

# SDN - Software Defined Networks

## (Sieci definiowane programowo)

Software Defined Network (SDN) – Sieci definiowane programowo - jest to zbiór technik, które umożliwiają bezpośrednio, w sposób programowy zarządzać, sterować zasobami sieciowymi. Ułatwiając tym samym projektowanie, dostarczanie oraz utrzymanie usług sieciowych w dynamiczny i skalowalny sposób (bardzo często zamiast słowa **Network** używa się **Networking**).

Definicja ta pochodzi z rekomendacji ITU-T [SG13: Recommendation Y.3300], merytorycznie pokrywając się ze znaczą większością definicji, z którymi można się spotkać w literaturze. Charakteryzuje się tym, że nie utożsamia SDN z podstawowym założeniem jakim jest separacja płaszczyzny transportu od płaszczyzny sterowania, traktuje ją jedynie jako jedną z technik.

Komu potrzebna jest ta koncepcja? Przede wszystkim operatorom telekomunikacyjnym aby można było optymalizować wykorzystanie zasobów, niezależnie od ich dostawców, przy dynamicznie zmieniającym się rynku usług telekomunikacyjnych i obsługiwanego ruchu.

Podstawowym założeniami SDN, które jednocześnie umożliwiają programowalność to:

- separacja płaszczyzny transportu oraz sterowania,
- scentralizowanie funkcji sterowania oraz zarządzania siecią,
- abstrakcja sprzętu, także poprzez wirtualizację,
- niezależność rozwiązań od dostawców sprzętu,
- płaszczyzna transportu zorientowana połączeniowo (oparta na przepływach),
- programowalność zasobów.

Spełnienie ich, jest nie lada wyzwaniem dla przyszłych projektantów, ponieważ sieci radiowe, optyczne lub tradycyjne kablowe charakteryzują się skrajnie różnymi własnościami, które należy ukryć pod warstwą abstrakcji.

Potrzeby nowej architektury są tak wielkie, że największe przedsiębiorstwa branży telekomunikacyjnej oraz IT, powołały specjalne konsorcjum o nazwie „Open Networking Foundation” (ONF), którego głównym celem

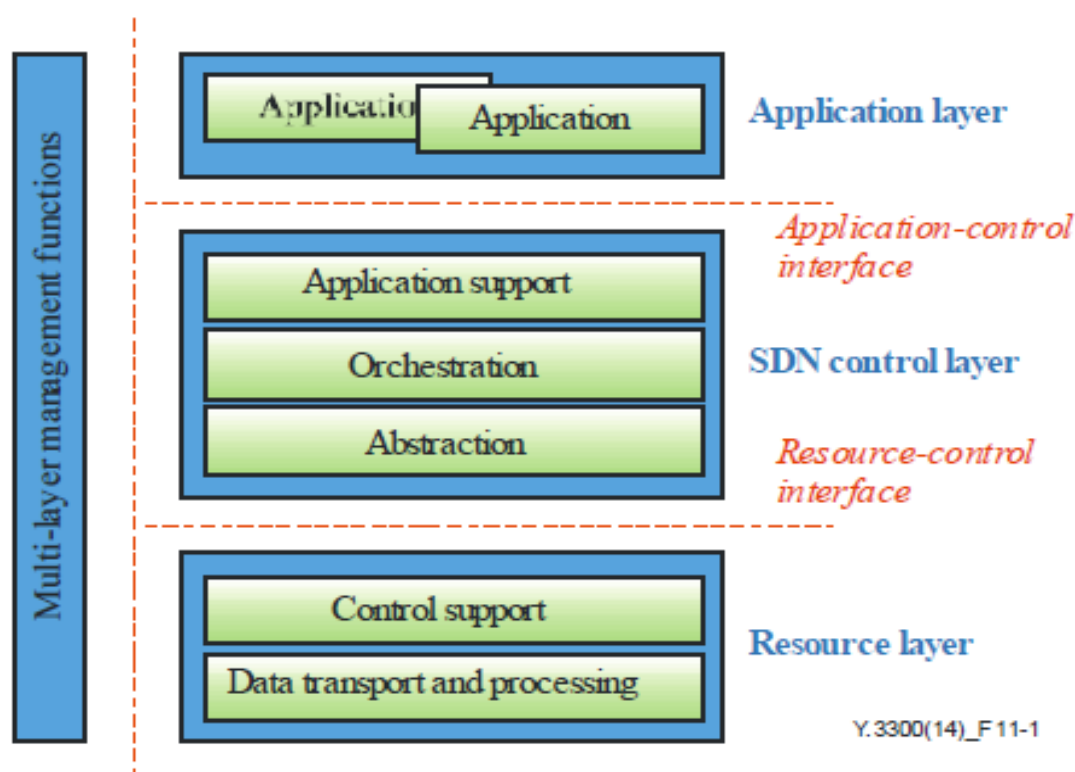
jest przyspieszenie wprowadzenia rozwiązań Sieci Definiowanej Programowo oraz technologii jej pokrewnych.

Dodatkowo zajmuje się standaryzacją protokołu „Openflow”.

Nad standaryzacją Sieci Definiowanych Programowo pracują także inne organizacje jak ITU-T, IETF, MEF, Linux Foundation oraz wiele innych.

## SDN według ITU-T

W 2014 roku ITU-T wydało rekomendację [SG13: Recommendation Y.3300] w której została opisana koncepcja Sieci Definiowanej Programowo.



Warstwa zasobów (Resources layer) – Składa się z różnego rodzaju zasobów sieciowych, wirtualnych lub fizycznych, które odpowiadają za transport i przetwarzanie danych w sposób narzucony przez warstwę wyższą. Zasoby sieciowe powinny być programowalne, po to by móc rozszerzać ich funkcjonalność, bez potrzeby instalowania wyspecjalizowanego sprzętu. Wymieniona własność umożliwia w dynamiczny sposób dodawać lub modyfikować skomplikowane przetwarzanie, takie jak przechowywanie, kodowanie, kompresja pakietów lub nawet współpracowanie z nowoczesnymi

protokołami.

**Warstwa sterowania** (Control layer) – Zapewnia dynamiczne i deterministyczne sterowanie zasobami sieciowymi (znajdujących się w warstwie zasobów) w taki sposób w jaki zostanie poinstruowana przez warstwę aplikacji. Komunikacja z zasobami odbywa się za pomocą interfejsu „resource-control”. Informacje oraz konfiguracje są przedstawione warstwie najwyższej za pomocą ustandaryzowanych modeli danych. Poziom abstrakcji tych informacji zależy od typu aplikacji, oraz natury świadczonych przez tą sieć usług.

**Warstwa aplikacji** (Application layer) – Warstwa, która umożliwia specyfikować funkcje sieciowe zorientowane na usługi w sposób programowy. Wykorzystuje do tego abstrakcyjny model danych, który jest mu udostępniany przez warstwę niższą za pomocą interfejsu „Application-Control”. Z punktu widzenia użytkownika, jest to najbardziej czytelna warstwa. Wynika to z poziomu abstrakcji, który pomija wiele niepotrzebnych szczegółów.

**Wielowarstwowe funkcje zarządzania** (Multilayer management functions) – Funkcje takie jak: wykaz sprzętu, uaktualnienie oprogramowania, naprawianie uszkodzeń, optymalizacja wydajności, inicjalizacja ustawień, konfiguracja poszczególnych warstw. Wszystkie te czynności z reguły są wykonywane przez człowieka i dotyczą każdej z warstw.

**Resource-control interface** – Jest to interfejs, który służy do interakcji warstwy sterowania z warstwą zasobów. Informacje wymieniane podczas tej interakcji uwzględniają:

- informacje sterujące dostarczane z warstwy wyższej,
- powiadomienia wynikające z działania systemu, czyli: informacje o przeciążeniach, zmiany topologii sieci itp.

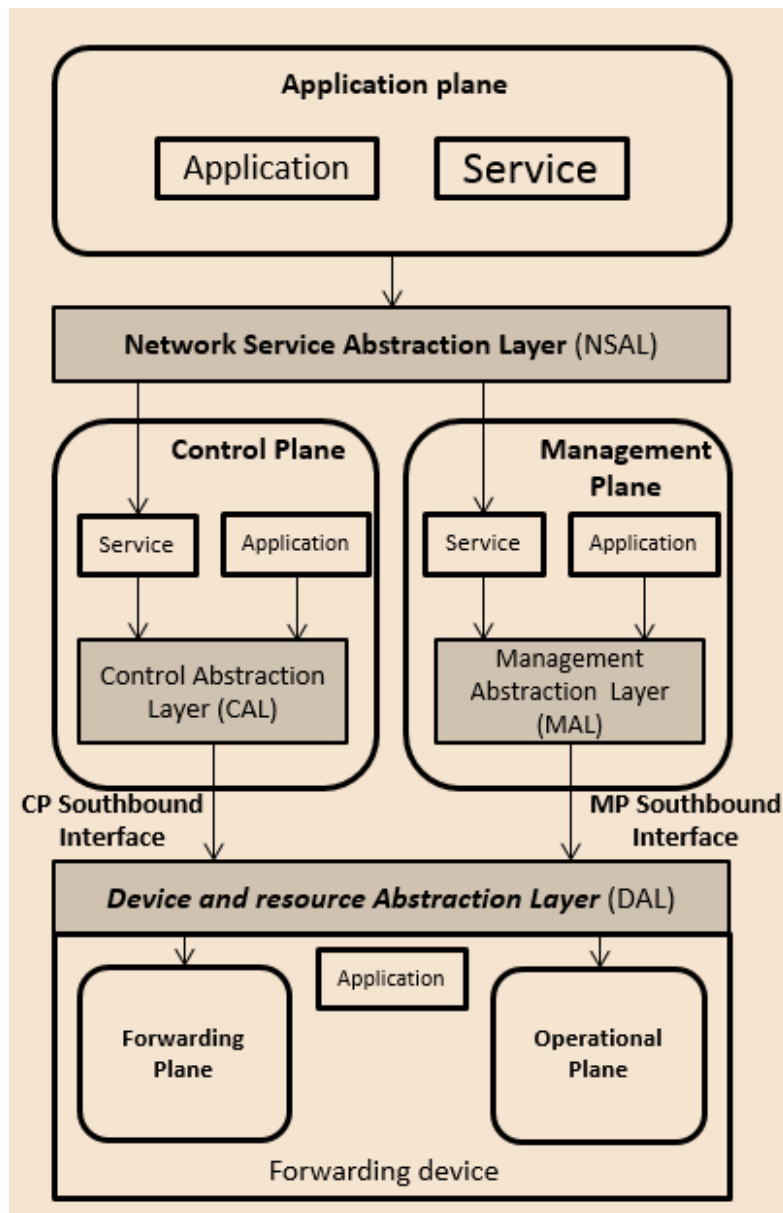
Interfejs dostarcza uniwersalny, wysokopoziomowy dostęp do zasobów sieciowych, niezależny od wykorzystywanych technologii.

**Application-control interface** – Jest to interfejs, który służy do komunikacji warstwy aplikacji z warstwą sterowania. Informacje wymieniane podczas tej interakcji uwzględniają:

- informacje zawierające sposób w jaki warstwa sterowania ma działać,
- abstrakcyjne modele sieci udostępnione przez warstwę sterowania.

## SDN według IETF

Internet Engineering Task Force (IETF) pracuje nad wydaniem standardu RFC na temat Sieci Definiowanych Programowo. Na dzień dzisiejszy dostępny jest tylko szkic, który wstępnie definiuje architekturę oraz opisuje poszczególne jej bloki funkcjonalne [IETF: SDN Layers and Architecture Terminology, Internet Draft].



### Podstawowe pojęcia

Zasób – Fizyczny lub wirtualny składnik, występujący wewnątrz systemu. Charakteryzuje się różnym poziomem złożoności. Prosty czyli port, kolejka lub złożony, które są połączeniem wielu prostych zasobów.

Przykładem złożonego zasobu może być urządzenie sieciowe.

Urządzenie sieciowe – jest to urządzenie, które wykonuje jedną bądź wiele funkcji związanych z przekazywaniem lub przetwarzaniem pakietów.

Architektura nie odróżnia urządzeń wirtualnych od fizycznych. Z punktu widzenia operacji sterujących nie ma to żadnego znaczenia.

Aplikacja (w kontekście architektury SDN) – jest to część programu, która wykorzystuje udostępnione jej usługi, do zrealizowania różnych funkcjonalności. Przykładem aplikacji, może być wyznaczanie trasy (ruting) w płaszczyźnie sterowania.

Usługi (w kontekście architektury SDN) – jest to część programu, która udostępnia API dla aplikacji lub dostęp do innych usług, aplikacjom, które znajdują się logicznie w innej warstwie.

Płaszczyzna przekazywania (Forwarding Plane) – Jest to zbiór zasobów odpowiadający za przekazywanie pakietów według zasad jakie zostaną narzucone przez płaszczyznę sterowania. Do jej zadań należą: przekazywanie, odrzucanie, modyfikacja pakietów. Omawiana płaszczyzna jest punktem końcowym dla usług oraz aplikacji płaszczyzny sterowania. W dokumencie jest podkreślone, że nie odróżnia się czy zasób jest implementacją sprzętową, programową lub hybrydową. Z punktu widzenia wyższych płaszczyzn nie ma to żadnego znaczenia.

Płaszczyzna operacyjna (Operational Plane) – Punkt końcowy dla usług płaszczyzny sterowania. Odpowiada za zarządzanie operacyjnym stanem urządzenia, czyli za wykonanie zadań utrzymaniowych narzuconych przez płaszczyznę zarządzania. Przykładem czynności wykonywanych przez płaszczyznę operacyjną mogą być:

- ustawienie portu w stan aktywny,
- monitorowanie stanu portu,
- alarmowanie o uszkodzeniach,
- ustalanie parametrów technicznych urządzenia np. moc nadajnika.

Płaszczyzna sterowania (Control Plane) - Sprawuje sterowanie urządzeniami sieciowymi. Skupia się głównie na nadzorowaniu pracy płaszczyzny przekazywania, ale ma również wpływ na płaszczyznę operacyjną. Do zadań płaszczyzny sterowania należą:

- monitorowanie topologii sieci,
- ustalanie tras (ruting) oraz przekazywanie ich płaszczyźnie przekazywania,

- dynamiczna reakcja na uszkodzenia sieci (tylko jeśli dotyczą, nadzorowanych przez nią tras),
- przetwarzanie pakietów, dla których płaszczyzna przekazywania nie zna reguły.

**Płaszczyzna zarządzania** (Management Plane) – Zbiór funkcji odpowiadających za monitorowanie, konfigurowanie oraz utrzymanie urządzeń sieciowych. W architekturze występuje jako jednostka scentralizowana. Historycznie, w większości bazowała na ludzkich działaniach. Z biegiem czasu, wprowadzono uniwersalne protokoły, które umożliwiły częściową automatyzację tych czynności. Płaszczyzna zarządzania skupia się głównie na płaszczyźnie operacyjnej, nie mniej jednak ma również wpływ na płaszczyznę przekazywania. Do jej zadań należą:

- wykrywanie uszkodzeń oraz znaczących wydarzeń,
- monitorowanie zasobów,
- konfiguracja zasobów.

**Płaszczyzna aplikacji** (Application Plane) - Wysokopoziomowe aplikacje i usługi, które korzystają z usług płaszczyzn sterowania oraz zarządzania. Przykładem aplikacji może być graficzny interfejs użytkownika, który umożliwia operatorowi rezerwowanie ścieżek, planowanie sieci itp.

**Warstwa abstrakcji usług sieciowych** (Network Service Abstraction Layer – NSAL) – Jest to warstwa umożliwiająca komunikację płaszczyzn sterowania, zarządzania oraz aplikacji. Usługi, które świadczą warstwy wzajemnie, są dostępne za pośrednictwem tej warstwy. W zależności od potrzeb, NSAL może przyjmować różne formy. Przykładowe interfejsy do usług to: NETCONF, CORBA, RESTful API, IPC oraz wiele innych. Jednak szczególnym zainteresowaniem cieszą się dwa ostatnie, które działają w trybie „Client-Server” oraz korzystają z XML/JSON do przesyłania wiadomości.

**Warstwa abstrakcji urządzeń oraz zasobów** (Device and resource Abstraction Layer – DAL) - Jest to warstwa, która przedstawia płaszczyznę operacyjną oraz przekazywania za pomocą jednego lub wielu z abstrakcyjnych modeli. Umożliwia to niezależność rozwiązań od producentów sprzętu telekomunikacyjnego. Przykładowe modele abstrakcji płaszczyzny przekazywania to ForCES oraz Openflow. W przypadku Płaszczyzny Operacyjnej jest to ForCES, model YANG opisu danych lub SNMP MIB.

# NFV - Network Function Virtualization (Wirtualizacji funkcji sieciowych)

Realizacje sprzętowe rozwiązań sieciowych są aktualnie obarczone dwoma wadami:

- Krótki czas życia (Short Life Cycle) - Operatorzy są zmuszeni do wymiany sprzętu ze względu na rosnące wymagania usług np. na przepustowość. Obecne rozwiązania równie szybko się starzeją, głównie ze względu na wydajność. Często się zdarza, że nowy sprzęt realizuje swoje funkcje w ten sam sposób. Oznacza to, że logika urządzenia się nie zmienia, a jedynie używane podzespoły elektroniczne.
- Skalowalność – Dane urządzenie w przybliżeniu pobiera tyle samo energii elektrycznej, bez względu na to jak bardzo jest obciążone. Jeśli w trakcie eksploatacji, okazuje się, że liczba urządzeń jest niewystarczająca, operator zmuszony jest dokupić kolejną „porcję” sprzętu. Co jest kłopotliwe z najprostszych przyczyn, trzeba zapewnić zasilanie oraz przestrzeń dla takich urządzeń.

Wirtualizacja funkcji sieciowych (Network Function Virtualization - NFV) jest propozycją rozwiązania tych problemów.

## Istota NFV

Polega ona na dekompozycji urządzeń funkcjonalnych, poprzez odłączenie funkcji sieciowych (logiki urządzenia) od dedykowanych rozwiązań sprzętowych.

Technologia zakłada, że funkcje sieciowe zostaną zrealizowane za pomocą powszechnie używanej w branży IT wirtualizacji. Oznacza to, że wszystkie funkcje będą realizowane za pomocą środowiska serwerowego, składającego się z prostych zasobów takich jak:

- procesory,
- pamięci RAM oraz pamięci stałe,
- karty sieciowe.

Funkcje sieciowe będą instalowane w wirtualnym środowisku zlokalizowanym w centrum bazodanowym (chmura), węźle sieciowym, lub nawet w



obrębie użytkownika końcowego (mgła, brzeg).

Wybór platformy docelowej zależy jedynie od natury funkcji oraz wymaganych parametrów jakościowych.

Na przykład niemożliwe jest, aby umieścić „firewall” zdala od brzegu sieci, ponieważ samo przeznaczenie wymaga przetwarzania ruchu przychodzącego do sieci.

NFV jest technologią wywodzącą się ze standardowej wirtualizacji z której korzysta się w branży IT, dlatego może korzystać z podobnych technik oraz charakteryzuje się podobnymi zaletami.

Największą zaletą NFV jest możliwość dynamicznego zarządzania zasobami z których korzystają zwirtualizowane funkcje. Gdy wirtualnej maszynie braknie zasobów, to są jej dostarczane, a gdy ma ich nadmiar to są redukowane.

Zmniejsza to w sposób drastyczny koszty eksploatacyjne. Fakt, że zwirtualizowane funkcje sieciowe są niezależne od sprzętu sprawia, że gdy zaprojektowany system staje się niewydajny np. z powodu starzenia się sprzętu lub zwiększenia się potrzeb użytkowników, to operator nie jest zmuszony do kupienia kolejnego dedykowanego urządzenia, które jest drogie.

W tym przypadku jedynie nabywa lub wymienia standardowy sprzęt, taki jak procesory, pamięć lub przełączniki sieciowe.

Wymienione zasoby mogą być współdzielone pomiędzy wiele różnych funkcji, co sprawia, że rozwiązanie jest bardzo ekonomiczne.

Przedstawiona idea nie jest nowa, nie brano jej do tej pory pod uwagę, ze względu na wydajność, której nie dało się zapewnić.

Przewiduje się, że obecny stan technologiczny jest już w stanie spełnić wymagania jakościowe. Z tego względu NFV w telekomunikacji znajduje się w stanie „Proof of Concept”.

Oznacza to, że czeka się na pierwsze implementacje, które udowodnią, iż technologia NFV jest w pełni realizowalna. Na dzień dzisiejszy ukazało się wiele dokumentów, przedstawiających wyniki badań nad pierwszymi implementacjami NFV.



Okazuje się, że większość z nich przedstawia technologię w pozytywnym świetle.

Już powstały otwarte projekty (np. OPNFV), których celem jest wsparcie wdrożenia technologii.

### Relacja pomiędzy SDN a NFV

1. Określenie relacji pomiędzy SDN a NFV jest kwestią problematyczną.
2. Istnieje wiele dokumentów, które nie są ze sobą spójne.
3. ETSI to organizacja, która sprawuje pieczę nad standaryzacją NFV.
4. W białej księdze jest napisane [ETSI, Network Functions Virtualisation – Introductory White Paper], że Network Function Virtualization jest technologią niezależną od SDN.
5. Podkreślone jest jednak, że równoczesne współdziałanie technologii przynosi dużo lepsze efekty, dla każdej z nich, synergia technologii.
6. Nieścisłość wynika z faktu, że wiele organizacji (m.in. ETSI) traktuje SDN jako separację płaszczyzny przekazywania od płaszczyzny sterowania. A to było już założeniem w architekturze NGN; SDN to naśladuje bo inaczej nie wpisałoby się w NGN.
7. Jeżeli spojrzeć na sprawę pod tym aspektem, to obie technologie bez problemu są w stanie działać niezależnie od siebie.