https://upel2.cel.agh.edu.pl/wiet/course/view.php?id=40
hasło: transparency

Rozprochy pytania:

“1) definicja systemów rozproszonych - czym jest system rozproszony, jakie warunki system rozproszony spełnia”

System rozproszony - zbiór niezależnych urządzeń technicznych postrzegany przez jego użytkowników jako pojedynczy, spójny system.

Aspekty
sprzęt: maszyny są autonomiczne
oprogramowanie: wrażenie pojedynczego systemu

Systemy rozproszone w przykładach
Firma, biuro, uczelnia, (np. PC, drukarki, ...)
Roboty/automaty w hali produkcyjnej
Systemy bankowe (wiele oddziałów)
Systemy rezerwacji biletów

Cechy systemów rozproszonych
Ukrycie przed użytkownikami:
- różnic pomiędzy poszczególnymi komputerami
- sposobów komunikowania się komputerów
- wewnętrznej organizacji systemu rozproszonego

Jednolity i spójny interfejs dla użytkownika —niezależnie od czasu i miejsca interakcji
Nieustanna dostępność zasobów i usług
Łatwość rozszerzania (skalowania)

Charakterystyka systemów rozproszonych
duża wydajność w sensie dużej mocy obliczeniowej i maksymalnej przepustowości, krótkiego czasu odpowiedzi;
duża efektywność inwestowania w sensie kosztów niezbędnych do uzyskania wymaganej wydajności systemu;
wysoka sprawność wykorzystania zasobów w sensie stopnia wykorzystania zasobów, współczynnika jednoczesności;

Charakterystyka systemów rozproszonych
skalowalność w znaczeniu możliwości ciągłego i nieograniczonego rozwoju systemu bez negatywnego wpływu na jego wydajność i sprawność;
wysoka niezawodność w sensie odporności na błędy;
otwartość funkcjonalna w sensie łatwości realizacji nowych, atrakcyjnych usług komunikacyjnych, informatycznych i informacyjnych

Problemy związane z konstrukcją systemów rozproszonych

optymalne zrównoleglenie algorytmów przetwarzania
ocena algorytmów rozproszonych
alokacja zasobów rozproszonych
synchronizacja procesów
ocena globalnego stanu przetwarzania
niezawodność
bezpieczeństwo

Przezroczystość:

System przezroczysty — (transparentny, ang. transparent) sprawia wrażenie systemu scentralizowanego (dla użytkowników i aplikacji).

Przezroczystość dostępu (ang. access transparency) ujednolicanie metod dostępu do danych i ukrywanie różnic w reprezentacji danych.Różnice w reprezentacji danych mogą wynikać z:
Zastosowania różnych architektur komputerowych
Kodowanie liczb w procesorach Intela – little endian
Kodowanie liczb w procesorach Sun SPARC – big endian
Różnych konwencji nazewnictwa plików stosowanych w różnych systemach operacyjnych.

“2) czym sa gniazda - definicja gniazda”

Gniazdo (ang. socket) – pojęcie abstrakcyjne reprezentujące dwukierunkowy punkt końcowy połączenia. Dwukierunkowość oznacza możliwość wysyłania i odbierania danych. Wykorzystywane jest przez aplikacje do komunikowania się przez sieć w ramach komunikacji międzyprocesowej.

Gniazdo posiada trzy główne właściwości:

-typ gniazda identyfikujący protokół wymiany danych
-lokalny adres (np. adres IP, IPX, czy Ethernet)
-opcjonalny lokalny numer portu identyfikujący proces, który wymienia dane przez gniazdo (jeśli typ gniazda pozwala używać portów)

Definicja od strony “java owej”:
A socket is one end-point of a two-way communication link between two programs running on the network. Socket classes are used to represent the connection between a client program and a server program. The java.net package provides two classes--Socket and ServerSocket--that implement the client side of the connection and the server side of the connection, respectively. - strona Oracle

https://pl.wikipedia.org/wiki/Gniazdo_(telekomunikacja)

3)” jaka warstwa socketow, jakie protokoly tam sa, jakie roznice” - ??

Warstwa 7: aplikacji

Warstwa aplikacji jest warstwą najwyższą, zajmuje się specyfikacją interfejsu, który wykorzystują aplikacje do przesyłania danych do sieci (poprzez kolejne warstwy modelu ISO/OSI). W przypadku sieci komputerowych aplikacje są zwykle procesami uruchomionymi na odległych hostach. Interfejs udostępniający programistom usługi dostarczane przez warstwę aplikacji opiera się na obiektach nazywanych **gniazdami** (ang. socket).

?? - nwm czy to o te porty chodzi??

Warstwa 4: transportowa

Warstwa transportowa segmentuje dane oraz składa je w tzw. strumień. Warstwa ta zapewnia całościowe połączenie między stacjami: źródłową oraz docelową, które obejmuje całą drogę transmisji. Następuje tutaj podział danych na części, które są kolejno indeksowane i wysyłane do docelowej stacji. Na poziomie tej warstwy do transmisji danych wykorzystuje się dwa protokoły TCP (ang. Transmission Control Protocol) oraz UDP (ang. User Datagram Protocol). W przypadku gdy do transmisji danych wykorzystany jest protokół TCP stacja docelowa po odebraniu segmentu wysyła potwierdzenie odbioru. W wyniku niedotarcia któregoś z segmentów stacja docelowa ma prawo zlecić ponowną jego wysyłkę (kontrola błędów transportu). W przeciwieństwie do protokołu TCP w protokole UDP nie stosuje się potwierdzeń. Protokół UDP z racji konieczności transmisji mniejszej ilości danych zazwyczaj jest szybszy od protokołu TCP, jednakże nie gwarantuje dostarczenia pakietu. Oba protokoły warstwy transportowej stosują kontrolę integralności pakietów, a pakiety zawierające błędy są odrzucane.

4) “czemu uzywamy wciaz zarowno tcp i udp - kiedy uzyc jednego kiedy drugiego”

TCP and UDP: What are the differences?

This graph sums up the differences, but we go into detail for each section below.

	TCP	UDP
Connection	Connection-oriented	Connectionless
Sequencing	TCP numbers each packet so they can be arranged in a sequence by the recipient	UDP sends the packets without numbering
Speed	Slower	Faster
Reliability	High	Low
Header size	Packets are heavy because of overheads	Lightweight packets with minimal headers
Error detection/correction	Error checking and error recovery	Error checking but no recovery. Corrupted packets are simply discarded and not requested again
Acknowledgement	Acknowledgement sent by the recipient	No acknowledgement is sent
Transfer method	Stream	Individual packets
Congestion control	Yes	No
Applications	File transfer, email, web browsing	Video conferencing, gaming, broadcasts

5) “przybliżenie portów, jaki zakres, kiedy do czego używamy, efemeryczne, zarezerwowane”

Port protokołu – pojęcie związane z protokołami używanymi w Internecie do identyfikowania procesów działających na odległych systemach. Jest to jeden z parametrów gniazda.

Numery portów reprezentowane są przez liczby naturalne z zakresu od 0 do 65535 ({\displaystyle 2^{16}-1}2^{16}-1) . Niektóre numery portów (od 0 do 1023) są określone jako ogólnie znane (ang. well known ports) oraz zarezerwowane na standardowo przypisane do nich usługi, takie jak np. WWW czy poczta elektroniczna. Dzięki temu można identyfikować nie tylko procesy, ale ogólnie znane usługi działające na odległych systemach. Można więc powiedzieć, że numer portu to identyfikator danej usługi. Numery od 1024 do 49151 są określone przez IANA jako zarejestrowane, (ang. registered), a od 49152 do 65535 jako dynamiczne/prywatne, (ang. dynamic/private).

Różne usługi mogą używać tego samego numeru portów, pod warunkiem że korzystają z innego protokołu (TCP lub UDP), chociaż istnieją także usługi korzystające jednocześnie z jednego numeru portu i obu protokołów. Przykładem takiej usługi jest DNS – korzysta z portu 53 za pomocą TCP i UDP jednocześnie. Zdarza się także, że jedna usługa może korzystać z dwóch różnych portów używanych do innych zadań, jak to jest w przypadku FTP czy SNMP.

Poszczególne numery portów przydzielone są przez IANA.

Port	Protokół
53	DNS
20	FTP – przesyłanie oraz pobieranie plików i folderów
21	FTP – przesyłanie poleceń
67	DHCP – serwer
68	DHCP – klient
79	Finger
70	Gopher
80	HTTP, dodatkowe serwery, np. proxy, są najczęściej umieszczane na porcie 8080
443	HTTPS (HTTP na SSL)
143	IMAP
220	IMAP3
6661 – 6668	IRC
5222	XMPP – dla serwera sieci Jabber
389	LDAP
636	LDAPS (LDAP na SSL)

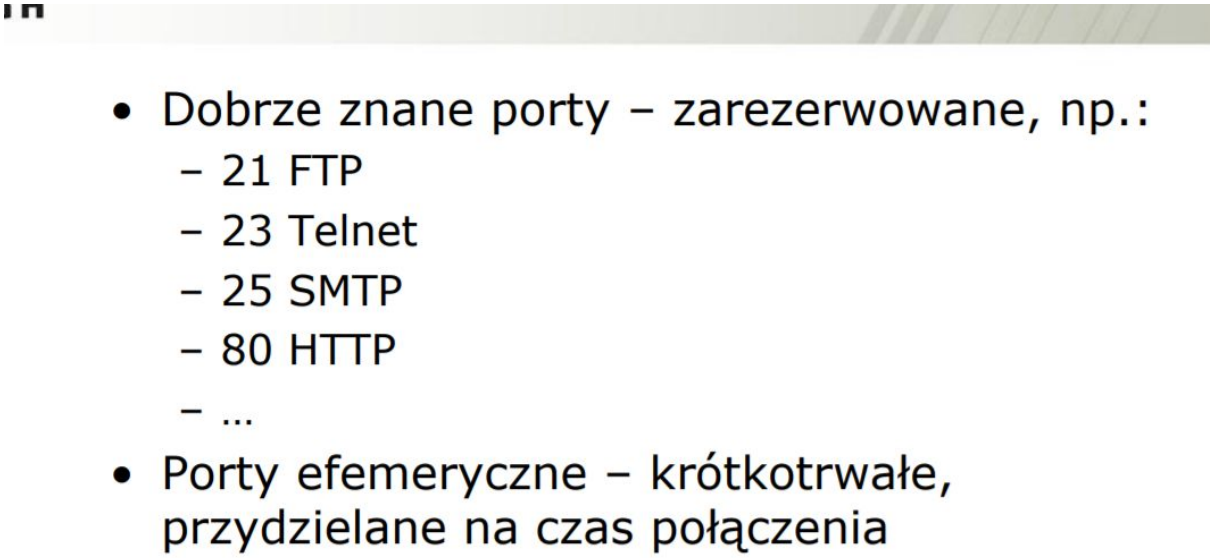
3306	MySQL
119	NNTP
110	POP3
995	POP3S (POP3 na SSL)
5432	PostgreSQL
873	Rsync
25	SMTP
22	SSH
514	Syslog
23	Telnet
69	TFTP
6000 – 6007	X11
161	SNMP
3389	RDP (Remote Desktop Connection)

Jeśli gniazdo używa numerów portów, to lokalny numer portu może zostać przydzielony automatycznie i nosi wtedy nazwę **efemerycznego numeru portu** (ang. ephemeral port number). Lokalny numer portu może też zostać wymuszony przez wykonanie przypisania (ang. bind) gniazdu numeru pożądanego przez twórcę aplikacji. Próba użycia gniazda, które wymaga zdefiniowanego lokalnego numeru portu bez uprzedniego przypisania mu go, spowoduje automatyczne przydzielenie numeru efemerycznego przez system operacyjny lub bibliotekę (zależnie od implementacji).

Grupy portów:

Do dyspozycji jest ogółem 65 535 portów TCP i UDP. Aby zachować nad nimi kontrolę, a także by móc przydzielać aplikacjom stałe numery, podzielono je na trzy grupy.

- **Dobrze znane porty (well known ports)** – zarezerwowane, standardowe numery portów od 1 do 1023. Ułatwiają nawiązanie połączenia, ponieważ zarówno nadawca, jak i odbiorca z góry wiedzą, że dane muszą być przesłane dla określonego procesu pod określony numer portu. Serwery Telnetu używają na przykład portu nr 23. Dobrze znane porty umożliwiają klientom nawiązywanie połączeń z serwerami bez dodatkowej konfiguracji. Zarządzaniem tymi portami zajmuje się Internet Assigned Numbers Authority (IANA). Listę aktualnie przydzielonych numerów portów można znaleźć pod adresem http://www.iana.org/assignments/port-numbers . Do roku 1992 dobrze znane porty ograniczały się do zakresu 1 do 255. Porty o numerach od 256 do 1023 były stosowane do usług uniksowych.
- **Zarejestrowane porty (registered ports)** – porty o numerach od 1024 do 49.151 przewidziane są dla usług, które zwyczajowo korzystają z określonych portów. Przykładem może być port 3128, często wykorzystywany przez serwery proxy jako alternatywny port HTTP.
- **Porty przydzielane dynamicznie (dynamically allocated ports, również ephemeral ports)** – jak wskazuje nazwa, zawsze przydzielane dynamicznie. Są to porty o numerach od 49.152 do 65.535. Każdy klient może korzystać z nich tak długo, jak długo kombinacja protokołu transportowego, adresu IP i numeru portu jest jednoznaczna. Proces, który potrzebuje dostępu do portu, żąda go od swojego hosta



W systemie Windows polecenie ***netstat*** wyświetla listę aktualnie uruchomionych programów (usług) korzystających z zasobów sieci, adresy IP do których nastąpiło połączenie, aktualne porty zarówno lokalne jak i zdalne z którymi łączy się dany program (usługa) oraz stan połączenia. Program *netstat* posiada duże możliwości w wyświetlaniu danych. Dane formuje się poprzez przełączniki (dodatkowe argumenty podawane podczas wywoływania programu). Wszystkie przełączniki wyświetlane są po podaniu polecenia:

>netstat /?

6) “jak przesyłamy (problemy nad ktorymi trzeba panowac) - rozne formaty zapisu danych, dlugosc bajtu”

??

Dane liczbowe - kolejność bajtów

- Kolejność bajtów w pamięci (*byte order / endianness)*
  - **Big endian** (Motorola, Sparc, **sieć**)
  - **Little endian** (Alpha, Intel)
  - Przykład:

1 507 634 416 (dec) = 59DC ACF0 (hex)

Big endian	59	DC	AC	F0
Little endian	F0	AC	DC	59