

SDN/NFV 기반 5G 통신망 인프라의 진화

| 작 성 | SKT 신상호 매니저 (sanghoshin@sk.com)

- 『AI Network Lab 인사이트』는 인공지능, 클라우드, 5G 등 4차 산업혁명의 핵심인 지능정보기술과 네트워크 신기술에 대한 동향을 간략하고 심도 있게 분석한 보고서입니다.
- 본 연구보고서는 과학기술정보통신부의 방송통신발전기금조성사업, 한국정보화진흥원의 미래네트워크선도시험망 구축운영사업의 연구과제 결과이며, 한국정보화진흥원/경기도경제과학진흥원과 공동 기획하였습니다.
- 본 보고서의 내용의 무단 전재를 금하며, 가공인용할 때는 반드시 출처를 『한국정보화진흥원(NIA)』이라고 밝혀 주시기 바랍니다.

발행처 한국정보화진흥원

발행인 문용식

기획 한국정보화진흥원 지능형인프라본부 인프라기획팀

보고서 온라인 서비스 www.nia.or.kr



Contents

보고서 요약

(1) 보고서 요약	5
------------------	---

보고서 주요 내용

(1) 5G 망의 Web Scale로의 진화	7
(2) NFV (Network Function Virtualization) 및 SDN (Software Defined Network) 화 ...	8
(3) VNF (Virtual Network Function)의 마이크로 서비스화 (MSA)	10
(4) VNF의 컨테이너화 (Containerization)	12
(5) 5G VNF의 컨테이너화 이슈 I - Kubernetes Networking	13
(6) 5G VNF의 컨테이너화 이슈 II - Kubernetes 도입에 따른 운영 환경 변화 ...	15
(7) 5G 표준화와 MSA	15
(8) 결론 및 시사점	17
참고문헌	18

개요

- 지난 4월 세계 최초로 우리나라에서 5G 서비스가 상용화가 되었다. 물론 코어망은 LTE망을 활용하는 NSA (Non-Standalone) 방식이지만 세계 최초로 5G 단말을 통한 5G 서비스의 상용화라는 점에서는 의미가 크다고 할 수 있다.
- 하지만 5G 통신망의 구축은 이제부터이다. 5G SA (Standalone) 방식을 지원하는 코어망의 통신 장비가 내년부터 상용화가 되기 시작하고, Radio 또한 초저지연을 지원하는 28GHz를 사용하는 mmWave 또한 내년부터 상용화 될 예정이다.
- 5G가 지향하는 초저지연, 초고속, 초대용량 서비스를 원활하게 지원하기 위해서는 이러한 코어 통신 장비, Radio 장비의 진화뿐만 아니라 통신 인프라 자체의 진화 또한 필수적이다. 기존 LTE 망에서 시도되었던 SDN/NFV를 통한 인프라의 가상화를 뛰어넘어 Cattle 시스템을 통한 WebScale로의 진화가 요구 된다.
- 이번 이슈 리포트를 통해서 5G망 인프라의 WebScale로의 진화를 위해서 어떤 요소들이 필요하며, 현재 이러한 요소들의 진화 상태 및 개발 및 도입에 걸림돌이 되는 이슈들에 대해서 알아보려고 한다.

보고서 요약

(1) 5G 망의 Web Scale로의 진화

- 5G 서비스를 위해서는 통신 장비의 진화뿐만 아니라 통신망의 인프라 또한 기존의 Pets 시스템에서 Cattle 시스템에 기반 한 WebScale로의 진화가 필수적이다.

(2) NFV (Network Function Virtualization) 및 SDN (Software Defined Network) 화

- LTE망에서 처음으로 시도된 NFV 및 SDN 도입을 통한 인프라의 가상화가 필수적이다.
- SDN/NFV의 도입을 통해 SW와 HW 분리를 통한 TCO 절감 또는 SW와 HW의 독립적 진화에 그치지 않고 가상 네트워크의 도입 및 인프라의 클라우드화까지 진화가 되어야 한다.

(3) VNF (Virtual Network Function)의 마이크로 서비스화 (Micro-Service Architecture - MSA)

- 통신 인프라의 Pets 시스템에서 Cattle 시스템으로의 변화를 위해서는 VNF의 MSA화를 통한 각 모듈의 간소화 (lightweight) 및 각 모듈 간의 SBA (Service Based Architecture)에 기반한 통신 방식이 필요하다.

(4) VNF의 컨테이너화 (Containerization)

- VNF의 MSA 기반으로 설계된 모듈을 컨테이너를 통해 개발하면 각 모듈의 배포 및 장애 관리, 업그레이드가 용이해져, 컨테이너는 통신망 인프라의 WebScale화에 최적의 배포방식이다.
- 몇몇 주요 통신 장비 벤더에서는 이미 VNF의 컨테이너화를 시작하였으며 내년에 구축되는 5G SA 장비 중 일부는 컨테이너 기반으로 구축될 것으로 예상된다.

(5) 5G VNF의 컨테이너화 이슈 I - Kubernetes Networking

- 가상화된 통신 장비는 TCO 절감을 위해서 물리 장비와 동일한 네트워크 성능을 보장해야하며, 이를 위해서 SR-IOV, OVS-DPDK와 같은 가속 기술을 사용한다.
- 그러나 컨테이너의 기본 관리 툴로 사용되는 Kubernetes의 네트워크는 아직까지

OVS-DPDK를 완전하게 지원하지 않아, 이를 위한 상용 수준의 기술 개발이 필요하다.

(6) 5G VNF의 컨테이너화 이슈 II - Kubernetes 도입에 따른 운영 환경 변화

- 현재 통신 장비는 Pets 시스템으로 시스템 장애 시에 알람이 발생하고 즉각적으로 복구가 진행되고 다시 서비스에 투입되는 반면, Kubernetes는 Cattle 시스템 기반으로 컨테이너 장애 발생 시 즉각 새로운 컨테이너로 대체되어 서비스를 유지하게 된다. 통신망의 운영에서도 이러한 환경을 받아들이기 위한 교육 및 정책의 재정의가 필요하다.

(7) 5G 표준화와 MSA

- 올해 말에 5G 표준의 Phase II인 Release 16이 배포가 될 예정이다. 이미 지난 Release 15에 5G 코어 모듈의 MSA가 어느 정도는 반영이 되어 있지만, Release 16에는 더욱 더 그 모듈이 세분화 될 예정이다.

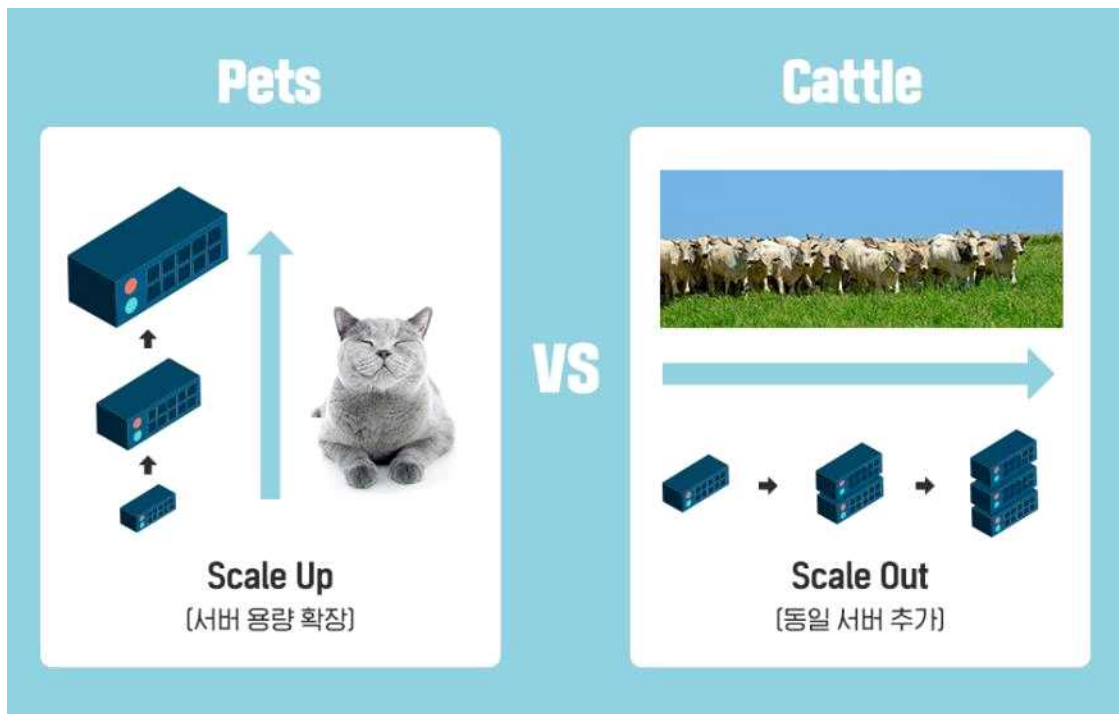
※ 시사점

- 5G 서비스를 위해서는 통신 장비 뿐 만 아니라 통신 인프라의 WebScale로의 진화가 필수적이다.
- 통신 인프라의 WebScale로의 진화를 위해서는 통신 인프라의 SDN/NFV에 기반한 인프라의 가상화, 5G 표준에 기반한 가상 통신 장비의 MSA화 및 컨테이너화, 이를 통한 통신 인프라의 완전한 클라우드화가 필수적이다.
- 이를 위한 기술적인 개발이 더욱 가속화 되어야 하면 통신사의 운영 정책의 변화 또한 같이 수반이 되어야 한다.

주요 내용

(1) 5G 망의 Web Scale로의 진화

- Web Scale 이란 구글, 아마존, 넷플릭스, 페이스북과 같은 초대규모 클라우드 시스템 또는 그 아키텍처를 의미하며, WebScale은 Cattle 시스템으로 구축된다.
- ‘Cattle과 Pets’, 이 용어는 대규모 시스템을 구축하는 서로 다른 두 접근 방식을 설명할 때 많이 사용되었던 비유이며, Cattle은 ‘가축’이라는 뜻으로 들판에서 키우는 수백 수천 마리의 들소 떼를 의미하며, Pets는 집에서 애지중지 키우는 평생을 주인과 같이 하는 반려 동물이다. 이를 IT적으로 비유하여 Pets는 다른 것으로 쉽게 대체 될 수 없는, 결코 죽어서는 안 되는 시스템을 의미하고, 이에 반해 ‘Cattle’은 자동화된 툴을 이용하여 두 개 이상의 서버로 구성되며 운영자의 간섭 없이 자동으로 새로운 서버로 대체될 수 있는 시스템을 말한다.



〈Pets와 Cattle 시스템 개념도 [1]〉

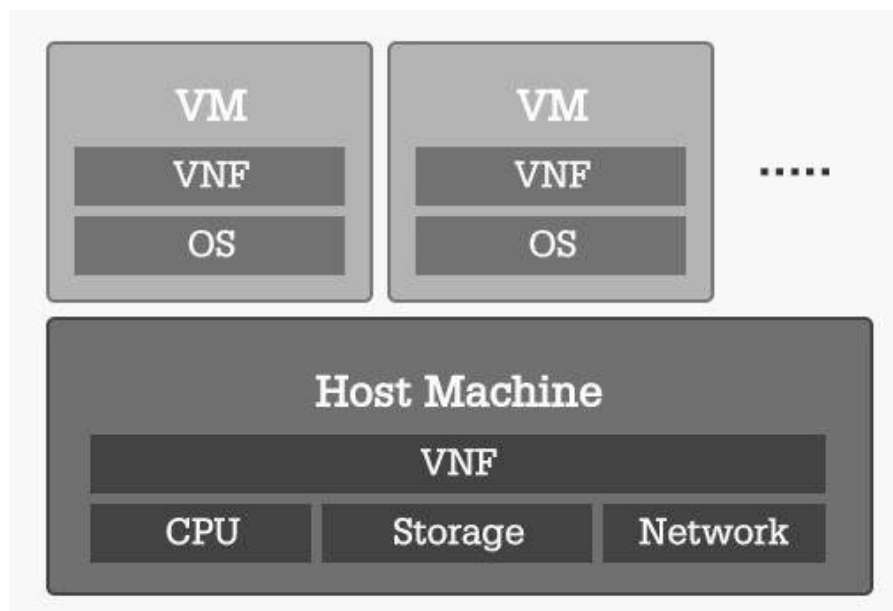
- 결국 Pets 시스템에서는 시스템에 장애가 생기면 시스템을 즉각적으로 장애를 복구하

여 서비스에 영향도를 최소화하지만, Cattle 시스템에서는 하나의 시스템에 장애가 생기면 장애 시스템의 즉각 복구 대신 해당 시스템의 동일한 기능을 가진 시스템이 서비스를 이어받아서 서비스의 연속성을 보장하게 된다.

- 현재의 통신망은 절대 죽어서는 안 되는 Pets 구조에 가까운 시스템으로 구성되어 있으며, 5G와 같은 초연결, 초대용량 시스템을 초고속으로 처리하기 위해서는 Cattle 시스템 기반으로 구성된 Web Scale로 진화해야 함. 그럼 통신망 시스템이 Web Scale로 진화하기 위한 요건 및 이슈에 대해서 알아보도록 한다.

(2) NFV (Network Function Virtualization) 및 SDN (Software Defined Network) 화

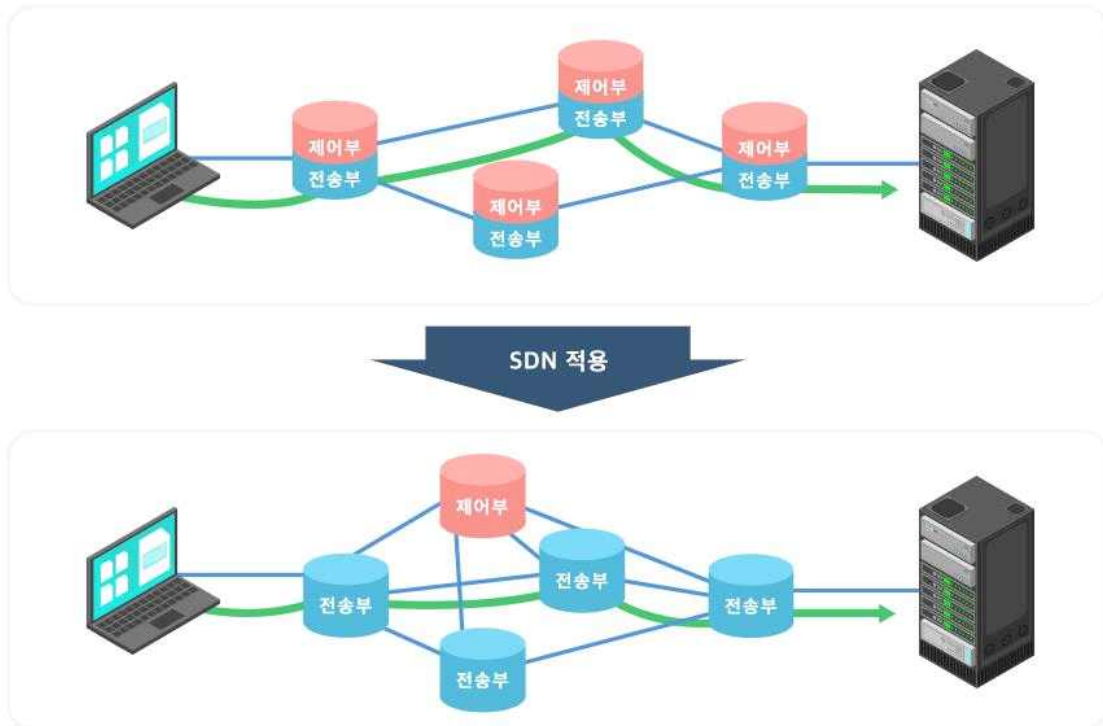
- 통신망 시스템의 WebScale화를 위한 첫 번째 요건은 NFV화 및 SDN의 도입이다. SDN/NFV에 대해서는 깊게 다루는 대신 간단한 정의 및 이슈에 대해서만 언급하고자 한다.
- NFV란 통신 시스템 구축에 필요한 구성 컴포넌트 (Network Function)을 가상화 (Virtualization) 하는 것이다. 즉, NFV란 Network Function SW를 HW 독립적으로 개발하고 이를 x86 기반의 일반 서버 상에 설치하여 운영할 수 있도록 하여 Network Function의 SW와 HW를 분리하여 SW와 HW를 독립적으로 유지 보수 및 업그레이드 할 수 있도록 하여 각각의 성능은 최적화하고 비용을 낮추는 구조이다.



< NFV 개념도 [2] >

- 국내 통신사를 비롯한 세계 주요 통신사는 지난 몇 년 전부터 통신 및 네트워크 장비에 대한 가상화를 진행해왔다. 하지만 그 가상화 정도는 가상화 기술 및 운영비용 이슈로 인해 초기 수준에 머물러 있다. 기존 물리 통신 장비에 설치된 소프트웨어를 가상 머신 상에 올릴 수 있도록 하여 물리 장비와의 분리에는 성공하였으나 완전한 클라우드화에는 이르지 못하였다.
- 여기에서 클라우드화란 Network Function VM이 동일 스펙을 가지는 어떤 HW에든지 옮겨 다니며 용량 증설이 필요할 경우 VM을 쉽게 증설 (Scale-Out) 함으로써 서비스 용량을 늘리며 사용량이 줄면 VM을 다시 회수 (Scale-In) 하여 Cloud 자원을 효율성 (Utilization)을 높일 수 있는 것을 의미한다.
- 현재의 많은 가상화된 통신 장비 (Virtualized Network Function - VNF)는 가상화로 인한 성능 저하를 막기 위해 PCI-Passthrough 또는 SR-IOV와 같은 가상화 레이어를 우회하는 기술을 사용한다. 이로 인해 해당 기술을 사용하는 VNF는 미리 지정된 서버에만 설치가 가능하고, 서버가 변경되거나 미리 설정된 서버 외의 다른 서버에 설치될 경우에는 NFV의 설정을 다시 해주어야 하여, 자동화된 Scale-Out 및 Scale-In이 불가능하다. 이러한 제약을 완화하기 위해서 OVS-DPDK와 같은 기술이 개발되었으나 여전히 패킷을 전달하는 Process에 전용 CPU Core를 할당하는 CPU pinning과 같은 설정이 필요하여 완전 자동화는 쉽지 않다.
- SDN은 네트워크 장비의 제어부 (Control Plane)과 전송부 (Data Plane)을 분리하여 제어부와 전송부를 독립적으로 진화 및 업그레이드 할 수 있도록 하여 그 성능은 높이고 비용은 낮추자는 개념이다. 원칙과 목적 차원에서는 SW 및 HW를 분리하는 NFV와 유사하다고 할 수 있다.
- SDN의 목적은 단순하게는 제어부와 전송부 분리를 통한 비용 절감이라고 할 수 있으나 SDN을 도입하는 Service Provider 입장에서는 네트워크의 제어권을 독립적으로 구성함에 따라서 네트워크의 제어권을 Service Provider가 가지고 오는 의미가 더 크다고 할 수 있다.
- SDN은 Google이나 Facebook과 같은 글로벌 서비스 업체에서 도입하여 그 효율성은 입증하였으며 몇 년 전부터 통신사에서도 물리 서버 간의 물리 네트워크 설정 및 운영 효율화를 위해 도입되고 있으나, 아직까지 가상 머신 간의 네트워크를 위한 가상 네트워크에는 그 적용이 미미한 상황이다. 그 이유는 앞서 이야기한 성능 이슈로 인한 가상

화 레이어의 우회로 인해 가상 네트워크를 적용한다고 해도 VM의 자유로운 이동 (Migration) 이나 Scale-In/Out이 불가능하기 때문에 그 효용성이 거의 없기 때문이다.



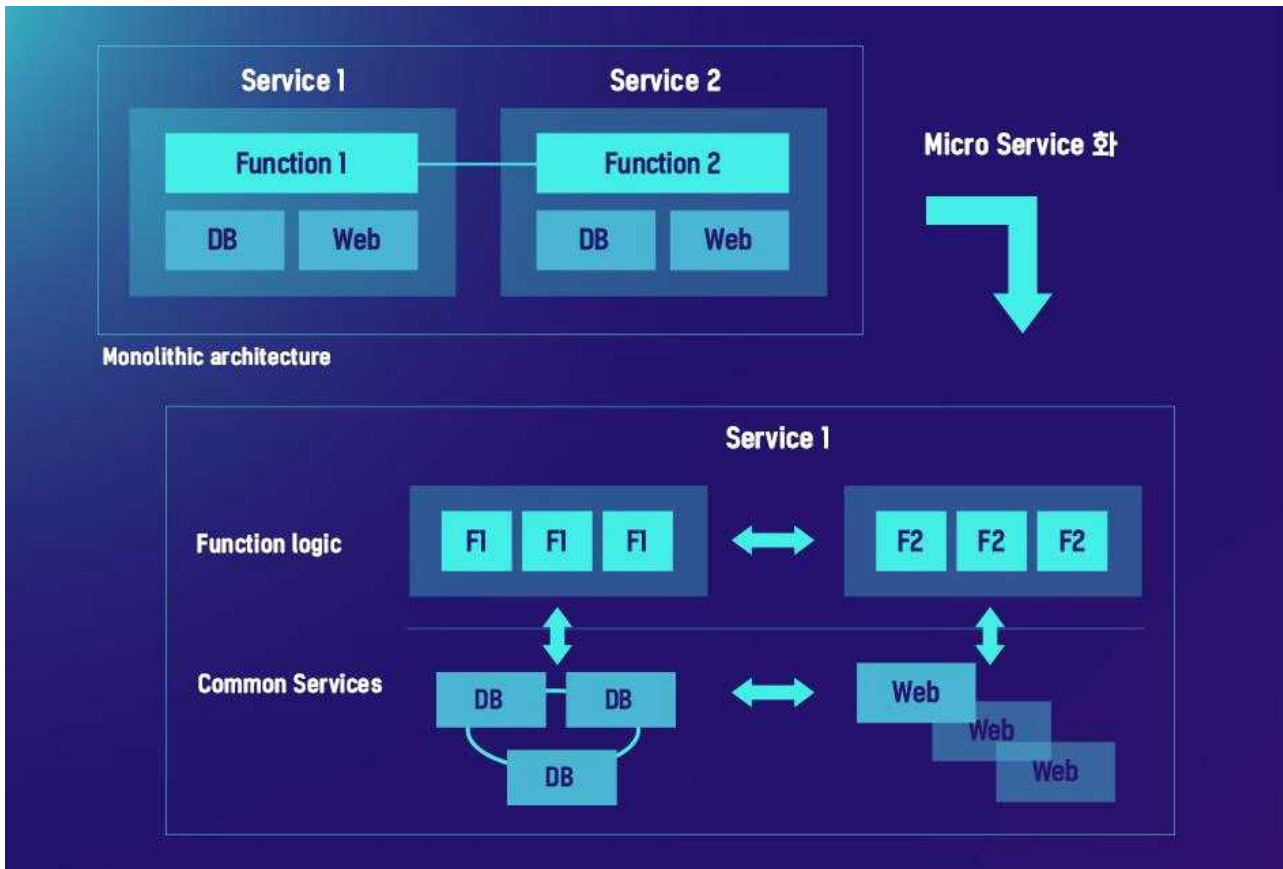
〈 SDN의 개념도 [3] 〉

- 결국 SDN/NFV는 통신망에 적용이 되어 있지만 앞서 설명한 여러 가지 이슈로 인해 그 적용 정도는 초기 수준이며 그 효과 또한 SW 및 HW의 분리 구매로 인한 TCO 절감 정도로 한정되어 있다. 이를 해결할 수 있는 것이 통신망의 클라우드화이며 WebScale화이다.

(3) VNF (Virtual Network Function)의 마이크로 서비스화 (Micro-Service Architecture - MSA)

- MSA란 최소 기능 단위를 독립적으로 실행 또는 배포가 가능한 모듈로 구성하며 각 모듈 간에는 잘 정의된 인터페이스를 이용하여 통신하는 소프트웨어 아키텍처를 의미한다. 이와 반대의 개념으로는 모든 기능을 단일 모듈 안에 다 구현하는 모놀리틱 (Monolithic) 아키텍처가 있다.

- MSA에서는 각 모듈은 모듈 간 인터페이스만 유지하면 독립적으로 진화가 가능하며, 단일모듈 장애 시에 다른 모듈에 직접적인 영향을 주지 않는다. 또한 Scale-Out 시에는 모듈 독립적으로 Scale-Out이 가능해야 한다. 따라서, 이러한 특성은 앞서 설명한 Cattle 시스템을 구성하는 데 가장 적합한 소프트웨어 아키텍처라고 할 수 있다.



〈 Mirco-Service Architecture의 개념도 [5] 〉

- MSA의 또 하나의 장점은 공통 모듈 구성을 통한 모듈의 공유이다. 위의 그림과 같이 Monolithic Architecture 에서는 각 서비스 기능 마다 각자의 DB와 Web 서버를 가지고 있어서 서비스 모듈이 나중에 추가될 때 마다 DB와 Web 서버가 새로 추가되어야 한다. 그 반면에 MSA에서는 DB와 Web 서버를 Common Service로 구성하며 Common Bus (일반적으로 Web 기반 Service 통신 사용)를 이용하여 서비스 기능과 통신을 하게 되며, 새로 추가되는 서비스 기능 또한 DB와 Web 서버 등을 새로 구성하는 대신 이 Common Service를 사용하게 된다.

(4) VNF의 컨테이너화 (Containerization)

- 통신망의 WebScale화를 위한 마지막 구성 요소는 VNF의 컨테이너화이다.
- 컨테이너란 하나의 단위 기능을 실행하기 위해 필요한 실행 코드, 라이브러리, 및 설정을 모두 포함한 패키징 (Packaging) 방식을 의미한다. 컨테이너는 다른 컨테이너들과 운영체제(정확히는 OS Kernel)를 공유하며 다른 컨테이너들과 독립된 프로세스로 실행이 된다.
- 이에 반해 가상 머신 (Virtual Machine 또는 VM)은 독립적인 운영체제의 모든 기능들을 포함하고 있으며 여러 개의 응용 프로그램 또는 프로세스를 실행할 수 있다. 따라서, 컨테이너는 가상머신에 비해 용량이 작아 가벼우며 (lightweight) 생성속도 또한 훨씬 빠르다.
- 반면, 가상 머신은 독립적인 운영체제를 사용하기 때문에 보안이나 안정성에 있어 컨테이너보다 우수한 장점도 있다. 이러한 컨테이너의 특성은 앞서 설명한 MSA 소프트웨어 아키텍처 및 Cattle 시스템의 요구 사항을 모두 만족하는 소프트웨어 패키징 방식이다.
- VNF에 이러한 아키텍처를 적용하게 되면, VNF 기능을 작은 독립적인 모듈로 재설계하고 이 모듈들을 각각 컨테이너화 하여 배포할 수 있게 된다. 여러 개의 동일한 기능을 갖는 컨테이너가 하나의 모듈 기능을 하고, 이 컨테이너 중 하나 또는 여러 개가 장애를 일으키면 바로 컨테이너를 제거 (Terminate) 하고 아주 짧은 시간 내에 필요한 수만큼의 컨테이너를 재생성하여 대체를 하게 된다.
- 시스템 업그레이드가 필요하면 여러 모듈 중 필요한 모듈만 수정하여 업그레이드가 가능하고, 업그레이드 방식 또한 여러 개의 컨테이너를 한 번에 업그레이드 하는 대신 하나씩 순차적으로 업그레이드 (Rolling Upgrade) 함으로써 서비스의 지속성도 유지할 수 있다. 시스템의 Scale-Out 또한 기능 확장이 필요한 모듈의 컨테이너 수만 늘려주면 쉽고 빠르게 가능하게 된다.



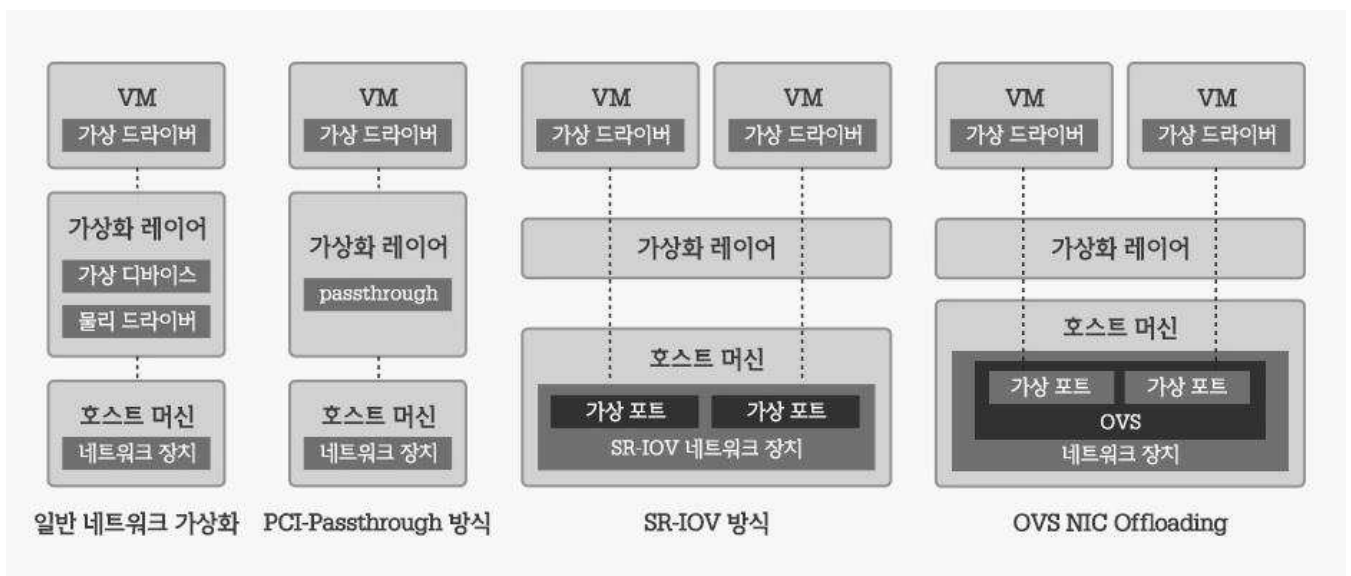
〈 VM 및 Container 아키텍처의 비교 [4] 〉

(5) 5G VNF의 컨테이너화 이슈 I - Kubernetes Networking

- VNF의 컨테이너화에 대한 이슈는 크게 컨테이너 네트워크 문제 및 운영의 이슈로 나눌 수 있다.
- 그 중 가장 큰 이슈는 컨테이너 네트워크 이슈이며, 정확하게는 컨테이너의 관리 솔루션인 Kubernetes 네트워크의 이슈이다. Kubernetes는 2015년에 오픈 소스로 출시된 이후 현재는 전 세계적으로 가장 많이 사용되는 컨테이너 관리 솔루션이며, VNF 벤더들 또한 Kubernetes를 사용하여 VNF를 컨테이너화하는 것을 최우선으로 고려하고 있다.
- Kubernetes는 CNI (Container Networking Interface) [6] 라고 하는 네트워크 인터페이스를 제공하며 현재 십여 가지가 넘는 Kubernetes 네트워크 솔루션으로 등록이 되어 있다. 그런데 문제는 앞서 언급한 VNF의 네트워크 성능 저하를 막기 위해서 사용되는 OVS-DPDK [7]를 지원하는 Kubernetes용 네트워크 솔루션이 없

다는 것이다.

- SR-IOV (Single-Root I/O Virtualization) 와 같은 가상 네트워크 레이어를 완전히 우회하는 솔루션을 위한 Kubernetes 네트워크 솔루션은 개발이 되어 일부 솔루션에는 이용되어 있으나, 이를 사용하면 앞서 이야기 한 바와 같이 이 기술을 사용해서 개발된 컨테이너화된 VNF는 미리 설정된 서버에서만 실행이 되고 다른 서버에 설정의 변경 없이는 실행이 될 수는 없다. 이는 서버 오류나 서버 다운으로 해당 서버의 컨테이너가 죽으면 Kubernetes가 이를 인지하여 바로 다른 서버에 띄워주는 Kubernetes의 가장 큰 장점을 활용할 수 없게 된다.
- 따라서, 제어를 위한 Signal 처리를 담당하는 Control Plane VNF는 트래픽 처리 용량이 크지 않아서 OVS-DPDK를 사용할 필요가 없어 컨테이너화가 가능하나, 대용량 사용자 데이터 트래픽을 처리하는 Gateway 와 같은 VNF는 성능 극대화 차원에서 OVS-DPDK를 사용해서 개발이 되어야 하기 때문에 그 개발이 지연되고 있다.
- 다행히 Kubernetes용 OVS-DPDK Plugin [8] 은 Intel사에서 개발 중에 있기 때문에 5G SA 향 VNF의 도입 시기에 맞추어 그 기능과 성능이 상용 수준까지 개발이 된다면 5G SA향 Data Plane VNF의 컨테이너화도 가능할 것으로 보인다.



< 네트워크 가상화 가속 기술 [2] >

(6) 5G VNF의 컨테이너화 이슈 II - Kubernetes 도입에 따른 운영 환경 변화

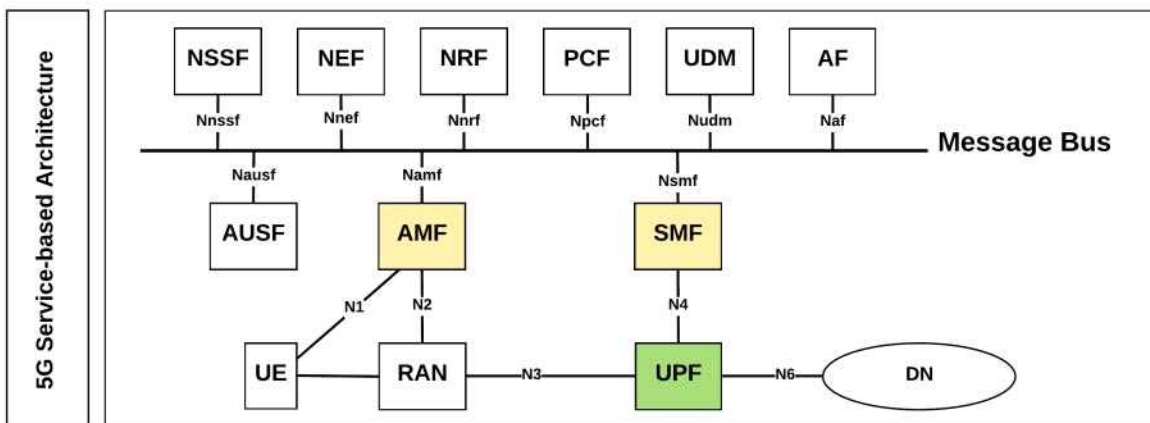
- Kubernetes의 장점은 자동 배포 또는 업그레이드 용이성 등 여러 가지가 있지만 그 중에서 대체 불가능한 장점은 자동 복구이다. Kubernetes는 Pod라는 컨테이너의 관리 단위를 사용하여 이를 모니터링하고 Pod가 어떤 이유로 동작을 멈추면 이를 바로 감지하여 삭제하고 새로운 Pod를 생성한다. 물론 Pod의 삭제 및 생성에는 어느 정도 (일반적으로 수초에서 수십초) 간의 시간이 걸리므로 이 동안의 서비스의 연속성을 위해 동일한 Pod를 다른 서버에 여러 개를 띄워서 HA (High Availability)를 보장한다.
- 통신망에 들어가는 VNF 또한 HA를 위해 하나 이상의 동일한 VM을 실행하여 하나의 VM이 동작을 멈추면 다른 VM이 서비스를 이어받는 HA 구조를 이미 지원한다. 동작을 멈춘 VM을 자동으로 살려서 다시 이를 서비스할 수 있게 하는 것은 기술적으로는 가능하나 일반적으로는 운영에는 이 정책을 적용하지는 않는다. 대부분의 경우는 이를 오류로 인지하고 알람이 발생하고 운영자가 문제를 분석 한 후 수동으로 조치를 취한 후에 복구 하여 서비스에 투입하게 된다. 여기에서 기존 운영 방식과 Kubernetes의 운영 방식과의 차이가 발생하게 된다.
- 본 리포트의 서두에서 언급한 바와 같이 통신망은 Pets 시스템이며 WebScale은 Cattle 시스템이다. 기본적으로 Cattle 시스템은 시스템은 죽을 수 있다는 것을 전제로 하고 설계를 한다. Pets 시스템 또한 시스템의 오류 또는 서버 고장 등을 고려하여 항상 HA 구조를 유지하지만, 이는 심각한 장애로 인식이 되는 반면, Cattle 시스템에서는 일상적인 오류로 인식되고 바로 새로운 시스템으로 대체된다.
- 따라서, 작게는 Kubernetes의 Pod의 재시작을 오류로 인식하고 알람을 띄우고 장애 상황을 발생시킬 것인지, 아니면 서비스에 지장이 없거나 잦은 반복적인 상황이 발생하지 않는 한 Warning 정도로 넘어 갈지에 대한 운영 방식의 변화 및 결정이 필요하다.
- 물론 이러한 이슈 및 대책은 실제 운영에서 발생한 상황이 아닌 예상이며, VNF의 컨테이너화에 따른 실제 이슈는 운영하면서 적지 않게 발생할 것으로 예상된다.

(7) 5G 표준화와 MSA

- 지금까지 설명한 바와 같이 통신망 인프라를 Web Scale 아키텍처로 변화하기 위해서

는 VNF의 아키텍처부터 운영 방식까지 많은 부분이 바뀌어야 한다. 따라서, 이를 상용화하기 위해서는 VNF의 개발을 담당하는 VNF 벤더, 통신 인프라 구축 및 운영을 책임지고 있는 통신사가 모두 함께 움직여야 한다.

- 우리나라는 이미 통신 3사에서 5G 서비스가 상용화가 되었다. 하지만 이는 4G인 LTE의 코어 망과 5G의 무선망을 결합한 NSA (Non-StandAlone) 방식이다. 5G SA 코어 망과 5G 무선망을 사용하는 SA (StandAlone) 방식은 내년에 도입될 예정이다. 현재 개발 중인 5G SA용 VNF는 글로벌 주요 통신사의 요구로 인해 마이크로 서비스화하고 컨테이너를 이용해서 준비 중이다. 따라서, 5G SA 향 인프라는 컨테이너 중심의 Web Scale 아키텍처로 구축될 것이다.
- 이를 위한 기술적 준비가 되는 시점은 빨라도 5G 기술 표준화 관점에서 살펴보면 2020년 말에 될 것으로 보인다. 현재 5G 표준은 3GPP라는 세계 통신 표준 단체에서 규정하고 있으며 5G의 첫 표준 규격인 Release 15 [9] 이 '18년 6월에 발표되었다. 이는 5G Phase I이며, 다음 5G 규격인 Release 16 [10] 이 5G Phase II인데 이는 올해 19년 12월에 발표가 될 예정이다. 5G 표준 회의에 참여하는 분들에 의하면 Phase I에도 어느 정도 모듈 분리가 이루어져 있지만, Phase II에서 MSA가 본격적으로 적용이 될 예정이며 이 표준을 이용하여 구현된 모듈을 컨테이너화를 했을 때 그 효과가 극대화될 것이라고 한다. 따라서, 실제로 5G 인프라가 Web Scale로 발전할 수 있는 계기가 마련되는 시점은 5G Phase II 발표 (2019년 12월) 이후인 2020년 말일 것이다.



AMF: Access & Mobility Management Function NEF: Network Exposure Function UDM: Unified Data Function
 SMF: Session Management Function NRF: NF Repository Function AUSF: Authentication Server Function
 UPF: User Plane Function NSSF: Network Slice Selection Function PCF: Policy Control Function

< 3GPP 5G 표준 코어 아키텍처 [12] >

(8) 결론 및 시사점

지난 4월 세계 최초로 우리 나라가 5G 서비스의 상용화에 성공하였다. 하지만 이는 NSA기반이며 5G SA 망에 대한 구축과 투자는 내년부터 본격화될 예정이다.

5G가 제공하는 초저지연, 초고속, 초대용량 서비스를 안정적으로 지원하기 위해서는 WebScale로 인프라가 구축이 되어야 한다. WebScale 구축을 위한 요건은 SDN/NFV의 도입을 통한 통신 인프라의 가상화, VNF의 MSA화 및 컨테이너화, Kubernetes와 같은 컨테이너 관리 툴을 이용한 운영 환경의 변화 등이 필요하다.

SDN/NFV는 우리 나라를 포함한 주요 통신사에서는 LTE 망에서부터 도입을 이미 시작하였으나, 그 가상화 정도는 여러 가지 요인으로 인해 SW와 HW의 분리를 통한 TCO 절감 정도에 그치고 있다. 다행히 5G 망은 LTE 망과는 달리 모든 인프라의 가상화가 기본 전제이며, 코어망의 표준부터 Web 기반 통신을 기반으로 한 SBA (Service-based Architecture) 및 MSA를 채택하고 있다.

가상화된 통신 장비를 개발하는 벤더에서도 이 표준을 따라서 모든 모듈을 MSA 기반으로 설계를 하고 더 나아가 모듈의 컨테이너화까지 진행하고 있다. 하지만, 아직까지 VNF의 완전화 컨테이너화에는 Kubernetes Network의 OVS-DPDK 지원과 같은 기술적으로 넘어야 할 이슈가 있다.

벤더 뿐만 아니라 통신사에서도 기존의 Pets 시스템 운영 방식에서 컨테이너화된 VNF 및 Kubernetes와 같은 컨테이너 관리 툴을 이용한 Cattle 시스템의 운영 방식로의 전환이 필요하다. 이는 새로운 기술에 대한 이해를 위한 운영자의 교육 뿐만 아니라 장애의 재정의의 포함한 전반적인 운영 방식 및 정책의 변화 또한 필요하다.

참 고 문 헌

- [1] Telco 인프라, Pets에서 Cattle로의 진화 <https://www.sktinsight.com/106424>
- [2] 다가올 5G 시대의 핵심? SDN 기반 가상 네트워크를 이용한 NFV
<https://www.sktinsight.com/100548>
- [3] 네트워크 세상을 바꾼 SDN (<https://www.sktinsight.com/98995>)
- [4] 5G 인프라와 마이크로 서비스 아키텍처 <https://www.sktinsight.com/112011>
- [5] Production-Grade Container Orchestration, <https://kubernetes.io/>
- [6] Kubernetes Network Plugins,
<https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/>
- [7] Open vSwitch with DPDK, <http://docs.openvswitch.org/en/latest/intro/install/dpdk/>
- [8] userspace-cni-network-plugin , <https://github.com/intel/userspace-cni-network-plugin>
- [9] 3GPP 5G Release 15, <https://www.3gpp.org/release-15>
- [10] 3GPP 5G Release 16, <https://www.3gpp.org/release-16>
- [11] 코어 네트워크 진화 - 5G 서비스 기반 구조,
<https://www.netmanias.com/ko/post/blog/13206/5g/core-network-evolution-5g-service-based-arhcitecture>