

SIECI KOMPUTEROWE NAJCZĘSTSZE PROBLEMY — I ICH — ROZWIĄZANIA

INNOWACYJNE PODEJŚCIE DO BUDOWANIA
ODPORNYCH, NOWOCZESNYCH SIECI

RUSS WHITE ETHAN BANKS

Tytuł oryginału: Computer Networking Problems and Solutions:
An innovative approach to building resilient, modern networks

Tłumaczenie: Witold Woicki (wstęp, rozdz. 1 – 10, 25 – 30), Lech Lachowski (rozdz. 11 – 24)

ISBN: 978-83-283-5043-4

Authorized translation from the English language edition, entitled: COMPUTER NETWORKING PROBLEMS AND SOLUTIONS: AN INNOVATIVE APPROACH TO BUILDING RESILIENT, MODERN NETWORKS, First Edition, ISBN 1587145049 by Russ White and Ethan Banks, published by Pearson Education, Inc, publishing as Cisco Press, Copyright © 2018 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Polish language edition published by HELION S.A., Copyright © 2018 Pearson Education, Inc.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/sienpr>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorach	17
Wprowadzenie	19
Część I. Płaszczyzna danych	23
Rozdział 1. Podstawowe pojęcia	27
Sztuka czy inżynieria?	28
Komutacja łączy	30
Przełączanie pakietów	33
<i>Działanie przełączania pakietów</i>	<i>34</i>
<i>Kontrola przepływu w sieci z przełączaniem pakietów</i>	<i>36</i>
Ramki o stałej a ramki o zmiennej długości	37
Obliczanie ścieżek pozbawionych pętli	40
Jakość usług	42
Zemsta scentralizowanych płaszczyzn sterowania	44
Złożoność	44
<i>Skąd ta złożoność?</i>	<i>45</i>
<i>Definiowanie złożoności</i>	<i>46</i>
<i>Zarządzanie złożonością poprzez talię osy</i>	<i>49</i>
Końcowe rozważania	51
Dalsza lektura	51
Pytania kontrolne	52
Rozdział 2. Problemy i rozwiązania związane z transportem danych	55
Cyfrowe gramatyki i organizowanie	57
<i>Cyfrowe gramatyki i słowniki</i>	<i>57</i>
<i>Pola o stałej długości</i>	<i>60</i>
<i>Format TLV</i>	<i>62</i>
<i>Współdzielone słowniki obiektów</i>	<i>63</i>

Błędy	64
<i>Wykrywanie błędów</i>	64
<i>Korekcja błędów</i>	69
Multipleksowanie	71
<i>Adresacja urządzeń i aplikacji</i>	71
<i>Multicast</i>	73
<i>Anycast</i>	76
Kontrola przepływu	78
<i>System okien dystrybucji</i>	79
<i>Negocjowane szybkości transmisji bitów</i>	83
Końcowe rozważania dotyczące transportu	84
Dalsza lektura	84
Pytania kontrolne	86
Rozdział 3. Modelowanie transportu sieciowego	89
Model Departamentu Obrony Stanów Zjednoczonych (DoD)	90
Model Open Systems Interconnect (OSI)	93
Model rekursywnej architektury internetowej (RINA)	97
Zorientowanie na połączenie i bezpołączeniowość	99
Końcowe rozważania	99
Dalsza lektura	100
Pytania kontrolne	100
Rozdział 4. Transport w niższych warstwach	103
Ethernet	104
<i>Multipleksowanie</i>	104
<i>Kontrola błędów</i>	110
<i>Organizowanie danych</i>	111
<i>Kontrola przepływu</i>	112
Sieć bezprzewodowa 802.11	112
<i>Multipleksowanie</i>	113
<i>Organizowanie danych, kontrola błędów i kontrola przepływu</i>	119
Końcowe rozważania dotyczące protokołów transmisji w niższych warstwach	120
Dalsza lektura	121
Pytania kontrolne	121
Rozdział 5. Transport danych w wyższych warstwach	123
Protokół internetowy	125
<i>Transport i organizowanie danych</i>	127
<i>Multipleksowanie</i>	130
Protokół kontroli transmisji (TCP)	135
<i>Kontrola przepływu</i>	135
<i>Kontrola błędów</i>	140
<i>Numery portów TCP</i>	140
<i>Konfiguracja sesji protokołu TCP</i>	141

QUIC	141
<i>Redukcja początkowego uzgadniania</i>	142
<i>Ograniczanie retransmisji</i>	142
<i>Zmniejszenie blokowania początku linii</i>	143
ICMP	146
Końcowe rozważania	147
Dalsza lektura	148
Pytania kontrolne	150
Rozdział 6. Odnajdowanie międzywarstwowe	151
Rozwiązania w zakresie odnajdowania międzywarstwowego	152
<i>Powszechnie znane lub ręcznie skonfigurowane identyfikatory</i>	153
<i>Mapowanie bazy danych i protokołu</i>	154
<i>Ogłaszanie mapowania identyfikatorów za pomocą protokołu</i>	155
<i>Wylizanie jednego identyfikatora z innego</i>	155
Przykłady odnajdowania międzywarstwowego	155
<i>System nazw domen (DNS)</i>	155
<i>DHCP</i>	157
<i>Protokół rozwiązywania adresów IPv4</i>	159
<i>IPv6 Neighbor Discovery — odnajdowanie sąsiadów</i>	161
Problem z bramą domyślną	163
Końcowe rozważania	166
Dalsza lektura	167
Pytania kontrolne	168
Rozdział 7. Przełączanie pakietów	169
Z medium fizycznego do pamięci	171
Przetwarzanie pakietu	172
<i>Przełączanie</i>	172
<i>Routing</i>	173
<i>Po co routować?</i>	174
<i>Wiele ścieżek o równym koszcie</i>	175
Przez magistralę	182
<i>Krzyżowe pola komutacyjne i rywalizacja</i>	184
Z pamięci do medium fizycznego	186
Końcowe rozważania dotyczące przełączania pakietów	187
Dalsza lektura	188
Pytania kontrolne	189
Rozdział 8. Jakość usług	191
Definiowanie zakresu problemu	192
<i>Dlaczego po prostu nie zrobić wystarczająco szybkich łącz?</i>	193
Klasyfikacja	194
<i>Zachowywanie klasyfikacji</i>	199
<i>Nieoznaczony Internet</i>	201

Zarządzanie zatorami	202
<i>Terminowość — kolejkovanie o niskim opóźnieniu</i>	202
<i>Uczciwość — uczciwe kolejkovanie ważne według klasy</i>	207
<i>Krytyczne przeciążenie</i>	208
<i>Inne narzędzia QoS do zarządzania zatorami</i>	208
Zarządzanie kolejką	209
<i>Zarządzanie pełnym buforem — ważne losowe wczesne wykrywanie</i>	209
<i>Zarządzanie opóźnieniami bufora, rozdęte bufor i CoDel</i>	210
Końcowe rozważania dotyczące QoS	212
Dalsza lektura	212
Pytania kontrolne	213
Rozdział 9. Wirtualizacja sieci	215
Zrozumieć sieci wirtualne	216
<i>Świadczenie usług Ethernet w sieci IP</i>	219
<i>Wirtualny prywatny dostęp do sieci korporacyjnej</i>	220
<i>Podsumowanie problemów i rozwiązań związanych z wirtualizacją</i>	222
Routing segmentowy	223
<i>Routing segmentowy z wieloprotokołowym przełączaniem etykiet (MPLS)</i>	224
<i>Routing segmentowy w IPv6</i>	228
<i>Sygnalizowanie etykiet routingu segmentowego</i>	229
Sieci rozległe definiowane programowo	230
Złożoność i wirtualizacja	232
<i>Powierzchnie interakcji i grupy łączy wspólnego ryzyka</i>	232
<i>Powierzchnie interakcji i nakładające się płaszczyzny sterowania</i>	234
Końcowe rozważania dotyczące wirtualizacji sieci	236
Dalsza lektura	236
Pytania kontrolne	238
Rozdział 10. Bezpieczeństwo transportu	239
Sformułowanie problemu	240
<i>Sprawdzanie prawidłowości danych</i>	240
<i>Ochrona danych przed badaniem</i>	240
<i>Ochrona prywatności użytkowników</i>	241
Dostępne rozwiązania	242
<i>Szyfrowanie</i>	242
<i>Wymiana kluczy</i>	249
<i>Skróty kryptograficzne</i>	252
<i>Zaciemnianie danych użytkownika</i>	252
Transport Layer Security	257
Końcowe rozważania dotyczące bezpieczeństwa transportu	259
Dalsza lektura	260
Pytania kontrolne	261

Część II. Płaszczyzna sterowania263**Rozdział 11. Wykrywanie topologii265**

Węzły, krawędzie i osiągalne miejsca docelowe	267
<i>Węzeł</i>	267
<i>Krawędź</i>	268
<i>Osiągalne miejsca docelowe</i>	268
<i>Topologia</i>	270
Poznanwanie topologii	271
<i>Wykrywanie innych urządzeń sieciowych</i>	271
<i>Wykrywanie łączności dwukierunkowej</i>	272
<i>Wykrywanie maksymalnej jednostki transmisji (MTU)</i>	274
Uczenie się osiągalnych miejsc docelowych	276
<i>Uczenie się reaktywne</i>	276
<i>Uczenie się proaktywnie</i>	277
Rozgłaszanie osiągalności i topologii	278
<i>Decydowanie, kiedy należy rozgłaszać osiągalność i topologię</i>	278
<i>Reaktywne rozpowszechnianie osiągalności</i>	280
<i>Proaktywne rozpowszechnianie osiągalności</i>	283
Redystrybucja między płaszczyznami sterowania	285
<i>Redystrybucja i metryki</i>	285
<i>Pętle redystrybucji i routingu</i>	287
Końcowe rozważania dotyczące wykrywania topologii	289
Dalsza lektura	290
Pytania kontrolne	291

Rozdział 12. Wolne od pętli ścieżki unicastowe (1)293

Która ścieżka jest wolna od pętli?	294
Drzewa	296
Alternatywne ścieżki wolne od pętli	299
<i>Model wodospadu (lub zlewiska)</i>	300
<i>Przestrzeń P/Q</i>	302
<i>Zdalne alternatywy bez pętli</i>	303
Obliczanie wolnej od pętli ścieżki za pomocą algorytmu Bellmana-Forda	304
Algorytm DUAL Garcii	310
Końcowe rozważania	315
Dalsza lektura	315
Pytania kontrolne	316

Rozdział 13. Wolne od pętli ścieżki unicastowe (2)317

Najkrótsza ścieżka Dijkstry	317
<i>Częściowy i przyrostowy SPF</i>	324
<i>Obliczanie LFA i rLFA</i>	325
Wektor ścieżki	327

Algorytmy rozłącznej ścieżki	330
<i>Sieć dwupołączona</i>	331
<i>Algorytm rozłącznej ścieżki Suurballe'a</i>	332
<i>Maksymalnie redundantne drzewa</i>	336
Łączność dwukierunkowa	339
Końcowe rozważania	340
Dalsza lektura	341
Pytania kontrolne	343
Rozdział 14. Reagowanie na zmiany w topologii	345
Wykrywanie zmian w topologii	347
<i>Odpytywanie w celu wykrycia awarii</i>	347
<i>Wykrywanie awarii oparte na zdarzeniach</i>	348
<i>Porównanie wykrywania opartego na zdarzeniach i odpytywaniu</i>	350
<i>Przykład: wykrywanie przekazywania dwukierunkowego</i>	351
Dystrybucja zmian	354
<i>Zalewanie</i>	354
<i>Przeskok po przeskoku</i>	358
<i>Scentralizowany magazyn</i>	359
Spójność, dostępność i odporność na partycjonowanie	362
Końcowe rozważania	364
Dalsza lektura	365
Pytania kontrolne	366
Rozdział 15. Płaszczyzny sterowania wykorzystujące protokoły wektora odległości	367
Klasyfikacja płaszczyzn sterowania	368
Protokół STP	371
<i>Budowanie drzewa wolnego od pętli</i>	371
<i>Poznananie osiągalnych miejsc docelowych</i>	375
<i>Podsumowanie protokołu STP</i>	377
Protokół RIP	378
<i>Powiązanie algorytmu Bellmana-Forda z protokołem RIP</i>	380
<i>Reagowanie na zmiany w topologii</i>	382
<i>Podsumowanie protokołu RIP</i>	383
Protokół EIGRP	383
<i>Reagowanie na zmianę topologii</i>	386
<i>Wykrywanie sąsiadów i niezawodny transport</i>	388
<i>Podsumowanie protokołu EIGRP</i>	390
Dalsza lektura	391
Pytania kontrolne	392

Rozdział 16. Płaszczyzny sterowania wykorzystujące protokoły stanu łącza i wektora ścieżki	395
Krótką historia OSPF i IS-IS	396
Protokół IS-IS	397
<i>Adresowanie OSI</i>	397
<i>Marshalling danych w protokole IS-IS</i>	399
<i>Wykrywanie sąsiadów i topologii</i>	399
<i>Niezawodne zalenie</i>	401
<i>Podsumowanie protokołu IS-IS</i>	403
Protokół OSPF	404
<i>Marshalling danych w protokole OSPF</i>	404
<i>Wykrywanie sąsiadów i topologii</i>	406
<i>Niezawodne zalewanie</i>	407
<i>Podsumowanie protokołu OSPF</i>	409
Wspólne elementy protokołów OSPF i IS-IS	409
<i>Łącza wielodostępowe</i>	410
<i>Konceptualizacja łączy, węzłów i osiągalności w protokołach stanu łącza</i>	412
<i>Sprawdzanie łączności dwukierunkowej w SPF</i>	414
Protokół BGP	414
<i>Peering BGP</i>	415
<i>Proces decyzyjny wyboru najlepszej ścieżki w protokole BGP</i>	417
<i>Reguły rozgłaszania BGP</i>	419
<i>Podsumowanie protokołu BGP</i>	421
Końcowe rozważania	421
Dalsza lektura	422
Pytania kontrolne	424
Rozdział 17. Reguły w płaszczyźnie sterowania	425
Przypadki użycia reguł płaszczyzny sterowania	426
<i>Routing i ziemniaki</i>	426
<i>Segmentacja zasobów</i>	428
<i>Przypinanie przepływów dla optymalizacji aplikacji</i>	429
Definiowanie reguł płaszczyzny sterowania	434
Reguły i złożoność płaszczyzny sterowania	435
<i>Routing i ziemniaki</i>	435
<i>Segmentacja zasobów</i>	436
<i>Przypinanie przepływów dla optymalizacji aplikacji</i>	438
Końcowe rozważania dotyczące reguł płaszczyzny sterowania	439
Dalsza lektura	439
Pytania kontrolne	440

Rozdział 18. Scentralizowane płaszczyzny sterowania	441
Definicja pojęcia software defined	442
Systematyka interfejsów	442
Podział pracy	444
BGP jako SDN	444
Fibbing	446
I2RS	449
Protokół PCEP	453
Protokół OpenFlow	455
Twierdzenie CAP i pomocniczość	457
Końcowe rozważania dotyczące scentralizowanych płaszczyzn sterowania	460
Dalsza lektura	461
Pytania kontrolne	462
Rozdział 19. Domeny awarii i ukrywanie informacji	463
Przestrzeń problemu	464
Definiowanie zakresu stanu płaszczyzny sterowania	464
Pętle dodatniego sprzężenia zwrotnego	465
Przestrzeń rozwiązań	468
Sumaryzacja informacji o topologii	469
Agregowanie informacji o osiągalności	470
Filtrowanie informacji o osiągalności	473
Uwarstwienie płaszczyzn sterowania	474
Buforowanie	475
Spowalnianie	479
Końcowe rozważania dotyczące ukrywania informacji	481
Dalsza lektura	481
Pytania kontrolne	482
Rozdział 20. Przykłady ukrywania informacji	483
Sumaryzacja informacji o topologii	484
Protokół IS-IS	484
Protokół OSPF	489
Agregacja	495
Uwarstwienie	496
Protokół BGP jako nakładka osiągalności	496
Routing segmentowy z nakładką kontrolera	498
Zmniejszenie prędkości stanu	500
Exponential backoff	500
Redukcja zalewania stanem łącza	503
Końcowe rozważania dotyczące domen awarii	505
Dalsza lektura	505
Pytania kontrolne	507

Część III. Projektowanie sieci	509
Trzy podstawowe modele	510
<i>Prawo nieszczelnych abstrakcji</i>	<i>510</i>
<i>Triada stan-optimalizacja-powierzchnia (SOS)</i>	<i>510</i>
<i>Triada spójność-dostępność-odporność na partycjonowanie (CAP)</i>	<i>511</i>
Rozdział 21. Kwestie bezpieczeństwa w szerszym ujęciu	513
Zakres problemu	514
<i>Zagadnienie tożsamości biometrycznej</i>	<i>514</i>
<i>Definicje</i>	<i>516</i>
<i>Przestrzeń problemów</i>	<i>517</i>
Przestrzeń rozwiązań	517
<i>Obrona w głąb</i>	<i>517</i>
<i>Kontrola dostępu</i>	<i>518</i>
<i>Ochrona danych</i>	<i>519</i>
<i>Gwarantowanie dostępności usług</i>	<i>523</i>
Pętla OODA jako model bezpieczeństwa	532
<i>Obserwuj</i>	<i>533</i>
<i>Zorientuj się</i>	<i>533</i>
<i>Zdecyduj</i>	<i>534</i>
<i>Działaj</i>	<i>535</i>
Końcowe rozważania dotyczące kwestii bezpieczeństwa	535
Dalsza lektura	536
Pytania kontrolne	538
Rozdział 22. Wzorce projektowania sieci	539
Przestrzeń problemu	540
<i>Rozwiązywanie problemów biznesowych</i>	<i>540</i>
<i>Przekładanie wymagań biznesowych na techniczne</i>	<i>544</i>
<i>Co to jest dobry projekt sieci?</i>	<i>546</i>
Projektowanie hierarchiczne	547
Powszechne topologie	549
<i>Topologie pierścienia</i>	<i>550</i>
<i>Topologie siatki</i>	<i>553</i>
<i>Topologie gwiazdy</i>	<i>555</i>
<i>Topologie planarne, nieplanarne i regularne</i>	<i>556</i>
Końcowe rozważania dotyczące wzorców projektowania sieci	558
Dalsza lektura	558
Pytania kontrolne	558
Rozdział 23. Redundancja i odporność	559
Przestrzeń problemu: jak aplikacje postrzegają awarie	560
Definiowanie odporności	561
<i>Inne „wskaźniki”</i>	<i>563</i>

Redundancja jako narzędzie do tworzenia odporności	563
<i>Grupy łączy współdzielonego ryzyka</i>	565
<i>Aktualizacja oprogramowania w trakcie działania i płynny restart</i>	566
<i>Rdzenie dwupłaszczyznowe i wielopłaszczyznowe</i>	566
Modułowość i odporność	567
Końcowe rozważania dotyczące odporności	569
Dalsza lektura	569
Pytania kontrolne	570
Rozdział 24. Rozwiązywanie problemów	571
Co jest celem?	572
Czym są komponenty?	573
Modele i rozwiązywanie problemów	574
<i>Budowanie modeli „jak”</i>	575
<i>Budowanie modeli „co”</i>	576
<i>Budowanie dokładnych modeli</i>	578
<i>Przełączanie się między modelami</i>	579
Podziel na pół i idź dalej	581
<i>Manipulowanie</i>	583
<i>Upraszczenie przed testowaniem</i>	585
Usuwanie problemu	585
Końcowe rozważania dotyczące rozwiązywania problemów	586
Dalsza lektura	587
Pytania kontrolne	588
Część IV. Aktualne tematy	589
Rozdział 25. Dezagregacja, hiperkonwergencja i zmieniająca się sieć	591
Zmiany w zasobach obliczeniowych i aplikacjach	592
<i>Konwergentne, zdezagregowane, hiperkonwergentne i kompozycyjne</i>	592
<i>Zwirtualizowane i zdezagregowane aplikacje</i>	595
Wpływ na projektowanie sieci	597
<i>Wzrost ruchu na linii wschód – zachód</i>	597
<i>Wzrost odchylen i opóźnień</i>	599
Sieci szkieletowe z przełączaniem pakietów	599
<i>Szczególne własności sieci szkieletowej</i>	599
<i>Gałęzie i liście</i>	603
<i>Inżynieria ruchu w sieci gałęzi i liści</i>	606
<i>Sieć gałęzi i liści o większej skali</i>	607
Dezagregacja w sieciach	607
Końcowe rozważania dotyczące dezagregacji	611
Dalsza lektura	612
Pytania kontrolne	613

Rozdział 26. Powody automatyzacji sieci	615
Koncepcje automatyzacji	617
Nowoczesne metody automatyzacji	620
NETCONF	620
RESTCONF	623
Automatyzacja z wykorzystaniem interfejsów programowalnych	624
Automatyzacja na poziomie urządzenia	627
Automatyzacja sieci z wykorzystaniem narzędzi automatyzacji infrastruktury	628
Kontrolery sieciowe i automatyzacja	629
Automatyzacja sieci na potrzeby wdrażania	629
Końcowe rozważania dotyczące przyszłości automatyzacji sieci: od zautomatyzowanej do automatycznej	630
Dalsza lektura	630
Pytania kontrolne	632
Rozdział 27. Zwirtualizowane funkcje sieciowe	633
Elastyczność w projektowaniu sieci	635
Tworzenie łańcucha usług	637
Skalowanie horyzontalne	642
Zmniejszenie czasu obsługi dzięki automatyzacji	643
Scentralizowane zarządzanie konfiguracjami	643
Sieć oparta na intencjach	645
Korzyści	646
Architektura i korzyści obliczeniowe	646
Poprawianie przepustowości VNF	647
Rozważanie kompromisów	647
Stan	647
Optymalizacja	648
Powierzchnia	648
Inne kompromisy do rozważenia	649
Końcowe rozważania	649
Dalsza lektura	649
Pytania kontrolne	651
Rozdział 28. Koncepcje i wyzwania przetwarzania w chmurze	653
Biznesowe powody korzystania z chmur publicznych	655
Od wydatków inwestycyjnych do operacyjnych	656
Czas wprowadzenia na rynek i zwinność biznesowa	656
Nietechniczne kompromisy związane z chmurami publicznymi	657
Kompromisy operacyjne	657
Kompromisy biznesowe	660
Techniczne wyzwania tworzenia sieci w chmurze	661
Opóźnienie	661
Wypełnianie zdalnej przestrzeni dyskowej	663

Ciężar danych	664
Wybór spośród wielu ścieżek do chmury publicznej	664
Bezpieczeństwo w chmurze	665
Ochrona danych przy transporcie przez sieć publiczną	665
Zarządzanie bezpiecznymi połączeniami	666
Chmura z wieloma podmiotami	667
Kontrola dostępu oparta na rolach	667
Monitorowanie sieci w chmurze	668
Końcowe rozważania	668
Dalsza lektura	669
Pytania kontrolne	669
Rozdział 29. Internet rzeczy	671
Wprowadzenie do internetu rzeczy	672
Bezpieczeństwo internetu rzeczy	673
Zabezpieczanie niezabezpieczalnych urządzeń poprzez izolację	674
IoT nie stanowi nowych wyzwań dla bezpieczeństwa	678
Łączność w internecie rzeczy	678
Bluetooth Low Energy (BLE)	678
LoRaWAN	680
IPv6 dla IoT	681
Dane w internecie rzeczy	683
Końcowe rozważania dotyczące internetu rzeczy	684
Dalsza lektura	685
Pytania kontrolne	686
Rozdział 30. Patrząc w przyszłość	687
Rozpowszechniona otwarta automatyzacja	688
Języki modelowania i modele	689
Krótkie wprowadzenie do YANG	689
Patrząc w przyszłość w stronę wszechobecnej automatyzacji	691
Sieci hiperkonwergentne	691
Sieć oparta na intencjach	692
Uczenie maszynowe i wąska sztuczna inteligencja	694
Sieci nazwanych danych i łańcuchy bloków	696
Działanie sieci nazwanych danych	697
Łańcuchy bloków	700
Przekształcenia Internetu	702
Końcowe rozważania dotyczące przyszłości inżynierii sieci	704
Dalsza lektura	704
Pytania kontrolne	705
Skorowidz	707

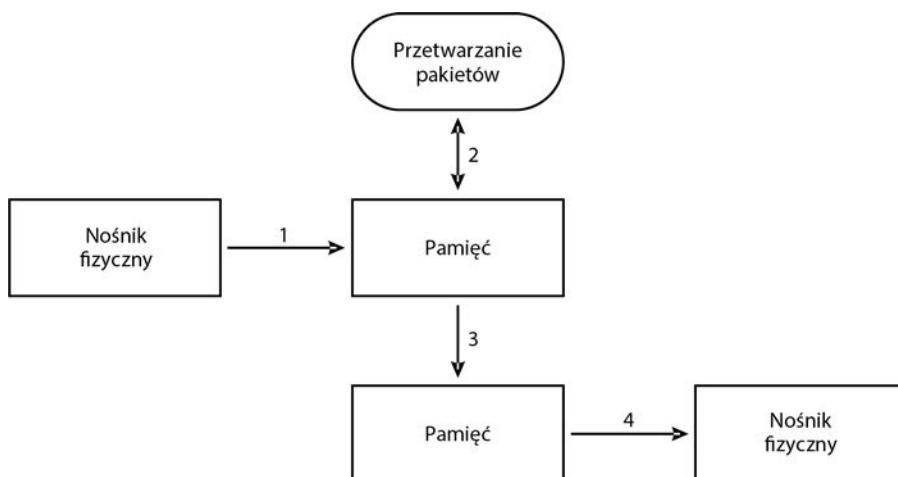
Przełączanie pakietów

Cele nauki

Po przeczytaniu tego rozdziału będziesz rozumieć:

- cztery kroki wymagane do przełączenia pakietu przez urządzenie sieciowe;
- w jaki sposób w procesie przekazywania pakietu są używane bufora cykliczne odbiorcze i nadawcze;
- podstawowy proces przełączania pakietu, w tym sposób tworzenia tabel przekazywania;
- czym routing różni się od przełączania i jakie są zalety routingu;
- pojęcie wielu ścieżek o równym koszcie;
- pojęcie agregacji łączy;
- pojęcia magistrali i krzyżowego pola komutacyjnego.

Urządzenia sieciowe są umieszczane w sieci w celu rozwiązania szeregu problemów, w tym problemu z łączeniem różnych rodzajów mediów i problemów ze skalowaniem sieci, poprzez przenoszenie pakietów tylko tam, gdzie muszą one się dostać. Routery i przełączniki są jednak złożonymi urządzeniami. Inżynierowie całą swoją karierę mogą oprzeć na specjalizacji w rozwiązywaniu tylko niewielkiego zestawu problemów napotkanych podczas przenoszenia pakietów za pośrednictwem urządzenia sieciowego. Rysunek 7.1 pokazuje przegląd zakresu tej problematyki.



Rysunek 7.1. Przenoszenie pakietu przez urządzenie sieciowe

Na rysunku 7.1 przedstawiono cztery wyraźne etapy:

1. Pakiet musi zostać skopiowany z medium fizycznego do pamięci w urządzeniu; czasami jest to nazywane odbieraniem pakietu z łącza.
2. Pakiet musi zostać przetworzony, co zwykle oznacza określenie właściwego interfejsu wyjściowego i niezbędną modyfikację pakietu. Na przykład w routerze nagłówek niższej warstwy jest usuwany i zastępowany nowym, w stanowym filtrze pakietów pakiet może zostać odrzucony na podstawie stanu wewnętrznego itp.
3. Pakiet musi zostać skopiowany z interfejsu wejściowego do interfejsu wyjściowego. Często wiąże się to z podróżą po wewnętrznych układach lub magistrali. Niektóre systemy pomijają ten krok, korzystając z pojedynczej puli pamięci dla interfejsów wejściowych i wyjściowej. Nazywane są one systemami pamięci współdzielonej (to charakterystyczne dla inżynierii sieciowej, że nazwy są albo zbyt sprytne, albo zbyt oczywiste).
4. Pakiet musi zostać skopiowany z powrotem na łącze wyjściowe; czasami jest to nazywane nadawaniem pakietu na łącze.

Uwaga

Mniejsze systemy, szczególnie te, które koncentrują się na szybkim, spójnym przełączaniu pakietów, często wykorzystują pamięć współdzieloną do przesyłania pakietów z jednego interfejsu do drugiego. Czas wymagany do skopiowania pakietu w pamięci jest często dłuższy, niż mogłoby to wynikać z prędkości interfejsów. Systemy pamięci współdzielonej unikają tego kopiowania pakietów wewnątrz pamięci.

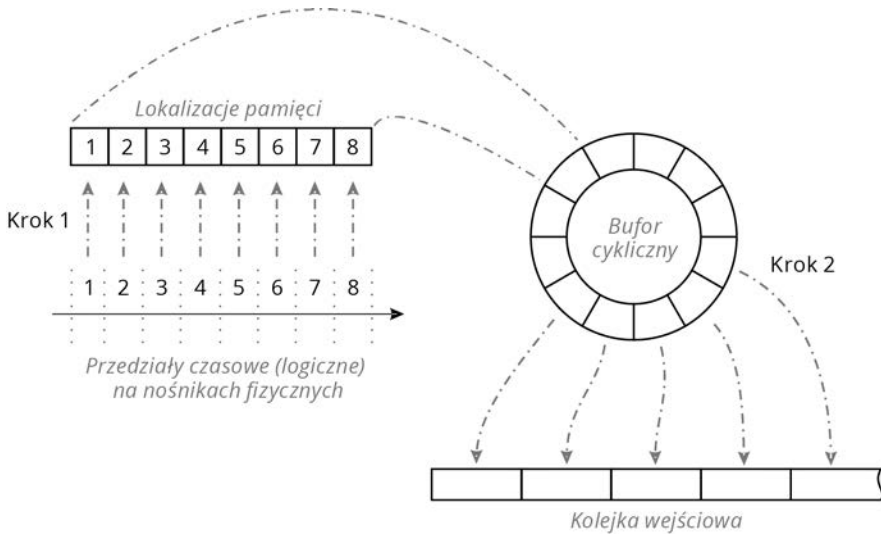
Problem omawiany w następnych podrozdziałach to:

W jaki sposób pakiety, które muszą być przekazane dalej przez urządzenie sieciowe, są przenoszone z przychodzącego do wychodzącego medium fizycznego, a także w jaki sposób te pakiety są przetwarzane po drodze?

Każdy z poniższych podrozdziałów omawia jedną część rozwiązania tego problemu.

Z medium fizycznego do pamięci

Pierwszym krokiem w przetwarzaniu pakietu przez urządzenie sieciowe jest skopiowanie tego pakietu z łącza do pamięci. Rysunek 7.2 pokazuje ten proces.



Rysunek 7.2. Kopiowanie pakietu do pamięci

Na rysunku 7.2 przedstawiono dwa kroki:

- **Krok 1.** Chipset medium fizycznego (*układ PHY*) kopiuje z medium do lokalizacji w pamięci każdy okres sygnalizacji, który reprezentuje pojedynczy znak danych. Ta lokalizacja w pamięci mieści się w cyklicznym buforze odbiorczym, który jest zbiorem komórek pamięci, zarezerwowanych wyłącznie na potrzeby odbierania pakietów z łącza (jest buforem pakietów). Bufor cykliczny wraz z całą pamięcią bufora pakietów jest zwykle umieszczony fizycznie w jednym rodzaju pamięci, który jest dostępny (współdzielony) dla wszystkich interfejsów wejściowych danego urządzenia.

Uwaga

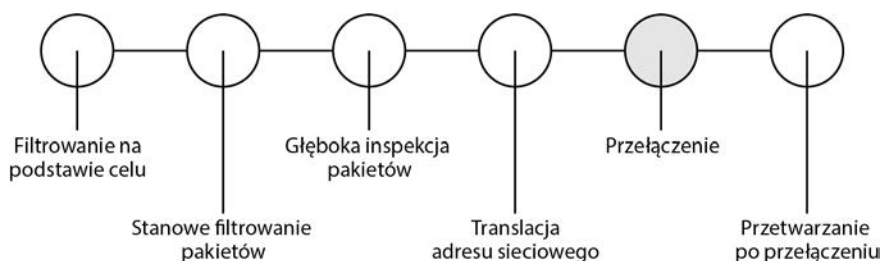
Bufor cykliczny bazuje na pojedynczym wskaźniku, który jest zwiększany za każdym razem, gdy nowy pakiet trafia do bufora. W przykładzie pokazanym na rysunku 7.2 wskaźnik zacząłby w przedziale 1 i zwiększałby się w kolejnych przedziałach, gdy pakiety są kopiowane do bufora. Jeśli wskaźnik dojdzie do gniazda 7 i pojawi się nowy pakiet, zostanie on skopiowany do przedziału 1 — niezależnie od tego, czy zawartość tego przedziału została już przetworzona, czy nie.

W przełączaniu pakietów najbardziej czasochłonnym i energochłonnym zadaniem jest kopiowanie pakietów z jednej lokalizacji do drugiej. Wskaźniki pozwalają tego uniknąć. Zamiast przenosić pakiet przez pamięć, w ścieżce przełączania przekazywany jest z procesu do procesu tylko wskaźnik do lokalizacji w pamięci.

- **Krok 2.** Po skopiowaniu pakietu do pamięci zostaje przerwana praca jakiegoś lokalnego procesora. Podczas tego przerwania lokalny procesor usuwa wskaźnik do bufora odbiorczego, który zawiera pakiet z bufora cyklicznego, i umieszcza wskaźnik do pustego bufora w buforze cyklicznym. Wskaźnik jest umieszczany na osobnej liście, nazywanej kolejką wejściową.

Przetwarzanie pakietu

Gdy pakiet znajduje się w kolejce wejściowej, można go przetworzyć. Przetwarzanie można postrzegać jako łańcuch zdarzeń, a nie jako pojedyncze wydarzenie. Pokazuje to rysunek 7.3.



Rysunek 7.3. Proces przełączania pakietów

Część przetwarzania musi odbywać się przed przełączeniem pakietu. Jest to na przykład translacja adresów sieciowych (NAT), ponieważ zmienia ona niektóre informacje o pakiecie, używane w samym procesie przełączania. Inne przetwarzanie może mieć miejsce po przełączeniu pakietu.

Przełączanie

Przełączanie pakietu jest dość prostą operacją:

1. Proces przełączania wyszukuje docelowy adres MAC lub docelowe urządzenie fizyczne w tabeli przesyłania (w przełącznikach jest to czasami nazywane tabelą nauki mostów lub po prostu tabelą mostów).
2. Na podstawie informacji w tej tabeli jest określany interfejs wychodzący.
3. Pakiet jest przenoszony z kolejki wejściowej do kolejki wyjściowej.

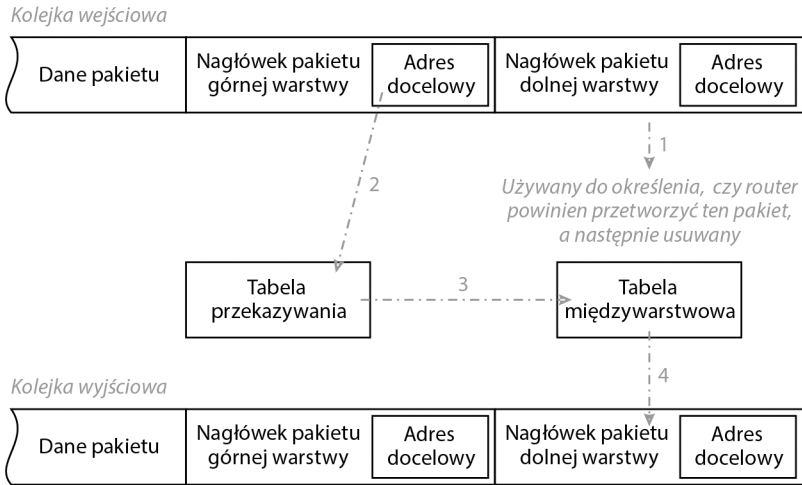
Pakiet nie jest w żaden sposób modyfikowany podczas procesu przełączania. Jest kopiowany z kolejki wejściowej do wyjściowej.

Uwaga

W jaki sposób jest budowana tabela przekazywania? Za pomocą płaszczyzny sterowania. Część II tej książki omawia szczegółowo płaszczyznę sterowania.

Routing

Routing jest procesem bardziej złożonym niż przełączanie. Pokazuje to rysunek 7.4.



Rysunek 7.4. Routing pakietu

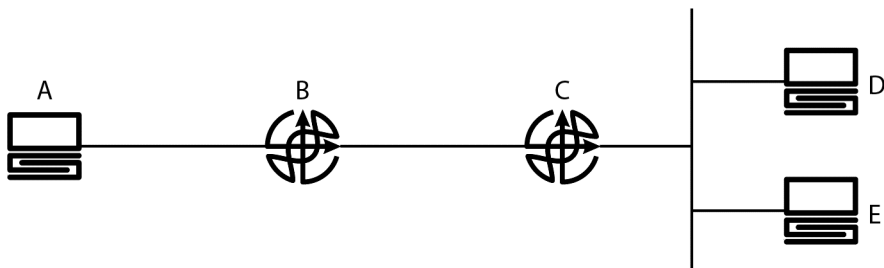
Na rysunku 7.4 pakiet zaczyna od kolejki wejściowej. Następnie procesor przełączający:

1. Usuwa (lub ignoruje) nagłówek niższej warstwy (na przykład ramkę Ethernet obejmującą pakiet). Ta informacja służy do określenia, czy router ma odbierać pakiet, ale nie jest używana podczas samego procesu przełączania.
2. Wyszukuje adres docelowy (i potencjalnie inne informacje) w tabeli przekazywania. Tabela przekazywania określa następny przeskok pakietu w kierunku miejsca docelowego. Następny przeskok może być albo następnym routerem na drodze do celu, albo samym celem.
3. Procesor przełączający analizuje następnie tabelę odnajdowania międzywarstwowego (taką jak te omawiane w rozdziale 6. „Odnajdowanie międzywarstwowe”), aby określić poprawny adres fizyczny, na który należy wysłać pakiet, żeby znalazł się o jeden przeskok bliżej miejsca docelowego.
4. Jest budowany i kopiowany na pakiet nowy nagłówek niższej warstwy, który używa tego nowego docelowego adresu niższej warstwy. Zwykle adres docelowy niższej warstwy jest buforowany lokalnie, wraz z całym nagłówkiem niższej warstwy. Cały nagłówek jest przepisywany w procesie nazywanym przepisywaniem nagłówka MAC.

Pakiet jest następnie w całości przenoszony z kolejki wejściowej do kolejki wyjściowej.

Po co routować?

Skoro routing jest procesem bardziej złożonym niż przełączanie, to po co go wykonywać? Można to zobaczyć na rysunku 7.5.



Rysunek 7.5. Po co routować?

Istnieją co najmniej trzy konkretne powody, aby w sieci raczej routować niż przełączać. Korzystając z sieci na rysunku 7.5 jako przykładu:

- Routing umożliwi A i D komunikowanie się bez obawy o różnice w typach łączy, jeśli łączy [B, C] jest innym rodzajem medium fizycznego niż dwa łącza do tych hostów — z odmiennymi kodowaniami, nagłówkami, adresowaniem itp. W sieci opartej tylko na przełączaniu można to obejść za pomocą translacji nagłówków, ale taka translacja nie wymaga wcale mniej pracy niż routing, tak więc nie ma sensu *rezygnować* z routingu przy rozwiązywaniu tego problemu. Innym rozwiązaniem mogłoby być uzgodnienie jednej adresacji i formatu pakietu dla wszystkich rodzajów mediów fizycznych, ale biorąc pod uwagę stały rozwój mediów i to, jak wiele ich jest, wydaje się ono mało prawdopodobne.
- Gdyby cała sieć była przełączana, B musiałby znać pełną informację o osiągalności dla D i E. W szczególności: D i E musiałyby znać adresy fizyczne lub adresy niższej warstwy dla każdego urządzenia podłączonego do segmentu za C. Może to nie być dużym problemem w mniejszej sieci, ale w większych sieciach, z setkami tysięcy węzłów, lub w globalnym Internecie to rozwiązanie byłoby nieskalowalne — byłoby po prostu zbyt wiele stanu do zarządzania. Możliwe jest łączenie informacji o osiągalności z adresowaniem niższych warstw, ale jest to trudniejsze niż użycie adresu wyższej warstwy, przypisanego na podstawie punktu przyłączenia urządzenia w topologii sieci, zamiast adresu, który jednoznacznie identyfikuje chipset interfejsu i który jest przydzielany na etapie produkcji tego chipsetu.
- Jeśli D wyśle rozgłoszenie do „wszystkich urządzeń w segmencie”, to A odbierze tę transmisję, gdy B i C są przełącznikami, ale nie wtedy, gdy B i C są routerami. Nie można wyeliminować pakietów rozgłoszeniowych, ponieważ są one istotną częścią prawie każdego protokołu transportowego, ale w sieciach wyłącznie przełączanych rozgłoszenia stanowią bardzo trudny do rozwiązania problem skalowania. Routery blokują (czy raczej konsumują) rozgłoszenia.

Uwaga

W świecie sieci komercyjnych terminy *routing* i *przełączanie* są często używane zamiennie. Powodem tego jest przede wszystkim historia marketingu: *routing* pierwotnie zawsze oznaczał „przełączanie programowe”, natomiast *przełączanie* zawsze oznaczało „przełączanie sprzętowe”. Gdy stały się dostępne urządzenia przełączające pakiety, które były zdolne do sprzętowego przepisywania nagłówka MAC, nazwano je „przełącznikami warstwy 3”, co ostatecznie zostało skrócone po prostu do *przełączników*. Większość „przełączników” w centrach danych to zwykle routery, ponieważ wykonują przepisywanie nagłówka MAC w przekazywanych pakietach. Jeśli ktoś nazywa jakieś urządzenie przełącznikiem, najlepiej wyjaśnić, czy jest to przełącznik warstwy 3 (czyli prawidłowo: router), czy przełącznik warstwy 2 (prawdziwy przełącznik).

Uwaga

Terminy *łączy* i *połączenie* są tu używane zamiennie. Łączy to fizyczne lub wirtualne połączenie przewodowe lub bezprzewodowe między dwoma urządzeniami.

Wiele ścieżek o równym koszcie

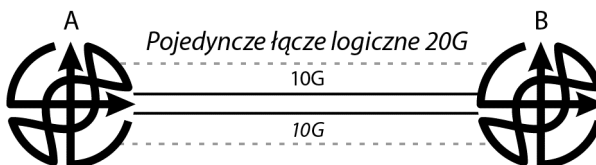
W niektórych projektach sieci inżynierowie wprowadzają równoległe połączenia między dwoma węzłami. Jeśli założymy, że te równoległe łączy są identyczne co do szerokości pasma, opóźnienia i tak dalej, to mają one taki sam koszt. W tym scenariuszu mówimy o wielu ścieżkach o równym koszcie (ang. *Equal Cost Multipath* — ECMP).

W takim przypadku w sieciach produkcyjnych spotykane są zwykle dwa warianty postępowania. Oba zachowują się podobnie, ale różnią się pod względem sposobu grupowania i zarządzania łączy przez system operacyjny.

Agregacja łączy

Mechanizmy agregacji łączy zbierają wiele łączy fizycznych i grupują je w jedno łączy wirtualne. Na potrzeby protokołów routingu i algorytmów zapobiegania pętli, takich jak drzewo rozpinające, łączy wirtualne jest traktowane tak, jakby było pojedynczym łączy fizycznym.

Agregacja łączy jest używana do zwiększenia przepustowości między węzłami sieci, bez konieczności zastępowania wolniejszych łączy fizycznych szybszymi. Na przykład dwa łączy o przepustowości 10 Gb/s mogą zostać połączone w jedno łączy 20 Gb/s, podwajając potencjalną przepustowość między dwoma węzłami, jak pokazano na rysunku 7.6. Słowo „potencjalną” zostało użyte celowo, ponieważ łączy zagregowane w praktyce nie skalują się liniowo.



Rysunek 7.6. Agregacja łączy

Problem, z którym mierzy się agregacja łączy, to określenie, które pakiety należy wysłać przez które z łączy członkowskich (w zagregowanym łączy). Na pierwszy rzut oka może to nie wyglądać problematycznie. Wystarczyłoby przecież używać zagregowanych łączy według algorytmu karuzelowego (ang. *round-robin*). Początkowa ramka zostałaby wysłana przez pierwsze łącze, kolejna przez drugie i tak dalej, aż w końcu znowu przez pierwsze. W ten sposób łącze powinno być idealnie równo obciążone, a przepustowość powinna się skalować linearnie.

Istnieje jednak bardzo niewiele rzeczywistych wdrożeń agregacji łączy, które używałyby algorytmu *round-robin*, a to z powodu ryzyka dostarczania pakietów w złej kolejności. Przyjmijmy, że ramka ethernetowa numer 1 jest wysyłana łączy pierwszym, a zaraz po niej ramka 2 łączy drugim. Z jakiegoś powodu ramka 2 dociera do celu przed ramką 1. Pakiety w tych ramach zostaną dostarczone do hosta odbierającego w złej kolejności — drugi przed pierwszym. A to stanowi problem, ponieważ host jest w takiej sytuacji obciążany obowiązkiem przeorganizowania pakietów w taki sposób, aby poprawnie odtworzyć cały datagram.

Dlatego też większość dostawców wdraża funkcje skrótu dla przepływów, co ma zapewnić, że cały dany przepływ pakietów używa tego samego łączy fizycznego. Dzięki temu nie ma ryzyka odbioru pakietów w miejscu docelowym w złej kolejności — są one wysyłane sekwencyjnie przez to samo łącze.

Funkcje skrótu

Skrót (ang. *hash*) to prosty koncept, który jest jednak dość trudny do wdrożenia w użyteczny sposób. Funkcja skrótu przyjmuje dowolnej długości ciąg liczb i zwraca numer o stałej długości (skrót), który (mniej więcej) unikatowo odzwierciedla oryginalny ciąg. Jest w tym część prosta do wdrożenia: przykładem dość banalnej funkcji skrótu byłoby zwykłe dodawanie liczb w zbiorze tak długo, aż uzyska się wynik jednocyfrowy, który nazwie się skrótem. Na przykład:

23523

$2 + 3 + 5 + 2 + 3 == 15$

$1 + 5 == 6$

Tak więc liczba 23523 może zostać *przedstawiona* jako 6. Ciekawą właściwością skrótu jest to, że nie istnieje sposób ustalenia na podstawie skrótu, co było oryginalną liczbą. To jedno z kluczowych spostrzeżeń dotyczących wielu użyci funkcji skrótu. Jeśli podam komuś jakąś liczbę, a ta osoba poda ją Tobie, to możesz mnie spytać o skrót tej liczby (bez mówienia mi, jaka to liczba), a następnie zweryfikować, czy to ta sama liczba, obliczając skrót i porównując go z uzyskanym ode mnie.

Opisany tu skrót jest banalny, ponieważ w zbyt łatwy sposób można w nim uzyskać *kolizję*. Innymi słowy: jest wiele różnych liczb, które w tych samych obliczeniach uzyskują skrót równy 6, na przykład 222, 33, 111111 i (prawdopodobnie) nieskończenie wiele innych. W niektórych sytuacjach kolizje są skrajnie niepożądane. Na przykład, jeśli chcesz przechowywać pary liczb, w których będziesz znajdować drugą z liczb przez wyszukanie pierwszej (kwestia indeksowania), chcesz zminimalizować ryzyko kolizji. Nie chcesz, aby skrót obliczony dla numeru, którego używasz jako indeksu, odsyłał Cię do zbioru skrótów z wieloma wpisami, ponieważ w takiej sytuacji trzeba jeszcze przeszukać każdy skrót w tym zbiorze, aby odnaleźć indeks. W ekstremalnej sytuacji każdy numer może być indeksowany jako zbiór skrótów, co uczyni funkcję skrótu całkiem nieefektywną przy wyszukiwaniu.

W innych przypadkach, takich jak opisywane tutaj równoważenie obciążenia, istotne jest, aby skrót możliwie równomiernie rozkładał wpisy pomiędzy zbiorami. Chcesz upewnić się, że każdy zbiór zawiera mniej więcej tyle samo wpisów, ponieważ każdy zbiór oznacza pojedyncze łącze, a łącza powinny obsługiwać ruch do mniej więcej równej liczby celów.

Funkcje skrótu dla przepływów bazują na działaniach matematycznych na przynajmniej dwóch statycznych komponentach przepływów, takich jak adresy MAC źródła i celu, adresy IP źródła i celu czy też numery portów TCP lub UDP. Z tych komponentów wyliczane jest łącze fizyczne, którego użyje przepływ. Ponieważ charakterystyki przepływu są statyczne, algorytm skrótu daje identyczne wyniki obliczeń dla każdej ramki lub pakietu w przepływie ruchu, gwarantując, że to samo łącze będzie używane przez cały czas przepływu.

Obliczanie skrótu dla przepływu rozwiązuje problem złej kolejności pakietów, wprowadza jednak nowy problem. Nie wszystkie przepływy są tej samej wielkości. Niektóre przepływy wykorzystują dużą szerokość pasma, na przykład do transferu plików, tworzenia kopii zapasowych lub przechowywania. Są one czasami nazywane **przepływami słoniowatymi** (ang. *elephant flows*). Inne przepływy są dość małe, takie jak te używane do ładowania strony internetowej lub komunikacji głosowej przez IP; te nazywa się czasami **przepływami myszowatymi** (ang. *mouse flows*). Ponieważ przepływy mają różne rozmiary, niektóre łącza członkowskie mogą być mocno obciążone, podczas gdy inne są niedostatecznie wykorzystywane.

To niedopasowanie wykorzystania przenosi nas z powrotem do punktu dotyczącego skalowania liniowego. Jeśli ramki byłyby równomiernie rozłożone na zagregowanym łączu, to dodanie nowych łączy członkowskich równomiernie zwiększałoby przepustowość. Ale połączenie algorytmów skrótu z nieprzewidywalną ilością przepływów ruchu oznacza, że łącza fizyczne nie będą równomiernie obciążane.

Zadaniem inżyniera sieci jest zrozumienie rodzaju ruchu przepływającego przez zagregowane łącze i dobranie takiego algorytmu skrótu, który zapewni najbardziej równomierny rozkład obciążenia. Można przy tym rozważyć przykładowo:

- Czy wiele hostów w tej samej domenie rozgłoszeniowej komunikuje się ze sobą poprzez zagregowane łącze? Możliwym rozwiązaniem jest obliczanie skrótu na podstawie adresów MAC z nagłówka ramki Ethernet, ponieważ te adresy MAC będą się różnić.
- Czy poprzez zagregowane łącze niewielka liczba hostów komunikuje się z pojedynczym serwerem? W tym scenariuszu może nie być wystarczającej różnorodności adresów MAC lub adresów IP. Obliczanie skrótów na podstawie numerów portów TCP lub UDP może dać największą różnorodność, a więc i najlepszą dystrybucję ruchu na łączach członkowskich.

Protokół kontroli agregacji łączy (ang. *Link Aggregation Control Protocol, LACP*)

Podczas grupowania łączy należy wziąć pod uwagę urządzenia sieciowe na każdym końcu łącza i zachować szczególną ostrożność, aby utworzyć zagregowane łącze przy jednoczesnym zachowaniu topologii bez pętli. Najczęstszym sposobem rozwiązania tego problemu jest zastosowanie standardowego protokołu kontroli agregacji łączy (ang. *Link Aggregation Control Protocol* — LACP), opisanego jako standard 802.3ad Instytutu Inżynierów Elektryków i Elektroników (IEEE).

Na wyznaczonych przez inżyniera sieci łączach LACP ogłasza drugiej stronie zamiar utworzenia zagregowanego łącza. Druga strona również uruchamia LACP, a jeśli podane parametry są poprawne, to akceptuje ogłoszenie oraz tworzy łącze. Po utworzeniu pakietu łączy zagregowane łącze zostaje przedstawione w stan przekazywania. Operatorzy sieci mogą następnie wysłać zapytania do LACP o status łącza zagregowanego i o stan elementów fizycznych.

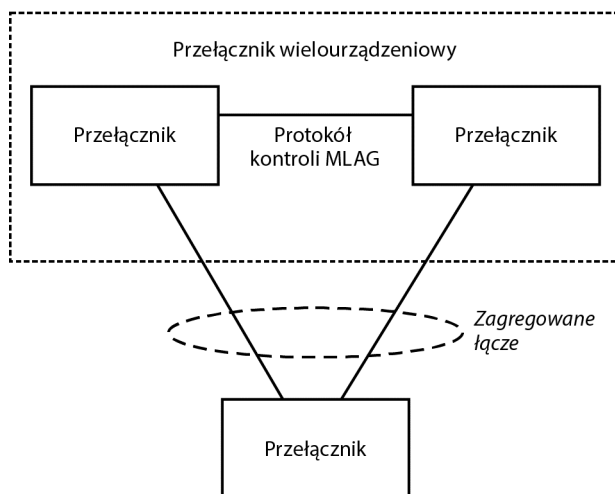
LACP dowiadyuje się również, że jedno z łączy w pakiecie przestaje działać, ponieważ nie przepływają przez nie pakiety kontrolne. Ta możliwość jest przydatna, ponieważ umożliwia procesowi LACP powiadomienie sieciowego systemu operacyjnego o konieczności ponownego obliczenia funkcji skrótu dla przepływów. Bez LACP sieciowemu systemowi operacyjnemu mogłoby zająć więcej czasu odnotowanie uszkodzonego łącza, co powodowałoby kierowanie ruchu do takiego elementu łącza, który nie jest już prawidłową ścieżką.

Istnieją też inne protokoły kontroli agregacji łączy. W niektórych przypadkach możliwe jest ręczne tworzenie pakietów łączy, bez ochrony protokołu kontrolnego. Jednak LACP dominuje jako standard używany do agregacji łączy przez dostawców sieci, a także przez systemy operacyjne hostów i przez dostawców hyperwizorów.

Agregacja łączy z wielu urządzeń

Niektórzy dostawcy sieci oferują też funkcję agregacji łączy z wielu urządzeń (ang. *Multi-chassis Link Aggregation* — MLAG). Pozwala ona na utworzenie pojedynczego, zagregowanego pakietu łączy na dwóch lub więcej przełącznikach sieciowych. Umożliwia to specjalny protokół kontrolny właściwy dla danego dostawcy, który działa pomiędzy przełącznikami należącymi do MLAG. Sprawia on, że wiele przełączników sieciowych może być widocznych dla LACP, protokołu STP i wszelkich innych protokołów tak, jakby były jednym przełącznikiem.

Zwykle stosowanie MLAG uzasadnia się redundancją fizyczną: inżynier sieciowy może wymagać, aby na niższych warstwach (np. Ethernet) urządzenia sąsiadowały ze sobą, a nie były połączone za pomocą routingu. Może też wymagać, aby pakiet łączy pozostawał aktywny, nawet jeśli zdalna strona połączenia ma awarię. Rozłożenie pakietu łączy na przynajmniej dwa przełączniki pozwala spełnić to wymaganie. Pokazuje to rysunek 7.7.

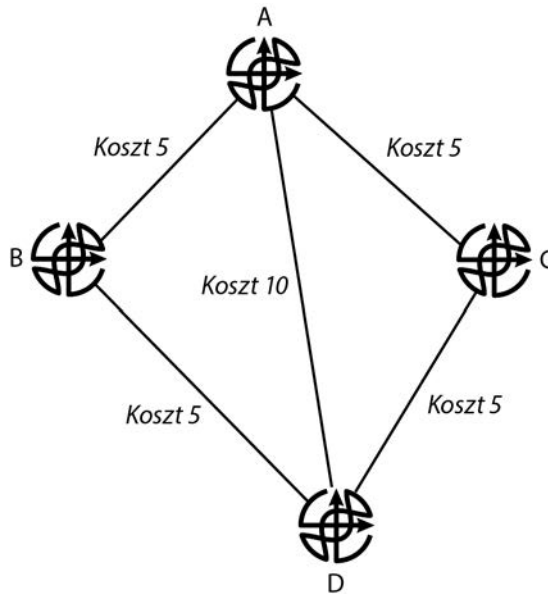


Rysunek 7.7. Agregacja łączy z wielu urządzeń

Wiele sieci produkcyjnych ma jakiś rodzaj MLAG, ale wiele innych odstąpiło od tej technologii, częściowo dlatego, że MLAG jest własnościowy — nie ma czegoś takiego jak międzyplatformowy MLAG. Lepsze projekty sieci odchodzą od szeroko rozproszonych domen przełączanych, w którym to scenariuszu MLAG jest korzystny. Zamiast tego projektowanie sieci zmierza w kierunku ograniczonych domen przełączanych, połączonych ze sobą poprzez routing, co eliminuje potrzebę stosowania technologii MLAG.

Routowane łącza równoległe

Routowane płaszczyzny sterowania, zwane protokołami routingu (w części II tej książki możesz uzyskać więcej informacji na temat routingu i obliczania ścieżek wolnych od pętli), czasami obliczają zbiór wielu ścieżek sieciowych o równym koszcie. W przypadku routingu łącza o tym samym koszcie mogą nawet łączyć więcej niż pojedynczą parę urządzeń. Pokazuje to rysunek 7.8.



Rysunek 7.8. Routowane ECMP

Na rysunku 7.8 są 3 ścieżki:

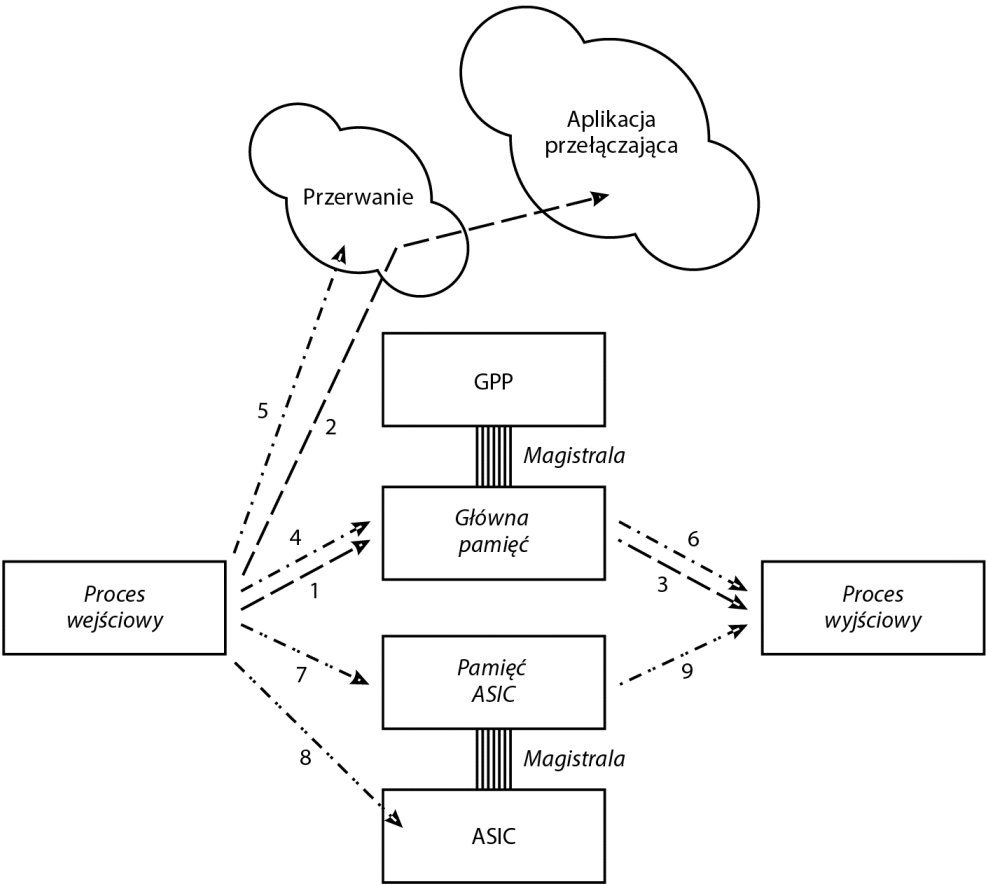
- [A, B, D] o łącznym koszcie 10,
- [A, D] o łącznym koszcie 10,
- [A, C, D] o łącznym koszcie 10.

Ponieważ te trzy ścieżki mają taki sam koszt, wszystkie trzy mogą być zainstalowane w lokalnej tabeli przekazywania w A i D. Na przykład router A może przekazywać ruch w kierunku D przez dowolne z tych trzech łączy. W jaki sposób router decyduje, którą fizyczną ścieżkę wybrać, gdy ma wiele opcji dotarcia do tego samego miejsca docelowego?

Tak samo jak jest w przypadku ECMP na niższych warstwach, odpowiedzią są funkcje skrótu. Obliczanie skrótu dla routowanego ECMP można wykonać w oparciu o wiele elementów. Typowymi są adres IP źródłowy lub docelowy oraz numer źródłowego i docelowego portu. Obliczenie skrótu skutkuje wyborem jednej, stałej ścieżki dla całego przepływu na warstwie trzeciej (L3). Ponowne przeliczenie skrótu i wybranie innej ścieżki jest konieczne tylko w przypadku awarii łącza.

Mechanizm przetwarzania pakietów

Kroki składające się na routowanie pojedynczego pakietu mogą wydawać się bardzo proste: wyszukaj cel w tabeli, stwórz (lub uzyskaj) nowy nagłówek MAC, nadpisz stary nagłówek MAC nowym, umieść pakiet w odpowiedniej kolejce do interfejsu wyjściowego. Może to być proste, ale przetworzenie pojedynczego pakietu zajmuje jednak trochę czasu. Rysunek 7.9 pokazuje trzy różne ścieżki, przez które można przełączyć pakiet w urządzeniu sieciowym.



Rysunek 7.9. Ścieżki przełączania

Rysunek 7.9 pokazuje trzy różne ścieżki przełączania przez urządzenie. Możliwe są też inne ścieżki, ale te trzy są najpopularniejsze. Na pierwszej ścieżce pakiet jest przetwarzany przez oprogramowanie działające na ogólnym procesorze (GPP). Dzieje się to w trzech krokach:

1. Pakiet jest kopiowany z łącza fizycznego do głównej pamięci, jak to opisano w poprzednich podrozdziałach.
2. Procesor sygnałów fizycznych, chipset PHY, wysyła do GPP (jest to prawdopodobnie główny procesor urządzenia sieciowego, choć nie musi tak być) sygnał, nazywany przerwaniem.
 - a. Przerwanie zmusza procesor do zatrzymania innych zadań (stąd jego nazwa) i do uruchomienia małego kawałka kodu, który zaplanuje późniejsze uruchomienie innego procesu: oprogramowania przełączającego.
 - b. Kiedy oprogramowanie przełączające zadziała, wykona właściwe wyszukiwania i dokona właściwych zmian w pakiecie.
3. Po przełączeniu pakiet jest kopiowany z głównej pamięci do procesu wyjściowego, jak to opisano w kolejnych podrozdziałach.

Przełączanie pakietu w ten sposób jest często nazywane przełączaniem procesowym (z oczywistych względów), a czasami — powolną ścieżką. Niezależnie od szybkości GPP uzyskanie pełnej, liniowej szybkości przełączania na szybkich interfejsach wymaga wielu optymalizacji — tak wielu, że jest prawie nieosiągalne. Druga ze ścieżek pokazanych na rysunku 7.9 została zaprojektowana, aby przyspieszyć przetwarzanie pakietów:

4. Pakiet jest kopiowany z łącza fizycznego do głównej pamięci, jak to opisano w poprzednich podrozdziałach.
5. Chipset PHY wysyła przerwanie do GPP. Zamiast wywoływać inny proces, kod odpowiedzialny za przerwanie samodzielnie przetwarza pakiet.
6. Po przełączeniu pakiet jest kopiowany z pamięci głównej do procesu wyjściowego, zgodnie z opisem poniżej.

Ten proces, z równie oczywistych powodów, jest często nazywany przełączaniem w kontekście przerwania. Wiele procesorów jest w stanie przetwarzać pakiety wystarczająco szybko, aby przenosić je w tym trybie pomiędzy interfejsami o niskiej i umiarkowanej prędkości. Sam kod przełączania musi być oczywiście wysoce zoptymalizowany, ponieważ przełączenie pakietu powoduje, że procesor przestaje wykonywać inne zadania (takie jak przetwarzanie aktualizacji protokołu routingu). Ta ścieżka była oryginalnie (a czasami nadal jest) nazywana szybką ścieżką.

W przypadku naprawdę szybkich aplikacji proces przełączania pakietów musi zostać przekazany z głównego procesora lub dowolnego innego GPP do wyspecjalizowanego procesora, zaprojektowanego właśnie do tego konkretnie zadania: przetwarzania pakietów. Czasami te procesory nazywane są procesorami sieciowymi (ang. *Network Processing Unit*, NPU), podobnie jak procesory zaprojektowane do obsługi tylko grafiki nazywa się procesorami graficznymi (ang. *Graphics Processing Unit*, GPU). Te wyspecjalizowane procesory są podzbiorem szerszej klasy procesorów, zwanych specjalizowanymi układami scalonymi (ang. *Application-Specific Integrated Circuits*, ASIC), i często są nazywane przez inżynierów po prostu układami ASIC. Przełączanie pakietu przez ASIC pokazano na rysunku 7.9 jako kroki od 7 do 9:

7. Pakiet jest kopiowany z łącza fizycznego do pamięci układu ASIC, jak to opisano w poprzednich podrozdziałach.
8. Chip PHY przerywa układowi ASIC. Ten obsługuje przerwanie, przełączając pakiet.
9. Po przełączeniu pakiet jest kopiowany z pamięci układu ASIC do procesu wyjściowego, zgodnie z dalszym opisem.

Wiele ASIC wyspecjalizowanych do przetwarzania pakietów ma różne interesujące funkcje, w tym:

- wewnętrzne struktury pamięci (rejstry), skonfigurowane specjalnie do obsługi różnego rodzaju adresów używanych w sieciach;
- zestawy specjalistycznych instrukcji, zaprojektowanych do obsługi różnych wymagań dotyczących przetwarzania pakietów, takich jak sprawdzanie wewnętrznych nagłówek w pakiecie i przepisywanie nagłówka MAC;
- specjalistyczne struktury pamięci i zestawy instrukcji, przeznaczone do przechowywania i wyszukiwania adresów docelowych w celu przyspieszenia przetwarzania pakietów;
- możliwość recyklingu pakietu przez potok w celu wykonania operacji, które nie mogą być obsługiwane w jednym przebiegu, takich jak głębokie inspekcje pakietów lub specjalistyczne zadania filtrowania.

Przez magistralę

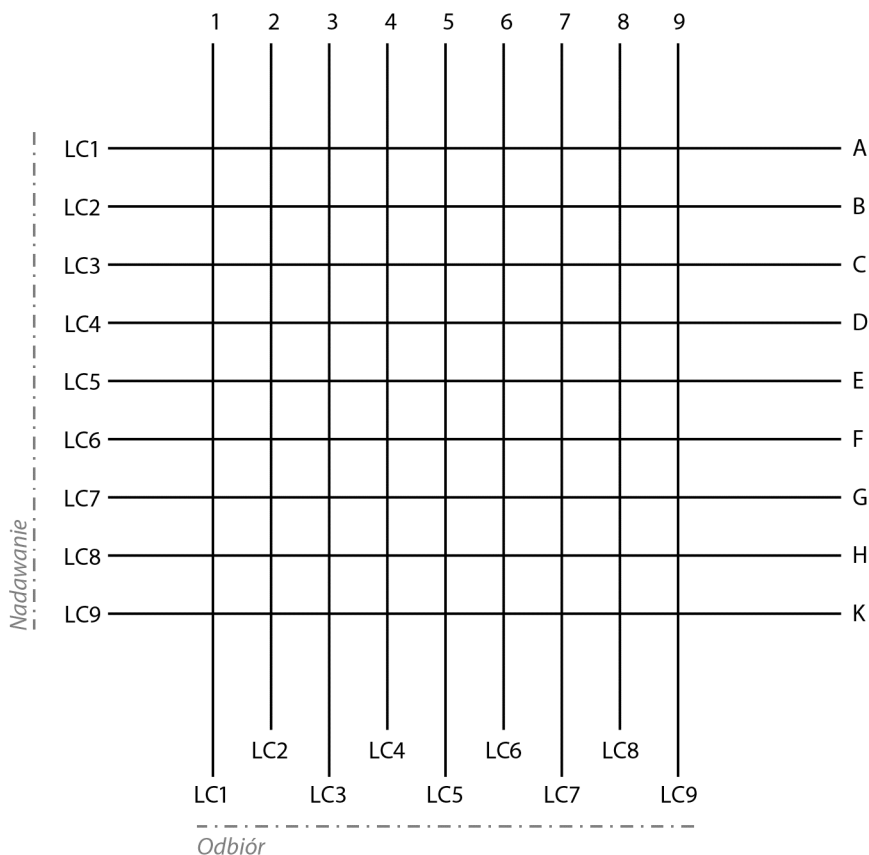
W mniejszych urządzeniach sieciowych, które mają tylko jeden proces sieciowy (opisany wcześniej ASIC lub NPU), przeniesienie pakietu z kolejki wejściowej do kolejki wyjściowej jest proste. Interfejsy wejściowe i wyjściowe dzielą wspólną pulę pamięci pakietów, więc wskaźnik do pakietu może być przeniesiony z jednej kolejki do drugiej.

Aby można było uzyskać większą liczbę portów i zbudować urządzenia o większej skali — w szczególności zestawy wielu urządzeń — musi istnieć wewnętrzna magistrala lub układ, który łączy wejściowe i wyjściowe mechanizmy przetwarzania pakietów. Jednym z popularnych układów, wykorzystywanych do łączenia mechanizmów przetwarzania pakietów w urządzeniu sieciowym, jest krzyżowe pole komutacyjne. Pokazuje je rysunek 7.10.

Rozmiar i struktura krzyżowego pola komutacyjnego są zależne od liczby połączonych portów. Jeśli w przełączniku jest więcej portów, niż może połączyć pojedyncze pole komutacyjne, przełącznik użyje wielu takich pól. Typową topologią dla tego rodzaju układu jest wieloetapowy system Clos, łączący pole komutacyjne wejścia i wyjścia. Możesz pomyśleć o tym jak o polu komutacyjnym dla pól komutacyjnych.

Uwaga

Układy rdzenia i liści, które są rodzajem Clos, są omówione w rozdziale 25. „Dezagregacja, hiperkonwergencja i zmieniająca się sieć”.



Rysunek 7.10. Krzyżowe pole komutacyjne

Krzyżowe pole komutacyjne wymaga do działania znajomości czasu (lub raczej ustalonego przedziału czasu) i harmonogramu pracy. W każdym przedziale czasu jeden port wyjściowy (nadający) jest połączony z jednym portem wejściowym (odbierającym), dzięki czemu w tym okresie nadajnik może przesłać pakiet, ramkę lub zestaw pakietów do odbiornika. Harmonogram „łączy” właściwe punkty w krzyżowym polu komutacyjnym, tak aby transmisja nastąpiła we właściwym przedziale czasowym.

Na przykład:

- karta liniowa 1 (LC1) chce wysłać pakiet do LC3;
- LC3 chce wysłać pakiet do LC5.

W następnym cyklu czasowym harmonogram może połączyć wiersz A z kolumną 1 („wykonać” połączenie na A3) i połączyć wiersz C z kolumną 5 („wykonać” połączenie na C5), aby między tymi parami kart liniowych został ustawiony kanał komunikacyjny.

Krzyżowe pola komutacyjne i rywalizacja

Co się stanie, jeśli dwa nadajniki chcą wysłać pakiet do jednego odbiornika? Na przykład, jeśli podczas jednego okresu zarówno LC1, jak i LC2 będą chciały wysłać pakiet do LC9 przez krzyżowe pole komutacyjne? Mówi się w takim wypadku o rywalizacji i jest to sytuacja, którą musi się zająć program harmonogramu łącza. Który z dwóch portów wejściowych powinien mieć możliwość wysyłania swojego ruchu do portu wyjściowego? I gdzie ma w tym czasie czekać kolejka ruchu przychodzącego?

Jedną opcją jest przechowywanie pakietów w kolejce wejściowej. Przełączniki korzystające z tej techniki nazywane są przełącznikami kolejkowania wejściowego. Tego rodzaju przełączniki są narażone na blokowanie początku linii (ang. *head-of-line* — HOL). Blokowanie początku linii następuje, gdy pakiet na początku kolejki, czekający na przekazanie dalej przez układ, blokuje inne pakiety w kolejce za nim.

Inną możliwością jest, aby przełącznik wykorzystywał wiele wirtualnych kolejek wyjściowych (ang. *virtual output queues* — VOQ) na każdy port wyjściowy.

VOQ dają krzyżowemu polu komutacyjnemu wiele miejsc do przechowywania pakietów wejściowych podczas ich oczekiwania na dostarczenie do portów wyjściowych. W licznych przełącznikach zaprojektowana jest jedna VOQ na każdy port wyjściowy, dla którego przeznaczony jest ruch wyjściowy. Dlatego też port wyjściowy może mieć kilka pakietów ustawionych w kilku różnych VOQ, o ile tylko czekają one w kolejkach do kilku różnych portów wyjściowych.

Każda z tych VOQ może być obsługiwana podczas jednego cyklu zegara. Oznacza to wyeliminowanie blokowania początku linii, ponieważ kilka różnych pakietów z tej samej kolejki wejściowej może być przekazanych przez krzyżowe pole komutacyjne w tym samym czasie. Zamiast pojedynczej kolejki dla portu wejściowego istnieje kilka różnych kolejek. Pomyśl o tym jak o otwarciu dodatkowych kas w sklepie spożywczym.

Nawet przy VOQ pozostaje potencjał do rywalizacji w krzyżowym polu komutacyjnym. Najczęstszym przykładem jest sytuacja, gdy dwa lub więcej pakietów wejściowych musi opuścić przełącznik tym samym portem wyjściowym w tym samym czasie, a dokładniej: w tym samym cyklu zegara. Port wyjściowy może wysłać tylko jeden pakiet na cykl zegara.

Ustaleniem, która kolejka wejściowa będzie mieć pierwszeństwo w dostarczaniu ruchu do portu wyjściowego, zajmuje się algorytm ustawiany przez producenta przełącznika tak, aby zapewnić maksymalne wykorzystanie sprzętu. Jednym z algorytmów planowania używanych przez przełączniki do rozwiązania tego problemu jest iSLIP.

Przegląd algorytmu iSLIP

Algorytm iSLIP rozstrzyga rywalizację w krzyżowym polu komutacyjnym, planując ruch tak, aby urządzenie sieciowe osiągnęło niezablokowaną przepływność. Na potrzeby tego omówienia pomocne będzie przeanalizowanie iSLIP w najprostszej postaci, poprzez przesledzenie, co dzieje się, gdy algorytm iSLIP jest wykonywany jednorazowo.

Podczas wykonywania iSLIP mają miejsce trzy kluczowe zdarzenia:

1. **Żądanie.** Wszystkie punkty wejściowe (wchodzące) do krzyżowego pola komutacyjnego z zakolejkowanym ruchem pytają swoje odpowiednie punkty wyjściowe (wychodzące), czy mogą wysłać.

2. **Zgoda.** Każdy punkt wyjściowy, który otrzymał takie żądanie, musi określić, który punkt wejściowy będzie mógł wysłać. W przypadku pojedynczego żądania zgoda jest udzielana bez dalszego rozważania. Jeśli jednak istnieje wiele żądań, punkt wyjściowy musi określić, który punkt wejściowy może wysłać. Odbywa się to za pośrednictwem algorytmu karuzelowego, który przyznaje jedną zgodę pierwszemu żądaniu, podczas kolejnego wykonania iSLIP przyznaje zgodę kolejnemu żądaniu i tak dalej, w sposób cykliczny. Po podjęciu decyzji dotyczącej tego konkretnego wykonania iSLIP każdy punkt wyjściowy wysyła do odpowiedniego punktu wejściowego komunikat zgody, tym samym sygnalizując pozwolenie na wysyłanie.
3. **Akceptacja.** Punkt wejściowy rozpatruje komunikaty zgody, które otrzymał od punktów wyjściowych, wybierając zgodę za pomocą algorytmu karuzelowego. Po dokonaniu wyboru punkt wejściowy informuje punkt wyjściowy, że zgoda została zaakceptowana. Wtedy i tylko wtedy, gdy punkt wyjściowy zostanie poinformowany o zaakceptowaniu zgody, przejdzie on do następnego żądania. Jeśli nie ma odebranego komunikatu akceptacji, punkt wyjściowy spróbuje obsłużyć poprzednie żądanie podczas kolejnego wykonania iSLIP.

Zrozumienie procesu żądania, zgody i akceptacji daje nam wgląd w to, w jaki sposób pakiety mogą być dostarczane jednocześnie za pośrednictwem krzyżowego pola komutacyjnego bez kolizji. Jeśli jednak zastanowisz się nad złożonym zestawem wejść, VOQ i wyjść, możesz zdać sobie sprawę, że pojedynczy przebieg iSLIP nigdy nie planuje do wysyłki maksymalnej możliwej liczby pakietów.

Z pewnością niektóre wejścia uzyskały dostęp do wyjść i niektóre pakiety można przekazać dalej, ale możliwe jest także, że niektóre wyjścia nigdy nie zostały dopasowane do oczekujących danych wejściowych. Innymi słowy, jeśli ograniczysz iSLIP do pojedynczego wykonania na cykl zegara, zostanie pewna nieużywana przepustowość wyjściowa.

Dlatego normalną praktyką jest uruchamianie iSLIP w wielu iteracjach. Rezultatem jest zmaksymalizowana liczba dopasowanych wejść i wyjść. Można przesłać przez krzyżowe pole komutacyjne więcej pakietów naraz. Ilu przebiegów iSLIP potrzeba, aby zmaksymalizować liczbę pakietów, które mogą być przełączane przez krzyżowe pole komutacyjne w cyklu zegara? Badania sugerują, że w odniesieniu do wzorców ruchu panujących w większości sieci czterokrotny przebieg iSLIP najlepiej dopasowuje wejścia i wyjścia w całym krzyżowym polu komutacyjnym. Wykonanie iSLIP więcej niż cztery razy nie daje istotnie większej liczby dopasowań. Innymi słowy, w większości środowisk sieciowych nie ma nic do zyskania dzięki uruchomieniu iSLIP pięć, sześć lub dziesięć razy.

Wychodząc poza iSLIP

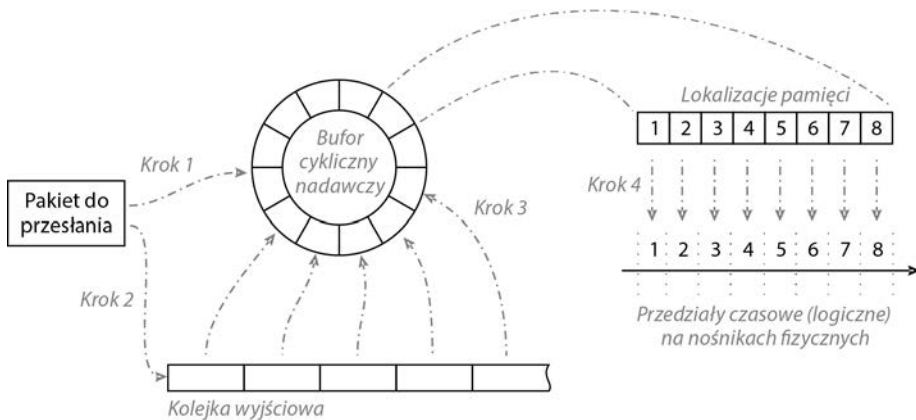
Przyjęliśmy dotychczas założenie, że cały ruch przepływający przez krzyżowe pole komutacyjne jest równie ważny. Jednak w nowoczesnych centrach danych niektóre klasy ruchu mają priorytet nad innymi. Na przykład ramki Fibre Channel przez Ethernet (ang. *Fibre Channel over Ethernet* — FCoE) muszą przechodzić przez łącze bezstratnie, podczas gdy sesja TCP należąca do klasy QoS ruchu śmieciowego nie musi.

Czy iSLIP obsługuje ruch o różnych priorytetach, przyznając niektórym żądaniom pierwszeństwo przed innymi? Tak, ale w zmodyfikowanej formie algorytmu, który widzieliśmy. Istnieją warianty iSLIP: priorytetyzujący, progowy i ważony.

Wychodząc poza iSLIP, używany tutaj jedynie jako wygodny przykład zarządzania rywalizacją, producenci będą pisać własne algorytmy, aby dopasować je do możliwości sprzętowych swoich krzyżowych pól komutacyjnych. W tym podrozdziale omówiono tylko krzyżowe pole komutacyjne z kolejkami wejściowymi, ale wiele łączy oferuje również kolejkowanie wyjściowe po stronie wychodzącej łącza.

Z pamięci do medium fizycznego

Kiedy już pakiet jest przeniesiony poprzez magistralę do wychodzącej karty liniowej lub wskaźnik do pakietu jest przeniesiony z kolejki wejściowej do kolejki wyjściowej, urządzenie sieciowe nadal ma coś do zrobienia. Pokazuje to rysunek 7.11.



Rysunek 7.11. Kopiowanie pakietu z powrotem na łącze

Zwróć uwagę, że bufor cykliczny pokazany na rysunku 7.11 to bufor nadawczy, a nie odbiorczy. Na rysunku 7.11 są cztery kroki:

Krok 1. Pakiet jest przekazywany do strony nadawczej routera w celu przekazania dalej. W zależności od platformy i specyficznych funkcji tutaj może zostać wykonane jakieś przetwarzanie po przełączeniu. Nie jest ono pokazane na tym rysunku. Najpierw zostanie podjęta próba umieszczenia pakietu bezpośrednio w buforze nadawczym, skąd może zostać przesłany. Jeśli bufor zawiera już pakiet lub jest pełny (w zależności od implementacji), pakiet nie zostanie umieszczony w buforze nadawczym. Jeśli pakiet jest umieszczony w buforze nadawczym, krok 2 jest pomijany (co oznacza, że pakiet nie będzie przetwarzany przy użyciu wychodzących zasad *Quality of Service* [QoS]). W przeciwnym razie pakiet zostanie umieszczony w kolejce wyjściowej, gdzie będzie czekał na przeniesienie do bufora nadawczego.

Krok 2. Jeśli pakiet nie może zostać umieszczony w buforze nadawczym, zostanie umieszczony w kolejce wyjściowej i przetrzymany tam przez jakiś czas.

Krok 3. Okresowo kod transmisji przenosi pakiety z kolejki wyjściowej do bufora nadawczego. Kolejność pobierania pakietów z kolejki wyjściowej zależy od konfiguracji QoS. W rozdziale 8. „Jakość usług” znajdziesz więcej informacji na temat zastosowania QoS do kolejek w różnych sytuacjach.

Krok 4. W pewnym momencie po przeniesieniu pakietu do bufora nadawczego układ transmitujący PHY, który odczytuje każdy bit z bufora pakietów, koduje go we właściwym formacie dla typu wychodzącego medium fizycznego i kopiuje pakiet na łącze.

Końcowe rozważania dotyczące przełączania pakietów

Szczegóły przełączania pakietów mogą wydawać się zbyt drobiazgowo. Bo czy ostatecznie ma znaczenie, w jaki sposób pakiet lub ramka przemieszczają się między dwoma urządzeniami? Czy naprawdę tak ważne jest zrozumienie serializacji i deserializacji, wielu ścieżek o równym koszcie, rywalizacji w krzyżowym polu komutacyjnym, pierścieni nadawczych i tym podobnych?

W pewnym sensie szczegóły te nie mają znaczenia dla przeciętnego inżyniera sieci. Kiedy urządzenie sieciowe wykonuje swoje zadanie przekazywania danych, rzeczywiste procesy przełączania, które się na to zadanie składają, są błahostkami. „To po prostu działa”.

Jednak wewnętrzne elementy przełączające często mają duży wpływ na projektowanie sieci. Rozważmy na przykład opóźnienie między portami. W niektórych sieciach o dużym natężeniu ruchu czas potrzebny na przełączenie ramki z portu wejściowego na port wyjściowy ma wpływ na ogólną wydajność aplikacji. W nowoczesnych przełącznikach opóźnienie między portami mierzone jest w mikrosekundach lub setkach nanosekund. Jeśli jeden przełącznik wykona zadanie w ciągu 1 mikrosekundy, podczas gdy inny może je wykonać w ciągu 400 nanosekund, może to wpłynąć na wybór sprzętu.

Kolejną kwestią jest rozwiązywanie problemów. Co się dzieje, gdy urządzenie sieciowe wydaje się nie przekazywać wszystkich odebranych pakietów, tzn. jest więcej danych na wejściu niż na wyjściu? Niewielkie straty pakietów w sieci są kłopotliwe do wysledzenia. Zrozumienie wewnętrznego procesu przełączania pakietów w urządzeniu sieciowym rzuca dużo światła na to, gdzie może wystąpić awaria.

Dlatego nie należy odrzucać przełączania pakietów jako czegoś „zbyt blisko okablowania”, aby było istotne dla ambitnego inżyniera sieci. Zamiast tego warto skorzystać z wiedzy na temat przełączania pakietów, aby uzyskać głęboki wgląd w ogólną wydajność sieci.

Dalsza lektura

- 1.5. *Basics of How Operating Systems Work. Operating Systems Study Guide*, <http://faculty.salina.k-state.edu/tim/oss/Introduction/OSworking.html> [dostęp: 22 kwietnia 2017].
- Bollapragada Vijay, White Russ, Murphy Curtis, *Inside Cisco IOS Software Architecture*, Cisco Press, Indianapolis 2000.
- Cisco Nexus 5548P Switch Architecture, Cisco, http://www.cisco.com/c/en/us/products/collateral/switches/nexus-5548p-switch/white_paper_c11-622479.html [dostęp: 29 lipca 2017].
- Fast Ethernet | Integrating 100mbps into Existing 10mbps Networks, Savvius, https://www.savvius.com/resources/compendium/fast_ethernet/overview [dostęp: 22 kwietnia 2017].
- Heineman George T., Pollice Gary, Selkow Stanley, *Algorytmy. Almanach*, Helion, Gliwice 2010.
- Inniss Daryl, Rubenstein Roy, *Silicon Photonics: Fueling the Next Information Revolution*, wyd. I, Morgan Kaufmann, 2016.
- Intel Ethernet Switch Family Hash Efficiency, Intel, kwiecień 2009, <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ethernet-switch-hash-efficiency-paper.pdf>.
- Interrupt, Wikipedia, <https://en.wikipedia.org/w/index.php?title=Interrupt&oldid=763436239> [dostęp: 3 lutego 2017].
- Kloth Axel K., *Advanced Router Architectures*, CRC Press, Boca Raton 2005.
- Konheim Alan G., *Hashing in Computer Science: Fifty Years of Slicing and Dicing*, wyd. I, Wiley-Interscience, 2011.
- Lekkas Panos, *Network Processors: Architectures, Protocols and Platforms*, wyd. I, McGraw-Hill Education, New York 2003.
- Meiners Chad R., Liu Alex X., Torng Eric, *Hardware Based Packet Classification for High Speed Internet Routers*, wyd. 2010, Springer, New York 2010.
- Noubir Guevara, *Signal Encoding Techniques*, <http://www.ccs.neu.edu/home/noubir/Courses/CS6710/S12/slides/signals-encoding.pdf> [dostęp: 22 kwietnia 2017].
- Scudder F.J., Reynolds J.N., *Crossbar Dial Telephone Switching System*, „BSTJ 18”, 1 stycznia 1939, <http://archive.org/details/bstj18-1-76>.
- Stringfield Nakia, White Russ, McKee, Stacia *Cisco Express Forwarding*, wyd. I, Cisco Press, Indianapolis 2007.
- Thakur Dinesh, *Encoding Techniques and Codec*, Computer Notes, <http://ecomputernotes.com/computernetworkingnotes/communication-networks/encoding-techniques-and-codec> [dostęp: 22 kwietnia 2017].
- Understanding IEEE 802.3ad Link Aggregation — Technical Documentation — Support — Juniper Networks, 26 marca 2013, https://www.juniper.net/documentation/en_US/junose14.2/topics/concept/802.3ad-link-aggregation-understanding.html.

Pytania kontrolne

1. Co się stanie, jeśli jeden koniec łącza zostanie skonfigurowany jako pakiet łączy zagregowanych, a drugi nie? Konkretnie: co się stanie, jeśli jedno urządzenie uważa, że STP działa, a drugie nie?
2. Dlaczego obliczanie funkcji skrótu dla przepływu jest zwykle używane zamiast *round-robin* jako algorytm przesyłania w ECMP?
3. Jaki jest cel istnienia wieloetapowych układów? Podaj przykład.
4. Krótko podsumuj techniki stosowane w krzyżowych polach komutacyjnych do łączenia rywalizacji.
5. Algorytm iSLIP zawiera kroki: żądania, zgody i akceptacji. Wyjaśnij krótko — po jednym zdaniu na krok — co dzieje się na każdym etapie.
6. Ile razy iSLIP musi zostać uruchomiony, zanim przestanie skutecznie poprawiać dopasowania wejścia-wyjścia?
7. Ile pakietów można jednocześnie umieścić w buforze cyklicznym?
8. Dlaczego nie ustawia się wystarczająco dużych buforów nadawczych i odbiorczych, aby zapobiec nadpisywaniu pakietów z powodu zbyt wolnego przetwarzania pakietów będących w buforze cyklicznym? Jakie są kompromisy pod względem szybkości przełączania poprzez przełącznik, wykorzystania pamięci i innych czynników?
9. Zbadaj i opisz wpływ burzy rozgłoszeniowej w sieci. Jak routing zapobiega burzom rozgłoszeniowym?
10. Wymień kilka zalet korzystania z MLAG do budowy bardzo dużych, płaskich sieci bez routingu. Jakie mogą być wady?

Skorowidz

A

ABR, Area Border Router, 490

adres

IPv4, 126

IPv6, 128, 130

MAC, 108

randomizacja, 253

adresacja urządzeń i aplikacji, 71

adresowanie OSI, 397

agregacja, 495

BGP, 495

informacji o osiągalności, 470

łączy, 175, 178

aktualizacja oprogramowania, 566

algorytm

Bellmana-Forda, 304, 377–381

Diffusing Update, 310

Dijkstry, 317, 318

DUAL Garcii, 310

exponential backoff, 500

iSLIP, 184

przeszukiwania w głąb, 336

rozłącznej ścieżki Suurballe'a, 332

SPF Dijkstry, 319, 321, 401

Suurballe'a, 336

algorytmy zachłanne, 298

anycast, 76

aplikacje, 23

awarie sieci, 560

zdezagregowane, 595

zwirtualizowane, 595

architektura

hiperkonwergentna, 594

I2RS, 449

konwergentna, 594

sieci, 509

zdezagregowana, 594

ARP, 159

ASIC, Application-Specific Integrated Circuits, 181

atak, 516

odbicia DDoS, 523, 525

siłowy, 244

spalenia, 525

typu man-in-the-middle, 255, 520

typu odmowa usługi, 523

wzmocnienia, 525

automatyzacja, 643

infrastruktury, 628

interfejsy programowalne, 624

na potrzeby wdrażania, 629

na poziomie urządzenia, 627

narzędzia, 628

otwarta, 688

sieci, 615

awaria, 463, 505, 560

łącza, 382

pamięci podręcznej, 477

płaszczyzny sterowania, 467

B

baza danych FIB, 357

bezpieczeństwo, 513

IoT, 673

przez zaciemnienie, 247

transportu, 239, 259

w chmurze, 665

warstwy transportowej, TLS, 257

bezpieczna sesja TLS, 258
 bezpołączeniowość, 99
 BGP, Border Gateway Protocol, 97, 414, 421, 444
 agregacja, 495
 nakładka osiągalności, 496
 proces peeringu, 416
 reflektory tras, 420
 reguły rozgłaszania, 419
 wybór najlepszej ścieżki, 417
 bit dołączenia, 487
 BLE, Bluetooth Low Energy, 678
 blokowanie
 początku linii, 143
 sesji, 525
 upstreamu DDoS, 529
 błędy, 64
 brama domyślna, 163
 Broadcast, 107
 budowanie dokładnych modeli, 578
 buforowanie, 475, 478
 bufory rozdęte, 210
 burze broadcastowe, 377

C

CAP, Consistency, Accessibility, and Partition
 tolerance, 457, 511
 charakterystyka
 konwergencji, 551
 odporności, 551
 skalowania, 550
 chipsety Ethernet, 107
 chmury publiczne, 589, 653
 bezpieczeństwo, 665
 ciężar danych, 664
 kompromisy
 biznesowe, 660
 operacyjne, 657
 kontrola dostępu, 667
 monitorowanie sieci, 668
 ochrona danych, 665
 opóźnienie, 661
 wydatki
 inwestycyjne, 656
 operacyjne, 656
 wypełnianie zdalnej przestrzeni, 663
 wyzwania techniczne, 661
 z wieloma podmiotami, 667

CoDel, 210
 CRC, 66
 CSMA/CD, 105, 109
 CSNP, Complete Sequence Number Packet, 400
 cyfrowe gramatyki, 57
 czarne dziury routingu, 283
 czynnik zagrożenia, 516

D

dane
 ochrona, 240
 sprawdzanie prawidłowości, 240
 dane IoT, 673, 683
 DDoS, distributed denial of service, 523
 definiowanie
 odporności, 561
 złożoności, 46, 48
 deskryptor bazy danych, 405
 dezagregacja, 591, 607, 611
 DHCP, 157
 DMZ, 535
 DNS, 155
 DoD, 90
 warstwa aplikacji, 91
 warstwa fizyczna, 91
 warstwa Internetu, 91
 warstwa łącza, 92
 warstwa narzędziowa, 92
 warstwa sieci, 92
 warstwa transportowa, 91
 domeny
 awarii, 463, 505
 floodingowe, 486
 w IS-IS, 486, 488
 w OSPF, 489
 dostęp do sieci korporacyjnej, 220
 dostępność, 363
 usług, 523
 dostrajanie MTU, 222
 drzewo, 296
 maksymalnie redundantne, MRT, 336
 MST, 298
 najkrótszych ścieżek, SPT, 318, 323, 377, 413
 skrótów, 700
 wolne od pętli, 371
 DSCP, Differentiated Service Code Points, 197
 mutacja, 199
 translacja, 199

dystribucja zmian, 354
działanie

- NETCONF, 622
- protokołów warstwowania, 545
- sieci na domeny floodingowe, 545
- sieci nazwanych danych, 697

E

ECMP, Equal Cost Multipath, 175
efektywność, 45
EIGRP, Enhanced Interior Gateway Routing Protocol, 383, 444

- działanie protokołu, 385
- metryki, 384
- niezawodny transport, 388
- reagowanie na awarię, 387
- wykrywanie sąsiadów, 388
- zasięg zapytań, 388

 ekspluata, 516
elastyczność, 58

- skalowania sieci, 544

 Ethernet, 104, 219

F

FaaS, Functions as a Service, 654
fibbing, 446
filtrowanie

- informacji o osiągalności, 473
- nieroutowalnych adresów, 528
- tras, 473

 firewall, 535
format TLV, 41, 62
fragmentacja, 128, 361

- pakietów, 144

 funkcje

- jako usługi, FaaS, 654
- skrót, 176
- zwirtualizowane, 633

G

gałęzie, 603
grupy łącz, 565

- wspólnego ryzyka, 232

 gwiazda, 555

H

hierarchia

- dwuwarstwowa, 548
- rekurencyjna, 549

 hiperkonwergencja, 591

I

I2RS, 449
IaaS, 654
ICMP, Internet Control Message Protocol, 124, 146
identyfikatory, 153
IKE, Internet Key Exchange, 522
informacje

- o osiągalności, 470, 473
- o topologii, 469, 484

 infrastruktura

- jako usługa, IaaS, 654
- klucza publicznego, PKI, 251

 interfejs

- południowy, 442
- północny, 442

 interfejsy programowalne, 624
Internet, 201
internet rzeczy, 589, 671

- bezpieczeństwo, 673
- BLE, 678
- dane, 673, 683
- IPv6, 681
- izolacja, 674
- LoRaWAN, 680
- łączność, 673, 678
- unikernel, 676

 IoT, Internet of Things, *Patrz: internet rzeczy*
IP, Internet Protocol, 124, 219
IPsec, 97, 666
IPv4

- rozwiązywanie adresów, 159

 IPv6

- agregacja adresów, 131
- format nagłówka, 129
- fragmentacja, 128
- internet rzeczy, 681
- multipleksowanie, 130
- multipleksowanie między procesami, 134
- Neighbor Discovery, 161
- routing segmentowy, 228
- ustalanie stanu połączenia, 165

IS-IS, Intermediate System to Intermediate System, 396, 403, 484
 adresowanie OSI, 397
 domeny floodingowe, 488
 łączy wielodostępowe, 410
 marshalling danych, 399
 wykrywanie sąsiadów, 399
 zalewanie, 402
 zalewanie kopiami, 484
 iSLIP, 184
 izolacja
 na poziomie usługi, 674
 punktu końcowego, 675

J

jakość usług, 42
 język YANG, 452, 689
 języki modelowania, 689

K

klasyfikacja, 199
 klucze
 prywatne, 248
 publiczne, 248, 521
 kodowanie flowspec, 530
 kolejki, 209
 kolejkovanie
 o niskim opóźnieniu, 202
 ważone według klasy, 207
 kolizje, 105
 kombinacje sygnałów, 116
 komponenty, 573
 routera, 609
 TLS, 257
 kompromisy
 biznesowe, 660
 operacyjne, 657
 komutacja łączy, 30
 konceptualizacja łączy, 412
 konfiguracja
 IPsec, 666
 OpenFlow, 455
 sesji protokołu TCP, 141
 kontrola
 błędów, 110, 119, 140
 dostępu, 518
 przepływu, 36, 78, 112, 119
 transmisji, 135

kontrolery sieciowe, 32, 629
 konwergencja, 346, 551
 pierścienia, 552
 korekcja błędów, 69
 korzyści obliczeniowe, 646
 krawędź, 268
 kryptografia klucza, 248
 krytyczne przeciążenie, 208
 krzyżowe pole komutacyjne, 183
 kształtowanie wiązki, 117

L

LACP, Link Aggregation Control Protocol, 177
 LFA, Loop-Free Alternates, 295, 324
 lista węzłów, 296
 liść, 268, 603
 LoRaWAN, 680
 losowe wczesne wykrywanie, RED, 209
 LSA, Link State Advertisement, 489
 routera międzyobszarowego, 493
 LSP, Link State Packet, 402, 485
 luki w zabezpieczeniach, 516

Ł

łańcuch
 bloków, 696, 700
 usług, 637–640
 łączy
 konceptualizacja, 412
 równoległe routowane, 179
 wielodostępowe, 410
 współdzielonego ryzyka, 232, 437, 565
 wirtualne, 217
 łączenie usług, 637
 łączność
 dwukierunkowa, 339, 414
 IoT, 673, 678

M

magistrala, 182
 maksymalna jednostka transmisji, MTU, 274
 maksymalnie redundantne drzewa, 336
 manipulowanie, 583
 mapowanie
 bazy danych, 154
 identyfikatorów, 155

- portów TCP, 154
- QoS, 43
- marshalling danych
 - w protokole IS-IS, 399
 - w protokole OSPF, 404
- mechanizm przetwarzania pakietów, 180
- metadane, 34
- metody automatyzacji, 620
- metryki, 285
 - EIGRP, 384
- miejsce docelowe, 268
- mierzenie odporności, 561
- mikropętla, 356
- model, 574, 689
 - „co”, 576
 - DoD, 50, 90
 - hybrydowy, 443
 - iteracyjny, 99
 - OSI, 93, 578
 - podmiany, 443
 - RINA, 97, 98, 578
 - rozproszony, 443
 - rozszerzony, 443
 - scentralizowanej płaszczyzny sterowania, 443
 - wodospadu, 300
- modelowanie transportu sieciowego, 89
- modułowość, 544, 567
- modyfikacje systemów operacyjnych, 525
- monitorowanie sieci, 668
- MPLS, Multiprotocol Label Switching, 223, 227, 498
 - stos etykiet, 227
- MRT, Maximally Redundant Trees, 336
- MST, Minimum Spanning Tree, 298
- MTU, Maximum Transmission Unit, 124, 274
- multicast, 73, 75, 107
- multipleksowanie, 71, 104, 130, 266
 - między procesami, 134
 - przestrzenne, 114, 117
- mutacja DSCP, 199

N

- nakładka
 - kontrolera, 498
 - osiągalności, 496
- NDN, Named Data Networking, 696
- negocjowane szybkości transmisji, 83

- NETCONF, 620
 - działania, 622
 - warstwy, 621
- NFV, Network Function Virtualization, 635, 647, 649
- nieplanarne topologie sieci, 556
- niezawodność, 45
- NPU, Network Processing Unit, 181

O

- obliczanie
 - CRC, 66
 - klucza sesji, 521
 - LFA, 325
 - metryki EIGRP, 384
 - rLFA, 325
 - ścieżek, 40
- obrona w głąb, 517, 518
- obszar
 - normalny, 490
 - stub area, 491, 494
 - stubby area, 492
 - totally not-so-stubby area, 493
- ochrona
 - danych, 240, 519
 - prywatności użytkowników, 241
- odcięcie ogona, 203
- odcisk palca, 514
- odnajdowanie
 - międzywarstwowe, 151, 152
 - DHCP, 157
 - DNS, 155
 - sąsiadów, 161
- odporność, 551, 561, 563, 567
 - na partycjonowanie, 363
 - sieci, 559
- odpytywanie, 347, 350
- ograniczanie przepustowości żądań połączeń, 526
- OODA, 532
- OpenFlow, 455, 457
- oprogramowanie
 - jako usługa, SaaS, 653
 - sieciowe, 23
- optymalizacja aplikacji, 429, 438
- organizacja danych, 57
- organizowanie danych, 111, 119, 127

OSI, Open Systems Interconnect, 93, 397, 485
 warstwa
 aplikacji, 96
 fizyczna, 95
 łącza danych, 95
 prezentacji, 96
 sesji, 96
 sieci, 95
 transportowa, 95
 osiągalne miejsca docelowe, 269, 375
 OSPF, Open Shortest Path First, 396, 409, 442, 489
 agregacja, 495
 domeny floodingowe, 489
 łącza wielodostępowe, 410
 marshalling danych, 404
 obszary, 489, 494
 wykrywanie sąsiadów, 406
 zalewanie, 407
 OTT, over-the-top, 219

P

PaaS, Platform as a Service, 654
 pakiet, 33, 111
 CSNP, 400
 LSP, 402, 485
 PSNP, 400
 pakiety
 kopiowanie
 do pamięci, 171
 na łączy, 186
 przekazywanie, 217
 przełączanie, 169, 172
 przenoszenie, 170
 przetwarzanie, 172
 routing, 173
 partycjonowanie, 362
 PCEP, Path Computation Element Protocol, 496
 PCEP, Path Control Element Protocol, 453
 peering BGP, 415
 pętla
 OODA, 532
 sprzężenia zwrotnego, 465
 redystrybucji i routingu, 287
 pierścień, 550
 PKI, Public Key Infrastructure, 251
 planarne topologie sieci, 556

platforma jako usługa, PaaS, 654
 płaszczyzna
 danych, 23, 33
 możliwego, 48
 sterowania, 33, 234, 263, 284, 361, 464
 awaria, 467
 klasyfikacja, 368
 modele, 443
 protokoły wektora odległości, 367
 przestrzeń rozwiązań, 468
 reguły, 425
 rozproszona, 368
 scentralizowana, 44, 359, 368, 441, 444, 460
 ukrywanie informacji, 474
 zakres stanu, 464
 złożoność, 435
 zarządzania, 33
 płaszczyzny sieciowe, 32
 płynny restart, 566
 PN, Programmable Network, 442
 podzielony horyzont, 359
 pola o stałej długości, 60
 pole
 DSCP, 195, 199
 typu usługi, 195
 połączenie, 99
 pomiar rozciągnięcia, 47
 pomocniczość, 457
 porty TCP, 140
 potwierdzenie
 negatywne, 82
 pozytywne, 82
 wybiórcze, 82
 zbiorcze, 82
 powierzchnia
 ataku, 516
 interakcji, 49, 234
 powtórne przetwarzanie, 225
 prawo nieszczelnych abstrakcji, 510
 priorytetyzacja QoS, 194
 proaktywne rozpowszechnianie osiągalności, 283
 problemy, 571
 biznesowe, 540
 ukrytego węzła, 118
 z bramą domyślną, 163
 z wirtualizacją, 222
 programowalność, 627

projektowanie
 hierarchiczne, 547
 sieci, 509, 546, 558, 597
 automatyzacja, 643
 elastyczność, 635
 łączenie usług, 637
 skalowanie horyzontalne, 642

protokoły
 kontroli błędów, 98
 stanu łącza, 41, 340, 395, 412
 transmisyjne niższych warstw, 120
 transportu danych, 98
 wektora odległości, 41, 340, 367, 377, 390
 wektora ścieżki, 41, 395

protokół
 ARP, 155, 159
 BGP, 97, 414
 DHCP, 157
 drzewa rozpinającego, STP, 371, 377
 EIGRP, 383, 388
 IKE, 522
 ICMP, 124
 IP, 124
 organizowanie danych, 127
 transport, 127
 IS-IS, 396, 397, 484
 LACP, 177
 OpenFlow, 455
 OSPF, 396, 404, 489
 PCEP, 453, 496
 QUIC, 124
 RIP, 378, 380, 383
 SNMP, 618
 TCP, 36, 124, 135
 UDP, 405

przeciążenie łącza, 208
 przekazywanie pakietu, 217
 przekształcenia Internetu, 702
 przeładowanie kodu, 29
 przełączanie
 etykiet, 37
 pakietów, 33, 169, 599
 się między modelami, 579

przenoszenie danych, 23

przepływy
 myszowate, 177, 429
 słoniowate, 177, 208, 429

przepustowość, 120
 VNF, 647

przeskok po przeskoku, 358
 przestrzeń P/Q, 302
 przetwarzanie
 pakietu, 172
 w chmurze, 653

przypinanie przepływów, 429, 438
 PSNP, Partial Sequence Number Packet, 400

punkty
 awarii, 563
 przewężenia, 545

Q

QoS, Quality of Service, 43, 186, 193, 212
 schematy priorytetyzacji, 194
 wybór ścieżki, 201
 zarządzanie zatorami, 208

QUIC, Quick User Datagram Protocol Internet
 Connections, 124, 141
 blokowanie początku linii, 143
 ograniczanie retransmisji, 142
 redukcja początkowego uzgadniania, 142

R

ramka Ethernet, 111

ramki
 o stałej długości, 37
 o zmiennej długości, 37

randomizacja adresu MAC, 253

rdzenie wielopłaszczyznowe, 566

reaktywne rozpowszechnianie osiągalności, 280

RED, Random Early Detection, 209

redukcja zalewania, 503

redundancja, 559, 563

redystrybucja, 285

reguła
 podzielonego horyzontu, 359
 zatruwania zwrotnego, 359

reguły
 BGP, 435
 płaszczyzny sterowania, 425, 439
 definiowanie, 434
 routing i ziemniaki, 426
 segmentacja zasobów, 428

rekursywna architektura internetowa, 97

RESTCONF, 623

RINA, Recursive Internet Architecture, 97, 123

RIP, Routing Information Protocol, 305, 378, 444
 rLFA, remote Loop-Free Alternates, 324
 router
 granicy obszaru, 490
 komponenty, 609
 routing, 173
 cebulowy, 254
 czarne dziury, 283
 i ziemniaki, 426, 435
 segmentowy, SR, 223, 496, 498
 w IPv6, 228
 z MPLS, 224
 sygnalizowanie etykiet, 229
 routowane
 ECMP, 179
 łącza równoległe, 179
 rozciąganie, 47
 rozgłaszanie
 BGP, 419
 osiągalności i topologii, 278
 rozkładanie
 ataku DDoS, 527
 ruchu, 526
 rozpowszechnianie osiągalności
 proaktywne, 283
 reaktywne, 280
 rozwiązywanie problemów, 571, 581
 przełączanie się między modelami, 579
 usuwanie problemu, 585
 równoległe uruchamianie, 566
 ruch
 północ – południe, 598
 VoIP, 205
 wschód – zachód, 597
 rywalizacja, 184
 ryzyka, 516

S

SaaS, Software as a Service, 653
 scentralizowane
 płaszczyzny sterowania, 281, 441, 460
 zarządzanie konfiguracjami, 643
 scentralizowany magazyn, 359
 SDN, Software-Defined Network, 442, 444
 SD-WAN, Software-Defined Wide Area
 Network, 231
 segmentacja zasobów, 428, 436
 sesja TLS, 258

siatka, 553
 sieci
 fizyczne, 24
 hiperkonwergentne, 691
 nazwanych danych, NDN, 696
 oparte na intencjach, 645, 692
 pakietowe, 35
 promieniste, 230
 routingu segmentowego, 498
 rozległe, 230
 definiowane programowo, SD-WAN, 231
 rozwiązywanie problemów, 571, 581
 szkieletowe, 606
 z przełączaniem pakietów, 34, 599
 sieci wirtualne, 216
 sieć
 bezprzewodowa 802.11, 112
 kontrola błędów, 119
 kontrola przepływu, 119
 multipleksowanie, 113
 multipleksowanie przestrzenne, 114
 organizowanie danych, 119
 problem ukrytego węzła, 118
 współdzielenie kanału, 118
 dwupołączona, 331
 gałęzi i liści, 607
 skalowanie, 550
 horyzontalne, 602, 642
 sieci w górę, 602
 skrót, hash, 176
 skróty kryptograficzne, 252
 słowniki, 57, 59
 obiektów, 63
 współdzielone, 63
 SNMP, Simple Network Management
 Protocol, 618
 software defined, 442
 SOS, State/Optimization/Surface, 457, 510
 SPF
 częściowy, 324
 przyrostowy, 324
 spłukiwanie, 403
 spowalnianie prędkości stanu, 480
 spójność, 362
 sprawdzenie łącza zwrotnego, 414
 sprzężenie zwrotne, 465
 SPT, Shortest Path Tree, 318, 377, 413
 SR, Segment Routing, 223, 496
 SRLG, Shared Risk Link Group, 437, 565

SSL, Secure Socket Layer, 257
 stan- optymalizacja-powierzchnia, 510
 sterowanie przepływem, 82
 STP, Spanning Tree Protocol, 371
 strefa zdemilitaryzowana, DMZ, 535
 style degradacji sieci, 546
 sumaryzacja informacji o topologii, 469, 484
 sygnał, 115
 synchronizacja TCP, 210
 system
 kompozycyjny, 596
 nazw domen, DNS, 155
 okien dystrybucji, 79
 sztuczna inteligencja, 694
 szybkość transmisji bitów
 dostępna, ABR, 83
 stała, CBR, 83
 zmienna, VBR, 83
 szyfrowanie, 242, 248

Ś

ścieżki, 370
 najkrótsze, 293, 295
 algorytm Dijkstry, 317
 o równym koszcie, 175
 pozbawione pętli, 40
 przełączania, 180
 redundantne, 564
 rLFA, 325
 rozgłaszania, 370
 sieciowe, 268
 wąskie gardło, 192
 wektor, 327
 wolne od pętli, LFA, 294, 325, 417
 algorytm Bellmana-Forda, 304
 alternatywne, 299
 unicastowe, 293, 317
 zdalne alternatywne, 303

T

tabela Unicode, 59
 tablica routingu, 390
 talia osy, 50
 TCP, Transmission Control Protocol, 36, 124, 405
 konfiguracja sesji, 141
 kontrola błędów, 140
 kontrola przepływu, 135

numery portów, 140
 okno odbioru, 138
 okno zatoru, 138
 próg powolnego startu, 139
 TE, Traffic Engineering, 496
 testowanie, 583
 timer
 blokowania, 383
 nieaktywności, 375
 nieważności, 382
 retransmisji, 375
 splukiwania, 383
 wstrzymania, 283
 TLS, Transport Layer Security, 257
 TLV, Type Length Value, 62
 TLV, Type Length Vector, 452
 topologie sieci, 265, 271
 fizyczne i wirtualne, 216
 gwiazdy, 555
 nieregularne, 600
 pierścienia, 550
 planarne, 556
 reagowanie na zmiany, 345, 386
 regularne, 556, 600
 siatki, 553
 ToS, Type of Service, 195
 tożsamość biometryczna, 514
 translacja DSCP, 199
 transmisja danych
 błędy, 64
 transport danych, 55, 127
 bezpieczeństwo, 239
 w wyższych warstwach, 123
 w niższych warstwach, 103
 tryb
 asynchroniczny, 352
 z echem, 353
 awarii pamięci podręcznej, 477
 na żądanie, 353
 z echem, 353
 nieograniczony, 107
 PIM Sparse Mode, 75
 tunel, 227
 twierdzenie CAP, 362, 457, 511
 tworzenie
 łańcucha usług, 637
 odporności, 563
 warstw wirtualnych, 545
 wiązki, 115

U

ucho, 339
 uczenie
 maszynowe, 694
 się proaktywnie, 277
 się reaktywne, 276
 UDP, User Datagram Protocol, 405
 ukrywanie informacji, 463, 481, 483
 unicastowe przekazywanie przez ścieżkę
 powrotną, 528
 Unicode, 59
 unikernel, 676
 upraszczanie przed testowaniem, 585
 uRPF, Unicast Reverse Path Forwarding, 528
 urządzenia sieciowe, 24, 271, 646
 usługa, 42, 191
 ochrony przed DDoS, 531
 OTT, 219
 usługi
 pole ToS, 195
 bezserwerowe, 654
 usuwanie problemu, 585
 uwarstwienie, 496
 uwierzytelnianie, authentication, 519

V

VNF
 poprawianie przepustowości, 647
 VoIP, Voice over Internet Protocol, 204, 551
 VPN, 221
 VOQ, virtual output queues, 184

W

warstwa
 aplikacji, 91, 96
 bezpiecznych gniazd, SSL, 257
 fizyczna, 91, 95
 Internetu, 91
 łącza danych, 92, 95
 narzędziowa, 92
 prezentacji, 96
 sesji, 96
 sieci, 92, 95
 transportowa, 91, 95
 warstwy NETCONF, 621

wąskie gardło, 636
 wczesne wykrywanie
 losowe ważone, 210
 wektor
 odległości, 315
 ścieżki, 327
 węzeł, node, 267
 liścia, 268
 tranzytowy, 267
 wiązka, 115
 wieloprotokołowe przełączanie etykiet, MPLS, 223
 Wi-Fi, 112
 wirtualizacja, 222, 232, 236
 funkcji, 589
 sieciowych, NFV, 633, 635, 649
 sieci, 215
 wirtualne sieci prywatne, VPN, 221
 wirtualny prywatny dostęp, 220
 wskaźnik
 MTBM, 563
 MTTI, 563
 współdzielenie kanału, 118
 współdzielone bazy danych, 486
 wybór najlepszej ścieżki, 417
 wydajność zasobów, 58
 wydatki
 inwestycyjne, 656
 operacyjne, 656
 wykrywanie
 awarii, 347, 348
 błędów, 64
 kolizji, 106
 oparte na odpytywaniu, 350
 oparte na zdarzeniach, 350
 osiągalności, 276
 przekazywania dwukierunkowego, 351
 sąsiadów, 388, 399, 406
 topologii, 265, 289, 399, 406
 tras źródłowych, 281
 urządzeń sieciowych, 271
 zmian w topologii, 347
 wyliczanie identyfikatora, 155
 wymagania
 biznesowe, 544
 techniczne, 544
 wymiana kluczy, 249–251
 wzorce projektowania sieci, 539, 558

Y

YANG, 689

Z

zachowywanie klasyfikacji, 199

zaciemnianie danych użytkownika, 247, 252, 522

zagrożenie, 516

zalewanie, 484

niezawodne, 401, 407

stanem łącza, 503

urządzeń sieciowych, 355

zarządzanie

bezpiecznymi połączeniami, 666

kolejką, 209

konfiguracjami, 643

opóźnieniami bufora, 210

pełnym buforem, 209

zatorami, 202, 208

złożonością, 49

zasada

konwergencji pierścienia, 552

pomocniczości, 459

zasoby, 516

zator, 202

zatrutowanie zwrotne, 359, 383

złożoność, 44, 232

płaszczyzny sterowania, 435

przetwarzania, 249

sieci, 616

złudzenie optyczne, 533

zmiany

topologii, 360

w zasobach, 592

zmniejszenie prędkości stanu, 500


zwinność biznesowa, 656

Ż

źródła, 268

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

INNOWACYJNE PODEJŚCIE DO BUDOWANIA ODPORNYCH, NOWOCZESNYCH SIECI

Zrozumienie budowy i działania sieci komputerowych nie jest łatwe. Trzeba przyswoić wiele pojęć, zrozumieć bardzo zawiłe mechanizmy, a także przygotować się na gwałtowny rozwój technologii sieciowych. Mimo dostępności licznych podręczników i wypracowania różnych podejść do nauczania podstaw sieci komputerowych wciąż wielu inżynierów nie rozumie tych zagadnień. Te braki w wiedzy i w umiejętnościach należy traktować jako istotny problem: w przyszłości będą się pojawiać coraz to nowsze rozwiązania sieciowe, a ich znaczenie dla niemal każdego przedsiębiorstwa będzie rosnąć.

W książce zastosowano podejście problemowe, dzięki czemu łatwiej jest zrozumieć budowę oraz działanie współczesnych sieci komputerowych i protokołów, jak również wyzwania, z jakimi mierzą się dzisiejsze systemy. W praktyczny sposób opisano zagadnienia transportu danych i sterowania pracą sieci, przeanalizowano też kilka typowych projektów i architektur sieci, w tym sieci szkieletowe centrów danych i nowoczesne sieci rozległe definiowane programowo (SD-WAN). Szczegółowo zaprezentowano także technologie jak sieci definiowane programowo (SDN). Każdemu zagadnieniu towarzyszy omówienie typowych problemów i ich rozwiązań, a także sposobów ich implementacji w protokołach oraz metod wdrożenia.

W TEJ KSIĄŻCE MIĘDZY INNYMI:

- protokoły transportu w warstwach sieci i komunikacja międzywarstwowa
- pakiety, usługi, topologia sieci
- zabezpieczanie sieci, redundancja i odporność
- wzorce projektowe w sieciach
- automatyzacja zarządzania siecią
- internet rzeczy oraz inne nowości w sieciach

Russ White od ponad 30 lat projektuje wielkie sieci, zajmuje się też ich zabezpieczeniami i rozwiązywaniem problemów. Jest współautorem ponad 40 patentów na oprogramowanie. Obecnie jest architektem w LinkedIn, pracuje także w grupie doradczej do spraw routingu IETF i współprzewodniczy grupom roboczym IETF I2RS i Babel.

Ethan Banks pracuje w branży IT od 1995 roku. Był inżynierem systemów: Novell, Windows i Linux, zajmował się technologiami: DNS, SMTP, HTTP i powiązаныmi aplikacjami. Uzyskał wiele certyfikatów, w tym Microsoft Certified Systems Engineer, Cisco Certified Network Professional, Certified Ethical Hacker i Cisco Certified Security Professional.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI
Sięgnij po więcej!



ISBN 978-83-283-5043-4



INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 119,00 zł

Pearson