1. Podstawowe zagadnienia systemów rozproszonych (definicja, przeźroczystość, korzyści z rozproszenia i związane z tym wyzwania).

**System rozproszony:** Wiele procesów pracujących razem. Wiele niezależnych komputerów które jawią się użytkownikowi, jako jeden spójny system.

*A distributed operating system is one that looks to its users like an ordinary centrialized operation system, but runs on multiple, independent CPUs. The key concept here is transparency, in other words, the use of multiple processors should be invisible (transparent) to the user. Another way of expressing the same idea is to say that the user views the system as a virtual uniprocessor , not as a collection of distinct machines.*”

System rozproszony jest systemem który dla użytkownika jest system scentralizowanym jednak działającym na wielu procesorach. Kluczową koncepcja tutaj jest transparentność (**przejrzystość**). Fakt iż system działa na wielu procesorach jest niewidoczny dla użytkownika. Innymi słowy użytkownik postrzega system jako maszynę jednoprocesorową, nie jako kolekcję odrębnych maszyn.

**Zalety:**

* Możliwa komunikacja i współdzielenie zasobów
* Korzystna proporcja między ceną a wydajnością systemu.
* Niezawodność, skalowalność.
* Możliwość rozbudowy systemu.

**Wady**

* Aplikacje muszą być napisane pod system rozproszony.
* Zabezpieczenie połączenia sieciowego.
* Prywatność i bezpieczeństwo połączenia.
* Problemy skalowalności systemu:
  + Centralizacja usługi: pojedynczy serwer dla wielu użytkowników
  + Centralizacja danych: pojedyncza książka telefoniczna dostępna online
  + Centralizacja algorytmów: routing based on complete information.

**Modele systemów rozproszonych:**

* Minikomputer (każdy z użytkowników ma lokalna maszynę, ale może pobierać więcej danych)
* Workstation model
* Model: klient-serwer. Uzytkownik ma maszyne lokalną i łączy się z potężnym serwerem.

**Formy transparentności/przejrzystości**

* Dostęp. Różnica w reprezentacji danych oraz sposób jak zorganizowany jest dostęp do nich jest ukryty.
* Lokalizacja. Ukrycie gdzie zasób jest zlokalizowany.
* Migracja/Przeniesienie. Informacja o tym, że zasób jest przeniesiony na inną maszynę jest ukryta przez użytkownikiem.
* Replikacja. Ukrycie informacji o tym, że dany zasób może być replikowany.
* Współbieżność. Ukrycie informacji o tym, że zasób może być użyty przez wielu użytkowników.
* Awaria. Ukrycie faktu awarii i przywrócenia działającego zasobu,
* Persystencja. Ukrycie obecności oprogramowania na dysku.

Podstawowe zagadnienia związane z komunikacją sieciową:

* umiejętność efektywnego korzystania z analizatora pakietów (struktura pakietów, filtrowanie wiadomości)

<< ZROBIC TUTORIAL Z WIRESHARKA>>

**Umiejętność praktycznego wykorzystania narzędzi ping, traceroute, netstat**

* + ***nslookup*** : wyszukiwania szczegółowych informacji odnoszących się do serwerów DNS włączając adres IP poszczególnych komputerów, nazwę domeny, czy aliasy jakie posiada. Nazwa oznacza z angielskiego name server lookup.
  + ***Ipconfig*** : polecenie w systemach operacyjnych Microsoft Windows służące do wyświetlania konfiguracji interfejsów sieciowych
  + ***Ifconfig-a*** : polecenie konfigurujące interfejsy sieciowe w systemach Unix i Linux. Odpowiednie [skrypty](https://pl.wikipedia.org/wiki/J%C4%99zyk_skryptowy) wykorzystują ifconfig do "podniesienia" (uruchomienia) interfejsów sieciowych podczas startowania [systemu operacyjnego](https://pl.wikipedia.org/wiki/System_operacyjny). Opcja –a wyświetla wszystkie status wszystkich interfejsów sieciowych również tych nieaktywnych.
  + ***netstat*** : służy do wyświetlania aktywnych połączeń sieciowych [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol) a także: [portów](https://pl.wikipedia.org/wiki/Port_protoko%C5%82u), na których komputer nasłuchuje, tabeli [trasowania](https://pl.wikipedia.org/wiki/Trasowanie_(telekomunikacja)) [protokołu](https://pl.wikipedia.org/wiki/Protok%C3%B3%C5%82_komunikacyjny) [IP](https://pl.wikipedia.org/wiki/Protok%C3%B3%C5%82_internetowy), statystyki sieci [Ethernet](https://pl.wikipedia.org/wiki/Ethernet), statystyki protokołu [IPv4](https://pl.wikipedia.org/wiki/IPv4) (dla protokołów [IP](https://pl.wikipedia.org/wiki/Protok%C3%B3%C5%82_internetowy), [ICMP](https://pl.wikipedia.org/wiki/Internet_Control_Message_Protocol), [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol) i [UDP](https://pl.wikipedia.org/wiki/User_Datagram_Protocol)), statystyki protokołu [IPv6](https://pl.wikipedia.org/wiki/IPv6) (dla protokołów [IPv6](https://pl.wikipedia.org/wiki/IPv6), [ICMPv6](https://pl.wikipedia.org/wiki/ICMPv6), [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol) przez [IPv6](https://pl.wikipedia.org/wiki/IPv6) i [UDP](https://pl.wikipedia.org/wiki/User_Datagram_Protocol) przez [IPv6](https://pl.wikipedia.org/wiki/IPv6)) oraz połączeń [NAT](https://pl.wikipedia.org/wiki/Network_Address_Translation) i komunikatów netlinkowych. Polecenie netstat użyte bez parametrów powoduje wyświetlenie aktywnych połączeń protokołu [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol).
    - ***netstat –a***: służy do wyświetlania wszystkich aktywnych połączeń protokołu [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol), a także portów protokołu [TCP](https://pl.wikipedia.org/wiki/Transmission_Control_Protocol) i [UDP](https://pl.wikipedia.org/wiki/User_Datagram_Protocol), na których komputer nasłuchuje.
    - ***netstat –n***: służy do wyświetlania aktywnych połączeń protokołu TCP. Adresy i numery portów są wyrażane numerycznie i nie zostaną zmienione na nazwy.
    - ***netstat –o***: służy do wyświetlania aktywnych połączeń protokołu TCP, a także dołącza identyfikatory procesów (PID) poszczególnych połączeń, dzięki czemu można sprawdzić informacje o właścicielach portów dla każdego połączenia. Może być łączony z parametrami -a,-n i -p.
    - ***neststat-p tcp/udp***: W tym przypadku parametr protokół może przyjmować wartości: udp, tcpv6, tcp lub udpv6. Gdy parametr -p zostanie użyty jednocześnie z parametrem -s, aby wyświetlić statystyki poszczególnych protokołów, parametr ten może przyjąć wartości: tcp, udp, icmp, udpv6, ip, tcpv6, icmpv6 lub ipv6.
  + ***ping -*** polecenie używane w sieciach komputerowych TCP/IP (jak Internet) i służący do diagnozowania połączeń sieciowych. Pozwala na sprawdzenie, czy istnieje połączenie pomiędzy hostami testującym i testowanym. Umożliwia on zmierzenie liczby zgubionych pakietów oraz opóźnień w ich transmisji, zwanych lagami.
  + ***Tracert/traceroute*** - program służący do badania [trasy pakietów](https://pl.wikipedia.org/wiki/Trasowanie_(telekomunikacja)) w sieci [IP](https://pl.wikipedia.org/wiki/Protok%C3%B3%C5%82_internetowy). –d – do not reselve adresses to host names.

**Adresacja prywatna i publiczna, translacja adresów i jej konsekwencje**

**DNS (Domain Name System) –** usługa wykorzystywana przez aplikację od odzworowania adresu IP na nazwę domenową (i odwrotnie), wyszukiwania informacji na przykład adresów email, obsługi aliasów nazw komputerów.

Adresacja w sieci internet: nazwy domenowa, adresy warstwy sieciowej, porty warstwy transportowej.

Adresacja usług i zasobów:

* Gniazdo docelowe: (np. 200.200.200.200:80/TCP)
* URL (Uniform Resource Location)
  + wskazuje używany protokół
  + wskazuje gniazdo docelowe
  + wskazuje zasób lub usługę

Porty transportowe:

* Zakrey portów
* Porty efemeryczne
* Porty, tzw. Dobrze znane
* Gniazdo
* Asocjacja

Adresacja prywatna

Jako doraźne rozwiązanie problemu InternetNIC opracował system wielokrotnego wykorzystania adresów, polegający na zarezerwowaniu puli adresów IP do wykorzystania tylko w sieciach lokalnych.

**Tabela 2. Zakresy adresów IP zarezerwowane dla sieci lokalnych**

|  |  |
| --- | --- |
| Zakres adresów | Maska podsieci |
| 10.0.0.0–10.255.255.255 | 255.0.0.0 |
| 172.16.0.0–172.31.255.255 | 255.240.0.0 |
| 192.168.0.0–92.168.255.255 | 255.255.0.0 |

Ponieważ adresy IP z zakresów sieci lokalnej nie pozwalają na komunikację z Internetem, muszą być tłumaczone na adresy publiczne. Translacją adresów IP zajmuje się NAT. Źródłowe dane adresowe wychodzących pakietów są przekładane przez NAT na adresy publiczne. Z kolei docelowe dane adresowe pakietów nadchodzących z Internetu – na adresy lokalne.

Adresacja publiczna

W roku 1981 protokół IP został standaryzowany. Jego specyfikacja wymagała, że każdy system podłaczony do Internetu i oparty na IP będzie miał unikalny 32-bitowy adres. Pierwsza część adresu identyfikuje sieć, w której umiejscowiony jest host, natomiast druga część adresu identyfikuje konkretny host w danej sieci. Tworzy się więc dwu-poziomowa hierarchia adresowania.

Dostęp do Internetu jest możliwy tylko poprzez adresy przydzielane przez InterNIC (Internet Network Information Center). Ponieważ te adresy mogą również odbierać dane nadchodzące z Internetu, są określane jako adresy publiczne. Dostawca usług internetowych (ISP) dysponuje pulą publicznych adresów IP, które przydziela swoim klientom.

Przy standardowej konfiguracji komputer musi mieć własny publiczny adres IP, jeśli użytkownik chce mieć dostęp do Internetu. Jak już wspomnieliśmy, liczba dostępnych publicznych adresów IP jest ograniczona.

Kowersja prywatnego adresu IP na publiczny może zostać zrobiona na równych poziomach adresowania pakietu:

* Na poziomie adresu warstwy sieciowej
* Na poziomie warstwy transportowe

NAT dokonuje translacji tylko adresu IP prywatnego hosta. Informacja adresowa wygląda następująco:

* (adres IP źródła : numer portu źródłowego;
* adres IP przezn. : numer portu przeznaczenia)

Pakiet, który wychodzi w kierunku Internetu jest translowany do postaci:

* (adres IP źródła : numer portu źródłowego;
* adres IP przezn. : numer portu przeznaczenia)

W sieciach lokalnych używa się adresów IP z zarezerwowanych zakresów, wymienionych w tabeli 2. Przykładowo, Twoja sieć może korzystać z zakresu 192.168.0.0–192.168.255.255. Oprócz tego zakresu masz przydzielony przez ISP publiczny adres IP, np. 83.31.15.15. Serwer NAT dokonuje translacji wszystkich adresów z zakresu 192.168.0.0–192.168.255.255 na publiczny adres 83.31.15.15. Gdy wiele komputerów z sieci lokalnej korzysta z jednego publicznego adresu IP, NAT używa dynamicznie przydzielanych portów TCP i UDP, aby rozróżnić komputery w sieci lokalnej.

NAT konsekwencje:

* Problem z identyfikacja użytkownika. Wiele adresów IP jest konwertowanych i na jeden adres publiczny stad nie wiadomo które urządzenie/użytkownik łączył się z Internetem. Potrzeba mechanizmu które śledzi logowania wewnętrznych adresów IP.
* Blokada zewnętrznego adresu IP (przes stronę, usługę) skutkuje zablokowaniem wszystkich wewnętrznych adresów ip
* Zagnieżdżone adresy portów. Aplikacje sieciowe działają w ten sposób, ze traktują przydzielony adres ip i port jako ten służący do komunikacji z internetem. Aplikacja umieszcza ten adres i port w pakietach wysyłanych do serwera. Serwer mimo usilnych starań nie da rady nic wyslac na ten serwer.
* Aplikacja korzystają z roznych portow. Pakiet wysylany jest na porcie X a spodziewa się odpowiedzi na porcie Y. NAT konwerutje prywatny adres i port i odkodowuje na ten sam port i adres.
* Wiele programów chce uzyc tego samego portu. Jeżeli wiele komputerów używa jednego publicznego adresu IP to tylko jeden z nich może używać danego numeru portu.

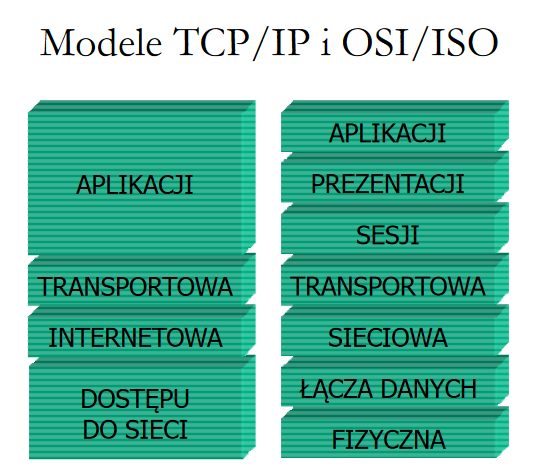
**Komunikacja unicast i multicast, jej przydatność w budowie systemów rozproszonych**

**Unicast** - to rodzaj transmisji, w której dokładnie jeden punkt wysyła pakiety do dokładnie jednego punktu - istnieje tylko jeden nadawca i tylko jeden odbiorca. Przydatność : transmisje wideo, telewizja intenetowa,

**Multicast** - to sposób dystrybucji informacji, dla którego liczba odbiorców może być dowolna. Odbiorcy są widziani dla nadawcy jako pojedynczy grupowy odbiorca (host group) dostępny pod jednym adresem dla danej grupy multikastowej. Multicast różni się od unicastu zasadą działania i wynikającą stąd efektywnością. W transmisji multicastowej po każdym łączu sieciowym dystrybuowana informacja jest przekazywana jednokrotnie, podczas gdy w unicastowej dystrybucji informacji do n odbiorców po niektórych łączach biorących udział w transmisji komunikat może być w najgorszym razie przesyłany nawet n razy. Wynika to z tego, że w transmisji unicastowej każdy komunikat ma przypisany dokładnie jeden adres docelowy, który identyfikuje jednoznacznie jeden węzeł sieci. Tak więc nawet jeśli dany komunikat po drodze do dwóch różnych węzłów sieci wykorzystuje to samo łącze, wysłane muszą zostać dwa niezależne komunikaty (o tej samej treści i innym adresie docelowym). W transmisji multicastowej unika się wielokrotnego wysyłania tego samego komunikatu do wielu odbiorców (na przykład dzięki adresowaniu grupowemu, tzn. posługiwaniu się adresami, które nie identyfikują pojedynczych węzłów sieci, lecz ich grupy). Zastoswanie: systemy konferencyjne, dystrybucja na zywo sygnalu wideo, aplikacje typu push: notowania giełdowe, wyniki, sportowe,

**Broadcast** – rozsiewczy (rozgłoszeniowy) tryb transmisji danych polegający na wysyłaniu przez jeden port pakietów, które powinny być odebrane przez wszystkie pozostałe porty przyłączone do danej sieci (domeny broadcastowej). W broadcascie nie ma weryfikacji poprawności przesłanych danych.

**Warstwa transportowa modelu OSI/ISO: podstawowe cechy protokołów TCP i UDP, port, gniazdo, asocjacja, adresacja usług sieciowych**



***TCP*** - (protokół kontroli transmisji) - strumieniowy protokół komunikacji między dwoma komputerami. TCP zapewnia wiarygodne połączenie dla wyższych warstw komunikacyjnych przy pomocy sum kontrolnych i numerów sekwencyjnych pakietów, w celu weryfikacji wysyłki i odbioru. Brakujące pakiety są obsługiwane przez żądania retransmisji. Host odbierający pakiety TCP porządkuje je według numerów sekwencyjnych tak, by przekazać wyższym warstwom modelu OSI pełen, złożony segment.

* jest zorientowany na połączenie: oznacza to, że program użytkowy, który chce skorzystać z protokółu TCP musi najpierw zwrócić się do odbiorcy z prośbą o uzyskanie połączenia i uzyskać jego zgodę;
* jest protokółem typu punkt-punkt: oznacza to, że każde połączenie TCP ma dokładnie dwa końce;
* zapewnia niezawodność: oznacza to, że protokół TCP zapewnia pełna niezawodność w dostarczaniu pakietów;
* zapewnia dwukierunkową komunikację: oznacza to, że komunikacja w połączeniu TCP odbywa się w dwu kierunkach, czyli zarówno od nadawcy do odbiorcy jak i od odbiorcy do nadawcy;
* zapewnia strumieniowy interfejs: oznacza to, że program może wysyłać połączeniem całą sekwencję bajtów, w konsekwencji prowadzi to do tego, że dane nie musza być dostarczane do odbiorcy w kawałkach tych samych wielkości, w których zostały wysłane;
* zapewnia łagodne kończenie połączenia: oznacza to, ze protokół gwarantuje niezawodne dostarczenie pakietów przed zamknięciem połączenia.
* Niezawodny
* Połączeniowy
* Strumieniowy

***UDP*** - Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do [TCP](https://pl.wikipedia.org/wiki/TCP_(protok%C3%B3%C5%82))). Nie ma też mechanizmów kontroli przepływu i retransmisji. Korzyścią płynącą z takiego uproszczenia budowy jest szybsza [transmisja](https://pl.wikipedia.org/wiki/Transmisja) danych i brak dodatkowych zadań, którymi musi zajmować się host posługujący się tym protokołem. Z tych względów UDP jest często używany w takich zastosowaniach jak [wideokonferencje](https://pl.wikipedia.org/wiki/Wideokonferencja), strumienie [dźwięku](https://pl.wikipedia.org/wiki/D%C5%BAwi%C4%99k) w [Internecie](https://pl.wikipedia.org/wiki/Internet) i gry sieciowe, gdzie [dane](https://pl.wikipedia.org/wiki/Dane) muszą być przesyłane możliwie szybko, a poprawianiem błędów zajmują się inne warstwy [modelu OSI](https://pl.wikipedia.org/wiki/Model_OSI). Przykładem może być [VoIP](https://pl.wikipedia.org/wiki/VoIP).

Protokół UDP jest pozbawiony wszystkich funkcji TCP. Oferuje usługę w której mogą wystąpić straty pakietów.

Protokół UDP jest bezpołączeniowy. Nie wymaga istnienia żadnego połączenia. Klient UDP może utworzyć gniazdo i wysłać datagram do jakiegoś serwera, po czym może natychmiast przez to samo gniazdo wysłać kolejne datagramy do różnych innych serwerów. Podobnie serwer przez jedno gniazdo może przyjmować datagramy od różnych klientów.

Należy podkreślić, że wiadomość zostanie odebrana tylko wtedy, gdy adresat oczekuje na odbiór datagramu, w przeciwnym wypadku wiadomość jest ignorowana.

Protokół jest zorientowany transakcyjnie, a dostarczenie wiadomości nie jest gwarantowane. Nie mamy żadnej informacji na temat tego, czy wysłane pakiety dotarły do celu. Nie ma też mechanizmy retransmisji (jak to było w przypadku TCP).

Komunikaty UDP mogą być gubione, duplikowane lub przychodzić w innej kolejności niż były wysłane, ponadto pakiety mogą przychodzić szybciej niż odbiorca może je przetworzyć.

Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP). Nie ma też mechanizmów kontroli przepływu i retransmisji. Korzyścią płynącą z takiego uproszczenia budowy jest większa szybkośćtransmisji danych i brak dodatkowych zadań, którymi musi zajmować się host posługujący siętym protokołem. Z tych względów UDP jest często używany w takich zastosowaniach jak wideokonferencje, strumienie dźwięku w Internecie i gry sieciowe, gdzie dane muszą byćprzesyłane możliwie szybko, a poprawianiem błędów zajmują się inne warstwy modelu OSI.

Oprócz wysyłanych danych, każdy komunikat UDP zawiera numer portu odbiorcy i numer portu nadawcy, dzięki czemu oprogramowanie UDP odbiorcy może dostarczyć komunikat do właściwego adresata oraz umożliwia wysłanie odpowiedzi.

***Socket*** - pojęcie abstrakcyjne reprezentujące dwukierunkowy punkt końcowy połączenia. Dwukierunkowość oznacza możliwość wysyłania i odbierania danych. Wykorzystywane jest przez aplikacje do komunikowania się przez [sieć](https://pl.wikipedia.org/wiki/Sie%C4%87_komputerowa) w ramach [komunikacji międzyprocesowej](https://pl.wikipedia.org/wiki/Komunikacja_mi%C4%99dzyprocesowa).

Gniazdo posiada trzy główne właściwości:

* typ gniazda identyfikujący [protokół](https://pl.wikipedia.org/wiki/Protoko%C5%82y_komunikacyjne) wymiany danych
* lokalny adres (np. [adres IP](https://pl.wikipedia.org/wiki/Adres_IP), [IPX](https://pl.wikipedia.org/wiki/IPX), czy [Ethernet](https://pl.wikipedia.org/wiki/Ethernet))
* opcjonalny lokalny numer [portu](https://pl.wikipedia.org/wiki/Port_protoko%C5%82u) identyfikujący [proces](https://pl.wikipedia.org/wiki/Proces_(informatyka)), który wymienia dane przez gniazdo (jeśli typ gniazda pozwala używać portów)

Gniazdo może posiadać (na czas trwania komunikacji) dwa dodatkowe atrybuty:

* adres zdalny (np. adres IP, IPX, czy Ethernet)
* opcjonalny numer portu identyfikujący zdalny proces (jeśli typ gniazda pozwala używać portów)

***Port*** – pojęcie związane z protokołami działającymi w intrenecie do identyfikowania procesów działających na odległych systemach. Jeden z parametrów gniazda. Zastaw dobrze znanych portów określa po rty na których typowo pracują powszechnie używane usługi. Porty efemeryczne, krótkotrwałe, określone przez moduł warstwy transportowej klienta na czas połączenia.

***Asocjacja*** - Aby 2 procesy mogły się skomunikować, należy określić elementy w następującej kolejności: protokół -> adres 1 -> port 1 -> adres 2 - > port 2. Jest to tzw. asocjacja (ang. association).

***Adresacja usług i zasobów:***

* Gniazdo docelowe (np. 200.200.200.200:80/TCP)
* URL (Uniform Resource Locator) (np. http://www.agh.edu.pl/index.html)
  + Wskazuje używany protokół
  + Wskazuje gniazdo docelowe
  + Wskazuje zasób (lub usługę)
* Inne sposoby

***Serializacja wiadomości –*** przekształcanie wiadomość tekstowej do postaci szeregowej (strumieni bajtów).