RAD
Your Network's Edge

# New Opportunities for Timing with SDN and NFV

ITSF-2015, Edinburgh

**Yaakov (J) Stein**
CTO

# Agenda

- Introduction to **SDN** and **NFV**
- **Dynamicity** and overlay networks
- **Function relocation** and distributed grand masters
- **Function repackaging** and decomposed network clocks

# Today's communications world

Your Network's Edge

Today's infrastructures are composed of many different Network Elements (NEs)

- sensors, smartphones, notebooks, laptops, desk computers, servers,
- DSL modems, Fiber transceivers,
- SONET/SDH ADMs, OTN switches, ROADMs,
- Ethernet switches, IP routers, MPLS LSRs, BRAS, SGSN/GGSN,
- NATs, Firewalls, IDS, CDN, WAN aceleration, DPI,
- VoIP gateways, IP-PBXes, video streamers,
- performance monitoring probes , performance enhancement middleboxes,
- etc., etc., etc.

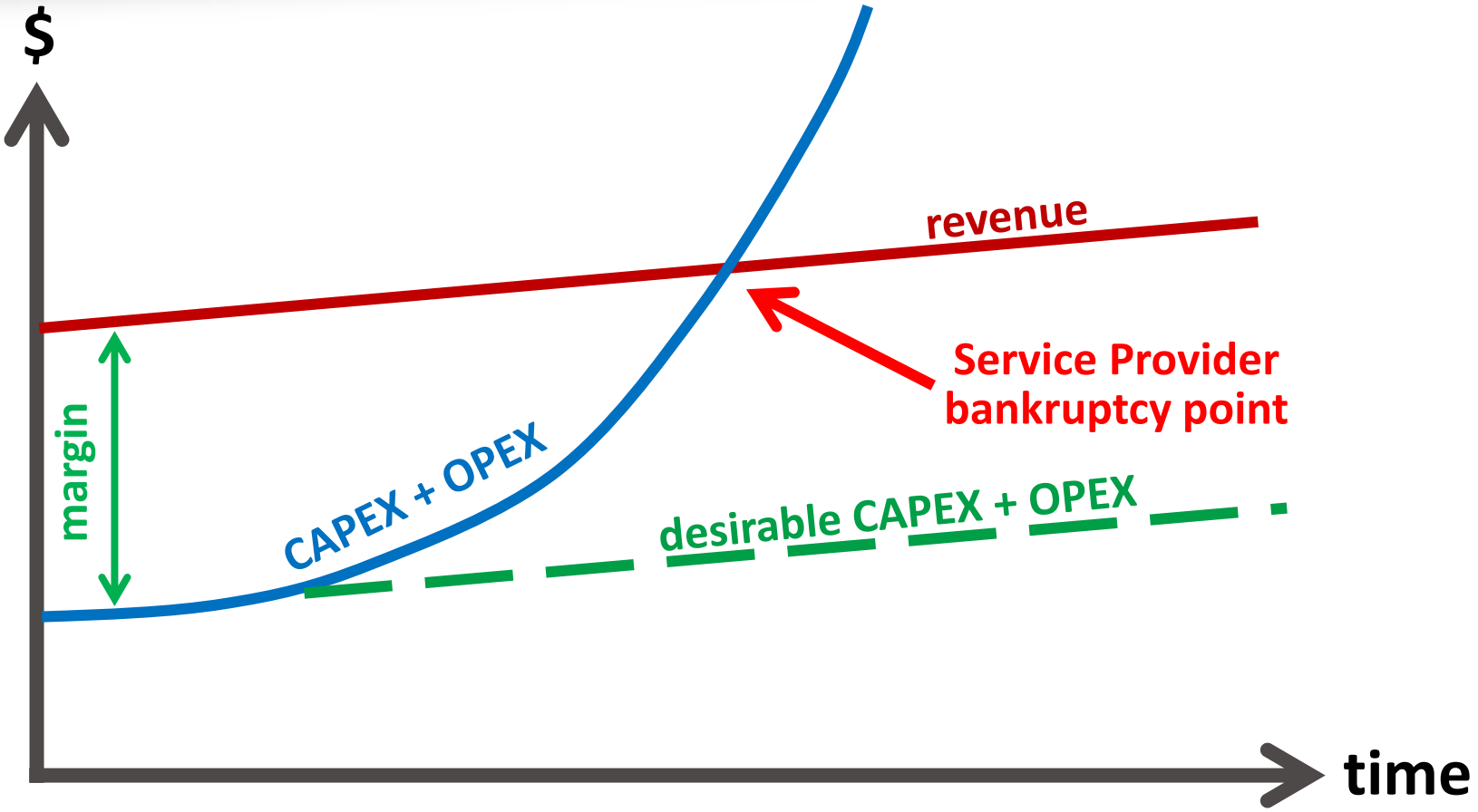
New and ever more complex NEs are being invented all the time
and while equipment vendors like it that way
Service Providers find it hard to shelve and power them all !

Service innovation is accelerating
but the lifetime of new services is decreasing

New services tend to be more challenging

- consuming more raw bandwidth
- requiring lower delay
- requiring tighter timing

# The service provider crisis




A *qualitative* picture of the service provider's world :

- revenue is *at best* increasing with number of users (a slow linear increase)
- expenses are proportional to bandwidth – doubling every 9 months

This situation obviously can not continue forever !

# Two complementary solutions

Your Network's Edge

**Software Defined Networks (SDN)**

*SDN* advocates replacing standardized networking protocols
with centralized software applications
and replacing complex NEs with dumb *whitebox switches*

Advantages:

- simplifies devices deployed in the network
- centralized control enables stronger optimization
- new functionalities may be speedily deployed, relocated, and upgraded

**Network Functions Virtualization (NFV)**

*NFV* advocates replacing hardware network elements
with software running on COTS *whitebox servers*
that may be housed in POPs, data centers, and/or customer premises

Advantages:

- COTS server price and availability scales with end-user equipment
- functionality can be located where-ever most effective or inexpensive
- new functionalities may be speedily deployed, relocated, and upgraded

# Why use SDN/NFV ?

Your Network's Edge

At first glance SDN and NFV do not seem relevant for network timing functions

- highly accurate timing requires hardware
  (compare software to hardware time-stamping !)
- timing needs to be delivered over the network to where it is needed
  (otherwise it is not *network* timing)
- timing is conventionally thoroughly planned and uses static routes

A closer look shows that there are three main reasons
people use SDN and/or NFV :

1. **dynamicity**
2. **function relocation**
3. **function repackaging**

We shall explain each of these
and show use cases for each in timing contexts

# Dynamicity

SDN differs from classical Network Management Systems
in the time scales over which it is designed to function

- NMS typically set up, in minutes, services that last from days to years
- SDN sets up, in milliseconds, services that last from seconds to hours

In fact, SDN frequently works in *reactive* mode
where the path is only determined after the data has already started flowing!

Conventional timing services are planned over long periods of time
and are expected to run effectively forever

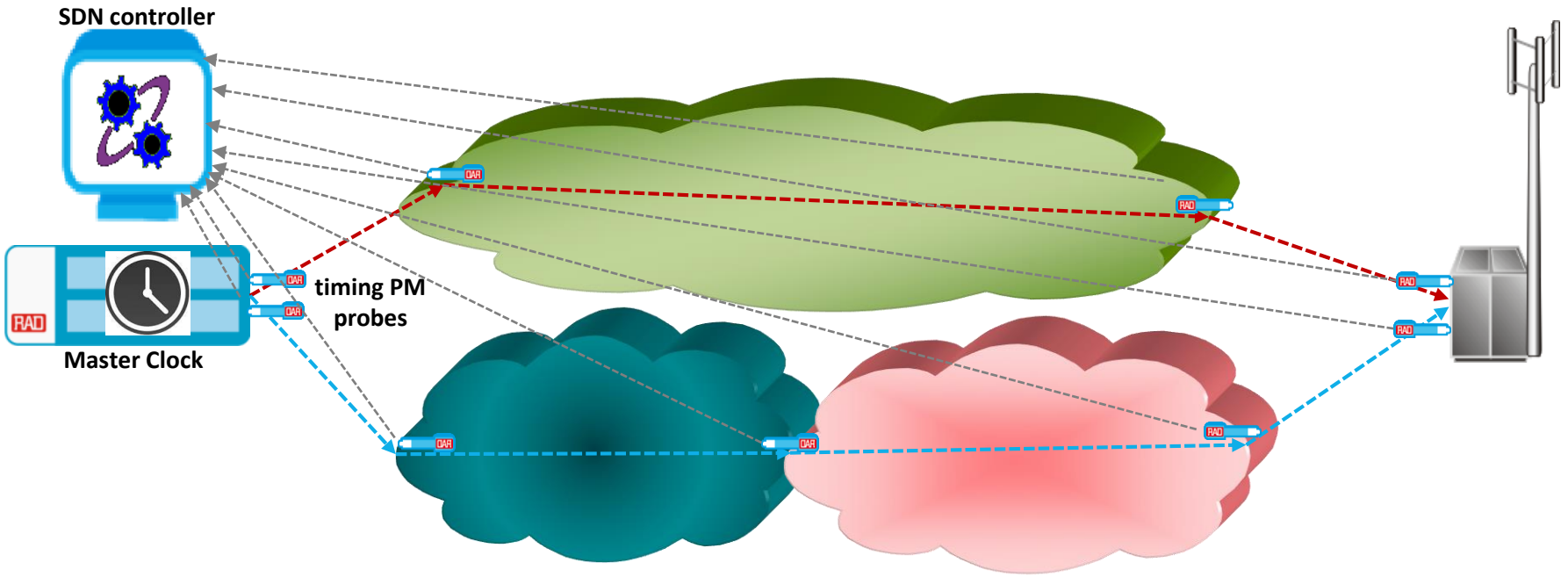
Yet, SDN can be relevant for overlay (Out of Footprint) timing distribution :

- the service provider does not control the physical network
- the characteristics of the physical network(s) may change over time
- there are assumed to be multiple operators offering connectivity alternatives

The service provider :

- monitors in real-time the expected timing performance of all alternatives
- selects the network (or networks) that provide the highest performance

# Overlay network timing distribution




Timing performance probes report to a timing PM portal
which computes the timing accuracies expected from each end-to-end path

The logic can now choose the path that will give the highest accuracy
or even propose simultaneously using multiple paths and weighted results

# Function relocation

Your Network's Edge

NFV and SDN facilitate (but don't require) ***relocation*** of functionalities to **P**oints **o**f **P**resence and **D**ata **C**enters

Many (mistakenly) believe that the main reason for NFV is to move networking functions to data centers where one can benefit from economies of scale

But even nonvirtualized functions *can* be relocated

Some telecomm functionalities need to reside at their conventional location

- Loopback testing
- E2E performance monitoring

but many don't

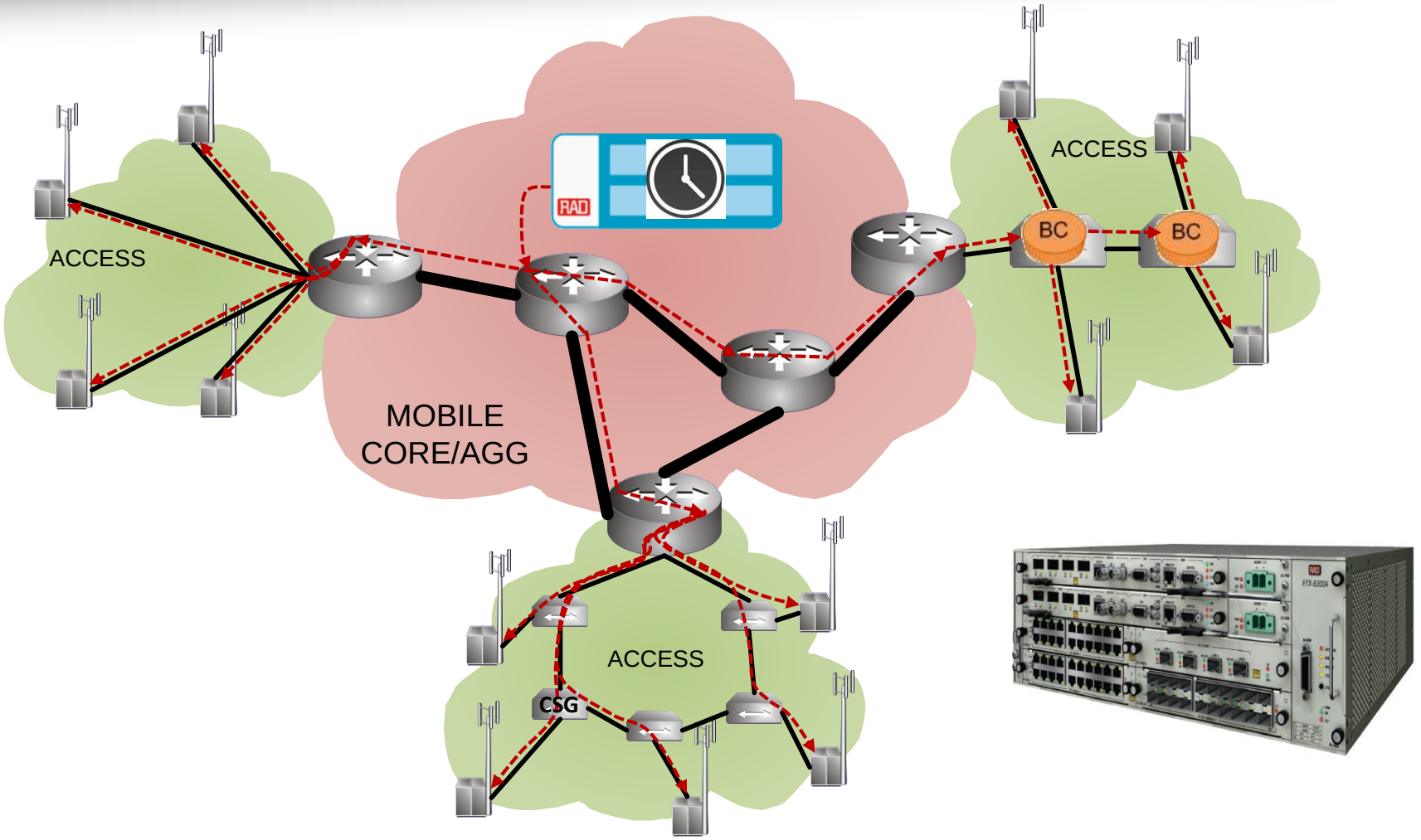
- routing and path computation
- billing/charging
- traffic management
- DoS attack blocking

Optimal location of a functionality needs to take into consideration:

- economies of scale
- *real-estate* availability and costs
- energy and cooling
- management and maintenance
- security and privacy
- regulatory issues

The idea of optimally placing virtualized network functions in the network is called **Distributed-NFV**

# Conventional GM-based timing architecture

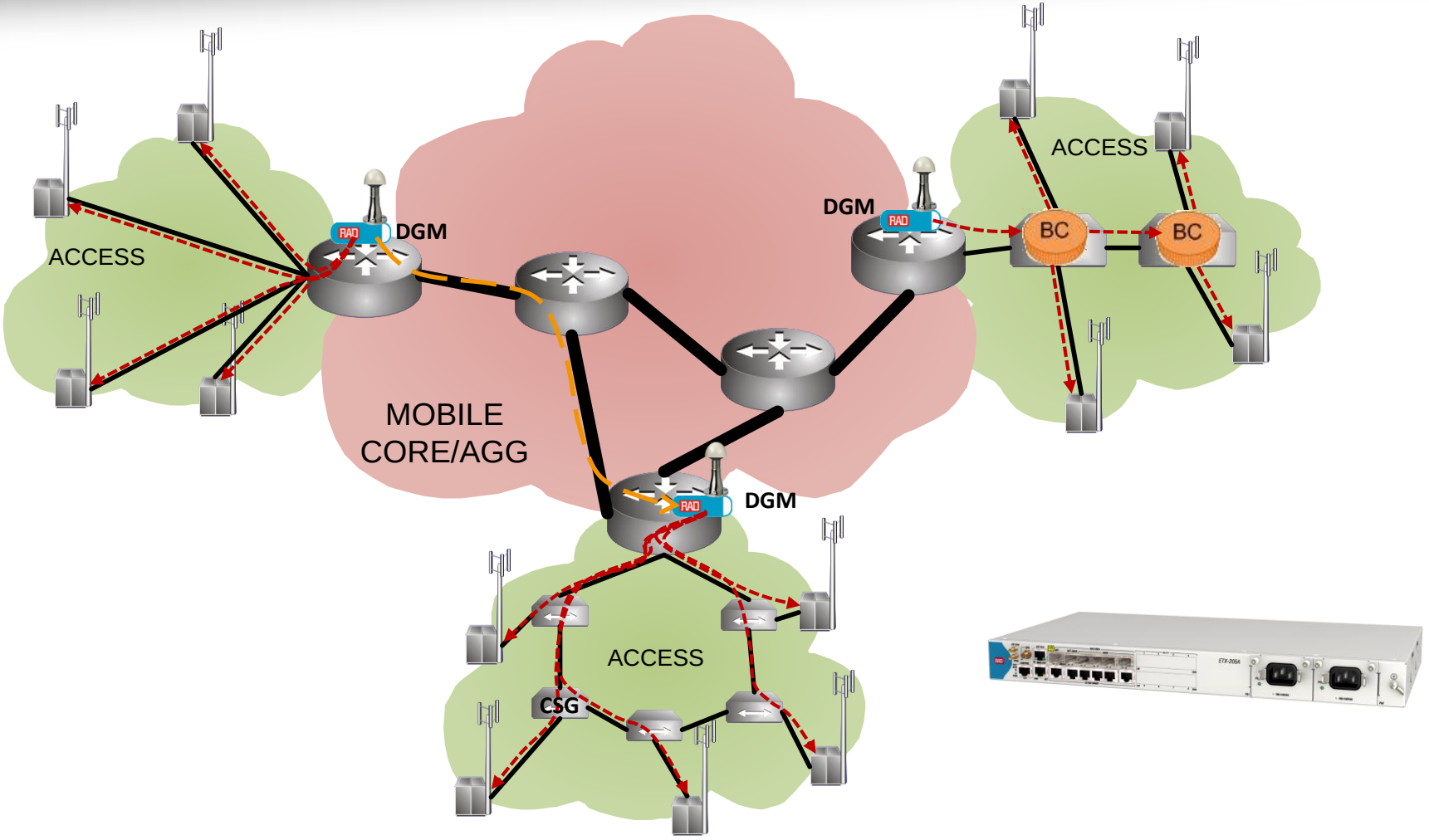



Conventionally there is one centrally located GM per network

One must ensure distribution does not overly degrade timing performance

On path support (BCs and/or TCs) used as needed

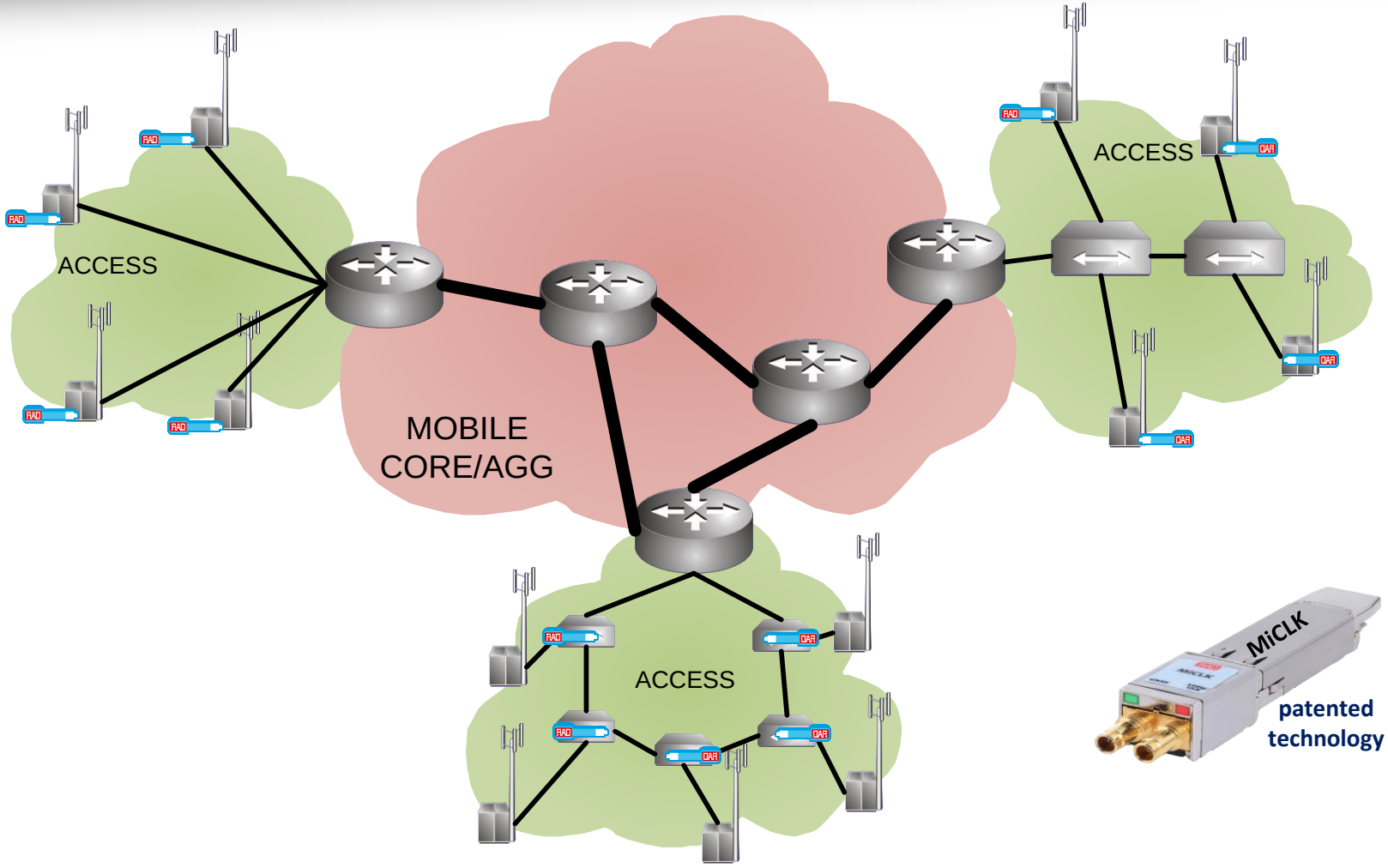
# Partially relocated GM architecture




Distribution is less critical if we use Distributed GMs closer to base stations
Failures cause less damage, and it is not difficult to share for backup

# Completely distributed architecture

Your Network's Edge




Maximum relocation is achieved by local miniature DGM at each cell site
Traceability still guaranteed by GNSS

# Is this relocation a form of SDN?

Your Network's Edge

Migrating from a single centralized master clock
to completely local distributed master clocks
*is* relocation, but *not* SDN (in fact, no network-wide timing distribution remains)

In fact, the fully distributed architecture
is architecturally identical to the common model of using GPS instead of ToP

However, that needn't be the end of the story

1. We needn't insist on a DGM for every cell site
   rather we can distribute from a DGN to several neighboring cell sites

2. GNSS may fail (due to line-of-sight problems or jamming)
   in which case we need suitable backup timing sources
   which can be neighboring DGMs or a centralized GM

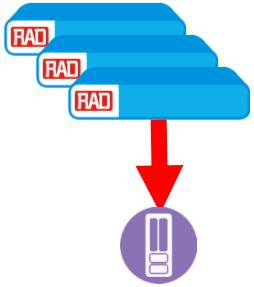
If the timing paths required are planned and pre-deployed
then no SDN is involved

An SDN approach would intelligently (and perhaps dynamically)
set up the needed timing flows

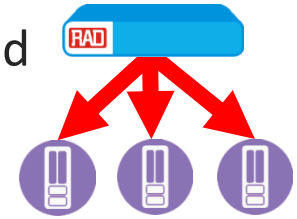
# Function repackaging

NFV and SDN facilitate (but don't require) ***repackaging*** of functionalities

- two or more functions can be collocated in one device




- conventional NEs can be decomposed into several atomic functionalities




Examples :

- a uCPE (universal CPE) can be configured to run multiple VASes
- a router that conventionally contains *forwarding* and *routing protocols* can be decomposed into *forwarding* and *path computation algorithms* and the path computation can be relocated to a data center

Repackaging is related to *Service Function Chaining*

- routing packets through complex sequences of network functionalities
- optimizing the use of networking and computational resources
- ensuring scalability and reliability

# Repackaging - decomposing a network clock

Your Network's Edge

SDN's premise is that a conventional forwarding NE has two parts :

1. smart but slow CPUs that engages in protocols and algorithms
2. fast but dumb switch fabric hardware that perform real-time operations

SDN separates the two :

1. the smart half is relocated to a data center
2. the dumb whitebox switch is left in the network

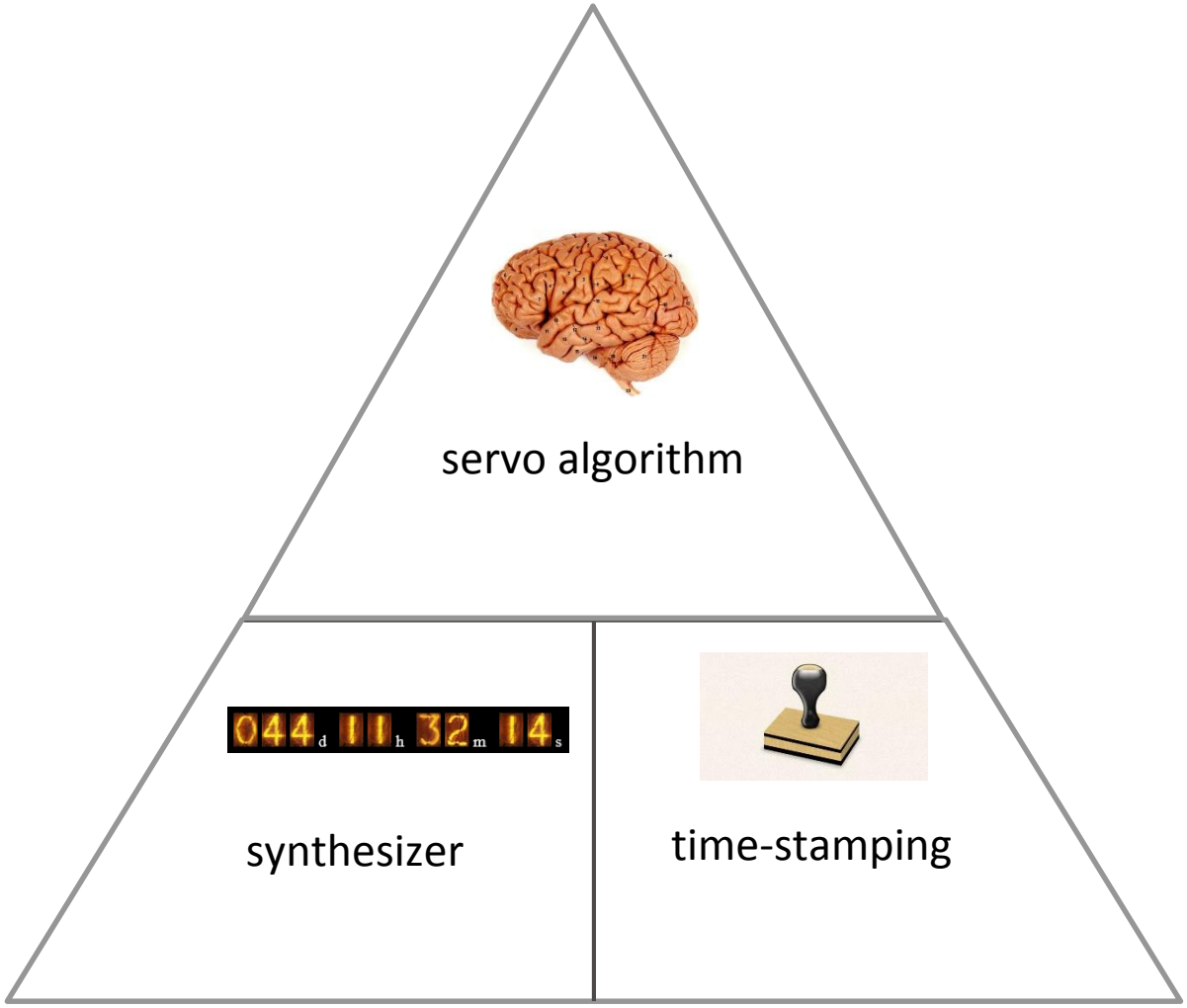
A little consideration brings us to the conclusion
that the same is true for a *network clock*

A network clock has two parts :

1. a servo algorithm that requires complex algorithms
2. hardware-based synthesizer and time-stamping

An SDN approach would explore separating the two !

# The decomposed network clock



# Decomposed timing distribution

Your Network's Edge

Instead of using true timing distribution (e.g., 1588 or NTP) flows
we send *time-stamp flow* packets (that contain the usual four time-stamps)

- from a *time-stamp flow generator* (TFG)
- to *time-stamp flow reflectors* (TFR)

TFRs perform time-stamping based on the local synthesizer
but no algorithmic processing is performed in the network

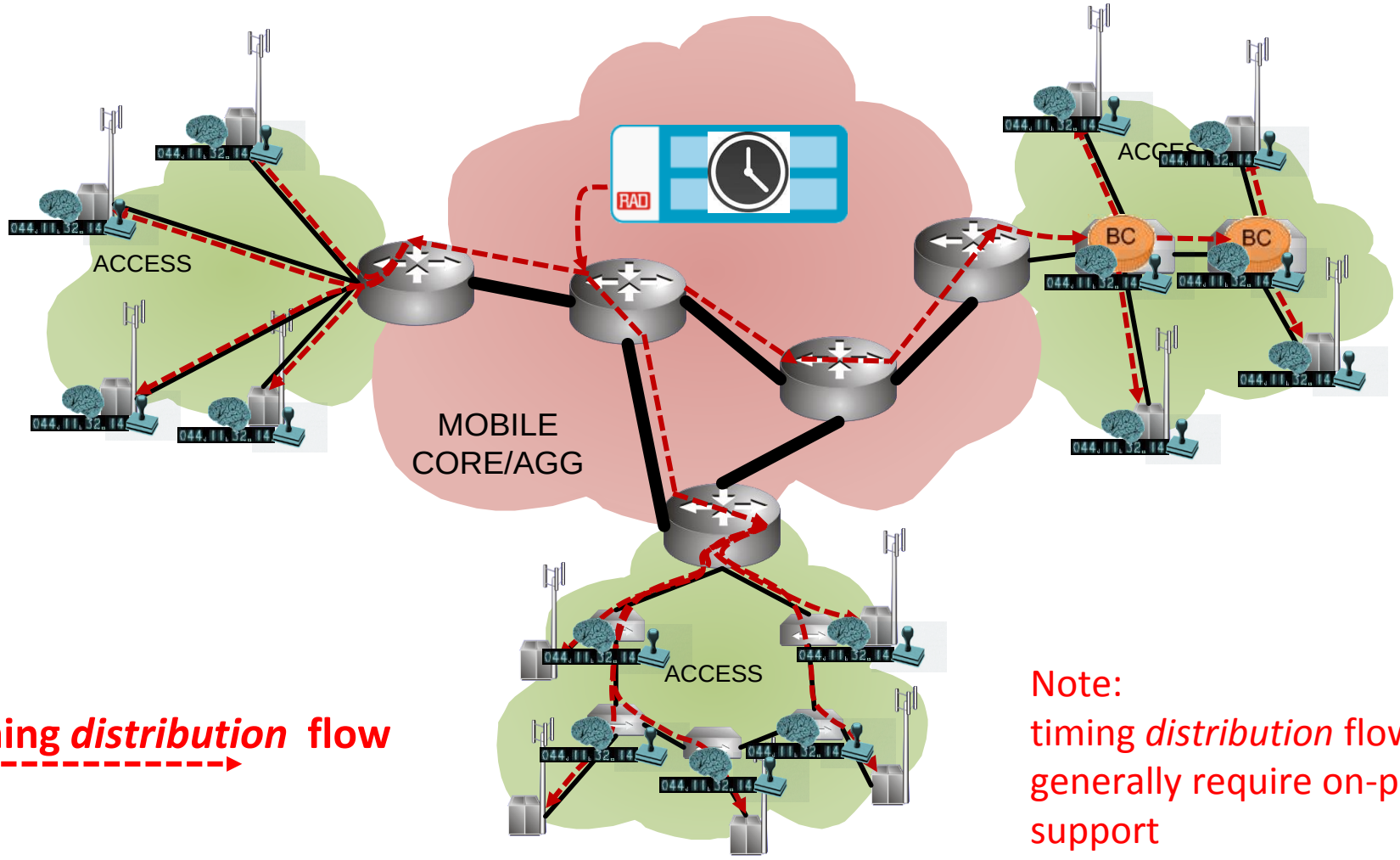
The TFG forwards the four timestamps per packet *as timing information*
to a data center where the timing recovery (servo) algorithm is performed

Periodically, frequency and time corrections are sent as *timing information*
to the TFG
which forwards them to the TFRs for forced correction of the synthesizer

Since the timing recovery algorithm is centrally performed in pure software

- the algorithm can be of arbitrary complexity and require arbitrary resources
- the algorithm may be upgraded as frequently as desired
- recovered parameters of neighboring TFRs can be compared
to detect faulty hardware or diagnose network anomalies

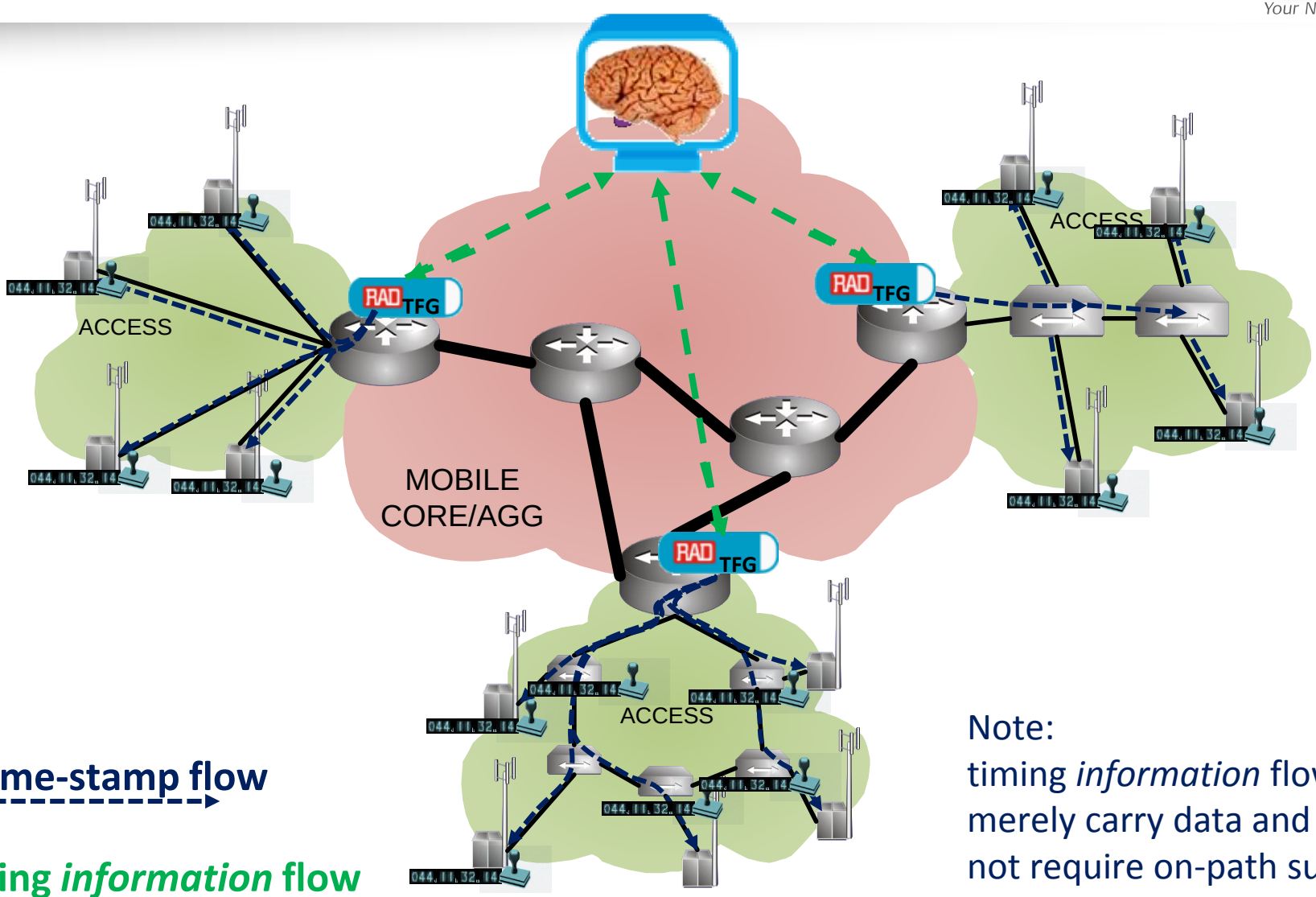
# Conventional timing distribution




**timing *distribution* flow**

Note:
timing *distribution* flows generally require on-path support

# Decomposed timing distribution




Note:
timing *information* flows merely carry data and do not require on-path support

# Summary

Due to their potential contribution to profitability

SDN and NFV technologies are being adopted by service providers

While network timing distribution

may not seem to be in accord with SDN/NFV approaches

we have seen that its main tenets:

- dynamicity
- function relocation
- function repackaging

may all be applied to network timing

While we only explored three simple applications

many more are possible

and may be deployed soon

Thank you
for your attention


*Your Network's Edge*

Yaakov (J) Stein

CTO

Yaakov_S@rad.com

For a 1 hour video tutorial on SDN and NFV go to :
http://recordings.conf.meetecho.com/Playout/watch.jsp?recording=IETF93_SDN&chapter=chapter_0