



SAMODZIELNY ZAKŁAD SIECI KOMPUTEROWYCH
Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej
POLITECHNIKA ŁÓDZKA

90-924 Łódź ul. Stefanowskiego 18/22
tel./fax. (42) 636 03 00
e-mail: szsk@zsku.p.lodz.pl

Paweł Szewczyk

**CDN (*Content Delivery Network*) jako mechanizm
wspomagania usług sieciowych.**

praca dyplomowa magisterska

Promotor:

dr inż. Michał Morawski

Dyplomant:

Paweł Szewczyk

nr albumu 104942

Łódź, październik 2005 r.

Spis treści

Słownik pojęć	4
Wstęp	7
Cel i zakres pracy	8
1. Wprowadzenie do technologii CDN	9
Elementy architektury CDN	9
Ogólna zasada działania sieci CDN	11
Dostawcy usług dostarczania treści	12
Rodzaje sieci CDN	14
Wydajność serwera WWW po zastosowaniu CDN	16
Zalety CDN	18
Wady CDN	19
Technologia Proxy Cache	19
2. Usługi świadczone w sieci CDN	21
Media strumieniowe	21
Dostarczanie treści dynamicznych	22
ESI	23
Delta Encoding	25
Zaawansowane możliwości sieci CDN	25
Szyfrowanie treści	25
Kontrola dostępu	25
Personalizacja treści (ang. <i>content targeting</i>)	26
DNS	27
„Speedera Smart Storage Manager”	27
Okresowe użytkowanie sieci CDN	27
Monitorowanie i raportowanie	28
3. Krawędź sieci	29
Dostarczanie treści do krawędzi sieci	29
Rozmieszczenie serwerów krawędziowych	29
Rozmieszczenie kopii treści	30
Współpraca serwerów krawędziowych	31
Samoorganizująca się Architektura Rozproszona	32
Oddelegowanie witryny do obsługi przez sieć CDN	32
4. Globalne kierowanie treści	35
Przekierowanie DNS	35
Proces pobierania treści z przekierowaniem DNS	36
Przekierowanie DNS na przykładzie urządzeń Cisco	39
Ruting statyczny	39
Ruting hybrydowy	40
Kodowanie typu obiektu w nazwie domenowej	41
Widoki	42
Ograniczenia przekierowania DNS	43
Przekierowanie HTTP 302	44
Adresowanie typu anycast	45

Anycast na podstawie nazwy domenowej.....	46
Łączenie kilku mechanizmów przekierowania	47
Współpraca sieci CDN	48
Wybór najlepszego serwera.....	50
Wybór najlepszego serwera na przykładzie urządzeń Cisco	52
5. Lokalne kierowanie treści	53
Równoważenie obciążenia na podstawie warstwy transportowej.....	53
Przełączanie na podstawie warstwy 7	57
Dwa połączenia	57
HTTP 1.0 i HTTP 1.1	58
Dystrybucja obciążenia na podstawie wiadomości HTTP.....	59
Kojarzenie kolejnych połączeń	61
Monitorowanie stanu serwera w czasie rzeczywistym.....	62
6. Testy programu „Content Switch”	63
Środowisko testowe.....	63
Test 1	63
Test 2	64
Test 3	65
Podsumowanie	67
7. Załącznik	68
Instrukcja użytkownika programu „Content Switch”.....	68
Wymagania programu.....	68
Instalacja sterownika.....	68
Konfiguracja sterownika	69
Plik logów	71
Deinstalacja sterownika	71
Szczegóły techniczne.....	72
Kierowanie pakietów	72
Nawiązywanie połączenia TCP/IP	73
Obsługa nawiązanych połączeń	75
Zamykanie połączenia.....	75
Wątki pomocnicze.....	75
Komunikacja ze sterownikiem.....	76
Włączanie i wyłączanie przełączania pakietów	77
Uszkodzenie serwera.....	77
Packet Stacking	77
Możliwości rozwoju.....	78
Bibliografia.....	79

Słownik pojęć

Krawędź sieci (ang. *edge*) - w odniesieniu do technologii CDN jest to miejsce, w którym użytkownicy łączą się z siecią. Krawędź sieci powinna znajdować się jak najbliżej użytkownika.

Serwer oryginalny – serwer dostawcy treści, z którego treść po raz pierwszy została przekazana do sieci CDN lub serwer w sieci CDN, który zawiera wiarygodną kopię treści.

Dostawca usług dostarczania treści, operator sieci CDN – właściciel sieci CDN, który umożliwia hosting usług w ramach swojej sieci.

Dostawca treści - firma lub osoba, która korzysta z usług dostarczania treści, przekazuje treści do obsługi przez sieć CDN.

Hosting – „usługa polegająca na udostępnieniu zasobów serwerowni oraz wynajmie platformy sprzętowej lub wirtualnej platformy systemowej. W ramach usługi klient może otrzymywać dostęp do środowiska systemu operacyjnego o określonych umową parametrach, dostęp do sieci, zasilanie i wsparcie administratorów” [15].

ISP (ang. *Internet Service Provider*), **Dostawca Usług Internetowych** – „firma, która (przeważnie odpłatnie) oferuje dostęp do Internetu” [15].

Sieć rozległa, WAN (ang. *Wide Area Network*) – „sieć łącząca sieci lokalne, inne (mniejsze) sieci rozległe, jak również pojedyncze komputery. Odbywa się to przy pomocy urządzeń sieciowych takich jak routery oraz urządzeń dostępowych takich jak modemy. Przykładami sieci rozległych są sieci miejskie, sieci korporacyjne, oraz Internet” [15].

E-learning – „oznacza wspomaganie dydaktyki za pomocą komputerów osobistych i Internetu. Pozwala na ukończenie kursu, szkolenia, a nawet studiów bez konieczności fizycznej obecności w sali wykładowej. Wspiera również tradycyjny proces nauczania” [15].

Internet (dosł. międzysieć; od ang. *inter* – między i ang. *net* – sieć) – „sieć komputerowa o światowym zasięgu łącząca sieci lokalne, sieci rozległe i wszystkie komputery do nich podłączone” [15].

Intranet – „jest siecią komputerową ograniczającą się do komputerów np. w firmie lub organizacji. Sieć taka zwana jest potocznie LAN. Po zamontowaniu serwera umożliwiającego korzystanie w obrębie sieci LAN z usług takich jak strony WWW,

poczta elektroniczna, czyli usług typowo internetowych, można mówić o sieci intranet. Do intranetu dostęp mają zazwyczaj tylko pracownicy danej firmy” [15].

Outsourcing (z ang. *out source*- zewnętrzne źródło) – „praktyka polegająca na przeniesieniu części zadań wykonywanych przez firmę lub organizację z własnych pracowników na zewnętrznych kontrahentów” [15].

Punkt prezentacji, POP (ang. *Point of Presence*) – „punkt prezentacji usług, stanowiący brzegowy węzeł istniejącej sieci telekomunikacyjnej, w którym jest zlokalizowana inteligencja do prowadzenia nowoczesnych usług o wartości dodanej” [17]. W punkcie prezentacji znajduje się serwer krawędziowy.

handel elektroniczny (ang. *e-commerce*) – „kupno i sprzedaż towarów lub usług poprzez Internet za pośrednictwem stron WWW. W skład *e-commerce* wchodzi sprzedaż towarów i usług, przyjmowanie i potwierdzanie zamówień oraz obsługa płatności bezgotówkowych” [15].

URL (ang. *Uniform Resource Locator*) – „zunifikowany format odnośników do zasobów (głównie Internetu). URL są zdefiniowane przez RFC 1738” [15]. Adres URL składa się z trzech części: identyfikatora usługi (np. *http:*), nazwy domeny (np. *www.networld.com.pl*) i ścieżki dostępu (np. */numery/*).

DNS (ang. *Domain Name System*) – system serwerów oraz protokół komunikacyjny zapewniający dwustronną konwersję numerycznych adresów internetowych (32 lub 128 bitów) na łańcuchy łatwych do zapamiętania nazw. Protokół DNS opisany został w dokumentach RFC o numerach 882, 883, 1034 i 1035.

Wirtualny adres IP (VIP) – adres nie przypisany na stałe do konkretnego interfejsu, system operacyjny akceptuje pakiety z wirtualnym adresem niezależnie od interfejsu, na którym przyszły.

HTTP (ang. *Hypertext Transfer Protocol*) – „protokół sieci WWW (ang. *World Wide Web*). Obecną definicję HTTP stanowi RFC 2616. Właśnie za pomocą protokołu HTTP przesyła się żądania udostępnienia dokumentów WWW i informacje o kliknięciu odnośnika oraz informacje z formularzy. Zadaniem stron WWW jest publikowanie informacji – natomiast protokół HTTP właśnie to umożliwia.

Protokół HTTP jest tak użyteczny, ponieważ udostępnia znormalizowany sposób komunikowania się komputerów ze sobą. Określa on formę żądań klienta dotyczących danych oraz formę odpowiedzi serwera na te żądania. Jest zaliczany do protokołów *stateless* (bezstanowy), z racji tego, że nie zachowuje żadnych informacji o poprzednich

transakcjach z klientem, po zakończeniu transakcji wszystko „przepada” - z tego powodu tak bardzo spopularyzowały się *cookies*” [15].

Model OSI, OSI (ang. *Open System Interconnection*) – „zdefiniowany przez organizacje ISO oraz ITU-T standard opisujący strukturę komunikacji sieciowej. Model OSI jest traktowany jako model odniesienia dla większości rodzin protokołów komunikacji. Podstawowym założeniem modelu jest podział systemów sieciowych na 7 całkowicie niezależnych warstw” [15].

WAP (ang. *Wireless Application Protocol*) – „zbiór otwartych, międzynarodowych standardów definiujących protokoły aplikacji bezprzewodowych. Stworzony w celu umożliwienia dostępu do usług WWW, uwzględniając ograniczenia techniczne urządzeń mobilnych korzystających z tego protokołu oraz ograniczeń łącza danych. WAP 2.0 został przystosowany do obsługi protokołów używanych w Internecie (IP, TCP, TLS, HTTP). Protokół definiuje nowy język opisu strony, którym jest XHTML Mobile Profile” [15].

Flash Crowd – w odniesieniu do sieci WWW terminem tym określamy sytuację, w której witryna WWW nagle z jakiegoś powodu przyciągnie uwagę dużej liczby użytkowników. Nagły wzrost liczby zapytań do serwera WWW może prowadzić do przeciążenia serwera, jednak za tym zjawiskiem nie kryje się umyślne i złośliwe działanie użytkowników.

Reguła Pareto – jest to prawidłowość nazywana również regułą 80/20, zgodnie z którą w zbiorze niejednorodnym 20% elementów tego zbioru reprezentuje 80% skumulowanej wartości cechy, ze względu na którą ta zbiorowość jest rozpatrywana.

Wstęp

W ostatnich latach nastąpił bardzo szybki rozwój Internetu i związanych z nim technologii. Stale wzrasta liczba osób posiadających dostęp do Internetu. Użytkownicy domagają się coraz krótszego czasu ładowania się stron WWW, jednocześnie strony zawierają coraz więcej grafiki oraz obiektów multimedialnych. W wyniku tego dostawcy usług muszą borykać się z problemami takimi jak zbyt długi lub zmienny czas ładowania się stron WWW. Przyczyną tego stanu są z kolei wąskie gardła na styku operatorów internetowych i w punktach dostępu do serwerów WWW oraz okresowe przeciążenie serwerów spowodowane dużą ilością żądań.

Próbując rozwiązać powyższe problemy, opracowano kilka metod usprawniających dostarczanie treści do użytkownika końcowego [1]. Najprostszym sposobem jest zwiększenie dostępnego pasma i zastosowanie szybszych urządzeń sieciowych. Powielanie zawartości, tworzenie tzw. farm serwerów zwiększa niezawodność systemu. Równoważenie obciążenia pozwala na zwiększenie szybkości reakcji systemu. Odejście od modelu scentralizowanego daje również wymierne rezultaty. Dystrybucja geograficzna serwerów oraz stosowanie urządzeń buforujących (ang. *cache*) w lokalnych sieciach przybliży zawartość do użytkowników końcowych zmniejszając tym samym czas odpowiedzi.

Powyższe metody stosowane pojedynczo nie zapewniają kompletnego rozwiązania. Dopiero stworzenie systemu, który łączy te techniki pozwoliło w dużym stopniu uporać się z problemami przedstawionymi na wstępie. Sieć Dostarczania Treści (ang. *Content Delivery Network*) jest właśnie takim systemem.

Tego typu sieci zapewniają:

- centralne sterowanie dystrybucją treści oraz siecią
- wysoką wydajność powielania treści do serwerów znajdujących się na krawędzi sieci
- automatyczne kierowanie zapotrzebowań na daną treść do najbliższego serwera znajdującego się na krawędzi sieci

Dzięki przybliżeniu treści do odbiorcy, Sieć Dostarczania Treści jest sposobem na pokonanie wielu ograniczeń współczesnego Internetu.

W dalszej części pracy zamiennie z terminem Sieć Dostarczania Treści używane będą określenia: sieć CDN, technologia CDN.

Cel i zakres pracy

Celem niniejszej pracy jest przedstawienie zagadnień związanych z budową i funkcjonowaniem sieci CDN. Ze względu na wzrastającą popularność i znaczenie technologii CDN, warto ją przedstawić. Dodatkowo dochodzi fakt, że niewiele jest opracowań w języku polskim szczegółowo omawiających ten temat.

Rozdział 1 omawia podstawowe pojęcia związane z architekturą sieci CDN oraz zalety wynikające z zastosowania tego typu sieci.

Rozdział 2 to przegląd najistotniejszych usług, jakie operatorzy sieci CDN świadczą przy pomocy swoich sieci.

Rozdział 3 rozwija pojęcie, jakim jest krawędź sieci, skupiając się na zagadnieniach rozmieszczenia serwerów na krawędzi sieci oraz dostarczania do nich treści.

Rozdział 4 zawiera opis metod kierowania żądań użytkowników do serwera, który będzie w stanie „najlepiej” obsłużyć dane żądanie.

Rozdział 5 opisuje, w jaki sposób działa kierowanie żądań użytkowników realizowane za pomocą przełącznika treści (ang. *content switch*).

Celem części praktycznej pracy jest implementacja przełącznika treści. Rozdział 6 zawiera testy programu, natomiast w załączniku umieszczono instrukcję użytkownika oraz opis techniczny programu.

1. Wprowadzenie do technologii CDN

Elementy architektury CDN

Architektura Sieci Dostarczania Treści składa się z następujących bloków funkcyjnych [2][3]:

- **Dystrybucja i zarządzanie treścią**

Głównym celem jest kontrola rozmieszczania treści na krawędzi sieci jak najbliżej użytkowników. Oprócz tego zapewnia systemy do obsługi i prawidłowego funkcjonowania usług CDN. Urządzenie pełniące tą funkcję nazywane będzie menadżerem dystrybucji treści (ang. *content distribution manager*).

- **Globalne kierowanie treści** (ang. *content routing*)

Kieruje żądania użytkowników w optymalne miejsce do obsługi określonego żądania. Lokalizuje miejsce pobrania treści w oparciu o topologię sieci, opóźnienia w sieci, obciążenie serwera i wybraną politykę. Urządzenie pełniące tą funkcję nazywane będzie ruterem treści lub węzłem kierowania żądań.

- **Lokalne przełączanie treści** (ang. *content switching*)

Polega na wyborze najlepszego serwera, który dostarczy żadaną treść. Wybór następuje nie tylko w oparciu o dostępność serwera i jego obciążenie, ale również weryfikację rodzaju żądanej treści. Umożliwia to usługi dostarczania treści bazujące na sesji użytkownika końcowego. Urządzenie pełniące tą funkcję nazywane będzie przełącznikiem treści (ang. *content switch*).

- **Serwowanie treści** (ang. *content edge delivery*)

Przechowuje treść przekazaną do obsługi przez sieć CDN. Dostarcza treści statyczne i strumieniowe z krawędzi sieci, a jeśli zajdzie potrzeba uaktualnia je odwołując się do oryginalnego serwera. Urządzenie pełniące tą funkcję nazywane będzie serwerem krawędziowym lub węzłem dostarczania treści.

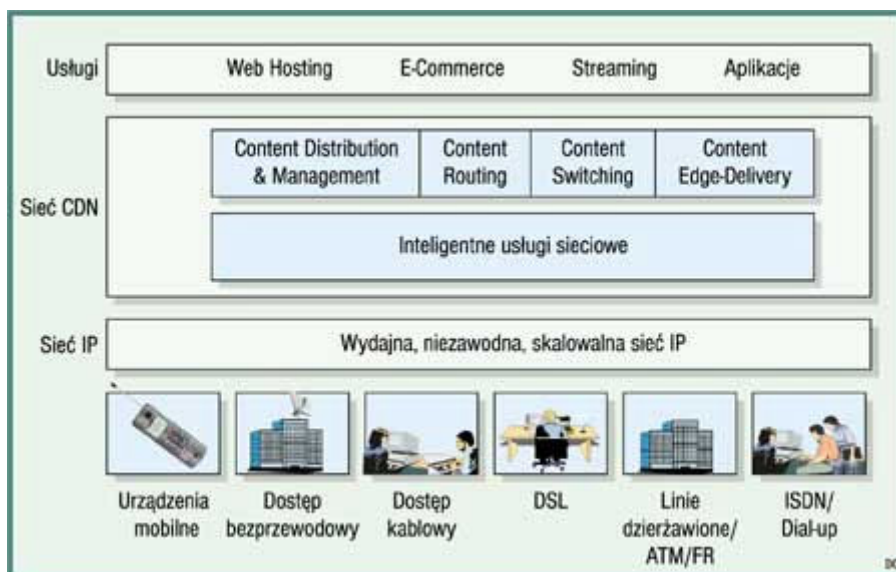
- **Inteligentne usługi sieciowe**

Wprowadza do sieci CDN usługi warstw 2 i 3, takie jak QoS, VPN, bezpieczeństwo i transmisje grupowe (ang. *multicast*).

Z powyższych elementów można zbudować w pełni funkcjonalną Sieć Dostarczania Treści. Należy jednak dodać, że bazuje ona na niezawodnej, szybkiej,

skalowalnej i zarządzalnej infrastrukturze warstw 2 i 3, która stanowi tzw. rdzeń sieci (ang. *core networking*). Przykładem może być sieć IP NTT/Verio [10], która swoim obszarem działania obejmuje Europę, Amerykę Północną oraz Azję. Zbudowana w technologii światłowodowej OC3, OC12, OC48 oraz przy użyciu szybkich ruterów takich firm jak Cisco i Juniper Networks pozwala na wydajne i niezawodne transmisje danych o globalnym zasięgu.

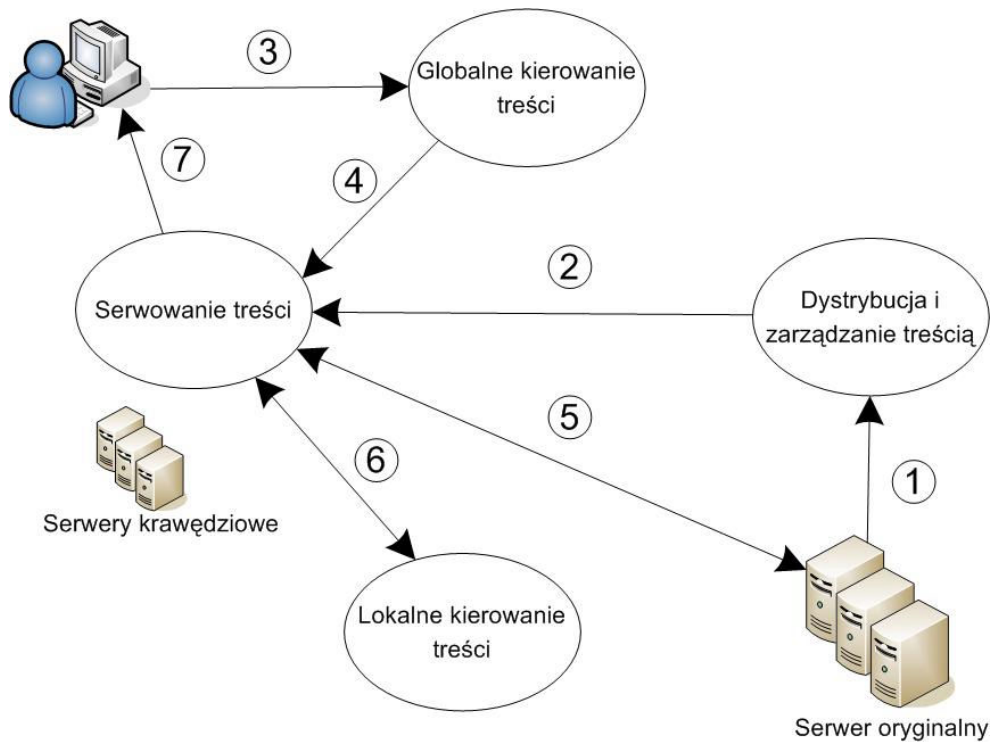
Na Rys. 1 przedstawione jest usytuowanie sieci CDN w modelu warstwowym. Zaznaczone są na nim opisane powyżej bloki funkcyjne.



Rys. 1. Elementy *Content Delivery Networks* w warstwowym modelu sieciowym [4]

Elementem, który jest niezbędny w komercyjnych sieciach CDN jest system monitorowania wykonywanych operacji (ang. *accounting system*) [13][14]. Tworzy on statystyki zdarzeń takich jak dystrybucja lub dostarczenie klientom końcowym określonej treści. Dzięki temu możliwe jest rozliczenie świadczonych usług. Statystyki te mogą zostać również wykorzystane do wspomaganie procesu globalnego kierowania treści.

Ogólna zasada działania sieci CDN



Rys. 2. Idea działania sieci CDN [2][4] (poszczególne kroki zaznaczone są w poniższym tekście)

W wielu miejscach sieci, możliwie blisko użytkownika końcowego rozmieszczone zostają urządzenia przechowujące i podające treść (serwery krawędziowe). Są one centralnie zarządzane i ładowane treścią przez menadżera dystrybucji treści (2), dzięki któremu administrator sieci CDN sprawuje kontrolę nad polityką dystrybucji treści. Użytkownik odwołujący się do zwykłej strony WWW, która przekazana została do obsługi jest przez sieć CDN (1), nie jest świadomy istnienia tej sieci. Kiedy nastąpi odwołanie do obiektu, który został rozesłany w sieci w procesie dystrybucji treści, użytkownik zostaje skierowany do węzła kierowania żądań (3). Zadaniem tego urządzenia jest znalezienie, przy pomocy jednego z wielu możliwych algorytmów, najbliższej użytkownikowi lokalizacji, która zawiera poszukiwany obiekt (4). W lokalizacji może znajdować się jeden lub wiele serwerów krawędziowych, przechowujących żadaną treść. Jeśli mamy do czynienia z farmą serwerów krawędziowych, stosowany jest przełącznik treści, który wybiera najbardziej odpowiedni serwer np. najmniej obciążony (6). Wybrany serwer podaje żądany obiekt użytkownikowi (7) lub ściąga go z oryginalnego serwera, jeśli zajdzie taka potrzeba (5).

Na tym kończy się cały cykl. Opisany proces jest całkowicie niewidoczny dla użytkownika.

Dostawcy usług dostarczania treści

Równoległe z pojawieniem się technologii CDN powstały firmy, które zbudowały sieci CDN o zasięgu globalnym. Dzięki stworzonej infrastrukturze świadczą one usługi dostarczania treści, zapewniając jej dostępność w każdym zakątku Internetu. Firma taka związana jest umowami z lokalnymi dostawcami usług internetowych (ISP), którzy w swoich sieciach odpłatnie umieszczają serwery krawędziowe należące do tej firmy. Właściciel sieci CDN dąży do zainteresowania dostawców treści możliwościami korzystania ze swojej sieci, która zapewni odbiorcom szybszy dostęp do treści.

Największą Sieć Dostarczania Treści posiada firma Akamai [5]. Sieć składa się z ponad 15000 serwerów rozmieszczonych w 1100 lokalizacjach na całym świecie. Całą infrastrukturę obsługuje jedno Centrum Zarządzania Siecią (ang. *Network Operations Center*). Umożliwia ono monitorowanie wszystkich serwerów oraz bieżącą kondycję całej sieci. Serwis dostarczania treści nosi nazwę „EdgeSuite” i wspiera efektywną dystrybucję wielu rodzajów treści: pliki html, graficzne, tekstowe, strumieniowe transmisje audio i wideo w czterech głównych formatach, QuickTime firmy Apple, Windows Media firmy Microsoft, G2 firmy RealSystem oraz Flash. Zaawansowane usługi umożliwiają przesyłanie szyfrowanych danych przy pomocy protokołu SSL [70], autoryzację użytkowników, personalizację treści.

Inni wiodący dostawcy usług dostarczania treści to:

- Speedera [6]

Udostępnia klientom zestaw serwisów o nazwie „SpeedSuite”, zapewniających niezawodne dostarczanie treści statycznych, dynamicznych oraz strumieniowych. Podstawą działania oprócz globalnie rozproszonej sieci jest opatentowany system globalnego zarządzania ruchem (ang. *Global Traffic Management*), który monitoruje wydajność poszczególnych serwerów krawędziowych, ocenia odległość pomiędzy serwerem i użytkownikiem ułatwiając kierowanie żądań. Specjalny portal zarządzający o nazwie „SpeedEye” umożliwia monitorowanie sieci i zarządzanie treścią.

- Mirror-Image [7]

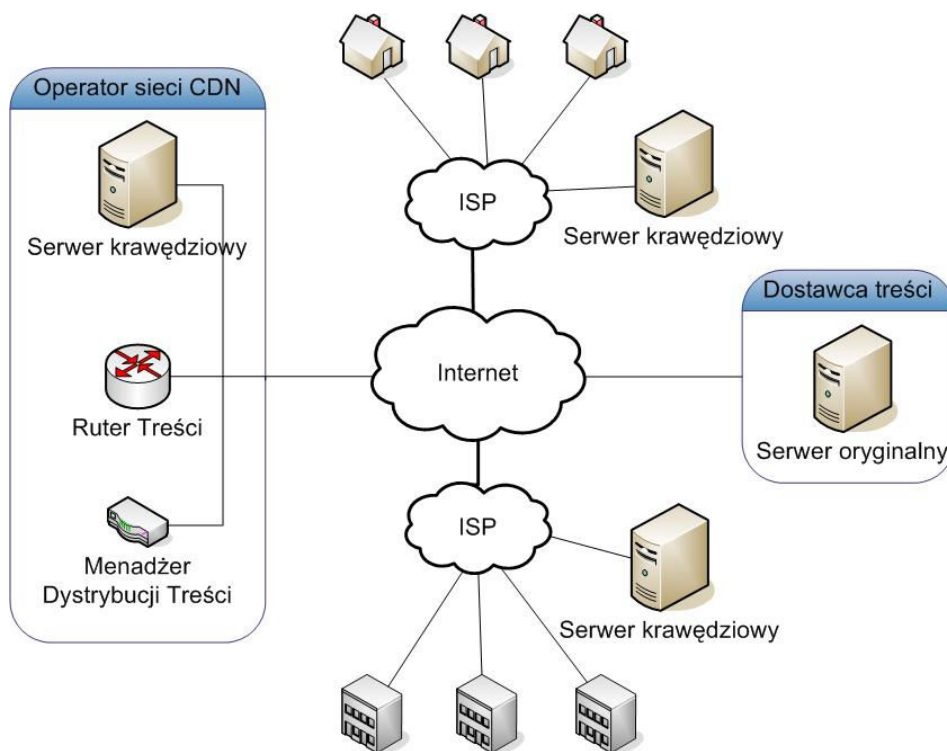
Firmowe rozwiązanie dostarczania treści nosi nazwę „Content Access Point”. Oprócz dostarczania statycznych i dynamicznych zawartości stron internetowych oferuje bogaty wybór przekazów strumieniowych: wideo na żądanie, przekazy na żywo (jeden do wielu) z konferencji, prezentacji lub materiały treningowe. Umożliwia również konwersję materiałów audio i wideo na inne formaty rozmiaru czy kodowania. Możliwe jest również różnicowanie dostarczanej treści w zależności od geograficznego położenia użytkownika końcowego, języka czy pory dnia.

- VitalStream [8]

Zapewnia platformę dostarczającą wyłącznie media strumieniowe. Wspiera technologię Flash oraz Windows Media, zapewniając klientom dystrybucję lub sprzedaż online ich treści.

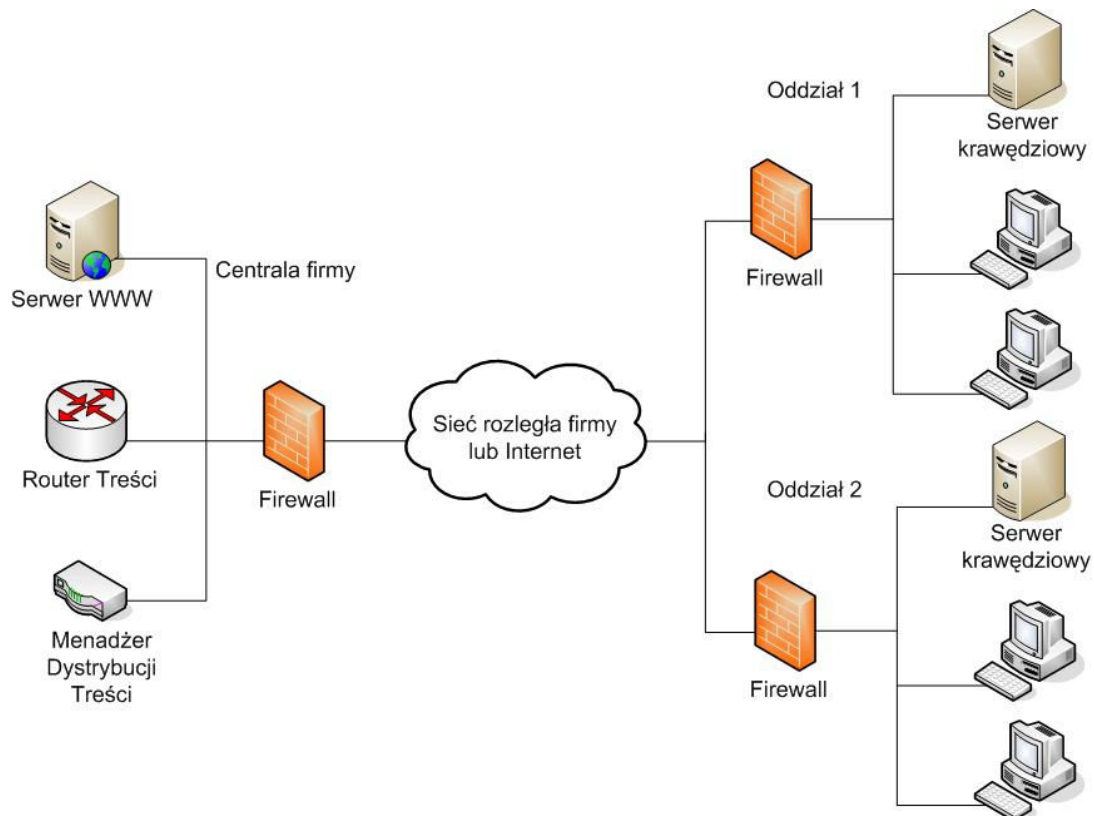
Rodzaje sieci CDN

Sieci CDN o zasięgu globalnym noszą nazwę Internet CDN. Charakteryzuje je to, że punkty prezentacji rozmieszczone są w wielu miejscach Internetu, w sieciach lokalnych ISP [8]. System dystrybucji i zarządzania treścią oraz węzeł kierowania żądań umieszczone są w internetowym centrum przetwarzania danych (ang. *data center*). Ten model sieci jest wykorzystywany przez dostawców usług dostarczania treści [16].



Rys. 3. Internet CDN

Sieci CDN często implementowane są wewnątrz przedsiębiorstwa. Ten rodzaj określany jest w literaturze jako korporacyjny CDN lub ECDN (ang. *Enterprise CDN*) [4][12]. Użytkownikami końcowymi są w tym wypadku pracownicy. Z tego powodu serwery krawędziowe umieszczone są w oddziałach firmy, podczas gdy system dystrybucji i zarządzania treścią oraz węzeł kierowania żądań umieszczone są w centrali firmy.



Rys. 4. ECDN [12]

Taka sieć pomaga efektywnie wspierać duży, intranetowy portal bogaty w grafikę i multimedia, dystrybucję dokumentów w firmie czy dystrybucję oprogramowania lub jego aktualizacji. Największe pliki audio, wideo, grafiki rozesłane zostają do serwerów krawędziowych umieszczonych w sieciach lokalnych użytkowników końcowych. Jednak największe korzyści przynosi wykorzystanie CDN do e-learningu, w którym najczęściej stosowane są transmisje wideo na żądanie.

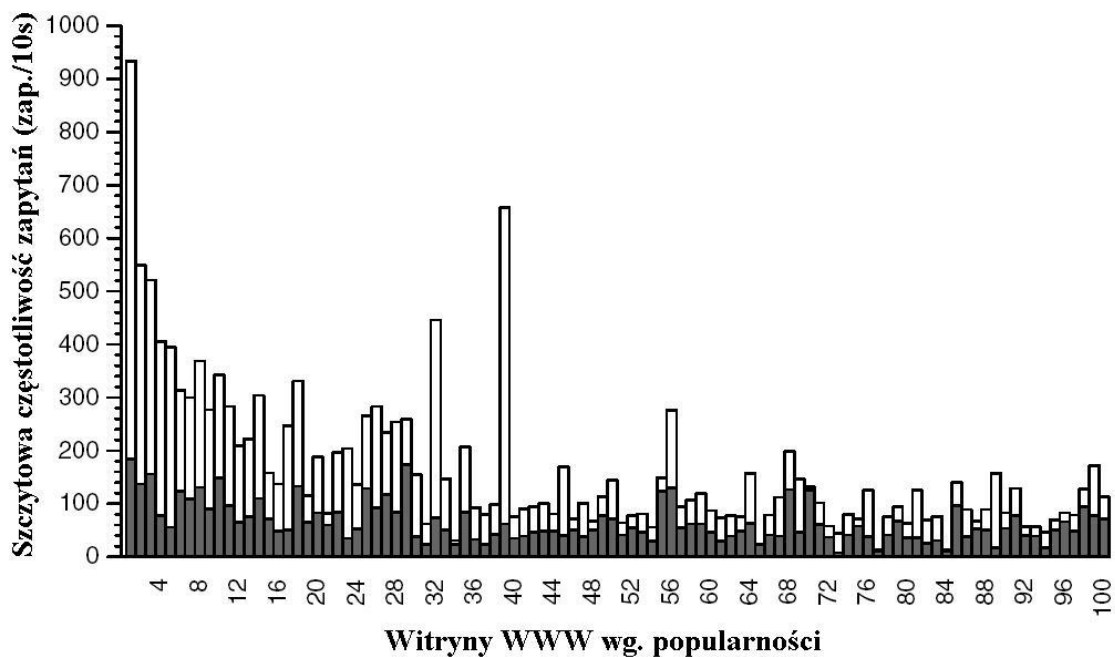
Wyobraźmy sobie pracownika, który w małym oddziale firmy dysponującym łączem 128kbps, chciałby uczestniczyć w przygotowanym przez centralę firmy szkoleniu multimedialnym. Łącze o podanej przepustowości nie nadaje się do korzystania z strumieniowych przekazów wideo przez Internet. Gdy stosujemy ECDN,

w oddziale znajduje się urządzenie buforujące (serwer krawędziowy). Materiały szkoleniowe mogą zostać umieszczone na tym urządzeniu poza godzinami pracy biura, dzięki temu pracownik uzyska do nich dostęp z sieci lokalnej i wąskie gardło jakim jest połączenie z Internetem nie będzie już przeszkodą.

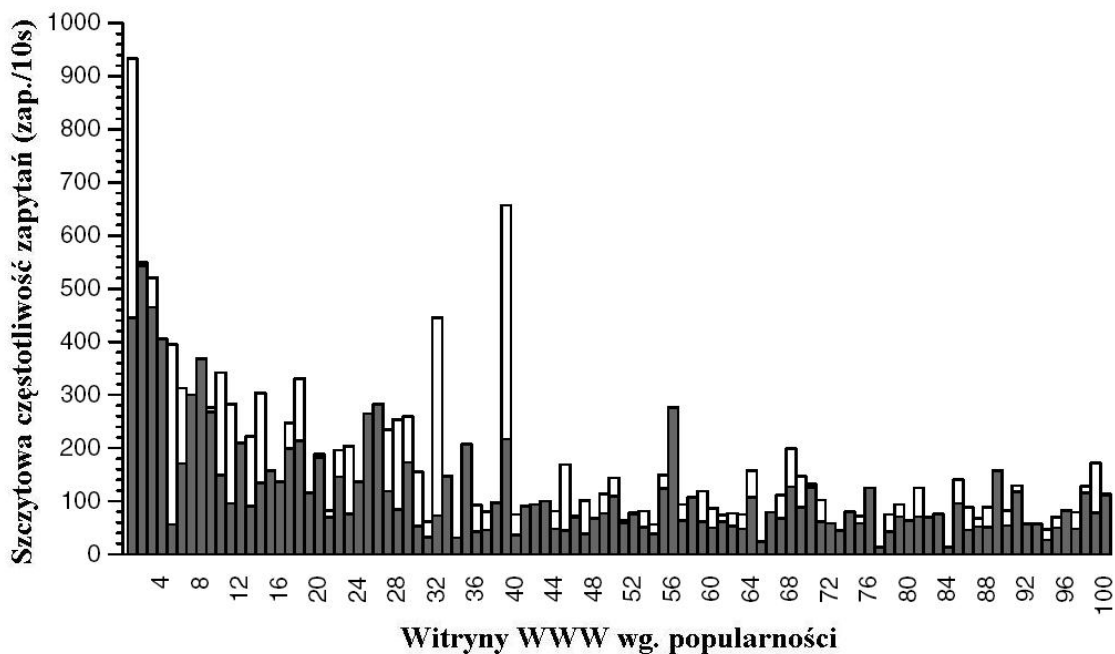
Warto wspomnieć o MECDN (ang. *Managed Enterprise CDN*). Model ten rozszerza ofertę operatorów sieci CDN. Łączy w sobie zalety modelu internetowego oraz ECDN [4]. Polega na outsourcingu zarządzania siecią CDN. Dostawca usług dostarczania treści przedłuża zasięg swojej sieci do wnętrza przedsiębiorstwa oraz przejmuje funkcje takie jak zamieszczanie, usuwanie i dystrybucja treści. Jednocześnie możliwe jest wdrażanie w firmie tych samych aplikacji co w ECDN.

Wydajność serwera WWW po zastosowaniu CDN

Zespół naukowców z Uniwersytetu Kalifornijskiego wraz z firmą AT&T stworzyli narzędzie o nazwie „Cassandra” służące do analizy korzyści płynących z zastosowania różnych mechanizmów poprawiających wydajność witryny internetowej [9]. Jednym z takich mechanizmów jest właśnie CDN. Przeprowadzono symulację zastosowania tego typu sieci do odciążenia serwera WWW. Do doświadczenia użyto prawdziwe dane z 3000 serwerów WWW [9]. Przedstawione na Rys. 5 i Rys. 6 wyniki odnoszą się do 100 najbardziej popularnych, najczęściej odwiedzanych witryn WWW. Wykresy obrazują częstotliwość zapytań do serwera oryginalnego. Wykres pierwszy odnosi się do przypadku idealnego, w którym wszystkie obiekty witryn skierowane do obsługi przez Sieć Dostarczania Treści są całkowicie buforowalne tzn. czas życia na serwerze krawędziowym jest z założenia nieskończony. Natomiast wykres drugi odnosi się do przypadku bliższego rzeczywistości, w którym zakładamy, że nie wszystkie z tych obiektów są całkowicie buforowalne, niektóre posiadają określony czas życia, przez co, co pewien czas muszą zostać ponownie pobrane z serwera oryginalnego. Na wykresach ciemne słupki odnoszą się do przypadku, w którym korzystamy z technologii CDN, a jasne słupki, gdy nie korzystamy z CDN.



Rys. 5. Porównanie szczytowych częstotliwości zapytań do serwera WWW dla przypadku idealnego [9]



Rys. 6. Porównanie szczytowych częstotliwości zapytań do serwera WWW dla przypadku rzeczywistego [9]

Z symulacji wynika, że zastosowanie technologii CDN do 100 najpopularniejszych witryn internetowych zmniejszyło szczytową ilość zapytań średnio 1,4 krotnie w przypadku rzeczywistym oraz niemal 2,5 krotnie w przypadku idealnym.

Rys. 5 pokazuje, że szczytowa ilość zapytań dla najbardziej popularnej witryny zmniejszyła się o 52% z wykorzystaniem CDN, natomiast dla drugiej najbardziej popularnej witryny tylko o 1,1%. Zatem nie zawsze użycie tej technologii daje oczekiwane efekty w postaci znacznego zmniejszenia obciążenia serwera WWW. W warunkach rzeczywistych, kiedy wiele elementów witryny internetowej ma charakter dynamiczny lub są to elementy niebuforowalne, korzystanie z CDN może okazać się po prostu nieopłacalne.

Zalety CDN

Technologia CDN przynosi wiele korzyści przedsiębiorstwom, które dążą do zoptymalizowania swoich sieci w kierunku obsługi mediów strumieniowych oraz dostawcom usług pragnącym oferować swoim klientom wydajne dostarczanie różnorodnych treści. Główne zalety używania technologii CDN [11]:

- Sieci budowane są ze skalowalnych komponentów, dzięki czemu można je łatwo rozszerzać i dostosowywać do rodzaju serwowanej treści.
- CDN znacząco zmniejsza zajętość pasma w sieci WAN, ponieważ użytkownicy końcowi pobierają treść z serwerów krawędziowych, które zlokalizowane są w sieciach lokalnych dostawców Internetu. Z tego samego powodu jest sposobem na ominięcie zatorów i wąskich gardeł w sieci WAN.
- Wzrasta szybkość oraz niezawodność dostępu do treści, co korzystnie wpłynie na zadowolenie klientów. W korporacyjnej sieci CDN może przyczynić się do wzrostu wydajności pracowników.
- Buforowanie treści w wielu miejscach na krawędzi sieci powoduje wzrost jej dostępności, a jednocześnie zapobiega przeciążeniom serwera oryginalnego, gdyż część żądań obsługiwana jest przez sieć CDN.
- CDN zapewnia hosting usług. Wiąże się z tym redukcja kosztów związanych z utrzymaniem rozbudowanej infrastruktury własnych serwerów w korporacyjnych centrach danych.
- Możliwość filtrowania zawartości pozwala firmie blokować niepożądane treści.

- Rozproszona struktura serwerów krawędziowych skutecznie zapobiega przeciążeniom serwera oryginalnego, które spowodowane mogą być nagłym wzrostem liczby żądań. Taki nagły wzrost wynikać może ze zwiększonego zainteresowania użytkowników treścią, mówimy wtedy o zjawisku *Flash Crowd*, lub może być wynikiem próby ataku typu DoS (ang. *Denial of Service*) [40]. A zatem sieć CDN uniemożliwia ataki typu DoS.

Wady CDN

W przypadku wystąpienia problemów w działaniu sieci CDN może okazać się trudne zlokalizowanie ich przyczyny ze względu na dużą liczbę urządzeń wchodzących w skład sieci. Ten aspekt należy starać się wyeliminować już w fazie projektowania i implementacji biorąc pod uwagę wiele czynników, które mogą zakłócić prawidłowe działanie sieci [11].

Drugą sprawą, o której należy wspomnieć, to niewielkie postępy poczynione w kierunku standaryzacji zagadnień związanych z CDN. Poszczególne firmy rozwijające tę technologię tworzą własne standardy, przez co utrudniona może być współpraca urządzeń różnych producentów.

Technologia Proxy Cache

W celu zmniejszenia obciążenia sieci oraz zmniejszenia liczby zapytań do serwera stosowana jest często technologia buforowania pośredniczącego (ang. *proxy cache*). Urządzenia buforujące (serwery *proxy*) umieszczane są blisko klienta, w jego sieci lokalnej lub w sieci lokalnego ISP. Po odpowiednim skonfigurowaniu przeglądarki internetowej ruch HTTP kierowany jest do bufora pośredniczącego, który w imieniu klienta przekazuje żądania do serwera oryginalnego lub, gdy posiada aktualną kopię żądanej treści, dostarcza ją bezpośrednio bez kontaktowania się z serwerem. Przekierowanie do serwera *proxy* może odbywać się również bez wiedzy użytkownika, mówimy wtedy o buforowaniu przezroczystym (ang. *transparent cache*). Serwery *proxy* mogą tworzyć struktury hierarchiczne. W takim przypadku, jeżeli serwer buforujący nie posiada w swoich zasobach żądanego obiektu przekazuje żądanie do serwera nadrzędnego. Gdy żądanie nie będzie mogło być obsłużone w najwyższym w hierarchii serwerze buforującym przekazywane jest do serwera oryginalnego. Odpowiedź wraca tą

samą drogą aktualizując lub wstawiając kopię żadanego obiektu we wszystkich buforach hierarchii, przez które przechodzi [52].

Technika *proxy cache* stosowana była przed pojawieniem się sieci CDN i nadal jest bardzo rozpowszechniona. Jednak posiada pewne ograniczenia w stosunku do sieci CDN:

- Serwery *proxy* obsługują żądania tylko swoich klientów, a nie dowolnych użytkowników Internetu.
- Dostawcy treści nie mają większego wpływu na istnienie serwerów *proxy*, przyspieszających dostęp do ich treści, ani na ich prawidłową implementację, jeśli takie istnieją.
- Dostawca treści często potrzebuje statystyki odwołań do swojej witryny. Tylko sieć CDN, dzięki możliwości monitorowania wykonywanych operacji (ang. *accounting*), może to zapewnić. Daje możliwość kontroli dostarczanych treści.
- Serwery *proxy* pobierają treść z serwera oryginalnego, gdy przyjdzie żądanie jej dostarczenia, a dana treść nie znajduje się obecnie w buforze. Natomiast w sieci CDN treść może zostać rozdystrybuowana do serwerów na krawędzi sieci zanim nadejdzie pierwsze żądanie dostarczenia tej treści.
- *Proxy cache* zaspokaja potrzeby tylko użytkowników końcowych, natomiast nie zaspokaja potrzeb dostawców treści.
- Serwery *proxy* używane są głównie przez ISP w celu zmniejszenia obciążenia łącza, podczas gdy z sieci CDN korzystają dostawcy treści chcący zapewnić swoim klientom lepszą jakość usług.

2. Usługi świadczone w sieci CDN

Pierwsze sieci CDN powstały w drugiej połowie lat 90, Akamai zaczęła świadczyć komercyjne usługi w 1999 roku [5]. Początkowo na krawędzi sieci buforowano tylko statyczne elementy stron HTML. Jednak korzyści płynące w używaniu technologii CDN spowodowały szybki jej rozwój.

Media strumieniowe

Drugim krokiem na drodze rozwoju sieci była potrzeba wprowadzenia obsługi mediów strumieniowych. W tradycyjnym rozwiązaniu zawartość pliku multimedialnego odtwarzana jest dopiero po całkowitym pobraniu jego źródła, natomiast nadawaną strumieniowo treść odtwarza się już po kilku sekundach od nawiązania połączenia. Występują dwie zasadnicze usługi:

- **Wideo na żądanie** – serwer krawędziowy przechowuje pliki mediów strumieniowych oraz dostarcza je do użytkowników końcowych, gdy tego żądają. Użytkownik może zatrzymać nagranie na chwilę, przewinąć do przodu lub do tyłu jak w odtwarzaczu kaset wideo. W tym trybie odtwarzanie rozpoczyna się i trwa niezależnie dla każdego użytkownika.
- **Wideo na żywo** – serwer krawędziowy odbiera transmisję strumieniową i jednocześnie replikuje przekaz do użytkowników końcowych, którzy zgłosili chęć jego odbioru. Użytkownik może odbierać transmisję tylko w czasie jej nadawania podobnie jak przekaz telewizyjny.

Operatorzy sieci CDN integrują komercyjne rozwiązania obsługi i przekazu mediów strumieniowych. Na przykład Speedera do obsługi formatu Flash używa kilku produktów firmy Macromedia: „Flash MX Professional”, „Dreamweaver”, „Video Kit”, a same serwery krawędzowe opierają się na platformie „Flash Communication Server MX”.

Dostarczanie treści dynamicznych

Coraz więcej stron WWW generowanych jest dynamicznie: katalogi produktów, aukcje internetowe, notowania giełdowe, serwisy informacyjne. Techniki takie jak ASP (ang. *Active Server Pages*) i JSP (ang. *Java Server Pages*) świetnie sprawdzają się w sytuacji, gdy na stronie wyświetlone muszą być aktualne dane zgromadzone w bazie danych. Jednak generowanie dynamicznych stron wprowadza dodatkowe obciążenie dla serwera WWW. Ten sam serwer aplikacji, który otrzymał żądanie komunikuje się z bazą danych, generuje stronę na podstawie otrzymanych wartości i wspólnych komponentów strony (np. menu, logo, reklama), a następnie dostarcza ją użytkownikowi. Strony generowane dynamicznie można bez problemu oddelegować do obsługi przez sieć CDN, z tym, że dostawca treści musi określić warunki buforowalności obiektu (czas ważności) [39]. Jeśli serwer krawędziowy otrzyma żądanie strony, której akurat nie ma w buforze (wcześniej nikt o nią nie pytał lub skończył się czas ważności), wygenerowana zostanie ona przez serwer oryginalny. Dopóki nie minie czas ważności, kolejne żądania realizowane będą przez serwer krawędziowy [39].

Wiele witryn internetowych wprowadziło personalizację stron w zależności od użytkownika (inna szata graficzna, język, reklamy). Powoduje to, że tych stron nie można zbuforować na krawędzi sieci.

Akamai umożliwia dostarczanie dynamicznych treści, których nie można zbuforować, przez swoje serwery krawędziowe. Serwer oryginalny utrzymuje stałe połączenia ze skończoną liczbą serwerów krawędziowych, zamiast z milionami użytkowników końcowych. Przyspiesza to realizację żądań, zwalnia serwer oryginalny z nawiązywania dużej ilości połączeń, co poprawia jego wydajność [39]. Dodatkowo przesłane treści mogą zostać skompresowane.

Na krawędź sieci przenoszone są również aplikacje przetwarzające i generujące dynamiczne strony. Powszechne stosowanie internetowych aplikacji J2EE wymusiło przeniesienie ich również na krawędź sieci. Akamai korzysta z platformy WebSphere stworzonej przez firmę IBM. Architektura tego typu składa się z aplikacji pracującej po stronie dostawcy treści i aplikacji pracującej na krawędzi sieci. Planując przeniesienie działającej aplikacji J2EE do sieci CDN należy wyznaczyć procesy, które będą wykonywane przez każdą ze stron. Na krawędzi sieci umieszczane są tzw. kontenery z komponentów J2EE, czyli JSP, JavaBeans, serwlety. Te kontenery realizują żądania

klientów, komunikując się w miarę potrzeby z aplikacją po stronie dostawcy treści. Komunikacja ta dotyczy pobrania na przykład aktualnych danych z bazy danych, realizowana jest za pomocą protokołów HTTP, SOAP[71], Java RMI (ang. *Remote Method Invocation*)[72], JDBC (ang. *Java Database Connectivity*)[73].

Częstotliwość dostępu do bazy danych decyduje o tym, czy umieszczenie aplikacji na krawędzi przyniesie pożądany wzrost wydajności. Dobrze sprawdzają się aplikacje sporadycznie komunikujące się z bazą danych, przesyłające niewiele danych. Jeśli aplikacja używa bazy danych tylko do odczytu, a sama baza nie zmienia się często oraz zmiany są niewielkie może zostać również przeniesiona na krawędź sieci. Natomiast dystrybucja aplikacji dokonujących częstych transakcji w bazie danych nie przynosi wzrostu wydajności, jednak korzyści płyną z innych zalet sieci CDN: większa dostępność, bezpieczeństwo, outsourcing.

ESI

ESI (ang. *Edge Side Includes*) to język znaczników. Z jego pomocą definiujemy fragmenty stron WWW, z których potem jest ona generowana na serwerze krawędziowym. ESI pozwala podzielić dynamiczną stronę na buforowalne i niebuforowalne fragmenty, każdy fragment ma przypisane własne warunki buforowalności. Dodatkowo generowanie strony odbywać się może na podstawie informacji z nagłówka żądania HTTP, ciasteczka użytkownika (ang. *cookie*), zmiennych środowiskowych [38]. Specyfikacja ESI pozwala stosować wyrażenia warunkowe oraz obsługę błędów i wyjątków. Dostępne znaczniki zamieszczone zostały w Tabeli 1, dokładny opis z przykładami użycia znaleźć można w specyfikacji ESI 1.0 [38].

Znacznik	Zastosowanie
<esi:include>	Dołączenie oddzielnie zbuforowanego obiektu.
<esi:choose> <esi:when> <esi:otherwise>	Używane do budowy wyrażeń warunkowych np. wybór jednej z kilku możliwości.
<esi:try> <esi:attempt> <esi:except>	Określenie alternatywnego postępowania, jeśli wykonanie żądania nie jest możliwe (nieodstępny serwer oryginalny).
<esi:vars>	Zastąpienie zmiennych środowiskowych.
<esi:remove>	Treść tej instrukcji zostanie wyświetlona tylko, gdy nie dokona się przetwarzanie ESI.
<!--esi ...-->	Określenie treści przetwarzanej przez ESI, ale ukrytej dla przeglądarki.
<esi:comment>	Komentarz.
<esi:inline>	Rozgraniczenie fragmentów osadzonych w odpowiedzi HTTP.

Tabela 1. Znaczniki dostępne w ESI

Dostawca treści tworzy szablon strony zawierający znaczniki HTML oraz fragmenty zawierające znaczniki ESI/HTML. Oto przykład takiego fragmentu [38]:

```
<table>
<tr>
<td colspan="2">
  <esi:attempt>
  <esi:include src=http://www.myxyz.com/news/top.html
onerror="continue"/>
  </esi:attempt>
  <esi:except>
  <!--esi
  <a href="www.myxyz.com/news/default.html">Brak strony</a>
  -->
  </esi:except>
<esi:try>
</td> </tr>
</table>
```

Gdy użytkownik zażąda dostarczenia dynamicznej strony, serwer krawędziowy sprawdza czy ma w buforze potrzebne obiekty, jeśli nie ma pobiera z serwera oryginalnego, a następnie w generuje stronę HTML i dostarcza użytkownikowi. ESI

eliminuje potrzebę uaktualniania całej strony, gdy zmienił się tylko jej fragment. Jest to otwarty standard ciągle podlegający rozwojowi.

Delta Encoding

Akamai stosuje *Delta Encoding*, alternatywne rozwiązanie do ESI pozwalające na podział strony na buforowalne i niebuforowalne fragmenty [5]. Polega ono na określeniu części strony zmienionej od ostatniej komunikacji z serwerem oryginalnym, który wysłała tylko różniący się fragment. Serwer krawędziowy tworzy stronę kompilując różniący się fragment ze statyczną częścią strony, którą przechowuje.

Zaawansowane możliwości sieci CDN

Dostawcy usług dostarczania treści chcą zdobyć jak największą liczbę klientów starają się sprostać ich wymaganiom i oczekiwaniom. Oferty poszczególnych dostawców są bardzo rozbudowane i dzięki temu można dostosować świadczoną usługę do potrzeb klienta.

Szyfrowanie treści

Coraz więcej transakcji dokonuje się przez Internet. Serwisy finansowe (obsługa konta bankowego przez WWW), szeroko pojęty *e-commerce* (np. zakupy w sklepie internetowym), wymagają przesyłania danych osobowych lub poufnych. Dlatego operatorzy sieci CDN wprowadzają do swoich sieci możliwość szyfrowania przy pomocy standardu SSL [70]. Treść jest szyfrowana na drodze pomiędzy użytkownikiem końcowym i serwerem krawędziowym, jak również pomiędzy serwerem krawędziowym i serwerem oryginalnym [5]. Szyfrowane mogą być całe strony WWW lub tylko ich elementy.

Kontrola dostępu

Firmy, aby sprostać konkurencji na rynku, często obsługują klientów czy partnerów biznesowych poprzez portal internetowy. Względny bezpieczeństwa oprócz szyfrowania treści wymuszają stosowanie kontroli dostępu. Serwery krawędziowe współpracują z serwerem oryginalnym zapewniając autoryzację użytkowników końcowych.

Akamai proponuje następujące możliwości kontroli dostępu [5]:

- Centralna autoryzacja – serwer krawędziowy odpytuje serwer oryginalny czy dany użytkownik może otrzymać dostęp do treści. W tej metodzie używa się głównie podstawową autoryzację HTTP [58] (ang. *HTTP Basic Authentication*) oraz formularz logowania.
- Autoryzacja może zostać przeniesiona na krawędź sieci wykorzystując zaszyfrowane ciasteczko (ang. *cookie*). Użytkownik uzyskuje dostęp, gdy jego żądanie zawiera prawidłowe ciasteczko.
- Serwer krawędziowy może blokować dostęp przy pomocy atrybutu określonego na podstawie żądania np. źródłowy adres IP lub URL.

Personalizacja treści (ang. *content targeting*)

Zróżnicowanie dostarczanej treści na podstawie informacji o użytkowniku, który jej zażądał. Dzięki takiej możliwości firma może budować strategię marketingową skierowaną do konkretnego klienta lub dostosowywać wygląd strony do potrzeb użytkownika końcowego. Najczęściej decyzje podejmowane są na podstawie następujących kryteriów:

- Lokalizacja geograficzna (ang. *geotargeting*)
- Preferencje językowe
- Dzień lub czas
- Typ i przepustowość połączenia
- Typ i wersja przeglądarki
- Typ i wersja systemu operacyjnego
- ISP użytkownika

Sieci CDN mają możliwość zbierania informacji o lokalizacji i przepustowości łącza użytkownika, sprzyja temu ich globalny zasięg. Dane wykorzystywane podczas personalizacji stron gromadzone są w rozproszonej bazie danych [5][6]. Serwer krawędziowy na podstawie tych informacji podejmuje decyzję o dostarczeniu użytkownikowi właściwej treści.

Często przyjmowaną strategią jest rotacja treści. W tej metodzie treść wybierana jest losowo lub sekwencyjnie z wcześniej określonego zbioru obiektów [7]. Na przykład, za każdym otwarciem strony zawierać ona będzie inny baner reklamowy.

DNS

Operatorzy sieci CDN umożliwiają outsourcing usługi DNS (ang. *Domain Name System*) [5][6]. Firma utrzymująca własny serwer DNS lub korzystająca z outsourcingu DNS lokalnego dostawcy internetu jest bardziej narażona na pojedyncze miejsce awarii (ang. *single point of failure*), spowodowanej uszkodzeniem sprzętu, brakiem prądu lub awarią sieci telekomunikacyjnej. Popularną metodą ataku typu *Denial of Service* na witrynę WWW, jest atak na jej serwery DNS. Unieruchomienie usługi DNS uniemożliwi użytkownikom wyświetlanie tej witryny. Globalnie rozproszona, redundantna architektura sieci CDN, w znacznym stopniu uniemożliwia przeprowadzenie tego typu ataku. Szczegóły implementacji usługi DNS w sieci CDN przedstawione zostały w [44].

„Speedera Smart Storage Manager”

W sieci CDN zauważalna jest prawidłowość 80/20 (zwana regułą Pareto) [6], to znaczy, że 80% ruchu generują zapytania o 20% treści. Z tego powodu warto tą „gorącą treść” rozmieścić na większej ilości serwerów krawędziowych niż pozostałe 80% mniej popularnych treści. Speedera udostępnia usługę, która pozwala dostawcy treści określić „wielkość” krawędzi sieci poprzez skonfigurowanie polityki określającej popularność treści oraz maksymalną powierzchnię dyskową zajmowaną przez tę treść. Na tej podstawie treść zostanie rozdystrybuowana do właściwych punktów prezentacji oraz możliwe są późniejsze automatyczne migracje treści bliżej lokalizacji, z których przyszło najwięcej zapytań.

Dzięki temu dostawca treści ma możliwość regulowania proporcji pomiędzy optymalną wydajnością dostępności treści dla użytkowników końcowych, a ponoszonymi kosztami przechowywania treści na serwerach krawędziowych.

Okresowe użytkowanie sieci CDN

Operatorzy sieci CDN proponują klientom usługę, która polega na okresowym wykorzystywaniu sieci CDN do rozszerzania dostępności ich serwisów [5][6]. Sieć CDN w takim przypadku staje się narzędziem pozwalającym na rozszerzenie własnych

zasobów serwerowych w momencie, gdy przestają być wystarczające. Może się tak stać podczas nagłego wzrostu zapytań do serwera. Wystarczające do tej pory łącze i zasoby sprzętowe serwera przestają w takiej sytuacji spełniać swoje zadania.

Dostawcy treści korzystający z tej usługi, muszą określić warunki (np. maksymalna ilość żądań na sekundę), w jakich ma być uruchomione przekierowanie żądań do obsługi przez sieć CDN.

Oryginalny serwer witryny WWW jest monitorowany przez system kierowania żądań (ang. *traffic management*) i po wystąpieniu szczególnych warunków część żądań zostanie skierowana do serwerów krawędziowych.

Dzięki takiemu rozwiązaniu witryna WWW będzie dostępna nawet podczas awarii oryginalnego serwera, gdyż kopia witryny znajduje się w sieci CDN (pod warunkiem, że zawiera statyczne treści [6]).

Takie podejście pozwala firmie utrzymującej witrynę WWW zredukować koszty związane z utrzymaniem rozbudowanej infrastruktury, jeśli jest ona potrzebna okresowo. Operator sieci CDN w rozliczeniu weźmie pod uwagę udostępnione pasmo i czas, w którym było ono wykorzystywane [18].

Monitorowanie i raportowanie

Dostawcy usług dostarczania treści dostarczają swoim klientom (dostawcom treści) różnorodne statystyki i raporty. Pozwala to lepiej zrozumieć zachowania użytkowników. I na tej podstawie dostosować treść do ich potrzeb. Statystyki mogą dotyczyć najpopularniejszych plików, użytkowników najczęściej odwiedzających witrynę WWW, geograficznego położenia użytkowników, przepustowości łącza do użytkownika, systemu operacyjnego czy typu przeglądarki [6][7]. Dane te zebrane są w plikach logów i udostępniane są dostawcom treści [6].

Dostępne są również statystyki w czasie rzeczywistym takich parametrów jak ruch generowany przez użytkowników, ilość żądań określonego zasobu w czasie.

3. Krawędź sieci

Dostarczanie treści do krawędzi sieci

Zasadniczo są dwa sposoby na dostarczenie treści do krawędzi sieci. Serwery krawędziowe same pobierają treść, gdy nadejdzie żądanie użytkownika. Treść jest równocześnie dostarczana do użytkownika i buforowana w celu obsługi kolejnych żądań. W drugim przypadku treść musi zostać rozdystrybuowana zanim nadejdzie żądanie jej dostarczenia. Tego sposobu używa się przy replikacji dużych obiektów, wskazane jest takie skonfigurowanie replikacji, aby proces ten odbywał się poza godzinami największego ruchu na łączach. Rozwiązanie firmy Cisco [12] pozwala określić dni tygodnia i godziny, w których może odbywać się replikacja oraz przepustowość łącza, jaką może zajmować ruch związany z replikacją treści pomiędzy menadżerem dystrybucji treści, a serwerem krawędziowym.

Dostawca treści może dokonywać zmian w treści, którą przekazał do sieci CDN. Aby mieć pewność, że na krawędzi sieci znajduje się aktualna kopia obiektu, serwery krawędziowe okresowo komunikują się z serwerem oryginalnym, aby sprawdzić czy obiekt nie został zmodyfikowany. Dostawca treści może życzyć sobie, aby zmodyfikowany obiekt został natychmiast rozesłany do wszystkich serwerów krawędziowych. SpeedEye [6] udostępnia aplikację, która unieważni obiekt na wszystkich serwerach krawędziowych, w wyniku tego pobiorą one nową jego wersję.

Rozmieszczenie serwerów krawędziowych

Serwery krawędziowe to urządzenia buforujące treść na krawędzi sieci. Powinny znajdować się jak najbliżej użytkownika. Ponieważ jest ich ograniczona ilość, trzeba je rozmieścić tak, aby zaspokajały potrzeby wszystkich użytkowników tzn. należy dążyć do zmniejszenia średniego czasu dostępu do treści i zajmowanego pasma w sieci na drodze pomiędzy klientem i serwerem. Aby rozwiązać ten problem powstały skomplikowane modele [46], które niestety ze względu na swoją złożoność pozostają tylko teoretyczne. Na potrzeby sieci CDN opracowano heurystyczne algorytmy.

W [46] przedstawiony został „Chciwy algorytm” (ang. *Greedy Algorithm*). Załóżmy, że potrzebujemy rozmieścić M serwerów, a do wyboru mamy N potencjalnych lokalizacji ($M < N$). Algorytm w pierwszej iteracji ocenia każdą z N

lokalizacji i określa, czy jest odpowiednia do ulokowania serwera tzn. oblicza koszt (np. zajętość pasma) związany z każdą lokalizacją przy założeniu, że żądania wszystkich klientów trafiają do tej lokalizacji i wybiera lokalizację o najmniejszym koszcie. W drugiej iteracji szuka drugiej lokalizacji, która w połączeniu z już wybraną, będzie miała najmniejszy koszt. Przy obliczaniu kosztu zakłada się, że żądania klientów kierowane są do najbliższego serwera. Algorytm wykonuje M iteracji. Algorytm ten wymaga wiedzy o lokalizacji klientów w sieci, a te dane nie zawsze mogą być dostępne.

Inny algorytm *Hot Spot* [46] sortuje potencjalne lokalizacje według żądań generowanych w ich pobliżu (parametr konfigurowalny). Następnie serwery umieszcza się blisko klientów generujących największy ruch.

W [47] zaprezentowano inną ciekawą technikę. Zakłada się, że węzeł z największą liczbą węzłów z nim połączonych (sąsiadów), może osiągnąć więcej węzłów z mniejszym opóźnieniem. Serwery umieszcza się zatem kolejno w węzłach według malejącej liczby ich sąsiadów. Powyższe założenia odnoszą się do istniejącej topologii systemów autonomicznych, gdzie węzeł oznacza pojedynczy system autonomiczny, a połączenie z sąsiednim węzłem odpowiada połączeniu z użyciem protokołu routingu BGP dwóch systemów autonomicznych.

Rozmieszczenie kopii treści

Ważnym zagadnieniem jest rozmieszczenie kopii obiektów na serwerach krawędziowych. Dąży się do tego, aby efektywne rozmieszczenie obiektów zapewniało równe obciążenie serwerów żądaniami użytkowników.

Rozważania w tym temacie poprzedzone powinny być pewnymi założeniami [48]. Serwery krawędziowe rozmieszczone są w systemach autonomicznych (SA), dla uproszczenia można przyjąć, że jeden SA to jeden ISP. Ilość SA wynosi I , każdy SA_i ma pojemność S_i bajtów i zagregowaną częstotliwość żądań klientów λ_i , ilość obiektów wynosi J , każdy obiekt j ma rozmiar b_j oraz prawdopodobieństwo p_j , że klient go zażąda (popularność obiektu). Replikację obiektów można przeprowadzić następującymi metodami [48]:

- Na chybił trafił (ang. *random*) – przypisuje obiekt do losowych SA, przypadkowy wybór podlega tylko ograniczeniom pojemności SA.
- Według popularności – każdy SA przechowuje najbardziej popularne obiekty spośród obiektów, których żądają od niego klienci. Oczywiście przechowuje je

w kolejności malejącej popularności, a ich ilość zależy od pojemności. SA sam musi określać popularność obiektów na podstawie żądań klientów, poza tym nie potrzebuje żadnych dodatkowych informacji z zewnątrz.

- Pojedynczy chciwy algorytm (ang. *Greedy-Single*) – każdy SA_i oblicza metrykę $C_{ij} = p_j d_{ij}(x_o)$ dla każdego obiektu j , gdzie $d_{ij}(x_o)$ to najkrótszy dystans potrzebny do skopiowania obiektu j do SA_i z lokalizacji x , w tym wypadku x_o oznacza serwer oryginalny. Następnie sortuje metryki obiektów w porządku malejącym i przechowuje tyle obiektów na ile pozwoli pojemność. SA sam musi określać popularność obiektów na podstawie żądań klientów, dodatkowo sieć CDN potrzebuje informacji o topologii sieci do wyliczeń d_{ij} . W tej metodzie metryki są obliczane tylko raz, a późniejsza kooperacja pomiędzy SA nie jest konieczna.
- Globalny chciwy algorytm (ang. *Greedy-Global*) – sieć CDN najpierw oblicza metrykę $C_{ij} = \lambda_i p_j d_{ij}(x_o)$ dla każdego SA_i i każdego obiektu j . Następnie wybierana jest para SA_i i j z największą metryką, obiekt j przechowywany jest w SA_i . Teraz obiekt j posiada nową lokalizację x_1 . Znowu obliczane są wszystkie metryki z uwzględnieniem nowej lokalizacji. Ponownie wybierana jest para z największą metryką i tworzy się nowa lokalizacja x_2 . Iteracje wykonywane są do momentu zapełnienia wszystkich SA.

Informacje dotyczące żądań użytkowników, o których była mowa powyżej, uzyskiwane są na podstawie logów serwera oryginalnego.

Przeprowadzone symulacje [48] wykazały, że największą wydajność uzyskuje się stosując „globalny chciwy algorytm”.

Współpraca serwerów krawędziowych

Zastosowanie prostej metody rozmieszczenia kopii treści w połączeniu z kooperacją pomiędzy serwerami znacząco poprawia wydajność [48]. Serwery krawędziowe zorganizowane mogą być postaci klastrów serwerów lub struktur hierarchicznych. Do ich utrzymania stosuje się między innymi *Internet Cache Protocol* (ICP). ICP pozwala serwerom buforującym treść na odpytywanie innych serwerów w tej samej sieci CDN. Jeśli do serwera krawędziowego przyjdzie żądanie dostarczenia treści, której nie posiada, używa protokołu ICP do znalezienia treści na innych

serwerach krawędziowych. W ten sposób treść zostanie pobrana z bliższej lokalizacji niż serwer oryginalny. ICP dokładnie opisany jest w [50] i [51].

Alternatywnymi protokołami do ICP są:

- *Hypertext Caching Protocol* (HTCP) [41]
- *Cache Array Routing Protocol* (CARP) [42]
- *Cache Digests* [43]

Samoorganizująca się Architektura Rozproszona

SODA (ang. *Self-Organizing Distributed Architecture*) to opatentowana technologia firmy Cisco wspomagająca efektywną replikację treści. Serwery krawędziowe firmy Cisco współpracując z menadżerem dystrybucji treści organizują się w hierarchiczną strukturę, tzw. drzewo dystrybucji. Menadżer jest korzeniem tego drzewa. Każdy węzeł monitoruje aktualny stan sieci oraz połączeń do sąsiednich węzłów i automatycznie dostosowuje się do zmieniających się warunków w sieci. Zmiany rozpoznawane są prawie natychmiast (np. utrata połączenia z sąsiednim serwerem) i następuje szybka reorganizacja struktury bez przerw w obsłudze sieci [19].

Proces dodawania nowego węzła do drzewa dystrybucji przebiega następująco [37]:

1. Serwer krawędziowy rozpoczyna proces rejestracji, od menadżera otrzymuje listę węzłów drzewa.
2. Kontaktuje się z każdym węzłem z listy, mierząc czas opóźnienia odpowiedzi.
3. Wybiera węzeł, który odpowiedział najszybciej i rejestruje się w hierarchii drzewa jako jego „dziecko”.

Stosowanie SODA daje największe efekty przy dystrybucji treści wymagających dużych przepustowości. Struktura hierarchiczna powoduje, że pobranie treści z serwera oryginalnego nastąpi tylko raz, dalej treść zostanie automatycznie powielona pomiędzy serwerami krawędziowymi drzewa dystrybucji.

Oddelegowanie witryny do obsługi przez sieć CDN

Do obsługi przez Sieć Dostarczania Treści mogą zostać przeznaczone całe witryny WWW lub pewne ich elementy [19].

W przypadku, gdy dostawca treści oddelegowuje swoją domenę do sieci CDN, serwer DNS operatora sieci CDN staje się serwerem autorytatywnym (ang. *authoritative*) dla tej domeny tzn. takim, który jest źródłem danych dotyczących tej domeny. Jednak w większości przypadków tylko wybrane elementy stron internetowych są wysyłane do obsługi przez sieć CDN. W takim wypadku dla danego dostawcy treści tworzona jest nowa domena, dla której serwer DNS operatora sieci CDN jest serwerem autorytatywnym. Na przykład domenie `www.portal.com` odpowiada domena `portal.cdn.net` w sieci CDN. Dostawca treści na oryginalnym serwerze musi zmodyfikować odnośniki do obiektów, które obsługiwane są przez sieć CDN, czyli zamiast:

```

```

powinno być:

```

```

Poniżej przedstawiono jak Akamai transformuje URL do obiektu przekazanego do obsługi w sieci CDN [36].

Przed transformacją: `www.foo.com/logo.gif`

Po transformacji:

```
a836.g.akamai.net/7/836/123/e358f5db0045e9/www.foo.com/logo.gif  
1      2      3 4 5      6      7
```

Oto co oznaczają zaznaczone części URL:

1. Numer seryjny – wirtualna grupa obiektów, które są serwowane zawsze z tego samego zbioru serwerów krawędziowych.
2. Domena Akamai – korzystając z DNS żądania trafiają do właściwych ruterów treści.
3. Oznaczenie typu – informacja dla serwerów krawędziowych, w jaki sposób mają interpretować URL.
4. Numer seryjny – ten sam co w punkcie 1.
5. Oznaczenie dostawcy treści – unikalny identyfikator dostawcy treści, wykorzystywany w raportowaniu, logach i bilingu.
6. Dopełnienie danych – używane do weryfikacji wersji obiektu. W zależności od oznaczenia typu może zawierać czas utraty ważności przez obiekt lub ciąg znaków jednoznacznie identyfikujący wersję obiektu (skrót MD5 z obiektu lub numer wersji). Gdy obiekt zostanie zmodyfikowany, pole to również się zmieni.

7. Oryginalny URL – używany przez serwery krawędziowe w celu pobrania obiektu z oryginalnego serwera.

Gdy istnieje już domena w sieci CDN mapująca obiekty z serwera oryginalnego dostawcy treści i nie cała witryna WWW ma zostać oddelegowana, dostawca treści powinien wskazać obiekty, które mają zostać rozdystrybuowane w sieci CDN. Cisco realizuje to w ten sposób, że dostawca treści wymienia obiekty oraz określa warunki bufrowalności w specjalnym pliku konfiguracyjnym napisanym w XML (ang. *Extensible Markup Language*). Pliki przeznaczone do dystrybucji mogą pochodzić z różnych lokalizacji (kilka serwerów oryginalnych). Zasady tworzenia plików konfiguracyjnych znajdują się w [23].

4. Globalne kierowanie treści

Globalne kierowanie treści jest kluczowym podsystemem sieci CDN. Realizuje on kierowanie żądań klientów do serwera krawędziowego, który zawiera żadaną treść i będzie w stanie najszybciej ją dostarczyć. Istnieje kilka metod realizacji systemu kierowania treści.

Przekierowanie DNS

Najczęściej spotykaną realizacją systemu globalnego kierowania treści jest integracja funkcjonalności kierowania żądań z systemem DNS. Implementowany jest głównie w modelu Internet CDN, gdyż DNS jest powszechnie stosowany w Internecie. Śmiało można powiedzieć, że DNS ma kluczowe znaczenie dla właściwego funkcjonowania Internetu. Rozwiązywanie nazw domenowych następuje zawsze, gdy maszyna klienta nie zna adresu IP powiązanego z daną nazwą domenową. Przejrzysty opis działania systemu DNS przedstawiony jest w [20]. Każdy serwer DNS posiada podręczną pamięć *cache*, którą wykorzystuje do tymczasowego zapamiętywania informacji o domenach, o które był „pytany”. Czas, jaki dana domena jest pamiętana określa zmienna TTL (ang. *Time to Live*) przypisana do rekordu. Czas ważności (TTL) jest parametrem konfiguracyjnym serwera DNS, określa jak długo dane mają być przechowywane w bazie danych DNS. Po upływie tego czasu, wyrażonego w sekundach, dane tracą ważność. Przez czas TTL nieautorytatywne serwery DNS rozwiązują zapytanie bez odpytywania serwera autorytatywnego. Nie należy mylić TTL rekordu DNS z TTL pakietu IP.

Do realizacji funkcjonalności rutera treści wykorzystać można rekordy typu A, NS, CNAME [22]. Z tego względu wyróżnia się kilka technik realizacji przekierowania DNS [24]:

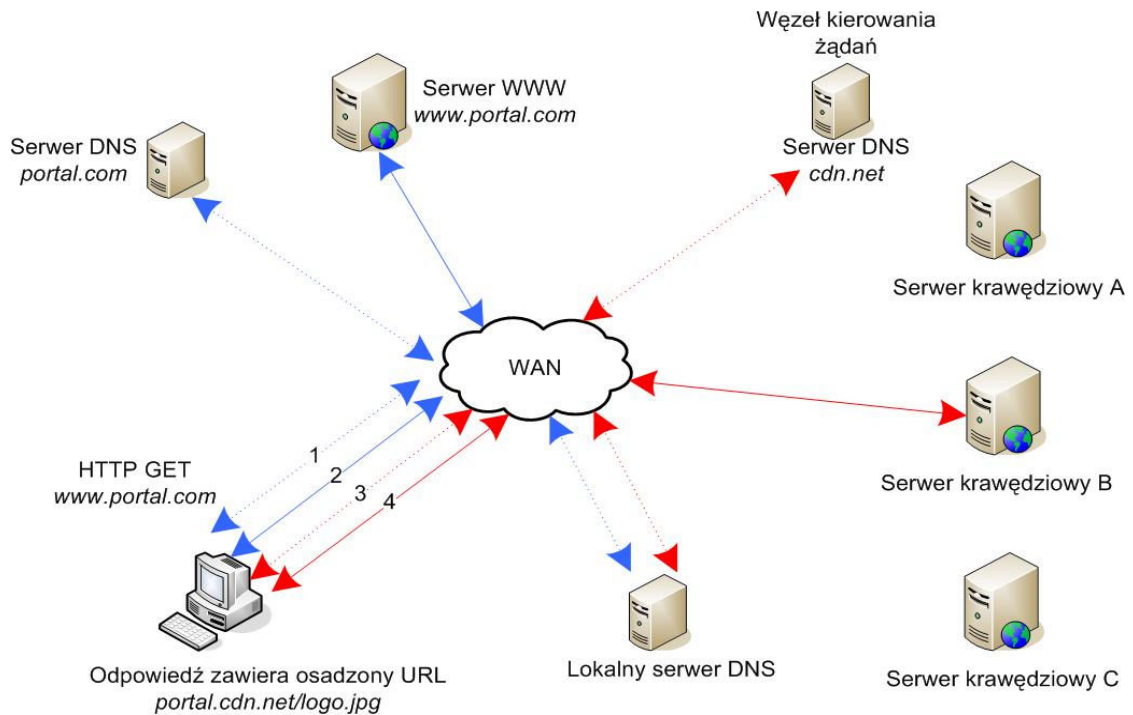
- Pojedyncza odpowiedź – serwer DNS zwraca adres IP najlepszego serwera krawędziowego w postaci rekordu A. Adres ten może być również wirtualnym adresem IP farmy serwerów.
- Mnoga odpowiedź – serwer DNS w odpowiedzi zwraca kilka rekordów z adresami IP serwerów krawędziowych.

- Przekierowanie wielopoziomowe – w procesie rozwiązywania nazw domenowych bierze udział więcej niż jeden serwer DNS funkcjonujący jako ruter treści. Podejmowanie decyzji staje się bardziej złożone, możliwa jest specjalizacja poszczególnych serwerów skierowana na obsługę określonych żądań. Na przykład serwer DNS wyższego poziomu obsługujący żądania z pewnego obszaru, kieruje je do ściśle określonego serwera niższego poziomu w celu podjęcia dokładniejszych decyzji. Przekierowanie wielopoziomowe realizuje się przy użyciu rekordów NS lub CNAME [22]. W pierwszym przypadku liczba ruterów treści jest ograniczona liczbą członów nazwy domenowej. Wynika to z faktu, że lokalny serwer DNS odrzuci żądanie, jeśli serwer autorytatywny nie rozwiąże kolejnego członu nazwy domenowej. W drugim przypadku przekierowanie następuje do całkiem nowej domeny, co wydłuża czas całego procesu o czas rozwiązania nazwy nowej domeny.

Proces pobierania treści z przekierowaniem DNS

Aby lepiej zrozumieć proces pobierania treści prześledzimy go od momentu wpisania w przeglądarkę adresu strony internetowej `www.portal.com` zawierającej osadzony URL `portal.cdn.net/logo.jpg` do obiektu, który przekazany został do obsługi przez sieć CDN. Proces ten podzielony został na cztery etapy, które zaznaczone są na Rys. 7. Linie przerywane odnoszą się do procesu rozwiązywania nazw DNS, a linie ciągłe do pobierania treści przy pomocy protokołu HTTP.

Warto w tym momencie przypomnieć, że ruter treści jest jednocześnie autorytatywnym serwerem DNS dla domeny, której nazwy rozwiązuje, w tym wypadku `cdn.net`.



Rys. 7. Pobieranie treści

Etap 1:

1. Do lokalnego serwera DNS wysyłane jest żądanie szukania nazwy `www.portal.com`.
2. Jeśli lokalny serwer DNS nie posiada aktualnie adresu IP dla `www.portal.com` w pamięci podręcznej, w wyniku działania mechanizmu protokołu DNS lokalny serwer DNS komunikując się z innymi serwerami DNS w Internecie uzyskuje poszukiwany adres IP. W przypadku, gdy dostawca treści oddelegował całą witrynę WWW do obsługi przez sieć CDN, serwer DNS dla domeny `portal.com` powinien zwrócić adres IP rutera treści w postaci rekordu NS (w opisywanym przypadku tak nie jest).
3. Lokalny serwer DNS podaje przeglądarce adres IP serwera `www.portal.com`.

Etap 2:

1. Przeglądarka wysyła pod podany adres IP żądanie HTTP GET
2. Serwer WWW otrzymuje żądanie i wysyłana jest do przeglądarki odpowiedź, która zawiera osadzony URL `portal.cdn.net/logo.jpg` do obiektu umieszczonego w sieci CDN

Etap 3:

1. Do lokalnego serwera DNS wysyłane jest żądanie szukania nazwy `portal.cdn.net`.
2. Lokalny serwer DNS nie posiada aktualnie adresu IP dla `portal.cdn.net` w pamięci podręcznej. W wyniku działania mechanizmu protokołu DNS lokalny serwer DNS komunikując się z innymi serwerami DNS w Internecie uzyskuje adres serwera DNS, który obsługuje domenę `cdn.net`, serwer ten jest jednocześnie węzłem kierowani żądań.
3. Lokalny serwer DNS wysyła żądanie do węzła kierowania zawartością dla adresu IP serwera `portal.cdn.net`.
4. Węzeł kierowania żądań podaje lokalnemu DNS adres IP serwera krawędziowego.
5. Lokalny serwer DNS podaje przeglądarce adres IP serwera `portal.cdn.net`.

Etap 4:

1. Przeglądarka wysyła do serwera krawędziowego żądanie pliku związanego z adresem URL `portal.cdn.net/logo.jpg`.
2. Serwer krawędziowy otrzymuje żądanie. Jeżeli treść została na nim umieszczona w procesie dystrybucji, jest ona natychmiast wysyłana do przeglądarki. Jeżeli nie jest aktualnie przechowywana, inicjowane jest ściągnięcie jej z oryginalnego serwera `www.portal.com` i następnie przesłanie do przeglądarki.

Przy następnym żądaniu zawartości związanej z `portal.cdn.net`, lokalny serwer DNS posiada adres IP optymalnego węzła dostarczania treści i pomijane są kroki od 2 do 4 etapu 3. Kiedy kończy się czas ważności (TTL) dla adresu IP serwera krawędziowego, powtarzane są wszystkie kroki.

Zazwyczaj dąży się do tego, aby czas ważności był bardzo krótki (20 sekund). Wynika to z faktu, że w kolejnych żądaniach tego samego klienta, ruter treści może skierować go do innego serwera krawędziowego. Jeśli czas ważności będzie zbyt długi, może okazać się, że lokalny serwer DNS zawiera już nieaktualne dane, a mimo to dostarczy je klientowi.

Rutery treści podejmują decyzję bazując na adresie IP lokalnego serwera DNS, od którego przyszło żądanie. System kierowania żądań podejmując decyzję zakłada, że

wybór najlepszego serwera krawędziowego dla IP lokalnego serwera DNS jest również poprawny dla IP klienta, który używa tego serwera do rozwiązywania nazw domenowych.

Przekierowanie DNS na przykładzie urządzeń Cisco

Rutery treści firmy Cisco wspierają przekierowanie DNS. W tym procesie możemy wyróżnić dwa etapy. Najpierw lokalny serwer DNS komunikuje się z ruterem treści. Po wybraniu najlepszego serwera krawędziowego, ruter treści zwraca jego adres IP w postaci rekordu NS.

Ruter treści w zależności od stopnia poprawności wyboru, określa liczbę rekordów NS, które zwróci i ich czas TTL. W przypadku pewnego wyboru zwraca dwa lub trzy rekordy NS z relatywnie długim czasem TTL, gdy wybór jest niepewny może zwrócić do ośmiu rekordów NS z relatywnie krótkim czasem TTL.

W drugim etapie lokalny serwer DNS kontaktuje się z jednym z serwerów krawędziowych podanych w odpowiedzi od rutera treści. Ten serwer zwraca mu swój adres IP w postaci rekordu A. Dopiero teraz adres IP serwera krawędziowego zostaje przekazany użytkownikowi, który go zażądał. Jeśli zapytany serwer krawędziowy nie odpowiedział, odpytywany jest następny.

W przypadku, gdy lokalny serwer DNS otrzymuje kolejne żądania rozwiązania nazwy domenowej, zanim skończy się czas TTL rekordów NS, kontynuuje kontaktowanie się z kolejnymi serwerami krawędziowymi. Ponieważ każdy rekord NS reprezentuje poprawny serwer krawędziowy zdolny do obsługi żądania klienta, więc zwracanie przez ruter treści kilku rekordów NS zwiększa odporność na usterki serwerów krawędziowych i umożliwia rozłożenie obciążenia pomiędzy nimi.

Ruting statyczny

Ponieważ decyzja o skierowaniu podejmowana jest na podstawie adresu IP lokalnego serwera DNS, może się zdarzyć, że wskazany serwer krawędziowy nie będzie w rzeczywistości najlepszym serwerem do obsługi żądania klienta. W przypadku strumieniowych przekazów wideo taka rozbieżność może niekorzystnie wpłynąć na jakość świadczonej usługi.

Firma Cisco proponuje dostawcom treści możliwość korygowania takich przypadków poprzez statyczne wpisy. Każdy serwer krawędziowy ma przypisaną „szczegółową strefę” (ang. *coverage zone*), czyli zbiór adresów IP komputerów

klientów, które są poprawnie przypisane do obsługi przez ten serwer. Wskazane są lokalne serwery DNS, których żądania powinny być specjalnie traktowane. Na tej podstawie żądania użytkowników końcowych są kierowane do właściwego serwera krawędziowego. Informacje o „szczegółowej strefie” zebrane są w pliku tekstowym, który jest dostarczany operatorowi sieci CDN. Rutery treści i serwery krawędziowe okresowo pobierają zawartość tego pliku.

Przykładowy plik „szczegółowej strefy” może wyglądać następująco [23]:

```
CZ1
DNS 10.89.11.1
network 10.89.0.0/12 Waltham

DNS 10.89.11.1 10.89.50.113
network 10.89.13.20/32 Plymouth; Boxborough
```

„CZ1” to nazwa, która identyfikuje plik w sieci CDN. Po nagłówku „DNS” podajemy adresy IP lokalnych serwerów DNS. Po nagłówku „network” adres lub zakres adresów IP komputerów klientów, a następnie nazwy serwerów krawędziowych, które powinny obsługiwać żądania z tych komputerów.

Ruting hybrydowy

W rozwiązaniu firmy Cisco wskazane jest używanie jednocześnie routingu dynamicznego i statycznego. W takim przypadku ruter treści sprawdza czy lokalny serwer DNS został wyszczególniony w pliku „szczegółowej strefy”. Jeśli nie, żądanie jest obsługiwane w sposób standardowy (ruting dynamiczny). Jeśli tak, ruter treści wybiera losowo jeden lub dwa serwery krawędziowe, które znajdują się w pliku „szczegółowej strefy”. Następnie ich adresy IP, w postaci rekordów NS, zostają zwrócone lokalnemu serwerowi DNS.

Kiedy wybrany serwer krawędziowy otrzyma żądanie od klienta, na podstawie informacji zawartych w pliku „szczegółowej strefy” określa, czy jest poprawnie przypisany do obsługi żądań tego klienta [23]. Jeśli tak jest, to serwer dostarcza treść. Jeśli nie, generuje przekierowanie zawierające adres IP klienta do rutera treści. Ruter treści znów dokonuje wyboru serwera krawędziowego na podstawie informacji z pliku „szczegółowej strefy”, tym razem znając już IP komputera klienta.

Kodowanie typu obiektu w nazwie domenowej

System DNS rozwiązuje nazwy domenowe, nie widzi natomiast całego URL i dlatego ruter treści nie może podejmować decyzji na podstawie typu żądanej treści. Istnieje możliwość umieszczania typu obiektu w nazwie domenowej [24]. Wystarczy dokonać prostego podziału treści ze względu na jej typ w momencie skierowania treści do obsługi przez sieć CDN. Obiekty różnych typów np. obrazy i filmy umieszczamy na różnych serwerach krawędziowych. Następnie wystarczy skonfigurować ruter treści tak, aby np. takie nazwy domenowe: `obrazy.portal.cdn.net` i `filmy.portal.cdn.net` rozwiązywane były do adresów IP tych serwerów krawędziowych.

Niewątpliwą zaletą jest to, że w momencie rozwiązywania nazwy domenowej, węzeł kierowania żądań posiada informacje o typie treści. Jest też znacząca wada takiego rozwiązania. Lokalny serwer DNS będzie musiał dokonać rozwiązania kilku nazw domenowych, aby użytkownik mógł wyświetlić stronę internetową zawierającą kilka typów obiektów.

Dystrybucja obciążenia przez rotowanie rekordów

Najbardziej popularnym rekordem DNS jest rekord typu A, który określa adres IP dla danej nazwy hosta. W niektórych przypadkach ta sama nazwa domenowa może zostać rozwiązana do kilku adresów IP [21]. Na przykład, jeśli 3 serwery wspierają witrynę `www.portal.com`, serwer DNS może zawierać 3 rekordy:

```
www.portal.com IN      A 192.167.0.1
www.portal.com IN      A 192.167.1.1
www.portal.com IN      A 192.167.2.1
```

Kiedy do serwera dotrze zapytanie klienta o nazwę `www.portal.com`, zwróci on wszystkie 3 rekordy. Większość przeglądarek internetowych za każdym razem skorzysta z pierwszego adresu w odpowiedzi. Aby tego uniknąć, serwer DNS używa algorytmu *Round Robin* [61]. Z każdym żądaniem, algorytm dokonuje rotacji kolejności zwracanych rekordów. Na przykład za pierwszym razem może zwrócić:

```
192.167.0.1      192.167.1.1      192.167.2.1
```

za drugim:

```
192.167.1.1      192.167.2.1      192.167.0.1
```

za trzecim:

```
192.167.2.1      192.167.0.1      192.167.1.1
```

i tak dalej.

Takie rozwiązanie jest przydatne, gdy mamy kilka równoważnych serwerów i chcemy rozdzielić ruch pomiędzy nimi.

Również ruter treści może w odpowiedzi zwracać więcej niż jeden adres serwera krawędziowego w postaci kilku rekordów A [24]. Rotacja rekordów następuje na lokalnym serwerze DNS klienta.

Widoki

Popularny serwer DNS, BIND (ang. *Berkeley Internet Name Domain*) od wersji 9 udostępnia mechanizm tzw. widoków (ang. *views*) [22]. Dzięki niemu, w zależności od klienta, który przysłał żądanie, zachowanie się serwera DNS może być zróżnicowane. Dla poszczególnych grup klientów można skonfigurować oddzielne strefy (ang. *zone*) [22], jak również oddzielną konfigurację serwera nazw. Widoki definiujemy za pomocą wyrażenia `view` w pliku konfiguracyjnym `named.conf`. Lista wszystkich opcji używanych wewnątrz wyrażenia `view` jest długa, dostępna na stronie internetowej [26] lub w pliku `doc/misc/options` w dystrybucji serwera BIND. Widok może, choć nie musi, mieć nadaną niepowtarzalną nazwę, aby można go było jednoznacznie identyfikować oraz podwyrażenie `match-clients` przyjmujące jako argument listę adresów IP. Zapytania, których adresy źródłowe IP pasują do tej listy, zostaną obsłużone tak, jak definiuje to ten widok. Gdy brak jest wyrażenia `match-clients`, widok pasuje do wszystkich adresów IP. Jeśli adres IP pasuje do kilku widoków, pod uwagę zostanie wzięty pierwszy z widoków. Z tego względu podczas konfigurowania należy zadbać o prawidłową kolejność definicji poszczególnych widoków [22].

Poniżej przedstawiono przykładową konfigurację widoków w pliku `named.conf`:

```
options { directory "/var/named"; };
view "oddzial_1" {
match-clients { 192.168.1.0/24; };
recursion no;
zone "firma.com" {
type master;
file "db.firma.com.oddzial_1";
};
view "oddzial_2" {
match-clients { 192.168.2.0/24; };
recursion no;
zone "firma.com" {
type master;
file "db.firma.com.oddzial_2";
```

};};

Pomimo, że nazwy stref są takie same w obu widokach, ich definicje różnią się, bo zapisane są w różnych plikach stref. Jeśli założymy, że w każdym oddziale firmy znajduje się urządzenie buforujące (serwer krawędziowy), którego chcemy się odwoływać poprzez nazwę `bufor.firma.com`, w pliku strefy dla oddziału pierwszego nazwę `bufor.firma.com` przypisujemy adres IP z sieci 192.168.1.0/24, a w pliku strefy dla oddziału drugiego nazwę `bufor.firma.com` przypisujemy do adresu IP z sieci 192.168.2.0/24. W zależności od adresu IP użytkownika (czy będzie z pierwszego czy drugiego oddziału), rozwiązanie nazwy `bufor.firma.com` wskaże serwer krawędziowy znajdujący się w jego sieci lokalnej.

Ograniczenia przekierowania DNS

- DNS pozwala na podejmowanie decyzji na podstawie nazwy domenowej. Idealny system powinien umożliwiać przekierowanie na podstawie informacji o obiekcie.
- Ruter treści zwraca rekordy DNS z krótkim czasem TTL, aby szybciej reagować na zmiany. Jednocześnie wzrasta ilość zapytań do serwerów DNS, przez co są one bardziej obciążone.
- Niektóre implementacje serwerów DNS odbiegają od przyjętych standardów, np. nie wszystkie honorują pole DNS TTL.
- Przekierowanie DNS bazuje na adresie IP lokalnego serwera DNS, przez co wyznaczana „odległość” klienta do serwera krawędziowego ma charakter przybliżony, a wyznaczony serwer krawędziowy może nie być najlepszym. Badania wykazały [49], że na podstawie czasu opóźnienia odpowiedzi lokalnego serwera DNS trudno przewidzieć rzeczywisty czas odpowiedzi klienta, natomiast odległość klienta od serwera nazw wynosi typowo 8 skoków lub nawet więcej. Sprawdzając programem `tracert` ilość skoków do serwerów DNS rekomendowanych przez ISP autora, otrzymano wynik 4 i 6 skoków.
- Klienci używający tego samego lokalnego serwera DNS, podczas okresu TTL przekierowywani będą do tego samego serwera krawędziowego, co może wywołać jego przeciążenie.

Przekierowanie HTTP 302

Przeglądarka internetowa i serwer WWW komunikują się za pomocą protokołu HTTP. Za każdym razem, gdy przeglądarka zażąda udostępnienia strony WWW, serwer WWW w odpowiedzi wysyła kod stanu HTTP (ang. *HTTP Status Code*) [59]. Kody stanu mają postać trzycyfrowych liczb. Wyróżniamy pięć grup kodów stanu:

- Informacyjne (1xx), np. kod 100 *Continue* – kontynuuj żądanie
- Udań (2xx), np. kod 200 *OK* – żądanie powiodło się
- Przekierowanie (3xx), np. kod 301 *Moved Permanently* – obiekt trwale przeniesiony
- Błąd klienta (4xx), np. kod 404 *Not Found* – serwer nie może znaleźć obiektu
- Błąd serwera (5xx), np. kod 500 *Internal Server Error* – wewnętrzny błąd serwera

W procesie kierowania treści wykorzystywane jest przekierowanie realizowane za pomocą kodu 302 *Found*, który oznacza tymczasowe przeniesienie obiektu. Żądany zasób znajduje się pod innym URL. Ze względu na to, że przekierowanie jest tymczasowe, czyli może się zmienić, klient w kolejnych odniesieniach tego zasobu nie powinien używać URL zwróconego w wyniku przekierowania [59]. Odpowiedź serwera powinna zawierać krótką wiadomość z linkiem do nowego URL. Poniżej przedstawiono przykładową odpowiedź serwera, wymuszająca przekierowanie:

```
HTTP/1.1 302 Found
Connection: close
Date: Sun, 25 Feb 2001 18:42:53 GMT
Location: http://webcab.de/
Server: Apache/1.3.9 (Win32) ApacheJServ/1.1.2
Content-Type: text/html
Client-Date: Sun, 25 Feb 2001 18:42:54 GMT
Client-Peer: 127.0.0.1:80
Title: 302 Found

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>302 Found</TITLE>
</HEAD><BODY>
<H1>Found</H1>
The document has moved <A HREF="http://webcab.de/">here</A>.<P>
</BODY></HTML>
```

Odpowiedź serwera w pierwszej linii zawiera kod 302, który informuje przeglądarkę, że plik znajduje się w innej lokalizacji. Pole nagłówka HTTP o nazwie

Location zawiera nową URL wskazujący nową lokalizację pliku. Jeśli przeglądarka wspiera automatyczne przekierowanie, wtedy wyśle ponownie żądanie używając nowego URL. Jeśli nie wspiera automatycznego przekierowania, w odpowiedzi serwera znajduje się prosta strona HTML informująca o potrzebie przekierowania i zawierająca odnośnik z nowym URL. Dopiero, gdy użytkownik kliknie na ten odnośnik, przeglądarka wyśle ponownie żądanie z nowym URL.

Opisana metoda ma tę przewagę nad przekierowaniem DNS, że ruter treści zna adres IP komputera, z którego zażądano dostarczenia treści. Wprowadza jednak opóźnienie związane z wysłaniem do klienta wiadomości o przekierowaniu.

Adresowanie typu anycast

Adresowanie typu *anycast* jest techniką zapewniającą redundancję oraz równoważenie obciążenia dla określonych usług sieciowych w Internecie. Polega na przypisaniu wspólnego adresu IP do wielu węzłów świadczących tą samą usługę, zlokalizowanych w różnych miejscach w sieci. Adres typu *anycast* może być zarówno adresem źródłowym jak i docelowym. Rutery, które wspierają *anycast*, kierują pakiety IP do „najbliższej” lokalizacji świadczącej daną usługę [29]. Używanie *anycast* znacznie upraszcza zadanie znalezienia właściwego serwera. Na przykład nie trzeba analizować listy dostępnych serwerów i wybierać najbliższy, użytkownik podaje nazwę serwera i zostanie połączony z „najbliższym”. Definicja „najbliższej” lokalizacji zależy od topologii sieci oraz używanych protokołów routingu [29]. Na skutek zmian w sieci lub awarii serwera, kolejne pakiety mogą zostać skierowane do innego serwera. Z tego względu adresowania tego typu nie powinno się używać, gdy komunikacja wymaga utrzymywania stanu sesji lub długotrwałej wymiany pakietów.

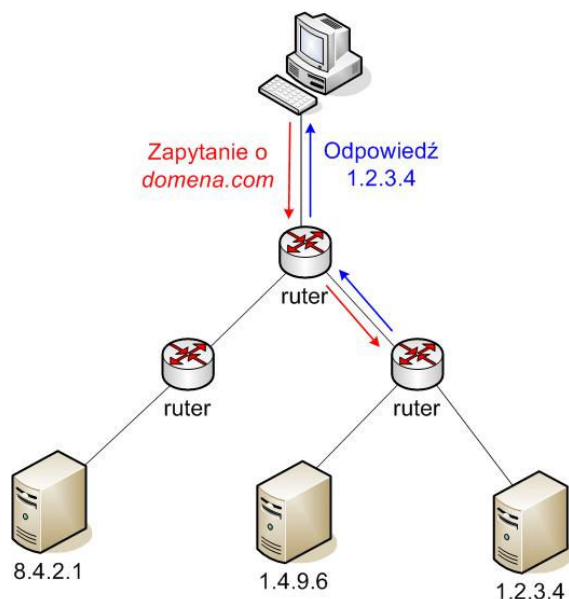
Prawdopodobnie najbardziej popularnym zastosowaniem adresowania typu *anycast* jest usługa DNS [29]. Prawie cała komunikacja pomiędzy klientem usługi DNS, a serwerem DNS odbywa się przy pomocy protokołu UDP. Węzły kierowania treści oparte na usłudze DNS mogą używać adresu *anycast*. W takim wypadku żądanie rozwiązania nazwy domenowej trafi do najbliższego rutera treści, który może ten fakt wykorzystać przy podejmowaniu decyzji [24].

Anycast na podstawie nazwy domenowej

Dla użytkownika nie jest ważne skierowanie go do określonego serwera, czy adresu IP, lecz do miejsca przechowywania konkretnej treści. Tę konkretną treść wyszczególniamy za pomocą nazwy (typowo jest to URL). Naukowcy z Uniwersytetu w Stanford zaproponowali interesujący mechanizm kierowania żądań dostarczenia treści do najlepszego serwera. Na potrzeby swojego projektu stworzyli protokół rozwiązywania nazw INRP (ang. *Internet Name Resolution Protocol*) oraz protokół routingu NBRP (ang. *Name-Based Routing Protocol*) [30].

Kierowanie żądań treści realizowane jest w szkieletowej sieci Internetu (ang. *core of the Internet*), poprzez rozszerzenie funkcji ruterów. Ruter w tym wypadku spełnia dwie funkcje, wspiera tradycyjny routing IP oraz dodatkowo routing na podstawie nazw domenowych. Te funkcje powinny spełniać bramy internetowe oraz rutery brzegowe (rutery poziomu BGP), a nie wszystkie rutery. Każdy ruter utrzymuje dodatkową tablicę routingu nazw, w której dla każdej nazwy domenowej przechowuje informacje o następnym węźle, do którego należy przekazać pakiet. Do utrzymywania tych tablic przeznaczony jest właśnie protokół NBRP. Jest on protokołem typu wektor odległości (ang. *distance-vector*) ze strukturą podobną do protokołu BGP (ang. *Border Gateway Protocol*). Rozgłasza wiadomości, które zawierają nazwę domenową, kolejny ruter na ścieżce w kierunku serwera oraz listę ruterów przez, które treść jest dostępna [30].

Protokół INRP został stworzony, w celu rozwiązywania nazw domenowych. Jest wstecznie kompatybilny z DNS tzn. używa takich samych typów rekordów i formatu pakietów.

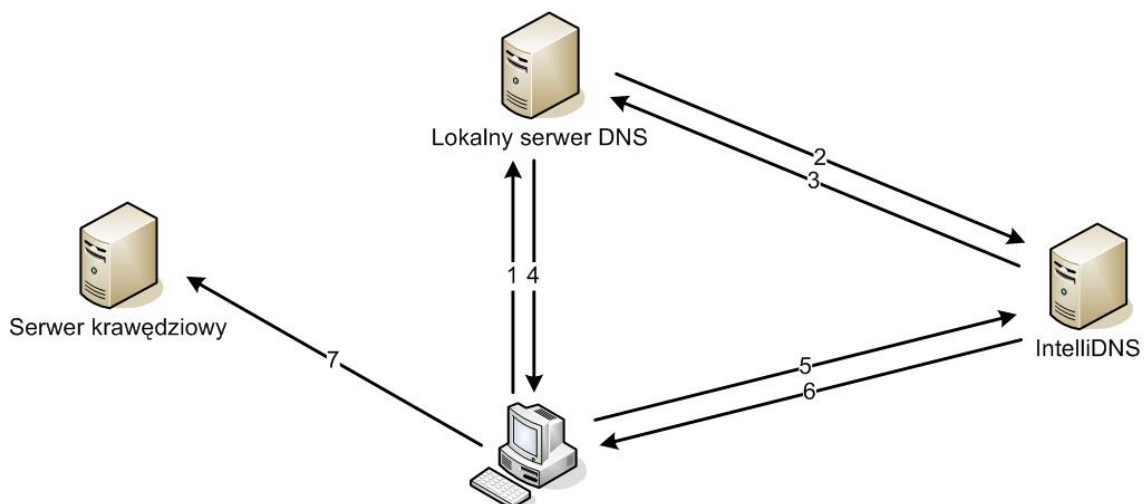


Rys. 8. Protokół INRP [30]

Gdy użytkownik ma potrzebę skontaktować się z serwerem krawędziowym, żądanie INRP zostanie wysłane do najbliższego rutera. Ruter wyszukuje podaną nazwę w tablicy routingu nazw i następnie skieruje żądanie do kolejnego rutera. W ten sposób poprzez kolejne routery żądanie INRP kierowane jest w stronę najbliższego serwera krawędziowego. Gdy dotrze do rutera sąsiadującego z serwerem, ruter ten wysła odpowiedź zawierającą adres IP serwera. Odpowiedź wraca tą samą drogą. Jeśli odpowiedź się nie pojawi, pośrednie routery mogą wybrać alternatywną drogę i powtórzyć wyszukiwanie. W ten sposób INRP zapewnia funkcjonalność podobną do adresowania typu „anycast”, ale na poziomie nazw domenowych.

Łączenie kilku mechanizmów przekierowania

W wielu przypadkach kombinacja kilku opisanych powyżej mechanizmów kierowania żądań może okazać się bardziej korzystna niż używanie tych mechanizmów oddzielnie. Firma Unitech [28] stworzyła oprogramowanie dla rutera treści o nazwie „InlelliDNS”. Mechanizm kierowania żądań użytkowników łączy cechy przekierowania DNS i przekierowania HTTP.



Rys. 9. „IntelliDNS” [28]

Przykład działania „IntelliDNS”:

1. Klient zgłasza do lokalnego serwera DNS żądanie rozwiązania nazwy domenowej.
2. Żądanie przekazane zostaje do rutera treści.
3. „IntelliDNS” jako rozwiązanie nazwy domenowej podaje swój adres IP.
4. Lokalny serwer DNS przekazuje adres IP do klienta.
5. Klient wysyła żądanie HTTP do rutera treści.
6. Następuje przekierowanie HTTP do najlepszego serwera krawędziowego.
7. Klient wysyła żądanie dostarczenia treści.

Dzięki temu, że „IntelliDNS” zwraca swój adres IP, następuje bezpośrednia komunikacja z klientem. Decyzja o przekierowaniu podejmowana jest na podstawie adresu IP klienta, co gwarantuje optymalny wybór serwera krawędziowego.

Współpraca sieci CDN

Operatorzy sieci CDN starają się rozmieścić swoje serwery krawędziowe tak, aby znajdowały się one możliwie najbliżej potencjalnych użytkowników końcowych, co z kolei zapewnia wyższą jakość świadczonych usług. Sprawność działania sieci można zwiększać poprzez dodawanie nowych węzłów, jednak jest to opłacalne pod warunkiem, że z danego węzła korzysta duża liczba użytkowników końcowych. Operatorzy sieci CDN chcąc zwiększyć zasięg swoich sieci mogą porozumieć się między sobą i udostępnić swoje serwery krawędziowe innej sieci. Kooperacja sieci

CDN określana jest w literaturze terminem *Content Distribution Internetworking* (CDI) lub *Content Peering*. Aby taka współpraca mogła być możliwa niezbędna jest realizacja trzech elementów [53]:

- System wzajemnej dystrybucji (ang. *Distribution Internetworking System*) zajmuje się dystrybucją treści pomiędzy lokalne systemy dystrybucji i zarządzania treścią.
- System wzajemnego kierowania żądań (ang. *Request-Routing Internetworking System*) kieruje użytkowników do najlepszej lokalizacji. W tym celu systemy poszczególnych sieci CDN wymieniają się niezbędnymi informacjami do podejmowania decyzji.
- System monitorowania i kontroli (ang. *Accounting Internetworking System*) zbiera informacje i przekazuje je do lokalnych systemów monitorowania i kontroli. Statystyki potrzebne są do rozliczeń pomiędzy operatorami współpracujących sieci CDN.

Sieć CDN widziana jest jako „czarna skrzynka” (ang. *black box*) z perspektywy pozostałych kooperujących z nią sieci. Każda z tych sieci posiada bramę CIG (ang. *Content Internetworking Gateway*), przez którą może komunikować się z innymi sieciami. Brama CIG utrzymuje drzewo dystrybucji i routingu żądań. Żądanie dostarczenia treści kierowane jest najpierw do autorytatywnego CIG dla danej treści. System wzajemnego kierowania żądań podejmuje decyzję, do której sieci (do którego CIG) skierować żądanie, aby mogło być najlepiej obsłużone. Jeśli stwierdzi, że najlepszy serwer znajduje się w jego sieci CDN, skieruje żądanie do lokalnego systemu kierowania żądań.

Zostały już poczynione prace w celu realizacji powyżej wymienionych elementów architektury CDI. CNAP (ang. *Content Network Advertisement Protocol*) jest protokołem umożliwiającym wymianę informacji potrzebnych w procesie kierowania żądań [54]. Jest to protokół typu punkt-punkt, uruchamianym na bramie CIG.

Ciekawe rozwiązanie o nazwie „Intelligent DNS” (IDNS) [55] zaproponowała firma AT&T. Realizuje ono kierowanie żądań klientów do właściwej sieci CDN, opierając swoje działanie na systemie DNS. Lokalny serwer DNS przesyła żądanie rozwiązania nazwy domenowej do autorytatywnego serwera DNS (tzw. brokera, przedstawiciela kilku sieci DNS) dla tej domeny. Serwer ten decyduje, która sieć CDN jest najlepsza do

obsługi żądania. Następnie przekierowuje żądanie zwracając rekordy typu CNAME lub NS, a jeśli wybrana sieć CDN jest jego własną siecią, zwraca rekordy typu A z adresem IP wybranego serwera krawędziowego.

Interesujące rozważania na temat współpracy sieci CDN można znaleźć w pracach [56][57].

Wybór najlepszego serwera

Systemy kierowania treści używają różnorodnych metryk do określenia najlepszego serwera, który zrealizuje żądanie użytkownika. Metryki opierają się na mierzalnych parametrach sieci oraz do ich wyznaczenia potrzebna jest współpraca serwerów krawędziowych.

Z metrykami związane jest pojęcie „odległości” dwóch punktów w sieci. Możemy je wyznaczyć biorąc pod uwagę:

- Opóźnienie – serwer krawędziowy wysyła, do komputera zgłaszającego żądanie, pakiet, na który oczekuje odpowiedzi. Mierzy się czas opóźnienia odpowiedzi.
- Liczba skoków – ilość ruterów na drodze pakietów pomiędzy serwerem krawędziowym, a komputerem zgłaszającym żądanie.
- Informacje BGP – atrybuty PATH i MED mogą posłużyć do wyznaczenia „odległości” [24][45].

Najbardziej miarodajnym i jednocześnie najczęściej stosowanym sposobem na wyznaczenie „odległości” w sieci jest pomiar czasu opóźnienia odpowiedzi. Zależy on bezpośrednio od możliwości sprzętowych serwera, aktualnego jego obciążenia oraz od obciążenia sieci. Technika ta jest względnie dokładna. „Odległość” serwera krawędziowego od lokalnego serwera DNS (lub bezpośrednio od klienta) można wyznaczyć używając następujących metod [25]:

- ICMP Echo – serwer krawędziowy wysyła pakiet ICMP typu *echo* [62] do serwera DNS, który powinien odpowiedzieć pakietem ICMP typu *echo reply*. Mierzony jest czas opóźnienia odpowiedzi.
- DNS RRT (ang. *request-response time*) – mierzony jest czas jaki zajmuje odwrotne zapytanie DNS (ang. *inverse query*) [22], do lokalnego serwera DNS.

- Traceroute – idea jego działania jest następująca: z każdym wysłanym pakietem związany jest licznik określający „czas życia” (ang. *time to live*). Licznik ten jest zmniejszany o jeden podczas przejścia przez każdy ruter. Gdy osiągnie on zero, ruter zwraca do nadawcy pakiet ICMP typu *time exceeded in-transit* [62]. Traceroute działa w ten sposób, że wysyła pakiet do wskazanego adresu z „czasem życia” równym 1, następnie z czasem życia równym 2, i tak dalej, aż osiągnie docelowy host lub limit. Pozwala to prześledzić trasę od nadawcy do odbiorcy. Serwer krawędziowy bada drogę jaką podążają pakiety do serwera DNS i zapamiętuje czas opóźnienia odpowiedzi od najodleglejszego rutera na trasie. Jeśli przynajmniej trzy serwery krawędziowe stwierdzą, że najodleglejszym ruterem jest ten sam ruter, brana jest pod uwagę „odległość” do tego rutera [25].

Metoda pierwsza jest najczęściej stosowana, pozostałe są używane tylko, gdy ta metoda zawiedzie. Może się tak zdarzyć, ponieważ zapory ogniowe i filtry pakietów mogą odrzucić pakiety ICMP, uniemożliwiając wykonanie próby wyznaczenia „odległości”.

Pomiar „odległości” może być również wykonywany tylko w jedną stronę, należy pamiętać, że droga pakietów w Internecie może być nie symetryczna.

Proces określenia „odległości” inicjuje ruter treści, który zgłasza żądanie wykonania pomiaru do serwera krawędziowego, a następnie odbiera od niego wynik.

Serwer krawędziowy może również sam monitorować swoją wydajność i wysyłać informacje do ruterów treści (np. korzystając z *multicastu*) w przypadku zaobserwowania znaczącej zmiany jakiegoś parametru [31]. Metoda ta jest skalowalna i zapewnia dokładny pomiar wydajności samego serwera, wymaga jednak modyfikacji serwera i nie uwzględnia stanu sieci.

Do monitorowania wydajności serwera można wykorzystać oddzielną aplikację tzw. sondę, współpracującą z ruterami treści. Sonda w imieniu dużej liczby klientów (ruterów treści) może wykonywać okresowe zapytania do serwera, w celu ustalenia aktualnych parametrów określających jego wydajność [31]. W tej metodzie możemy badać również parametry sieci.

Wybór najlepszego serwera na przykładzie urządzeń Cisco

Warto przyjrzeć się jak urządzenia firmy Cisco [23] realizują wybór najlepszego serwera. Rutery treści podejmują decyzje na podstawie bazy danych, która zawiera informacje o „odległości” serwera krawędziowego od konkretnego lokalnego serwera DNS. Informacje te zebrane są w tabelach. Każda tabela przechowuje informacje dotyczące grupy serwerów DNS, co ma na celu zmniejszenie liczby tabel. Serwery, dla których pierwsze 24 bity adresu IP są takie same, zaliczane są do jednej grupy.

Również serwery krawędziowe, które znajdują się w geograficznej bliskości np. w tym samym punkcie prezentacji, traktowane są jako jedna grupa, zwana dalej lokalizacją. Zmniejsza to rozmiar tabel. „Cisco Internet CDN Software” pozwala na maksymalnie 192 lokalizacje. Serwery krawędziowe są przypisywane do lokalizacji w momencie dodawania ich do sieci CDN.

Ruter treści co pewien czas kieruje do serwera krawędziowego polecenie określenia „odległości” od danego lokalnego serwera DNS. Tylko jeden serwer z lokalizacji jest uprawniony do wykonywania prób. Ten serwer wysyła do wskazanego serwera DNS wiadomość ICMP *echo request*. Jeśli serwer nie odpowie, wysyła zapytanie DNS. „Odległość” wyznacza czas odpowiedzi. Wartość ta jest zwracana ruterowi treści, który dzięki tej informacji uaktualnia tabele.

Zbieranie informacji odbywa się okresowo, w turach co około 2 minuty. Serwery, uprawnione do wykonywania prób, zwracają rezultaty do wszystkich ruterów treści, a nie tylko do tego, który zlecił wykonanie próby. Serwer krawędziowy podczas komunikacji z ruterem treści zwraca rezultaty wszystkich prób, a w odpowiedzi otrzymuje informacje potrzebne do próby w następnej turze.

Oprócz tego w każdej turze wszystkie serwery krawędziowe komunikują się z każdym ruterem treści, aby podać mu listę domen, których treść obsługują. Ma to na celu potwierdzenie, że są zdolne do obsługi żądań klientów.

Komunikacja pomiędzy serwerami krawędziowymi, a ruterami treści nie jest zsynchronizowana i może się zdarzyć, że tabele na różnych ruterach treści będą zawierały rozbieżne informacje. Dodatkowo komunikacja ta jest rozłożona jest równomiernie na cały okres tury, co zapobiega chwilowemu przeciążeniu rutera.

5. Lokalne kierowanie treści

W punkcie prezentacji zamiast jednego serwera, który może nie radzić sobie z natłokiem żądań klientów oraz stanowi pojedynczy punkt awarii, instaluje się kilka serwerów zdolnych dostarczać tę samą treść, taką strukturę nazywa się często farmą serwerów. Lokalne kierowanie treści realizowane jest przez urządzenie nazywane przełącznikiem treści. Przełącznik treści instaluje się przed farmą serwerów. Jego zadaniem jest kierowanie żądań klientów do odpowiedniego serwera krawędziowego oraz równoważenie obciążenia pomiędzy równorzędnymi serwerami. Często określane jest również jako przełącznik warstw 4-7, ponieważ decyzję o wyborze serwera podejmuje na podstawie informacji z warstw od 4 do 7 modelu sieciowego OSI. Jego działanie jest niewidoczne dla użytkowników, którzy znają tzw. wirtualny adres IP przełącznika (VIP) ogłaszany przez DNS [33]. Adresy IP serwerów zna tylko przełącznik. Używanie przełącznika treści pozwala łatwo dodawać nowe serwery, wystarczy tylko uaktualnić konfigurację przełącznika dla nowego serwera. Duża skalowalność jest niewątpliwą zaletą tego rozwiązania.

Równoważenie obciążenia na podstawie warstwy transportowej

Przełącznik treści podejmuje decyzję o przekierowaniu dla każdego nowego połączenia TCP lub strumienia UDP. Należy zrozumieć fakt, że po podjęciu tej decyzji, wszystkie kolejne pakiety należące jednej sesji muszą zostać przekazane do tego samego serwera. Aby to osiągnąć przełącznik treści dysponuje tablicą sesji. Oto jak może wyglądać przykładowa obsługa sesji protokołu transportowego [32]:

1. Pierwszy pakiet przybywa do przełącznika treści. W protokole TCP będzie to pakiet z flagą SYN.
2. Połączenie TCP w sieci IP można jednoznacznie zidentyfikować poprzez źródłowy i docelowy adres IP oraz źródłowy i docelowy port TCP. Przełącznik przeszukuje tablicę sesji w poszukiwaniu wpisu pasującego do tego pakietu.
3. Jeśli takie połączenie nie istnieje w tablicy sesji, stosowana jest metryka równoważenia obciążenia i następuje wybór serwera. Przełącznik dokonuje odpowiedniej translacji adresów i wysyła pakiet do serwera, a do tablicy sesji dodawany jest nowy wpis.

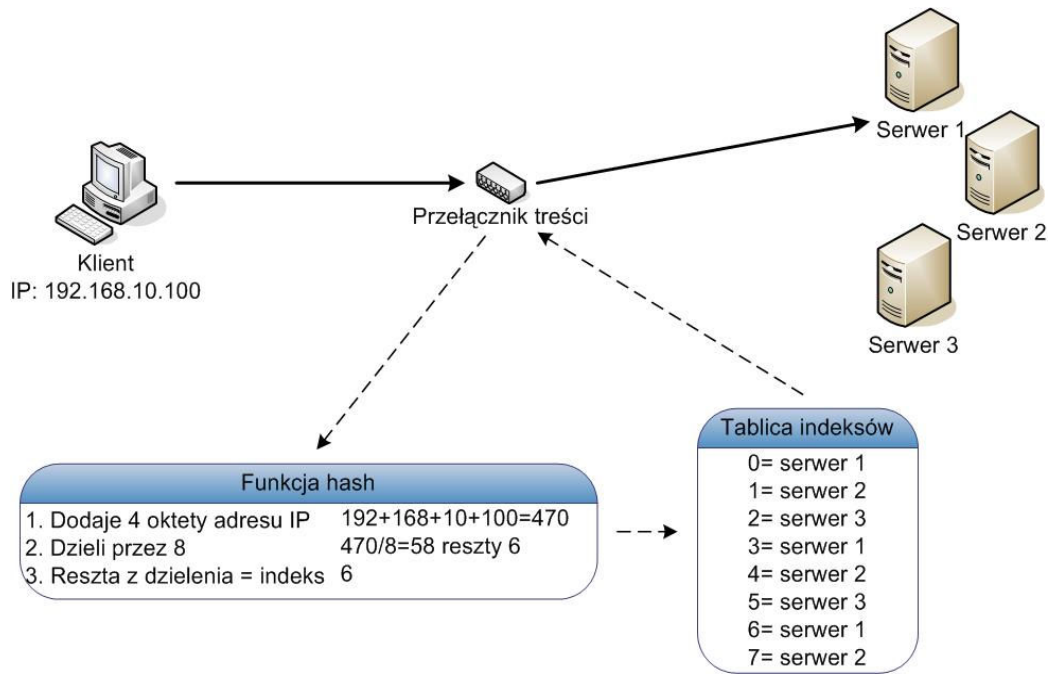
4. Gdy pakiet powrotny (z flagami SYN, ACK) od serwera do klienta pojawi się na porcie przełącznika, zachodzi odwrotna translacja adresów na podstawie informacji z tablicy sesji i wysyłanie pakietu do klienta.
5. Gdy trzeci pakiet (z flagą ACK) od klienta do serwera pojawi się na porcie przełącznika, odszukiwany jest pasujący wpis w tablicy sesji. Przełącznik dokonuje odpowiedniej translacji adresów i wysyła pakiet do serwera.

Przełącznik może na podstawie numeru portu docelowego rozpoznawać żadaną usługę (HTTP, FTP) i na tej podstawie kierować ruch do serwera, który ją świadczy.

Do realizacji równoważenia obciążenia stosuje się następujące metryki warstwy 4 [32]:

- Najmniej połączeń – jest to bardzo prosta metryka i często stosowana jako domyślana. Jak sugeruje nazwa, nowa sesja TCP lub UDP zostanie przekierowana do serwera, który aktualnie utrzymuje najmniejszą liczbę sesji. Zaletą wynikającą ze stosowania tej metody jest to, że na liczbę sesji, obsłużonych w określonym czasie przez dany serwer, ma wpływ wydajność tego serwera. Serwer posiadający szybszy procesor, więcej pamięci i szybszą kartę sieciową będzie w stanie poradzić sobie w większą liczbą sesji na sekundę. Wadą jest brak skojarzenia kolejnych połączeń (ang. *persistence*) pomiędzy klientem i serwerem. Na przykład, jeśli klient pobiera stronę WWW używając przeglądarki, która korzysta z protokołu HTTP w wersji 1.0, kolejne żądania GET obsłużone mogą być przez różne serwery.
- *Round Robin* – również bardzo prosta metoda dystrybucji obciążenia, rozdzielająca je równomiernie pomiędzy serwery. Nadchodzące sesje kierowane są kolejno: pierwsza do pierwszego serwera, druga do drugiego i tak dalej. Każdy serwer będzie musiał obsłużyć taką samą liczbę sesji w określonym czasie niezależnie od swojej wydajności sprzętowej. Dlatego powinno się używać tej metody, gdy dysponujemy serwerami o bardzo zbliżonej wydajności sprzętowej, w przeciwnym wypadku metoda staje się mniej efektywna. Tak jak w poprzedniej opisanym metodzie, również ta nie pozwala na skojarzenia kolejnych połączeń pomiędzy klientem i serwerem.
- *Hash z adresu IP* – ta metoda wprowadza możliwość skojarzenia kolejnych połączeń pomiędzy klientem i serwerem. Termin *hash* określa deterministyczną funkcję, która na podstawie adresu IP związanego z daną sesją, wyznaczy serwer

do obsługi żądania. Funkcja *hash* jako argument może przyjmować dowolną kombinację z następujących elementów: adres IP źródłowy i docelowy oraz port źródłowy i docelowy. Przykład koncepcji działania tego mechanizmu przedstawia Rys. 10:



Rys. 10. *Hash* z adresu IP [32]

Dostępne serwery reprezentowane są przez indeks w tabeli, w tym wypadku mamy 8 indeksów. Każdy indeks odwzorowany jest na jeden serwer. Indeks wyznaczmy używając numerycznych przekształceń, a zatem indeks w tabeli bezpośrednio zależy od adresu IP i na tej podstawie przełącznik treści może jednoznacznie przypisać serwer do przychodzącego połączenia. Również kolejne połączenia przychodzące z tego samego adresu IP będą wskazywały na ten sam indeks, a zatem będą kierowane do tego samego serwera.

- Mertyki ważone – większość przełączników treści umożliwia uwzględnienie w powyższych algorytmach wagi dla poszczególnych serwerów. Umożliwia to lepsze wykorzystanie zasobów sprzętowych. Na przykład w metodzie *Round Robin* do serwera o lepszej wydajności sprzętowej można skierować do obsługi dwa razy więcej sesji.
- Maksymalna liczba połączeń – jest to kolejne udogodnienie, które można zastosować do metryk warstwy 4. Dla każdego serwera można zdefiniować

maksymalną liczbę ustanowionych sesji. Taki mechanizm zabezpiecza przed przeciążeniem serwera lub grupy serwerów. Jeśli zostanie ustanowiona maksymalna liczba połączeń, nowe połączenia zostaną skierowane na podstawie używanej metryki do innego serwera w grupie. Jeśli stosowany jest mechanizm kojarzenia kolejnych połączeń pomiędzy klientem i serwerem, to w przypadku wyczerpania się limitu połączeń, przestanie on spełniać swoje zadanie. W przypadku, gdy ustalone limity połączeń są niewystarczające trzeba niestety dodać nowy serwer, aby zwiększyć zasoby grupy serwerów. W takim wypadku stosuje się również dodatkowy serwer generujący informacje o tym, że serwer jest zajęty lub przeciążony i nie jest możliwe zrealizowanie żądania klienta. Jednak z punktu widzenia sieci CDN taka sytuacja nie może mieć miejsca, zasoby muszą być wystarczające lub żądanie klienta musi zostać przekierowane do innej lokalizacji zdolnej je zrealizować.

- Czas opóźnienia odpowiedzi – powyższe metody nie uwzględniają faktu, że obciążenie serwera jest zmienne, a wraz ze zmianą obciążenia zmienia się jego wydajność. Dlatego badanie czasu opóźnienia odpowiedzi odzwierciedla aktualną zdolność serwera do obsługi nadchodzących żądań. Na tej podstawie można na bieżąco uaktualniać wagi dla poszczególnych serwerów. W niektórych implementacjach przełącznika treści badanie stanu wydajności serwera przeprowadza aplikacja, której sondy umieszczone na każdym serwerze przekazują informacje o użyciu procesora, zajętości dysku twardego i pamięci do przełącznika.
- Przepustowość – ilość pasma wykorzystywanego przez dany serwer może mieć wpływ na wydajność obsługi żądań klientów. Dlatego monitorowanie zajętości pasma jest często stosowaną techniką. Zebrane wartości mogą posłużyć do podejmowania decyzji o dystrybucji obciążenia i ustaleniu wag dla poszczególnych serwerów.

Przełączanie w warstwie 4 okazuje się stosunkowo proste w realizacji, gdyż informacja o adresie IP znajduje się zawsze w tym samym miejscu pakietu, odczytywanie tych informacji realizują sprzętowo specjalnie do tego celu zaprojektowane układy scalone [33].

Przełączanie na podstawie warstwy 7

W wielu przypadkach przełączanie na podstawie warstwy 4 nie wystarcza, potrzebne są bardziej złożone metody kierowania żądań do właściwego serwera, dające bardziej szczegółowe informacje o przychodzącym żądaniu klienta. Mowa tutaj o przełączaniu na podstawie warstwy 7, opartym na informacji o używanej aplikacji, a nie tylko na adresowaniu. Różne rodzaje treści wymagają zróżnicowanych parametrów serwera (użycie procesora, przepustowość). Daje to możliwość grupowania serwerów ze względu na rodzaj serwowanej treści. Serwery przeprowadzające transakcje i komunikujące się z bazą danych będą potrzebowały większej mocy obliczeniowej procesora niż serwery tylko dostarczające statyczne strony, ale te za to mogą wymagać większej przestrzeni dyskowej do przechowywania treści. Przełącznik treści może również kierować wyróżnioną grupę uprzywilejowanych użytkowników do bardziej wydajnych serwerów niż pozostałych użytkowników [34]. Wszystkie te zabiegi mają na celu podniesienie standardu świadczonej usługi.

Dwa połączenia

Przełączanie na podstawie warstwy 4 następuje już po otrzymaniu pierwszego pakietu od klienta. Aby przełącznik treści mógł odczytać informacje z warstwy aplikacji musi dojść do nawiązania pełnego połączenia TCP, dopiero czwarty pakiet od klienta zawiera nagłówek protokołu używanego w warstwie aplikacji w przykładzie będzie to protokół HTTP. Cały proces przebiega następująco:

1. Klient nawiązuje połączenie TCP z przełącznikiem treści. W protokole TCP zestawienie połączenia przebiega w trzech etapach: klient wysyła pakiet z ustawioną flagą SYN, przełącznik odpowiada pakietem z ustawionymi flagami SYN i ACK, klient wysyła pakiet z ustawioną flagą ACK.
2. W czwartym pakiecie przychodzą dane warstwy aplikacji. Przełącznik buforuje pakiet. W protokole HTTP może się zdarzyć, że pojedyncze żądanie HTTP (dane warstwy aplikacji) podzielone zostanie na kilka pakietów, a więc do podjęcia decyzji o przełączeniu nie wystarczy inwigilacja pierwszego pakietu.
3. Przełącznik przeszukuje pakiet pod kątem wymaganych informacji warstwy 7 zawartych w nagłówku protokołu HTTP.
4. Na podstawie znalezionych informacji przełącznik podejmuje decyzję o wyborze właściwego serwera, który najlepiej obsłuży to żądanie.

5. Przełącznik nawiązuje połączenie TCP z wybranym serwerem (komunikacja w trzech etapach jak w punkcie 1)
6. Przekazuje zbuforowane pakiety z żądaniem HTTP do serwera.
7. Przełącznik musi „złączyć” te dwa oddzielne połączenia, we wszystkich kolejnych pakietach musi zmieniać takie informacje jak numery portów, numery sekwencyjne, adresy IP w ten sposób, aby klient myślał, że komunikuje się bezpośrednio z serwerem, a serwer, że komunikuje się bezpośrednio z klientem. Klient i serwer nie mają świadomości istnienia przełącznika.

HTTP 1.0 i HTTP 1.1

Protokół HTTP obsługuje dwa rodzaje połączenia między klientem i serwerem. Pierwszy rodzaj pozwala klientowi podczas jednego połączenia TCP na pobranie jednego obiektu. Strona WWW składa się zazwyczaj z wielu obiektów i do pobrania całości strony konieczne jest nawiązanie oddzielnego połączenia TCP do pobrania każdego z obiektów. Przy dużej liczbie obiektów przypadających na jedną stronę WWW metoda ta staje się mało wydajna, pożera za dużo zasobów systemu operacyjnego.

Drugi rodzaj połączenia to połączenie stałe, podczas jednego połączenia TCP pozwala pobrać wiele obiektów strony.

Nagłówek HTTP zawiera pole o nazwie `Connection`. Pole to określa typ połączenia HTTP pomiędzy klientem i serwerem. Gdy pole to przyjmuje wartość `close` połączenie TCP jest zamykane po pobraniu jednego obiektu strony. Wersja 1.0 protokołu HTTP, domyślnie używa tego typu połączenia. Wersja 1.1 protokołu HTTP domyślnie stosuje stałe połączenia. Wersje wcześniejsze od 1.1 używają stałego połączenia ustawiając wartość `Keep-Alive` w polu `Connection` nagłówka HTTP [59].

Przełącznik treści powinien być przygotowany do obsługi stałych połączeń. Przy tym rodzaju połączenia przełącznik inwigiluje wszystkie przychodzące pakiety od klienta, gdyż mogą one zawierać kolejne żądania. Jeśli na przykład zdarzy się sytuacja, że w jednym połączeniu TCP od klienta otrzyma dwa żądania HTTP GET [59], z których każde powinien obsłużyć inny serwer, przełącznik treści dla każdego z tych żądań musi nawiązać oddzielne połączenie TCP do właściwego serwera. Po zrealizowaniu pierwszego żądania (punkty 1-7 opisane powyżej), przychodzi drugie żądanie HTTP GET, przełącznik zamyka połączenie TCP z serwerem, a następnie

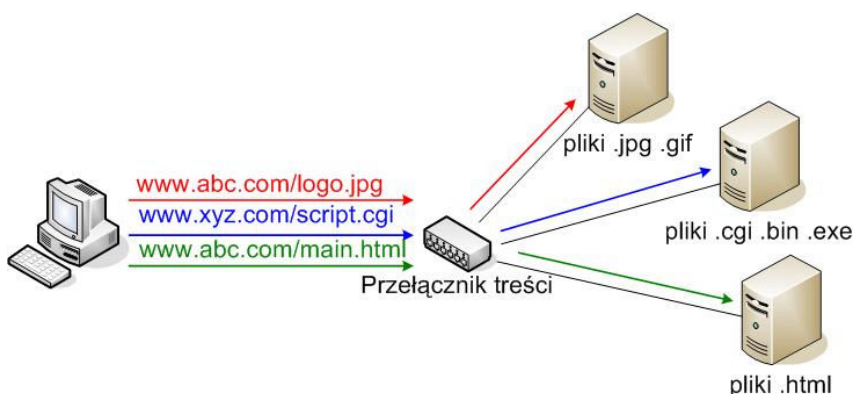
otwiera drugie połączenie TCP do serwera, który jest właściwym do obsługi drugiego żądania i ponownie musi „złączyć” połączenia do klienta i do serwera. Dla klienta zmiana serwera pozostaje niezauważona.

Dystrybucja obciążenia na podstawie wiadomości HTTP

Dystrybucję obciążenia zrealizować można analizując przesyłane od klienta do serwera wiadomości protokołu HTTP. Przykładowy nagłówek wiadomości HTTP wygląda następująco:

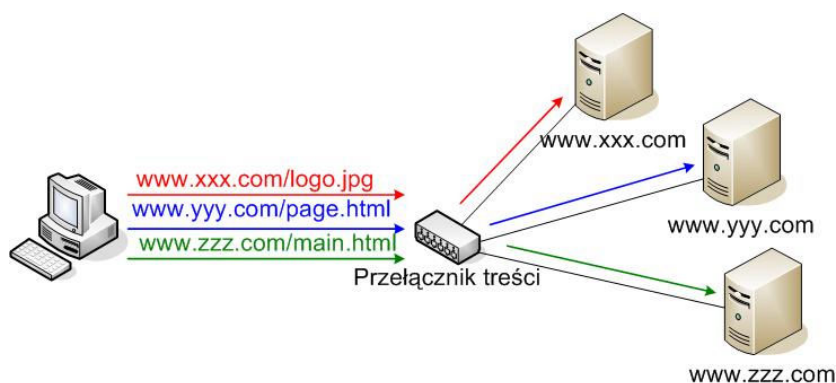
```
GET /podstrony/index1.html HTTP/1.0\r\n
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg\r\n
Accept-Language: en-gb\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT
5.0)\r\n
Host: www.portal.com\r\n
Connection: Keep-Alive\r\n
\r\n
```

Powszechnie w przełącznikach treści stosowane jest przekierowanie na podstawie URL. W pierwszej linii żądania HTTP znajdują się kolejno oddzielane spacją: metoda stosowana do zasobu (w powyższym przykładzie GET), identyfikator zasobu oraz używana wersja protokołu [59]. Analizując identyfikator zasobu możemy określić rodzaj treści np. pliki graficzne, html, skrypty wykonywalne. Każdy rodzaj treści obsługuje oddzielny serwer lub grupa serwerów, ilustruje to Rys. 11:



Rys. 11. Przełączanie na podstawie rodzaju treści

W żądaniu HTTP część domenowa adresu URL jest umieszczona w polu `Host`. Uwzględniając informacje z pola `Host` przełącznik kieruje żądania na podstawie nazw domenowych. Dzięki temu jeden adres IP (adres VIP przełącznika treści) może reprezentować wiele nazw domenowych. Rys. 12 przedstawia opisaną sytuację:



Rys. 12. Przełączanie na podstawie pola `Host`

Różne przeglądarki internetowe mogą wymagać innego formatu strony, aby ją poprawnie wyświetlić. Pole `User-Agent` informuje jaki jest typ przeglądarki oraz system operacyjny klienta. Przełącznik skieruje żądanie do serwera, który będzie mógł dostarczyć odpowiedni format treści. Jest to szczególnie przydatne, gdy chcemy, aby treść była dostępna dla użytkowników urządzeń przenośnych, telefonów komórkowych z przeglądarką WAP i palmtopów.

Pole `Accept-Language` pozwala kierować żądania klientów na podstawie preferowanego języka odpowiedzi. Jeśli ta sama strona dostępna jest w kilku językach, przełącznik sam skieruje klienta do właściwej wersji językowej.

Protokół HTTP posiada mechanizm pozwalający na przechowywanie po stronie klienta pewnej unikalnej informacji, a dostępnej zawsze podczas komunikacji klienta z serwerem, mowa tutaj o tzw. ciasteczkach (ang. *cookies*). Ciasteczka mogą być ulotne przechowywane w pamięci tylko w czasie sesji z serwerem lub stałe, zapisywane na twardym dysku komputera klienta w celu późniejszego użycia podczas kolejnych sesji. Ciasteczko posiada nazwę (identyfikator) oraz wartość. Stałe posiadają również datę oraz czas ważności. Ciasteczko może być przekazywane jako pole nagłówka HTTP lub połączone z URL. Zapisywane jest na dysku klienta podczas pierwszej komunikacji z

serwerem. Korzystanie z tego mechanizmu wymaga jednak odpowiedniej konfiguracji serwera oraz upewnienia się, że przeglądarka obsługuje ciasteczka.

Ciasteczka często wykorzystywane są do identyfikacji statusu użytkownika np. użytkownicy zwykli i uprzywilejowani. Status użytkownika przekazywany jest jako wartość ciasteczka. Przełącznik treści na podstawie wartości zapisanej w ciasteczku może podejmować decyzję o wyborze serwera. Na przykład użytkownicy uprzywilejowani kierowani mogą być do wydajniejszego serwera [32].

Powyżej przedstawione zostały możliwości przełączania treści na podstawie informacji znajdujących się w wiadomości protokołu HTTP. Jest on tylko jednym z wielu protokołów warstwy aplikacji. W książce [32] przedstawiono przykłady dystrybucji obciążenia dla protokołów FTP[63], DNS[22] i RTSP[64].

Kojarzenie kolejnych połączeń

Informacje zapisane w ciasteczkach mogą zostać wykorzystane przez przełącznik treści do kojarzenia kolejnych połączeń między klientem i serwerem. Ciasteczka tymczasowe przechowywane są w pamięci do momentu zamknięcia przeglądarki. Wartości ciasteczek dla każdej sesji mogą być inne. W momencie, gdy serwer ustanawia nowe ciasteczko przełącznik treści zapamiętuje dane charakterystyczne ciasteczka oraz serwer, aby kolejne żądania kierować właśnie do tego serwera.

Z faktu, że przełącznik treści pośredniczy w komunikacji pomiędzy klientem i serwerem wynika ciekawe udogodnienie. Przełącznik treści może sam ustanowić ciasteczko w momencie, gdy pierwszy pakiet od serwera przechodzi przez przełącznik. Wszystkie kolejne żądania od klienta będą zawierały to ciasteczko, rozpoznając je przełącznik skieruje żądanie do tego samego serwera. Przed tym jednak usuwa ciasteczko z żądania HTTP. W ten sposób serwer nie jest świadomy istnienia ciasteczka, nie są potrzebne zmiany w jego konfiguracji [32].

Aplikacje *e-commerce* używają szyfrowanych połączeń. Podczas połączenia do serwera z użyciem szyfrowania SSL [70], sesji SSL nadawany jest unikalny identyfikator (ID). Identyfikator ten może zostać wykorzystany do kojarzenia kolejnych połączeń sesji SSL i kierowania ich do tego samego serwera [35]. Gdy śledzenie ID sesji SSL jest aktywne przełącznik śledzi kolejne pakiety nowego połączenia TCP, aby ustalić czy sesja SSL już istnieje czy jest nowa. Jeśli jest nowa przypisuje ją do serwera wybranego na podstawie używanej metryki warstwy 4, jeśli pakiety są związane z istniejącą sesją SSL przekazywane są do serwera przypisanego do tej sesji.

Monitorowanie stanu serwera w czasie rzeczywistym

Przełącznik treści musi mieć pewność, że kieruje żądania do sprawnych serwerów. Może to zbadać używając następujących sposobów [35]:

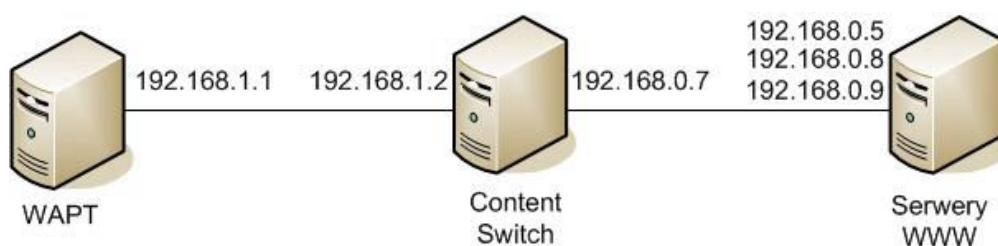
- Monitorowanie fizycznego połączenia – przełącznik monitoruje stan fizycznego połączenia na interfejsach sieciowych, do których podłączone są serwery. Jeśli połączenie zostanie utracone, przełącznik natychmiast nadaje temu serwerowi status „serwer zawiódł” i zaprzestaje kierować do niego żądania.
- Monitorowanie ARP (ang. *Address Resolution Protocol*) [60] – przełącznik co pewien ustalony okres czasu wysyła żądanie ARP dla adresu IP serwera [32]. Jeśli serwer nie odpowie otrzymuje natychmiast status „serwer zawiódł”.
- Monitorowanie ICMP – przełącznik co pewien ustalony okres czasu wysyła „ping” do każdego z serwerów. Brak odpowiedzi oznacza, że serwer zawiódł.
- Monitorowanie połączenia TCP – przełącznik próbuje nawiązać połączenie TCP (pakiet TCP SYN) do każdej aplikacji do każdego serwera i ustala czy serwer odpowiedział. W ten sposób przełącznik jest w stanie określić czy zawiódł serwer czy tylko konkretna aplikacja na serwerze.
- Aktywna weryfikacja treści – przełącznik w pewnych odstępach czasu wysyła do każdego serwera żądanie HTTP dostarczenia ustalonego, testowego obiektu. Odpowiedź serwera jest następnie weryfikowana. Pozwala to określić czy usługa HTTP działa prawidłowo [35]. Tą metodą można testować również usługi NNTP[65], FTP[63], SMTP[66], POP3[67], IMAP[68], DNS[22], RADIUS[69].
- Weryfikacja dynamicznych aplikacji – rozszerzenie powyższej metody. Przełącznik jest w stanie ustalić poprawność wykonywania skryptów CGI (ang. *Common Gateway Interface*), aplikacji ASP (ang. *Active Server Pages*) itp. na serwerach WWW.

6. Testy programu „Content Switch”

Środowisko testowe

Do testowania programu użyto narzędzia „WAPT 3.0” [77]. Umożliwia ono generowanie żądań HTTP w celu zbadania wydajności aplikacji WWW. Udostępnia różnorodne opcje umożliwiające skonstruowanie odpowiedniego scenariusza testu.

Podczas testów wykorzystano trzy komputery połączone siecią Ethernet 100Mb. Na jednym z komputerów zainstalowano program „Virtual PC” [78] pozwalający na uruchomienie na jednym fizycznym komputerze wielu wirtualnych komputerów. W przypadku, gdy do testu potrzebny jest więcej niż jeden serwer WWW, dodatkowy serwer WWW zostaje uruchomiany na wirtualnym komputerze. Wirtualne komputery posiadają adresy IP, przez co są jednoznacznie rozpoznawane w sieci. Rys. 13 przedstawia topologię sieci użytej do testów z wyróżnieniem adresów IP.



Rys.13. Topologia sieci używanej podczas testów

Program „WAPT” został zainstalowany na komputerze z procesorem Athlon 2,43 GHz i 1GB pamięci operacyjnej, program „Content Switch” na komputerze z procesorem Duron 700 MHz i 128MB pamięci operacyjnej, a serwery WWW na komputerze z procesorem Athlon 2 GHz i 640MB pamięci operacyjnej.

Test 1

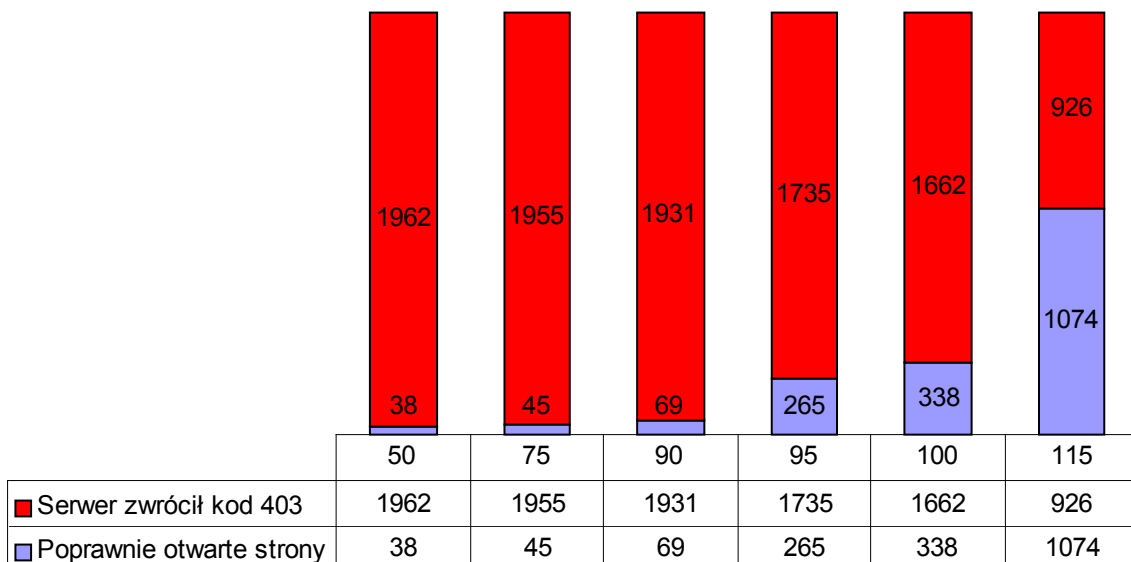
W celu zademonstrowania działania programu wygenerowany został plik logów. Program „WAPT” wygenerował 30 żądań HTTP w odstępach 5 sekund, każde w oddzielnym połączeniu TCP. W czasie trwania testu jeden z serwerów WWW został wyłączony, symulując w ten sposób awarię, a następnie po chwili ponownie uruchomiony.

Plik logów umieszczony został w katalogu `test1` na dołączonej do pracy płycie CD. Analiza tego pliku potwierdza prawidłowe zachowanie się programu. Doskonale widać zasadę *Round Robin* stosowaną przy wyborze serwera oraz awarię serwera WWW, a następnie przywrócenie go do pracy. Od momentu awarii żądania HTTP nie były kierowane do uszkodzonego serwera.

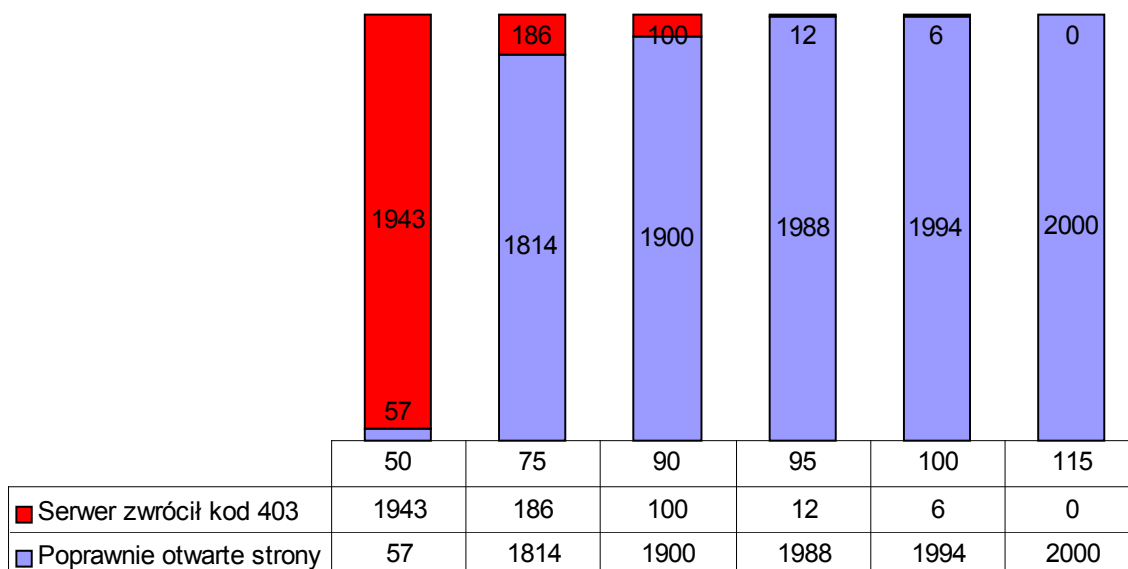
Test 2

Serwer IIS 5.1 dołączony do systemu operacyjnego „Windows XP” posiada ograniczenie pozwalające obsługiwać tylko 10 połączeń jednocześnie. Jeśli liczba ta zostanie przekroczona, serwer zwraca w odpowiedzi kod 403.

W doświadczeniu założono, że przeciążenie serwera objawia się właśnie zwracaniem kodu 403 zamiast realizacją żądania HTTP. Program „WAPT” nawiązuje 1000 połączeń TCP do serwera WWW. W każdym połączeniu zrealizowane zostaną 2 żądania HTTP dostarczenia strony o rozmiarze 7kB. Dodatkowo pomiędzy tymi żądaniami występuje losowy odstęp czasu z przedziału 0-1s. W kolejnych próbach zmienia się odstęp czasu (w ms) pomiędzy nawiązywaniem kolejnych połączeń TCP. Na wykresach 1 i 2 kolorem czerwonym zaznaczono liczbę żądań zakończonych niepowodzeniem, a kolorem niebieskim liczbę poprawnie otwartych stron WWW.



Wykres 1. Test 2 bez korzystania z „Content Switch” (1 serwer WWW)



Wykres 2. Test 2 z wykorzystaniem „Content Switch” (2 serwery WWW)

Wykres 1 przedstawia sytuację, gdy nie korzystano z programu „Content Switch”. Wszystkie żądania kierowane są do jednego serwera WWW. Wykres 2 prezentuje wyniki uzyskane po zastosowaniu programu „Content Switch”, który rozdziela żądania pomiędzy dwa serwery WWW.

Z wykresów wynika, że zastosowanie programu „Content Switch” w tym przypadku, korzystnie zwiększyło ilość poprawnie otwartych stron.

Test 3

Celem doświadczenia jest określenie jak duża liczba jednocześnie otwartych połączeń będzie mieć wpływ na wydajność przełączania pakietów.

Program „WAPT” nawiązuje połączenia TCP w odstępach 50ms. W każdym połączeniu TCP zostaje zrealizowane 5 żądań pobrania strony o rozmiarze 7kB. Liczba połączeń i odstęp pomiędzy żadaniami w połączeniu zmieniają się w kolejnych próbach. Wyniki doświadczenia zebrane są w tabeli 1. Rubryki 4 i 5 odnoszą się do sytuacji, w której pomiędzy klientem, a serwerem znajduje się komputer z zainstalowanym programem „Content Switch”. Natomiast rubryki 6 i 7 do sytuacji, gdy klient komunikuje się bezpośrednio z serwerem. W obu przypadkach serwer jest tylko jeden.

1	2	3	4	5	6	7
Liczba połączeń	Liczba jednocześnie otwartych połączeń	Odstęp pomiędzy kolejnymi żadaniami	Średnia liczba otwartych stron na sekundę	Czas testu	Średnia liczba otwartych stron na sekundę	Czas testu
1000	1000	10s	146,28	116s	144,31	117s
1000	1000	20s	48,78	165s	48,43	166s
5000	1000	10s	54,71	456s	54,51	458s
5000	2000	20s	48,78	512s	48,72	513s
10000	2000	20s	52,86	904s	51,66	922s

Tabela 2. Wyniki testu 3

Maksymalna liczba jednocześnie otwartych połączeń utrzymuje się przez pewien czas zależny od wartości z rubryk 1 i 3. Podczas ostatniej próby w obu przypadkach pojawiła się porównywalna liczba błędów (8% żądań nie powiodło się).

Na podstawie przeprowadzonego doświadczenia można uznać, że liczba 2000 jednocześnie otwartych połączeń nie wpływa negatywnie na wydajność przełączania pakietów przez program „Content Switch”. Wyniki uzyskane w obu przypadkach są niemal identyczne. Niestety ograniczenia sprzętowe uniemożliwiły zaostrenie warunków testu i dalsze kontynuowanie prób.

Podsumowanie

Content Delivery Network jest zagadnieniem bardzo rozległym. W niniejszej pracy starano się zwrócić uwagę na najistotniejsze i najciekawsze zagadnienia związane z tym tematem.

CDN jest technologią szybko rozwijającą się. Począwszy od dostarczania tylko statycznych treści ewoluowała w kierunku dostarczania mediów strumieniowych, a obecnie na krawędź sieci przenoszone są aplikacje dynamiczne. Operatorzy sieci CDN oferują bardzo duży wybór usług, które mogą być świadczone w sieci CDN. Z tego pewnie powodu technologia ta jest coraz częściej wybieranym mechanizmem wspomagającym usługi sieciowe. Pozwala zmniejszyć koszty utrzymania własnej infrastruktury, jednocześnie zapewniając lepszą jakość, a przede wszystkim niezawodność usług.

Budując sieć CDN należy brać pod uwagę przyszłe jej zastosowanie. Producenci urządzeń tacy jak Cisco Systems dostarczają kompleksowe rozwiązania zarówno dla modelu ECDN jak i Internet CDN. Tworzą również własne standardy, co z kolei uniemożliwia niekiedy współpracę urządzeń różnych producentów.

Trwają prace nad usprawnianiem systemów zarządzania treścią. Pojawiają się publikacje prezentujące wydajne algorytmy dostarczania treści do krawędzi sieci oraz przemieszczania treści w rejony zwiększonego zapotrzebowania na daną treść.

W dziedzinie globalnego kierowania żądań bezkonkurencyjna od początku istnienia tego typu sieci pozostaje technika integracji rutera treści z systemem DNS ze względu na powszechność jego stosowania.

W części praktycznej podjęta została próba zrealizowania przełącznika treści, czyli elementu lokalnego kierowania żądań. Główny problem stanowiło opracowanie metody pośredniczenia w nawiązywaniu i utrzymywaniu połączeń TCP pomiędzy klientem i serwerem. Zostało to z powodzeniem zrealizowane w postaci sterownika pośredniego NDIS przeznaczonego do systemu „Windows XP”.

7. Załącznik

Instrukcja użytkownika programu „Content Switch”

Na potrzeby niniejszej pracy dyplomowej stworzony został program „Content Switch”, dzięki któremu komputerowi klasy PC możemy nadać funkcjonalność przełącznika treści. Program realizuje kierowanie żądań HTTP na podstawie pola `Host`, które wchodzi w skład nagłówka żądania HTTP i określa nazwę domenową serwera WWW, którego żądanie to dotyczy. Jeśli do obsługi jednej nazwy domenowej przeznaczony zostanie więcej niż jeden serwer WWW, program „Content Switch” rozdzieli kolejne żądania pomiędzy te serwery w myśl zasady *Round Robin*. Obecnie powszechnie stosowane jest tzw. stałe połączenie HTTP, podczas jednej sesji TCP może przyjść kilka zapytań HTTP. Prezentowany „Content Switch” pozwala na wybór serwera tylko raz tzn. na podstawie pierwszego zapytania HTTP. Z tego względu należy zadbać o to, aby wszystkie serwery WWW przypisane do obsługi danej domeny posiadały tę samą zawartość.

Wymagania programu

Program przeznaczony jest dla systemu operacyjnego „Windows XP”. Zalecana konfiguracja komputera to procesor powyżej 500MHz i 128MB pamięci. Komputer na, którym uruchamiamy program wyposażony musi być w minimum dwa interfejsy sieciowe, z czego jeden musi obsługiwać standard 802.3 (Ethernet). Dodatkowo na komputerze uruchomiona musi być usługa „Ruting i dostęp zdalny”.

Instalacja sterownika

Na dołączonej płycie CD w katalogu `Driver` znajdują się pliki: `ContentSwitch.sys`, `netsf.inf`, `netsf_m.inf`, które potrzebne są w procesie instalacji. Sterownik należy zainstalować w następujący sposób:

- W „Panelu sterowania” wybieramy folder „Połączenia sieciowe”, a następnie wchodzimy we „Właściwości” dowolnej karty sieciowej.
- Klikamy przycisk „Zainstaluj”.
- W oknie „Wybieranie typu składnika sieci” wybieramy typ „Usługa” i klikamy przycisk „Dodaj”.

- W oknie „Wybieranie usługi sieciowej” należy wskazać katalog, w którym znajdują się w/w pliki.
- Podczas instalacji ukaże się ostrzeżenie informujące o tym, że oprogramowanie nie przeszło testów zgodności z systemem „Windows XP”. Należy wtedy wybrać przycisk „Mimo to kontynuuj”.

Jeśli instalacja przebiegła pomyślnie w oknie „Właściwości” wszystkich kart sieciowych widoczna będzie usługa „Content Switch Driver”.

Konfiguracja sterownika

Konfiguracja przechowywana jest w pliku `c:\CS_config.cfg`, który zostaje odczytany w momencie uruchamiania sterownika (podczas instalacji i później podczas startu systemu operacyjnego). Jeśli plik konfiguracyjny jest poprawny konfiguracja jest pobierana i działa przełączanie połączeń.

Do skonfigurowania sterownika przeznaczona jest oddzielna aplikacja użytkownika „Content Switch Manager”, którą można znaleźć na dołączonej płycie CD w katalogu `Manager`. Z pomocą tego programu wskazujemy adapter sieciowy, na który będą przychodziły żądania HTTP oraz podajemy nazwy domenowe i obsługujące je serwery WWW.

Po lewej stronie okna aplikacji znajduje się rozwijana lista drzewiasta zawierająca dwa główne węzły: „Adaptory sieciowe” i „Domeny”.

Po rozwinięciu węzła „Adaptory sieciowe” ukazuje się lista zarejestrowanych w systemie adapterów sieciowych. Każdy adapter reprezentowany jest na tej liście poprzez swój unikalny w systemie operacyjnym identyfikator. Wybranie jednego adaptera powoduje wyświetlenie dodatkowych informacji o tym urządzeniu:

- Opis producenta – zazwyczaj nazwa urządzenia (karty sieciowej) lub ciąg znaków charakteryzujący to urządzenie nadany przez producenta.
- Typ adaptera – rodzaj medium transmisyjnego np. 802.3, WAN.
- Adres MAC – fizyczny adres urządzenia.
- Adres IP – adres warstwy sieciowej, jeśli został nadany.
- Stan – stan adaptera, czy włączony do sieci czy wyłączony.
- Kontrolka typu *CheckBox* „Włącz kierowanie ruchu HTTP” określająca, czy na tym adapterze odbywa się kierowanie żądań HTTP.

Zaznaczenie na liście adaptera powoduje uzyskanie tych informacji bezpośrednio ze sterownika. Aby upewnić się, że żaden z tych parametrów nie uległ zmianie można użyć przycisku „Odśwież dane adaptera”. Użycie przycisku „Odśwież listę adapterów” spowoduje pobranie ze sterownika aktualnej listy adapterów.

Węzeł „Domeny” umożliwia podanie listy obsługiwanych domen i dla każdej domeny listy serwerów, które ją obsługują. Po zaznaczeniu węzła „Domeny” pojawia się pole tekstowe „Wpisz nazwę domeny” i przycisk „Dodaj domenę”. Wprowadzając tekst i klikając na przycisk dodajemy nową domenę jako liść węzła „Domeny”. Uwaga, program nie sprawdza poprawności wprowadzonej nazwy.

Zaznaczenie nazwy domeny powoduje pojawienie się pola tekstowego „Wpisz IP serwera” oraz przycisków „Dodaj serwer” i „Usuń domenę”. Wpisanie adresu IP oraz zatwierdzenie przyciskiem „Dodaj serwer” powoduje wyświetlenie adresu IP jako liść węzła zaznaczonej nazwy domeny. Kliknięcie na przycisk „Usuń domenę” powoduje skasowanie z listy wybranej domeny oraz przypisanych do niej serwerów.

Zaznaczenie adresu IP serwera powoduje pojawienie się przycisku „Usuń serwer”, którego użycie skasuje wybrany serwer z listy.

Niezależnie od wybranego węzła na liście drzewiastej zawsze dostępne są:

- Kontrolka „Zapis do pliku logów”, której zaznaczenie spowoduje zapisywanie komunikatów sterownika do pliku logów.
- Pole tekstowe „Wpisz IP bramy”, w które należy wpisać adres IP należący do tej samej sieci co adres IP adaptera, na którym ma się odbywać kierowanie ruchu HTTP. Dodatkowo powinien być to adres działającego fizycznego interfejsu sieciowego, czyli np. bramy (ang. *gateway*).

Wybranie adaptera oraz zaznaczenie kontrolki „Włącz kierowanie ruchu HTTP” powoduje zapisanie konfiguracji do pliku konfiguracyjnego oraz powiadomienie sterownika, aby odczytał nową konfigurację. Kierowanie ruchu HTTP może odbywać się tylko na adapterze typu 802.3.

Gdy kierowanie ruchu HTTP jest uruchomione pola edycji oraz przyciski są zablokowane. Aby móc zmienić konfigurację należy najpierw wyłączyć kierowanie ruchu HTTP.

Po zakończeniu konfiguracji sterownika program „Content Switch Manager” można wyłączyć.

Jeśli zmianie ulegnie adres IP lub MAC karty sieciowej, na której odbywa się kierowanie ruchu HTTP, należy ponownie skonfigurować sterownik.

Plik logów

W pliku logów C:\CS_log.txt zapisywane są następujące komunikaty:

- Dla każdego nowego połączenia zapisywane są: adres IP klienta, nazwa domeny oraz IP wybranego serwera.
- Awaria serwera, podawany jest adres IP uszkodzonego serwera.
- Powiadomienie o tym, że uszkodzony serwer zaczął działać.

Deinstalacja sterownika

Wchodzimy we „Właściwości” dowolnej karty sieciowej. Następnie zaznaczamy usługę „Content Switch Driver” i klikamy przycisk „Odinstaluj”.

Szczegóły techniczne

W skład projektu wchodzi sterownik pośredni NDIS „Content Switch Driver” oraz aplikacja umożliwiająca konfigurację sterownika „Content Switch Manager”. Sterownik napisany został w języku C. Jako edytora użyto „Microsoft Visual Studio .NET 7.0”, jednak kompilacja odbyła się przy pomocy „Microsoft Windows Driver Development Kit (DDK) 2600.1106”. Aplikacja „Content Switch Manager” napisana w języku C++, stworzona i skompilowana została w „Microsoft Visual Studio .NET 7.0”. Sterownik przeznaczony jest do współpracy z systemem „Windows XP”.

Bazą do rozpoczęcia pracy był sterownik „PassThru” znajdujący się w przykładach dołączonych do „Windows DDK” rozszerzony do „Extended PassThru”, którego kod źródłowy i opis modyfikacji można znaleźć stronie internetowej [74].

Moduł napisany na potrzeby niniejszej pracy dyplomowej mieści się w pliku `ContentSwitch.c`. Zmiany wprowadzane w pozostałych plikach zamieszczone zostały pomiędzy komentarze:

```
//ContentSwitch_BEGIN
...
//ContentSwitch_END
```

Niezwykle pomocna w zrozumieniu działania sterowników pośrednich NDIS okazała się praca dyplomowa [75]. Przejrzyście opisano w niej budowę sterownika pośredniego NDIS, sposoby komunikowania się ze sterownikiem oraz budowę i obsługę pakietów NDIS.

Kod źródłowy funkcji `PtTransferDataComplete()` i `PtReceive()`, znajdujących się w pliku `Protocol.c`, zapożyczony został z programu „BandMan” dołączonego do pracy [75].

Kierowanie pakietów

Sterownik pośredni zajmuje miejsce na stosie sterowników NDIS pomiędzy sterownikiem miniportu [75], który obsługuje kartę sieciową, a sterownikami protokołów [75], czyli stosem TCP/IP. Wszystkie pakiety sieciowe przechodzą przez stos sterowników NDIS. Sterownik „Content Switch Driver” wychwytuje ruch HTTP, funkcja `CS_is_interesting()` weryfikuje czy ruch jest interesujący tzn. przychodzący na port 80 (kierunek IN) oraz ruch wychodzący z karty sieciowej z portu 80 (kierunek OUT). Pozostały ruch jest przepuszczany.

Pakiet przechodzący w kierunku IN trafia do funkcji `CS_packet_dir_in()`, a w kierunku OUT do funkcji `CS_packet_dir_out()`. Dla każdego pakietu przeszukiwana jest lista połączeń w poszukiwaniu pasującego połączenia. Pojedyncze połączenie reprezentuje struktura `connection`. Na podstawie informacji odczytanych z pakietu oraz informacji zgromadzonych w strukturze `connection` podejmowana jest decyzja jak postąpić z pakietem.

Połączenia zebrane są w listę dwukierunkową cykliczną. Każdą pozycję na liście opisuje standardowa struktura `LIST_ENTRY`. W ten sam sposób zbudowane są pozostałe listy używane w programie.

Pakiety przychodzące i wychodzące obsługiwane są współbieżnie, natomiast lista połączeń jest jedna. Synchronizacja dostępu do współdzielonych zasobów realizowana jest za pomocą standardowego mechanizmu jądra systemu (ang. *kernel*) o nazwie *SpinLock* [76].

Nawiązywanie połączenia TCP/IP

Jeśli okaże się, że mamy do czynienia z nowym połączeniem, realizowany jest następujący scenariusz:

- Przychodzi pakiet z ustawioną flagą SYN: nowe połączenie dodawane jest do listy połączeń, zapamiętywany jest numer sekwencyjny (`connection::c_seq_number`) z pierwszego pakietu, źródłowy adres IP, źródłowy adres MAC oraz dane całego pakietu, oryginalny pakiet NDIS jest odrzucany.
- Wysłany zostaje pakiet z ustawioną flagą SYN+ACK. Źródłowy adres IP (`_ADAPT::AddressIP`), źródłowy adres MAC (`_ADAPT::AddressMAC`) znajdują się w konfiguracji adaptera wcześniej pobranej z pliku. Docelowy adres MAC jest taki sam jak źródłowy adres MAC z pakietu z flagą SYN. Adresy MAC muszą być ustawione, ponieważ przy wysyłaniu pakietu „w dół” nie działa protokół ARP. Numer sekwencyjny (`connection::cs_seq_number`) generuje funkcja `random()`.
- Teraz należy spodziewać się pakietu z ustawioną flagą ACK, a następnie pakietu zawierającego żądanie HTTP. Program przewiduje obsługę sytuacji, kiedy żądanie HTTP rozbite jest na kilka pakietów TCP. Kolejne pakiety dodawane są do listy pakietów HTTP (`connection::Pkt_cHTTP_list`), a

przechowywane są w postaci struktury `packet_HTTP`. Pakiety układane są na liście w rosnącej kolejności numerów sekwencyjnych i w takiej kolejności muszą przyjść. Funkcja `is_next_to_httpList()` sprawdza czy pakiet można umieścić na liście. Po każdym pakiecie umieszczonym na liście wysyłane jest potwierdzenie. Niewłaściwe pakiety (zła kolejność) są odrzucane i klient musi je ponownie wysłać. Pakiety buforowane są do momentu przyjścia całego nagłówka żądania HTTP. Rozpoznawane jest to po podwójnej nowej linii (`\r\n\r\n`), zajmuje się tym funkcja `isHttpDNL()`. Gdy dojdzie już cały nagłówek zapytania HTTP, funkcja `choose_server()` odczytuje domenę z pola `host`, a następnie funkcja `find_best_server()` wybiera serwer, który obsłuży to żądanie.

- Do wybranego serwera wysłany zostaje pakiet z ustawioną flagą SYN. Przy wysyłaniu „w górę” pakiet przechodzi przez stos protokołów TCP/IP, następuje ruting. Adresów MAC źródłowego i docelowego nie trzeba ustawiać, gdyż zadziała protokół ARP. Dla tego pakietu przewidziana jest retransmisja. Odstęp między retransmisjami wynosi tylko 2 sekundy. Wybrano krótki czas, aby klient długo nie czekał, gdyż do czasu nawiązania połączenia z serwerem WWW pakiety od klient będą odrzucane. Po dwóch nieudanych retransmisjach (6s) połączenie zostaje skasowane z listy , a do klienta odesłany pakiet z ustawioną flagą RST.
- Gdy serwer odeśle pakiet z ustawioną flagą SYN+ACK, zapamiętywany jest numer sekwencyjny (`connection::s_seq_number`). Do serwera wysłany zostaje pakiet z ustawioną flagą ACK, a następnie wszystkie pakiety z listy pakietów HTTP. Dla pakietów HTTP przewidziana jest retransmisja, która zaczyna się od pakietu z najwyższym numerem sekwencyjnym jaki został potwierdzony. Po dwóch nieudanych retransmisjach (6s) połączenie zostaje skasowane z listy , a do klienta odesłany pakiet z ustawioną flagą RST.
- Gdy serwer WWW potwierdzi otrzymanie wszystkich pakietów z listy pakietów HTTP, połączenie zostaje uznane za nawiązane. Następuje obliczenie różnicy numerów sekwencyjnych:

```
connection::seq_number_diff=cs_seq_number-s_seq_number
```

Do generowania pakietów używana jest funkcja `generate_packet_data()`. Wysyłaniem pakietów „w górę” i „w dół” zajmują się funkcje: `send_packet_up()` i `send_packet_down()`.

Obsługa nawiązanych połączeń

Od momentu nawiązania połączenia w pakietach wysyłanych między klientem i serwerem funkcja `packet_update()` podmienia odpowiednie pola:

- Pakiet od klienta do serwera: numer sekwencyjny pozostaje bez zmian, od numeru potwierdzenia odejmuje się różnicę (`seq_number_diff`), adres IP przeznaczenia zamienia się na adres IP wybranego serwera.
- Pakiet od serwera do klienta: numer potwierdzenia pozostaje bez zmian, do numeru sekwencyjnego dodaje się różnicę (`seq_number_diff`), adres IP źródłowy zamienia się na adres IP adaptera na, którym odbywa się kierowanie ruchu HTTP.

W obu przypadkach sumy kontrolne IP i TCP obliczają funkcje: `ip_sum_calc()` i `tcp_sum_calc()`.

Zamykanie połączenia

Połączenie kasowane jest z listy po otrzymaniu od jednej ze stron pakietu z ustawioną flagą RST lub po wzajemnej wymianie pakietów z flagami FIN+ACK, ACK. Dodatkowo każdy pakiet uaktualniania czas `connection::inactivity_time`, jeśli różnica między tym czasem, a aktualnym przekroczy 120 sekund, połączenie jest kasowane z listy. Zabezpiecza to przed pozostawianiem na liście nieprawidłowo zamkniętych połączeń.

Wątki pomocnicze

W momencie włączenia kierowania ruchu HTTP uruchamiane są cztery wątki pomocnicze, które po inicjalizacji zatrzymują się na semaforze typu `KSEMAPHORE`. Dla każdego wątku przeznaczony jest oddzielny semafor. Zostaje on zwolniony w momencie dodania nowego elementu na listę, którą obsługuje dany wątek. Charakterystyka poszczególnych wątków:

- `timeout_thread` – kasuje elementy listy połączeń, gdy `connection::MagicDEL` przyjmie odpowiednią wartość. Dodatkowo kasuje

element, gdy przez 120 sekund nie przyjdzie żaden pakiet należący do tego połączenia.

- `retransmission_thread` – obsługuje listę retransmisji, której elementy tworzy się za pomocą struktury `retransmission`. Retransmitowane są tylko pakiety z ustawioną flagą SYN oraz pakiety zawierające żądanie HTTP przeznaczone dla serwera WWW. Przewidziane są dwie retransmisje w odstępie 2 sekund.
- `server_health_thread` - obsługuje listę uszkodzonych serwerów, której elementy tworzy się za pomocą struktury `server_health`. Wątek co 60s wysyła pakiet z ustawioną flagą SYN do wszystkich uszkodzonych serwerów. Jeśli w ciągu 50ms przyjdzie pakiet z ustawionymi flagami SYN+ACK, serwer zostaje uznany za sprawny.
- `logging_thread` - obsługuje listę komunikatów, której elementy tworzy się za pomocą struktury `log_entry`. Nowy komunikat umieszcza się na początku listy wywołując funkcję `write_to_log_list()`. Wątek zdejmuje elementy z końca listy i zapisuje komunikat do pliku wywołując funkcję `write_to_log_file()`.

Komunikacja ze sterownikiem

Aplikacja użytkownika inicjuje komunikację ze sterownikiem używając mechanizmu IRP (ang. *I/O Request Packet*) [76]. Pierwszym krokiem jest otwarcie uchwytu do urządzenia, które w systemie reprezentuje sterownik, przy pomocy funkcji `CreateFile()`. Następnie funkcja `DeviceIoControl()` przekazuje do sterownika zdefiniowany w pliku `Iocommon.h` komunikat IOCTL. Na podstawie komunikatu IOCTL sterownik wywołuje przypisaną do komunikatu funkcję. Zdefiniowane zostały cztery takie funkcje:

- `DevEnumerateBindings()` – odczytanie listy adapterów.
- `DevOpenAdapter()` – przypisanie otwartego połączenia z aplikacją użytkownika do konkretnego adaptera.
- `DevQueryInformation()` – odczytywanie informacji o adapterze
- `CS_ConfigureAdapter()` – zatrzymuje i uruchamia kierowanie pakietów HTTP.

Plik `Ioccommon.h` jest wspólny dla sterownika i aplikacji użytkownika, zawiera definicje kodów IOCTL niezbędnych do komunikacji ze sterownikiem oraz definicje struktur używanych przy zapisywaniu i odczytywaniu konfiguracji.

Włączanie i wyłączanie przełączania pakietów

Przełączanie pakietów zostaje włączone po zaznaczeniu pola „Włącz kierowanie ruchu HTTP” w wyniku wywołania funkcji `OnBnClickedCheck1()`. Funkcja `ZapisDoPliku()` zapisuje konfigurację do pliku `c:\CS_config.cfg`, następnie poprzez wywołanie `KonfigurujSterownik()` powiadamiany jest sterownik, który przyjmuje zgłoszenie wywołując `CS_ConfigureAdapter()`. Następnie sterownik odczytuje konfigurację wywołując `read_config()`. Uruchamiane są wątki pomocnicze.

Przełączanie pakietów zostaje wyłączone po odznaczeniu pola „Włącz kierowanie ruchu HTTP” w wyniku wywołania funkcji `OnBnClickedCheck1()`. Poprzez wywołanie `KonfigurujSterownik()` powiadamiany jest sterownik, który przyjmuje zgłoszenie wywołując `CS_ConfigureAdapter()`. Wątki pomocnicze kończą swoje działanie, `domain_list_del()` kasuje konfigurację sterownika oraz kasowana jest lista połączeń.

Uszkodzenie serwera

Serwer zostaje uznany za uszkodzony, jeśli po upływie 2 sekund po drugiej retransmisji nie przyjdzie potwierdzenie lub gdy serwer wyśle pakiet z ustawionymi flagami RST+ACK. Do uszkodzonego serwera nie są kierowane nowe połączenia. Jednak jeśli na liście połączeń znajdują się połączenia z uszkodzonym serwerem to przychodzące pakiety są nadal normalnie przełączane. Jeśli taki pakiet przyjdzie od strony uszkodzonego serwera to serwer ten zostaje uznany za działający.

Packet Stacking

Sterowniki pośrednie w wersji NDIS wcześniejszej niż 5.1, aby przepuścić pakiet muszą skopiować jego zawartość do nowo utworzonego pakietu NDIS. Wersja 5.1 biblioteki NDIS (dla „Windows XP”) umożliwia zastosowanie mechanizmu *Packet Stacking*. Stosując ten mechanizm sterownik pośredni nie musi kopiować zawartości każdego pakietu NDIS do nowego pakietu. Można pracować na pakiecie, który

otrzymano od sąsiedniego sterownika NDIS. Oszczędza to zasoby sprzętowe i przyspiesza proces przekazywania pakietów. Każdy pakiet posiada domyślenie dwa wskaźniki do stosu (struktury `NDIS_PACKET_STACK`). Korzystanie z *Packet Stacking* warunkuje wynik wywołania funkcji `NdisIMGetCurrentPacketStack(Packet, &StacksRemaining)`. Jeśli `StacksRemaining` przyjmie wartość `false` to nie można skorzystać z tego mechanizmu, ponieważ na stosie nie ma już wolnego miejsca. Wielkość stosu nie może zmieniać się dynamicznie, można ją określić w rejestrze systemu operacyjnego. `StacksRemaining` przyjmie wartość `false`, gdy liczba zainstalowanych sterowników pośrednich przewyższy liczbę miejsc na stosie. Sterownik korzystający z *Packet Stacking* musi być przygotowany do obsługi pakietów w standardowy sposób.

„Content Switch Driver” w obecnej wersji nie został przystosowany do obsługi *Packet Stacking*. Kopiowaniem zawartości pakietów zajmuje się funkcja `copy_from_buffer()`.

Możliwości rozwoju

Prezentowany program w obecnej postaci pozostawia duże możliwości dalszego rozwoju. Oto kilka propozycji:

- Użycie mechanizmu *Packet Stacking* mogłoby korzystnie wpłynąć na szybkość przełączania pakietów.
- Rozszerzenie funkcjonalności o kierowanie na podstawie typu treści przy założeniu, że korzystamy z połączenia HTTP typu `close`.
- Opracowanie bardziej skomplikowanego mechanizmu wyboru najlepszego serwera, na przykład na podstawie czasu odpowiedzi serwera na pakiet z ustawioną flagą SYN.

Bibliografia

- [1] „Dystrybucja treści jako remedium na dostęp do witryn WWW (cz. I)”, Integrator Nr 9-10/2001
- [2] S. Herrmann, J. Laurenson, M. Recore „Cisco Internet Architecture Essentials Self-Study Guide: Cisco Internet Solutions Specialist”, Cisco Press 2002
- [3] „Content Delivery Network - usługi hostingowe”, Integrator Nr 5-6/2001
- [4] I.Sikorski „Sieć bliżej użytkownika”, Telenetforum Nr 11/2001
- [5] Akamai. <http://www.akamai.com>
- [6] Speedera. <http://www.speedera.com>
- [7] Mirror Image. <http://www.mirror-image.com>
- [8] VitalStream. <http://www.vitalstream.com>
- [9] L.Bent, M. Rabinovich, G. Voelker, Z. Xiao „Towards Informed Web Content Delivery”, October 2004.
- [10] Verio. <http://www.verio.com/about/network/gin.cfm>
- [11] OneStopClick „Content Delivery - Buyer's Guide”, 2004
- [12] „Cisco Enterprise CDN Software User Guide Version 3.0”, <http://www.cisco.com>
- [13] G. Peng „CDN: Content Distribution Network”, January 2003
- [14] M. Day, B. Cain, G. Tomlinson, P. Rzewski „A Model for Content Internetworking (CDI)”, RFC 3466, February 2003
- [15] Internetowa encyklopedia. <http://pl.wikipedia.org>
- [16] Nortel Networks „Content Delievery Network Solutions”, 2002
- [17] Internetowy słownik. <http://www.idg.pl/slownik/>
- [18] P.Szczepaniak „CDN w korporacjach”, NetWorld Nr 6/2003
- [19] „Dystrybucja treści jako remedium na dostęp do witryn WWW (cz. II)”, Integrator Nr 9-10/2001
- [20] A.Chrząszcz, W. Mikiciuk, B. Motoszko, M. Siciarek „DNS a usługa LDAP-analiza funkcjonalności i wymagań instalacyjnych”, <http://ldap.uni.torun.pl/raporty/ftp/nask/nask-dns.html>
- [21] J. Savill „What is DNS round robin and subnet prioritization?”, October 2002
- [22] P. Albitz, C. Liu „DNS and BIND, 4th Edition”, O'Reilly 2001
- [23] „Cisco Internet CDN Software User Guide”, <http://www.cisco.com>
- [24] A. Barbir, B. Cain, R. Nair, O. Spatscheck „Known Content Network (CN) Request-Routing Mechanisms”, RFC 3568, July 2003

- [25] A. Madhavapeddy, A. Crivelli „How to Build a Content Delivery Network”, Network Appliance April 2002
- [26] „DNS BIND view Clause”, <http://www.zytrax.com/books/dns/ch7/view.html>
- [27] Md. Humayun Kabir, Eric G. Manning, Gholamali C. Shoja „Request-Routing Trends and Techniques in Content Distribution Network”, December 2002
- [28] Unitech Networks. <http://www.unitechnetworks.com>
- [29] „Anycast Addressing on the Internet”, <http://www.kuro5hin.org/story/2003/12/31/173152/86>
- [30] M. Gritter, D. Cheriton „An Architecture for Content Routing Support in the Internet” March 2001
- [31] Z. Fei, S. Bhattacharjee, E. Zegura, M. Ammar „A Novel Server Selection Technique for Improving the Response Time of a Replicated Service”, 1998
- [32] M. Syme, P. Goldie „Optimizing Network Performance with Content Switch: Server, Firewall and Cache Load Balancing”, Prentice Hall 2003
- [33] L. McKeag „What you need to know about balancing your network for content delivery”, April 2004, <http://www.techworld.com>
- [34] L. McKeag „Techniques to spread the load across your network and your server farm”, April 2004, <http://www.techworld.com>
- [35] V. Subramaniam „Content Switching” Wipro White Paper <http://www.wipro.in>
- [36] Akamai „Fast Internet Content Delivery with FreeFlow”, April 2000
- [37] A. Czerwiński „CDN Solutions”, November 2002, cisco.parsek.tv/si/prezentacije/cdn.pdf
- [38] Specyfikacja języka ESI. <http://www.esi.org>
- [39] Akamai White Paper „Turbo-Charging Dynamic Web Sites with Akamai EdgeSuite”, <http://www.akamai.com>
- [40] J. Jung, B. Krishnamurthy, M. Rabinovich „Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites”, May 2002
- [41] P. Vixie, D. Wessels „Hyper Text Caching Protocol (HTCP/0.0)”, RFC 2756, January 2000
- [42] V. Valloppillil, K. Ross „Cache Array Routing Protocol v1.0”, Internet Draft, February 1998
- [43] M. Hamilton, A. Rousskov, D. Wessels „Cache Digest specification - version 5”, December 1998
- [44] J. Kangasharju „Internet Content Distribution”, April 2002

- [45] S. Hull „Content Delivery Networks, Web Switching for Security, Availability, and Speed”, Osborne/McGraw-Hill, February 2002
- [46] L. Qiu, V. Padmanabhan, G. Voelker „On the Placement of Web Server Replicas”, 2001
- [47] S. Jamin, C. Jin, T. Kurc „Constrained mirror placement on the internet”, 2001
- [48] J. Kangasharju, J. Roberts, K. Ross „Object Replication Strategies in Content Distribution Networks”, 2001
- [49] A. Shaikh R. Tewari, M. Agrawal „On the Effectiveness of DNS-based Server Selection”, 2001
- [50] D. Wessels, K. Claffy „Internet Cache Protocol (ICP), Version 2”, RFC 2186, September 1997
- [51] D. Wessels, K. Claffy „Application of Internet Cache Protocol (ICP), Version 2”, RFC 2187, September 1997.
- [52] P. Rodriguez, Ch. Spanner, E. Biersack, „Analysis of Web Caching Architectures: Hierarchical and Distributed Caching”, August 2001
- [53] M. Green, B. Cain, G. Tomlinson, M. Speer, P. Rzewski, S. Thomas „Content Internetworking Architectural Overview”, IETF Draft, June 2002
- [54] B. Cain, O. Spatscheck, K. van der Merwe, L. Amini, A. Barbir Barbir, M. May May, D. Kaplan „Content Network Advertisement Protocol (CNAP)”, IETF Draft, July 2002.
- [55] A. Biliris, C. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatscheck, W. Sturm „CDN Brokering”, 2002
- [56] M. Gan „A Hybrid Hierarchical Request-Routing Architecture For Content Internetworking”, July 2002
- [57] E. Turrini „An architecture for Content Distribution Internetworking”, March 2004
- [58] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart „HTTP Authentication: Basic and Digest Access Authentication”, RFC 2617, June 1999
- [59] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee „Hypertext Transfer Protocol – HTTP/1.1”, RFC 2616, June 1999
- [60] D. Plummer „An Ethernet Address Resolution Protocol”, RFC 826, November 1982
- [61] T. Brisco „DNS Support for Load Balancing”, RFC 1794, April 1995
- [62] J. Postel „Internet Control Message Protocol”, RFC 792, September 1981

- [63] J. Postel, J. Reynolds „File Transfer Protocol”, RFC 959, October 1985
- [64] H. Schulzrinne, A. Rao, R. Lanphier, „Real Time Streaming Protocol (RTSP)”, RFC 2326, April 1998
- [65] B. Kantor, P. Lapsley „Network News Transfer Protocol”, RFC 977, February 1986
- [66] J. Klensin „Simple Mail Transfer Protocol”, RFC 2821, April 2001
- [67] J. Myers, M. Rose „Post Office Protocol - Version 3”, RFC 1939, May 1996
- [68] M. Crispin „INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1”, RFC 350, March 2003
- [69] C. Rigney, S. Willens, A. Rubens, W. Simpson „Remote Authentication Dial In User Service (RADIUS)”, RFC 2865, June 2000
- [70] Specyfikacja SSL 3.0. <http://wp.netscape.com/eng/ssl3/>
- [71] Specyfikacja SOAP. <http://www.w3.org/TR/soap/>
- [72] Java RMI Tutorial. <http://java.sun.com/docs/books/tutorial/rmi/>
- [73] Dokumentacja JDBC 3.0 <http://java.sun.com/j2se/1.4.2/docs/guide/jdbc/index.html>
- [74] T. Divine „Extending The Microsoft PassThru NDIS Intermediate Driver”, July 2003, <http://www.wd-3.com/archive/ExtendingPassthru.htm>
- [75] K. Pijewski „Kształtowanie ruchu w sieciach komputerowych”, wrzesień 2004
- [76] W. Oney „Programming the Microsoft Windows Driver Model”, Microsoft Press, 1999
- [77] Prezentacja programu „WAPT”. <http://www.loadtestingtool.com>
- [78] Prezentacja programu „Virtual PC”, <http://www.microsoft.com/windows/virtualpc/default.msp>