

SDN

(SOFTWARE DEFINED NETWORKING)

**Konsep dan Framework Jaringan yang
Centralized, Flexible dan Programmable**

M. Azwir

DAFTAR ISI

1	SDN (SOFTWARE DEFINED NETWORK)	1
1.1	Konsep Jaringan Existing	2
1.2	Keterbatasan Teknologi Jaringan Existing	5
1.2.1	Kompleks dan Statis	5
1.2.2	Kesulitan dalam Implementasi Suatu Policy Jaringan	6
1.2.3	Kesulitan dalam Peningkatan Skalabilitas Jaringan	6
1.2.4	Ketergantungan pada Vendor	7
1.3	Kebutuhan akan Arsitektur Jaringan Baru	8
1.4	Apa itu SDN?	12
1.5	Standar SDN	18
1.5.1	OpenFlow	19
1.5.2	Keuntungan SDN Berbasis OpenFlow	21
1.6	Perkembangan dan Implementasi SDN	24
2	ARSITEKTUR SDN	26
2.1	Arsitektur Umum SDN	27
2.2	Prinsip Arsitektur SDN	31
2.3	Komponen Arsitektur	35
2.3.1	Data plane	35
2.3.2	Controller plane	38
2.3.3	Application plane	46
3	OPENFLOW	49
3.1	Komponen Switch	51
3.2	OpenFlow Ports	53
3.2.1	Port Physical	54
3.2.2	Port Logical	54
3.2.3	Port Reserved	55
3.3	OpenFlow Tables	57
3.3.1	Pipeline Processing	57
3.3.2	Flow Table	59

DAFTAR ISI

3.3.3	Matching	61
3.3.4	Table-miss	62
3.3.5	Flow Removal	63
3.3.6	Group Table	64
3.3.7	Meter Table	65
3.3.8	Counters	67
3.3.9	Instructions	68
3.3.10	Action Set	70
3.3.11	Action List	71
3.3.12	Actions	72
3.4	OpenFlow Channel	73
3.4.1	Pesan pada Channel OpenFlow	74
3.4.2	Asynchronous	75
3.4.3	Penanganan Pesan	76
3.4.4	Koneksi Channel OpenFlow	76
3.4.5	Bundle Messages	80
3.4.6	Protokol Konfigurasi dan Managemen OpenFlow	81
4	PCE (PATH COMPUTATIONAL ELEMENT)	83
4.1	Traffic Engineering di Jaringan MPLS	84
4.2	Apa itu PCE?	87
4.3	Arsitektur Dasar	91
4.4	Protokol PCEP	92
4.5	Traffic Engineering Database (TED)	95
4.6	Multi Domain Path Computation	96
4.7	PCE dalam Arsitektur SDN	101
5	IMPLEMENTASI SDN	105
5.1	WAN (Wide Area Network)	106
5.1.1	Permasalahan di WAN	107
5.1.2	SDN untuk WAN	108
5.2	Jaringan Kampus	113
5.2.1	Kebutuhan jaringan kampus	115
5.2.2	Implementasi SDN di Jaringan Kampus	115
5.3	Jaringan Mobile	119
5.3.1	Manajemen interferensi antar cell	119

DAFTAR ISI

5.3.2	Manajemen Traffik Mobile	121
5.3.3	Implementasi Lainnya	124
5.4	Fungsi In-Line network	126
5.4.1	Load Balancer menggunakan SDN	127
5.4.2	Firewall menggunakan SDN	128
5.4.3	Instrution Detection Menggunakan SDN	129
6	MIGRASI SDN	130
6.1	Metoda migrasi SDN	131
6.2	Faktor yang Mempengaruhi migrasi	135
6.2.1	Tipe Jaringan	135
6.2.2	Layer Jaringan	137
6.2.3	Arsitektur SDN yang akan Di-deploy	138
6.2.4	Pertimbangan Keamanan	139
6.3	Use Case Migrasi	140
6.3.1	Google	140
6.3.2	NTT	155
7	INOVASI LAYANAN JARINGAN	163
7.1	Pendorong Inovasi	164
7.2	Inovasi Jaringan Berbasis SDN	166
7.2.1	Internal Inovasi	167
7.2.2	Inovasi komunitas	168
7.3	Ragam Inovasi Layanan Jaringan	168
7.3.1	Traffik Steering	169
7.3.2	Bandwidth on Demand	170
7.3.3	Bandwidth Exchange	171
7.3.4	QoS Berbayar	174
7.3.5	Virtual Managed Router	177
8	NFV (NETWORK FUNCTION VIRTUALIZATION)	182
8.1	Pendorong NFV	184
8.2	Enabler NFV	185
8.2.1	Cloud Computing	186
8.2.2	Server Standard Industry	187
8.3	Keuntungan Fungsi Jaringan Berbasis Software	187

DAFTAR ISI

8.4	Peran Virtualisasi	188
8.5	NFV ETSI	189
8.5.1	Framework NFV	191
8.5.2	Arsitektur NFV	192
8.5.3	Use Case	196
8.6	Data Plane I/O (Input/Output)	197
8.7	Otomatisasi Operasi di NFV	198
8.8	Benefit NFV	199
8.9	NFV dan SDN	200

TENTANG PENULIS

Penulis sudah berkecimpung di dunia Telekomunikasi selama lebih 18 tahun dan 12 tahun diantaranya berada pada bagian riset. Beberapa riset yang telah penulis lakukan diantaranya tentang NGN (Next Generation Network), IP-PBX, Number Portability, Electronic Numbering (ENUM) dan IP Interconnection. Buku ini merupakan buku kedua dari penulis, buku penulis yang pertama berjudul IMS (IP Multimedia Subsystem) – Framework & Arsitektur Jaringan Telekomunikasi Masa Depan. Penulis dapat dihubungi melalui email ke alamat mazwir@telkom.co.id.

1

SDN (SOFTWARE DEFINED NETWORKING)

Internet telah mendorong pembentukan suatu masyarakat digital, dimana hampir segala sesuatu dapat dihubungkan dan diakses dari mana saja. Meskipun sedemikian besar adopsi internet, jaringan IP tradisional yang merupakan pembentuk internet sangat kompleks dan proses manajemen jaringan internet sulit dilakukan. Terdapat bermacam perangkat yang digunakan di jaringan untuk

memberikan layanan yang dibutuhkan oleh pengguna. Router, switch, NAT (Network Address Translation, Firewall, WAG (Wireless Access Gateway), IDS (Intrusion Detection System) dan banyak perangkat lainnya yang dibutuhkan untuk memungkinkan terjadinya komunikasi yang sesuai dengan kebutuhan dan karakteristik tertentu dari pengguna. Masing-masing perangkat dari suatu vendor mempunyai kapabilitas tertentu yang bersifat proprietary dan berbeda dengan perangkat dari vendor yang lain, bahkan kadang dari vendor yang sama sekalipun. Demikian juga dalam cara untuk melakukan manajemen atau konfigurasi terhadap perangkat, yang membutuhkan cara yang spesifik untuk perangkat tersebut.

Pada jaringan konvensional, terdapat kesulitan dalam membuat suatu konfigurasi jaringan sesuai dengan suatu aturan yang sudah ditentukan. Demikian juga kesulitan dalam melakukan konfigurasi ulang suatu jaringan untuk merespon terhadap fault, load yang tinggi dan perubahan jaringan. Kondisi ini semakin menyulitkan karena perangkat jaringan terintegrasi secara vertikal dimana control plane dan data plane di-bundling dalam satu perangkat.

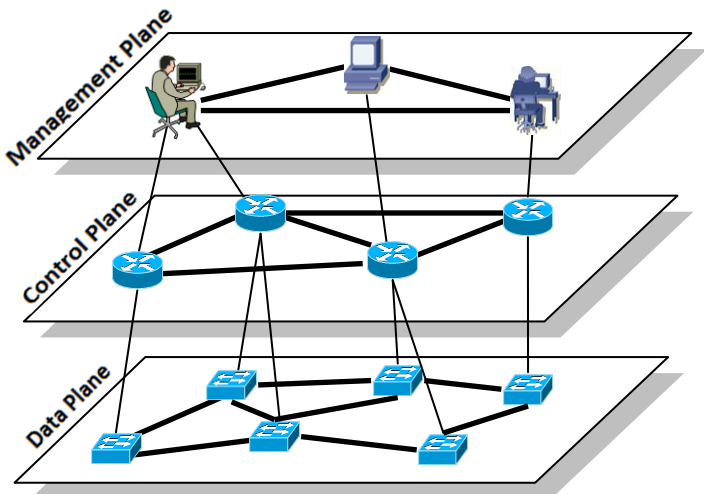
SDN merupakan suatu paradigma yang sedang berkembang yang menjanjikan solusi dari masalah yang dihadapi. SDN merupakan suatu metodologi yang memisahkan fungsi pengontrolan jaringan dan fungsi forwarding trafik, sentralisasi secara logika pengontrolan jaringan dan memperkenalkan kemampuan untuk melakukan pemrograman jaringan. Saat ini, SDN menjadi topik riset yang sedang banyak digeluti dan juga sudah mendapat penerimaan yang cukup luas di industri.

1.1 Konsep Jaringan Existing

Jaringan yang menghubungkan router dan switch menggunakan protokol transport yang terdapat di dalam

BAB 1 SDN (SOFTWARE DEFINED NETWORKING)

keduanya, merupakan teknologi kunci yang memungkinkan pengiriman informasi dalam bentuk paket digital, yang membuat dunia menjadi saling terhubung. Secara umum, jaringan komputer dapat dibagi dalam tiga bagian (plane) berdasarkan fungsi: data plane, control plane dan management plane seperti ditunjukkan pada gambar 1. Data plane terkait dengan perangkat forwarding (forwarding device), yang bertanggung jawab dalam meneruskan data. Control plane, merepresentasikan protokol yang digunakan untuk mengisi tabel forwarding dari elemen pada data plane. Management plane merupakan bagian dimana aplikasi (spt. tool berbasis SNMP), digunakan secara remote untuk memonitor dan mengkonfigur fungsi control. Network policy ditentukan di management plane, control plane melaksanakan policy dan data plane mengeksekusi policy dengan meneruskan data sesuai aturan yang telah dibuat.



Gambar 1. Hierarki Jaringan secara Layering

BAB 1 SDN (SOFTWARE DEFINED NETWORKING)

Pada jaringan IP tradisional, bagian control dan data merupakan dua bagian yang berpasangan dan digabungkan dalam perangkat jaringan yang sama, dan arsitektur secara keseluruhan merupakan suatu jaringan yang terdesentralisasi. Pendekatan ini dipandang penting pada saat-saat awal proses design internet karena merupakan cara terbaik untuk menjamin resilience jaringan yang merupakan hal krusial dari internet. Akibatnya, arsitektur jaringan yang dikembangkan menjadi sangat kompleks dan merupakan arsitektur yang relatif statik. Ini juga menjadi alasan fundamental kenapa jaringan tradisional sulit dalam proses manajemen dan pengontrolan. Ketika ingin mengimplementasikan suatu aturan tertentu pada jaringan, operator jaringan harus melakukan konfigurasi pada setiap elemen jaringan dengan menggunakan command line yang bersifat low-level dan kadang spesifik dari suatu vendor.

Untuk mendukung manajemen jaringan, beberapa vendor menawarkan solusi proprietary yang berupa hardware, sistem operasi dan aplikasi khusus. Operator jaringan harus memiliki tim dan manajemen yang khusus untuk solusi tersebut. Capex dan biaya operasi dalam membangun dan memelihara infrastruktur jaringan cukup tinggi dengan return dalam waktu yang lama. Keterbatasan ini menimbulkan hambatan dalam inovasi dan penambahan fitur dan layanan baru (seperti access control, load balancing, efisiensi energi dan rekayasa traffik).

Upaya peningkatan fungsionalitas jaringan dilakukan dengan menambah komponen khusus dan middleboxes ke jaringan seperti firewalls, Intrusion Detection System dan Deep Packet Inspection. Meskipun membantu meningkatkan fungsionalitas jaringan, efek dari perangkat tersebut menimbulkan kompleksitas dalam desain dan operasional jaringan.

1.2 Keterbatasan Teknologi Jaringan Existing

Arsitektur jaringan tradisional memiliki keterbatasan dalam memenuhi kebutuhan-kebutuhan baru baik dari enterprises, small business maupun dari user. Peningkatan kebutuhan pelanggan terkait dengan ketersediaan akses dimanapun dan kapanpun dan juga kebutuhan bandwidth yang semakin tinggi dengan tingkat keamanan yang tinggi merupakan tantangan besar yang dihadapi oleh penyedia jaringan dewasa ini. Terdapat beberapa keterbatasan arsitektur jaringan tradisional dalam memenuhi kebutuhan market yang semakin beragam.

1.2.1 Kompleks dan Statis

Seperti diuraikan di atas, kondisi jaringan saat ini sudah sangat kompleks karena faktor desain perangkat dan penambahan perangkat untuk mendukung kapabilitas tertentu. Faktor lain yang menyebabkan kompleksitas jaringan adalah banyaknya jenis protokol pada jaringan berbasis IP saat ini. Protokol-protokol ini dikembangkan untuk memenuhi kebutuhan bisnis dan teknis, seperti untuk mengirimkan data dengan performansi dan reliabilitas yang lebih tinggi, konektiviti yang lebih luas dan tingkat keamanan yang lebih tinggi. Protokol yang dikembangkan tersebut cenderung didesain secara tertutup yang dikhususkan untuk memecahkan suatu permasalahan yang spesifik.

Kondisi tersebut semakin meningkatkan keterbatasan jaringan saat ini, kompleksitas jaringan semakin tinggi sehingga jaringan menjadi tidak fleksibel. Sulit untuk memenuhi permintaan pelanggan yang dinamis dan membutuhkan dalam waktu yang cepat. Contohnya untuk menambah pelanggan baru yang membutuhkan penambahan perangkat, maka akan dibutuhkan konfigurasi pada switch, router, firewall dan juga portal autentifikasi. Serta mengupdate ACL (Access Control List), menambah

VLAN, menentukan QoS dan langkah lain yang terkait dengan mekanisme yang berhubungan dengan protokol. Selain itu juga perlu mempertimbangkan topologi jaringan, jenis switch dan bahkan versi dari software pada perangkat.

Sifat statik dari jaringan berbanding terbalik dengan sifat dinamik dari lingkungan server saat ini, dimana virtualisasi server telah menambah kebutuhan host yang membutuhkan konektiviti dan secara fundamental merubah lokasi fisik dari host. Sebelum virtualisasi, aplikasi berada pada satu server dan hanya mempertukarkan trafik dengan client tertentu. Saat ini, aplikasi terdistribusi di banyak virtual machine (VM), yang mempertukarkan trafik satu sama lain. VM sering berpindah untuk optimalisasi dan rebalancing workload server, menyebabkan end point fisik dari aliran trafik harus berubah dalam waktu singkat. Perpindahan VM merupakan tantangan bagi jaringan tradisional, baik dari sisi skema pengalamatan dan namespace sampai ke masalah segmentasi dan desain routing jaringan.

1.2.2 Kesulitan dalam Implementasi Suatu Policy Jaringan

Implementasi suatu policy jaringan dalam skala luas, harus dilakukan dengan merubah konfigurasi terhadap ribuan perangkat dan bermacam mekanisme aliran trafik. Sebagai contoh, setiap kali ingin memberikan suatu nilai QoS tertentu di jaringan, akan membutuhkan waktu berjam-jam bahkan bisa beberapa hari untuk melakukan konfigurasi nilai QoS baru untuk keseluruhan jaringan. Policy jaringan seringkali berubah untuk menyesuaikan dengan kebutuhan dan kondisi jaringan maupun karena kebutuhan pelanggan.

1.2.3 Kesulitan dalam Peningkatan Skalabilitas Jaringan

Kebutuhan akan data center saat ini tumbuh sangat cepat,

yang akan berdampak pada munculnya peningkatan kebutuhan akan jaringan. Jaringan menjadi lebih kompleks dengan penambahan ratusan hingga ribuan perangkat jaringan baru yang harus diatur dan dikonfigur. Beban jaringan menjadi sedemikian besar sehingga harus mampu memperhitungkan skalabilitas jaringan berdasarkan prediksi dari pola trafik yang terjadi. Akan tetapi pola trafik di data center saat ini luar biasa dinamik sehingga tidak dapat diprediksikan.

Raksasa-raksasa penyedia layanan internet seperti Google, Amazon dan Facebook, menghadapi kendala skalabilitas yang sangat besar. Pemain-pemain besar ini menggunakan algoritma pemrosesan parallel dalam skala yang besar dan menghubungkan dataset melalui keseluruhan pool computing. Dengan semakin meningkatnya scope dari aplikasi end-user (sebagai contoh, crawling dan indexing keseluruhan konten web untuk memberikan hasil pencarian ke user dengan sangat cepat), jumlah dari elemen computing sangat besar dan pertukaran dataset di antara node computing dapat mencapai hingga petabytes. Perusahaan-perusahaan ini membutuhkan yang disebut hyperscale network yang dapat menyediakan high-performance dan konektiviti yang berbiaya rendah antara ratusan hingga ribuan server fisik. Proses peningkatan skalabilitas seperti itu tidak dapat dilakukan dengan konfigurasi secara manual.

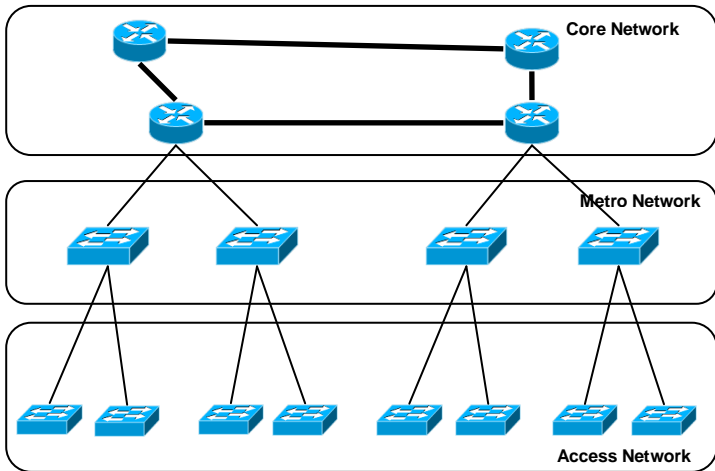
1.2.4 Ketergantungan pada Vendor

Operator dan enterprise berusaha untuk men-deploy kapabilitas dan layanan baru dengan cepat sebagai respon terhadap perubahan kebutuhan bisnis atau kebutuhan pelanggan. Akan tetapi, kemampuan mereka untuk merespon terkendala oleh ketersediaan perangkat dari vendor, permasalahan dari sisi standar, dan ketiadaan

interface yang bersifat terbuka (open interface) yang membatasi kemampuan operator jaringan untuk membentuk jaringan sesuai dengan kebutuhan operator.

1.3 Kebutuhan akan Arsitektur Jaringan Baru

Pertumbuhan yang luar biasa jumlah perangkat mobile dan konten, virtualisasi server dan munculnya layanan cloud merupakan beberapa faktor yang mendorong industri jaringan untuk meninjau kembali arsitektur jaringan tradisional. Kebanyakan jaringan konvensional dibentuk dalam struktur yang hierarki, dibangun dengan tier switch ethernet yang diatur dalam suatu struktur yang berbentuk pohon seperti ditunjukkan pada gambar 2. Desain seperti itu cocok ketika komputasi client-server bersifat dominan, sebaliknya arsitektur statik seperti itu kurang tepat untuk komputasi dan storage yang bersifat dinamik yang dibutuhkan oleh enterprise data center, kampus dan carrier. Beberapa dari trend perubahan yang mendorong kebutuhan terhadap paradigma jaringan baru diantaranya:



Gambar 2. Struktur Jaringan Pohon

- a. Peningkatan penggunaan trafik data
Jumlah user yang menggunakan layanan data meningkat sangat drastis yang didorong oleh banyak faktor seperti semakin murah biaya komunikasi dan perangkat, ketersediaan jaringan dan device, peningkatan konten dan aplikasi, digital lifestyle dan kemudahan interaksi dengan bermacam layanan yang tersedia. User sudah menggunakan akses data untuk segala macam kebutuhan dan beragam aktivitas yang berdampak pada trafik yang semakin tinggi yang harus dilayani oleh jaringan. Jaringan dituntut untuk memberikan kualitas terbaik sehingga semua aktivitas online user dapat berjalan dengan baik. Selain kualitas, jaringan juga harus berjalan secara efisien sehingga resource yang tersedia dapat dimanfaatkan secara maksimal. Beberapa keterbatasan pada jaringan

konvensional tidak mampu memenuhi kebutuhan jaringan yang semakin kompleks.

b. Perubahan pola trafik

Pola trafik yang terjadi ketika seorang user mengakses suatu layanan sudah berubah secara signifikan. Jika dahulu layanan diberikan secara client server dimana suatu komunikasi yang intens terjadi antara satu client dengan satu server. Saat ini aplikasi mengakses database dan server yang berbeda yang membentuk komunikasi machine-to-machine sebelum mengirimkan informasi ke user. Jenis perangkat yang digunakan user untuk mengakses suatu layanan sudah semakin banyak, baik dengan menggunakan notebook, smartphone maupun tablet. User juga semakin mudah mengakses ke jaringan sehingga dapat menggunakan layanan kapanpun dan dimanapun. Kebutuhan penggunaan jaringan tidak hanya untuk mengakses jaringan publik, tapi juga mengakses jaringan korporasi yang membutuhkan tingkat keamanan yang lebih tinggi. Kondisi ini menimbulkan perubahan pada pola trafik yang berjalan menjadi semakin bervariasi.

c. Munculnya BYOD

User yang menggunakan perangkat pribadi seperti smartphone, tablet, PC convertible dan notebook untuk mengakses jaringan korporasi semakin meningkat, yang memunculkan gaya hidup baru di dunia kerja atau dikenal dengan BYOD (Bring Your Own Device). Administrator jaringan di perusahaan harus mampu mengakomodasikan akses dari perangkat tersebut dengan tetap memproteksi data yang dimiliki perusahaan sehingga tidak menimbulkan dampak negatif seperti kebocoran data perusahaan yang bersifat rahasia atau tersebarnya malware dan virus di server perusahaan.

d. Tumbuhnya Layanan Cloud

Layanan cloud mulai tumbuh cukup baik terutama digunakan oleh perusahaan skala besar (enterprise). Pengguna layanan cloud menginginkan untuk dapat mengakses aplikasi, infrastruktur dan resource IT lainnya yang tersedia di cloud dengan lebih mudah dan secara on demand. Penyediaan fungsi provisioning yang dapat dilakukan secara mandiri (self-service) akan membutuhkan skalabilitas yang elastis dari komputansi, storage dan resources jaringan. Teknologi saat ini masih sangat terbatas dari sisi kemampuan untuk mendeliver jaringan secara elastis dan fleksibel.

e. Big Data

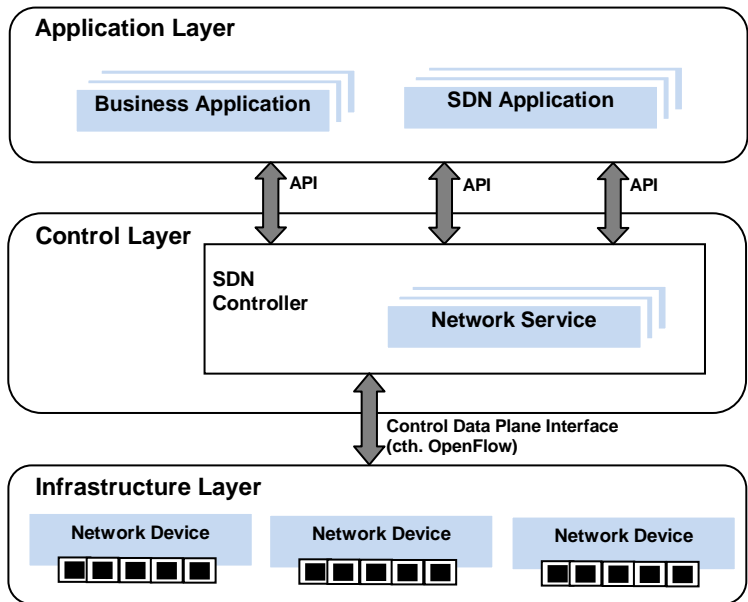
Big data berarti bandwidth yang lebih besar, menangani big data saat ini atau mega dataset membutuhkan pemrosesan secara paralel pada ribuan server, semuanya membutuhkan koneksi langsung antara satu dengan lainnya. Lonjakan pada jumlah data diiringi oleh kebutuhan terhadap penambahan kapasitas jaringan di data center. Pada data center yang sangat besar, jaringan yang digelar juga demikian besar yang menghubungkan koneksi any-to-any tanpa ada yang terputus.

1.4 Apa itu SDN?

SDN merupakan paradigma jaringan yang baru berkembang untuk mengubah keterbatasan yang ada pada infrastruktur jaringan saat ini. Terminologi SDN mula-mula ditujukan untuk merepresentasikan ide dan pengembangan OpenFlow di Universitas Stanford. SDN me-refer pada suatu arsitektur jaringan dimana fungsi forwarding yang terdapat pada data plane diatur oleh control plane yang terpisah. Pemisahan fungsi kontrol yang sebelumnya melekat di perangkat jaringan menjadi tersentralisasi secara logic akan

BAB 1 SDN (SOFTWARE DEFINED NETWORKING)

memungkinkan abstraksi infrastruktur underlying. Infrastruktur yang diabstraksi ini dapat digunakan oleh aplikasi dan layanan jaringan, dimana jaringan diperlakukan sebagai suatu entitas virtual.



Gambar 3. Arsitektur SDN

Gambar 3 di atas menunjukkan arsitektur logika SDN yang memusatkan intelegensia jaringan di controller, yang dapat melihat kondisi jaringan secara menyeluruh. Kondisi dan elemen jaringan akan diabstraksi ke aplikasi sebagai suatu elemen logika dari switch. SDN akan sangat menyederhanakan kapabilitas perangkat jaringan, perangkat tidak perlu lagi harus mengerti dan memproses ratusan protokol tapi hanya perlu menunggu perintah dari controller.

BAB 1 SDN (SOFTWARE DEFINED NETWORKING)

SDN mengkonsolidasi control plane dimana satu aplikasi melakukan pengontrolan terhadap beberapa elemen data plane. SDN control plane akan melakukan pengontrolan langsung terhadap status pada elemen data plane seperti switch dan router, melalui suatu API yang sudah ditentukan. Salah satu contoh API yang digunakan untuk fungsi tersebut adalah OpenFlow.

Definisi SDN sebagai suatu arsitektur jaringan mempunyai beberapa karakteristik utama, yaitu:

1. Pemisahan antara control plane dan data plane. Fungsi control dihilangkan dari perangkat jaringan sehingga perangkat tersebut menjadi elemen forwarding yang sederhana. Control plane berfungsi dalam menentukan keputusan terhadap penanganan trafik, sedangkan data plane yang meneruskan trafik berdasarkan keputusan dari control plane. Pemisahan antara control plane dan data plane dapat direalisasikan dengan menggunakan interface pemrograman antara switch dan controller SDN yang dapat berupa suatu API. Salah satu API yang banyak digunakan sebagai interface tersebut adalah OpenFlow. Suatu switch yang mendukung OpenFlow akan dapat diatur oleh controller untuk bertindak sebagai sebuah router, switch, firewall atau melakukan peran lainnya seperti load balancer, traffic shaper dan lainnya.
2. Keputusan dalam meneruskan paket yang datang berdasarkan pada flow. Suatu flow terdiri dari match field yang menjadi kriteria dalam menfilter paket yang data dan instruksi dimana di dalamnya terdapat tindakan (action) yang akan dilakukan jika paket yang diterima sesuai dengan kriteria. Abstraksi flow memungkinkan penggabungan sifat dari tipe perangkat jaringan yang berbeda seperti router, switch, firewall dan middlebox. Pemrograman flow memungkinkan fleksibilitas yang belum pernah ada sebelumnya, dibatasi hanya oleh

kapabilitas dalam mengimplementasi flow table.

3. Fungsi kontrol dari perangkat jaringan dipindahkan ke entitas eksternal, yang disebut SDN Controller atau Network Operating System (NOS). Controller adalah suatu platform software yang berjalan pada server dan menyediakan abstraksi dari resource jaringan untuk memfasilitasi pemrograman terhadap forwarding device berdasarkan pada view jaringan secara menyeluruh.
4. Jaringan dapat diprogram melalui aplikasi atau software yang berjalan di atas controller dan berinteraksi dengan perangkat data plane. Ini merupakan karakteristik fundamental dari SDN, yang dapat dianggap sebagai nilai utama dari arsitektur SDN.

Salah satu yang paling penting bagi operator jaringan dan administrator adalah dapat melakukan konfigurasi melalui aplikasi terhadap abstraksi jaringan daripada harus mengetik puluhan sampai ribuan baris konfigurasi ke perangkat jaringan yang tersebar dimana-mana. Dengan pemanfaatan controller SDN, administrator jaringan dapat mengubah sifat dan perilaku jaringan secara riil time dan mendeploy aplikasi baru dan layanan jaringan dalam kurun waktu yang sangat cepat. Dengan melakukan sentralisasi status jaringan di control layer, SDN meningkatkan fleksibilitas dalam mengkonfigur, mengatur, mengamankan dan mengoptimalkan resources jaringan melalui aplikasi SDN yang otomatis dan dinamik. Penyedia jaringan dapat menulis program sesuai dengan kebutuhan tanpa harus menunggu penambahan fitur oleh vendor perangkat.

Arsitektur jaringan SDN mendukung satu set API yang memungkinkan untuk implementasi layanan jaringan yang bersifat umum seperti routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, optimalisasi processor dan storage, penggunaan energy dan semua bentuk policy management dan

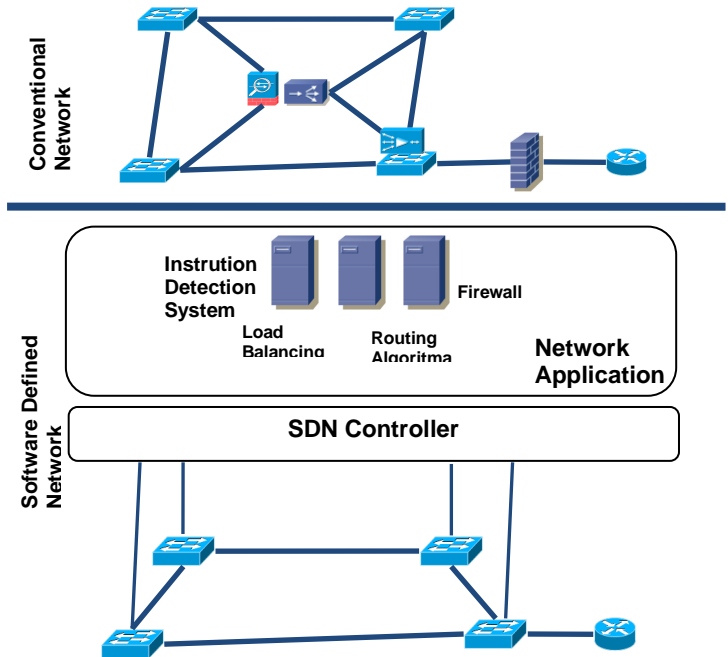
BAB 1 SDN (SOFTWARE DEFINED NETWORKING)

kustomisasi layanan untuk memenuhi kebutuhan bisnis. Layanan jaringan ini dapat digunakan untuk berbagai macam keperluan, seperti penerapan suatu policy penggunaan jaringan bagi user yang menggunakan koneksi wireline dan wireless di suatu kampus, aturan routing berdasarkan jenis aplikasi dan QoS yang fleksibel untuk layanan komunikasi riil time.

SDN memungkinkan juga untuk mengontrol keseluruhan jaringan melalui sistem orkestrasi dan provisioning yang memiliki tingkat intelegensia yang tinggi. ONF (Open Networking Foundation) sedang mengembangkan suatu API yang bersifat terbuka untuk memperkenalkan manajemen yang multi-vendor, yang membuka kesempatan untuk mengalokasikan resources secara on-demand, provisioning dengan cara self-service, jaringan yang benar-benar tervirtualisasi, dan layanan cloud yang aman.

Sentralisasi secara logic dari fungsi kontrol secara khusus memberikan beberapa keuntungan tambahan. Pertama, lebih sederhana dan lebih sedikit error untuk memodifikasi network policy melalui bahasa pemrograman highlevel, dibandingkan konfigurasi spesifik perangkat yang bersifat low level. Kedua, sebuah program control dapat secara otomatis bertindak terhadap perubahan yang tidak sesuai di jaringan sehingga menjaga policy berjalan dengan baik. Ketiga, sentralisasi dari fungsi kontrol di controller yang mengetahui seluruh kondisi dan status jaringan menyederhanakan pengembangan layanan, aplikasi dan fungsi jaringan yang lebih canggih.

BAB 1 SDN (SOFTWARE DEFINED NETWORKING)



Gambar 4. Perbandingan antara Arsitektur Layanan Jaringan SDN vs Konvensional

Seperti dijelaskan sebelumnya penyatuan antara data plane dan control plane menyulitkan dalam penambahan fungsi baru pada jaringan tradisional. Pengenalan fitur baru akan mengalami kesulitan dalam proses konfigurasi perangkat ke dalam jaringan, load balancer, Intrusion Detection System dan firewall merupakan contoh-contoh yang umum. Middlebox harus ditempatkan pada lokasi yang strategis di jaringan, sehingga semakin sulit untuk melakukan perubahan terhadap topologi jaringan, konfigurasi dan

fungsionalitas. Kondisi ini ditunjukkan pada gambar 4. Sebagai contoh, suatu Intrusion Detection System mungkin mengharuskan untuk menerima suatu traffic yang diduplikasi dari semua perangkat switching dari jaringan melalui link fisik atau logic. Sebaliknya, pengenalan fungsi baru di SDN dilakukan secara sederhana dengan menambah suatu aplikasi software baru yang berjalan di atas Network Operating System (NOS) atau controller. Pendekatan ini memiliki beberapa keuntungan :

1. Pengembangan aplikasi menjadi lebih mudah karena abstraksi yang disediakan oleh control platform dapat dishare.
2. Semua aplikasi dapat memanfaatkan informasi jaringan yang mencakup keseluruhan kondisi dan status jaringan.
3. Aplikasi dapat menjalankan suatu action (contohnya menkonfigur ulang perangkat forwarding) dari lokasi manapun di jaringan.
4. Integrasi dari aplikasi berbeda menjadi lebih mudah. Contohnya aplikasi load balancing dan routing dapat dikombinasikan secara sekuensial, dimana keputusan load balancing dilakukan terlebih dahulu sebelum penerapan aturan routing.