- Ferreira, A. B. d. H. (2004). Novo dicionário aurélio da língua portuguesa. In *Novo dicionário Aurélio da língua portuguesa*.
- Garfinkle, N. (1996). Video on demand. US Patent 5,530,754.
- Hsiang, H.-C. and Shih, W.-K. (2009). Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, 31(6):1118–1123.
- Krishnamurthy, B., Wills, C., and Zhang, Y. (2001). On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182. ACM.
- Leighton, F. T., Lewin, D. M., et al. (2007). Html delivery from edge-of-network servers in a content delivery network (cdn). US Patent 7,293,093.
- Metz, C. (1999). Aaa protocols: authentication, authorization, and accounting for the internet. *IEEE Internet Computing*, 3(6):75–79.
- O’Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040.
- Pathan, A.-M. K. and Buyya, R. (2007). A taxonomy and survey of content delivery networks. *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report*, 4.
- Pomelo, L. (2009). Analysis of netflix’s security framework for watch instantly service.
- Stallings, W. (1995). *Network and internetwork security: principles and practice*, volume 1. Prentice Hall Englewood Cliffs.
- Stuttard, D. and Pinto, M. (2011). *The web application hacker’s handbook: Finding and exploiting security flaws*. John Wiley & Sons.
- Wein, J. M., Kloninger, J. J., Nottingham, M. C., Karger, D. R., and Lisiecki, P. A. (2012). Content delivery network (cdn) content server request handling mechanism. US Patent 8,122,102.
- Zambelli, A. (2009). Iis smooth streaming technical overview. *Microsoft Corporation*, 3:40.