DNS TCP fallbackの教訓

NTTドコモ
國友宏一郎　奥田兼三

2025/6/27

目次

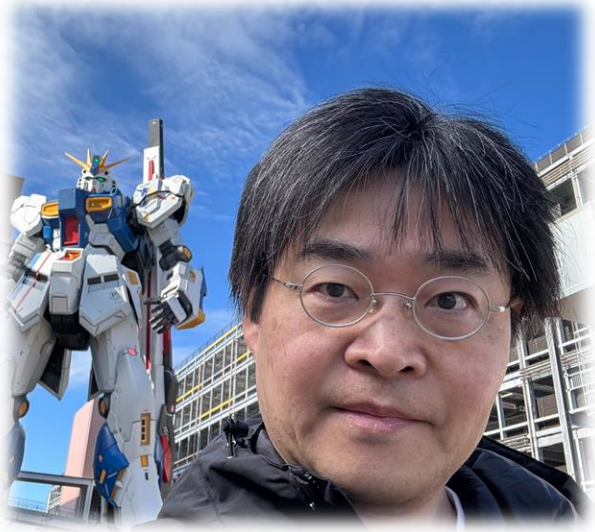
- 発生した事象とロジック
- モバイルコアでのDNSの使い方
- DNSレコードの解説
- DNSの性能
- DNSで行った対策

発表者紹介

國友　宏一郎

奥田　兼三

自己紹介



國友　宏一郎

くにとも　こういちろう

NTTドコモ

コアネットワークデザイン部 5Gコア担当

これまでの経歴	・PDC-P(2Gパケット)の保守運用 ・3Gパケットコアネットワーク開発 ・4G(LTE)コアネットワーク開発 ・5Gコアネットワーク開発 ドコモのパケット系のコアネットワーク担当です！ DNSによくつかわれるプロダクトを導入した犯人です
最近の業務	5GC on AWS 5GCスライス
趣味	社内でのIPv6布教活動 アニメ鑑賞 　(GQuuuuuuX面白かったです)

自己紹介

奥田　兼三 おくだ けんぞう

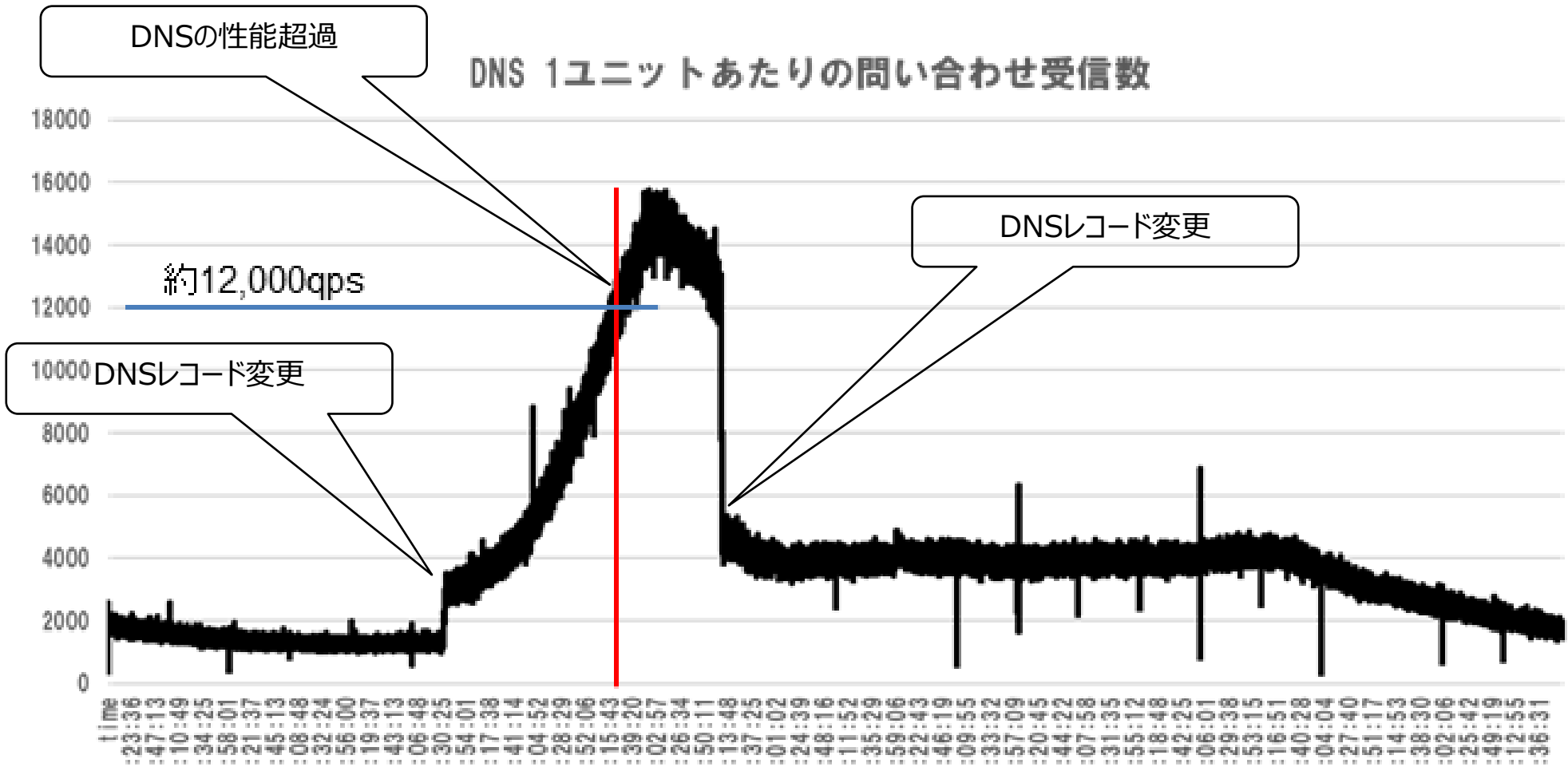
- NTTドコモ
  - コアネットワークデザイン部 5Gコア担当
- 経歴
  - NTT研究所に入所し、SDN、オーケストレータ等のNWソフトウェア制御方式や将来網方式検討を経て、2020年にドコモに転籍
  - 現在は5Gコアネットワーク開発に従事
    - 5G SA
    - ドコモMEC
- モバイルむつかしすぎなんもわからん

愛読書	3GPP TS 29.303 DNS Procedures 3GPP TS 23.502 3GPP TS 29.244 3GPP TS 29.510 3GPP TS 29.516 風の谷のナウシカ
口癖	コアネットワークのスライスとIPのスライスは別物です！
社内所属	IPv6教 3GPP標準原理主義集会

発生した事象

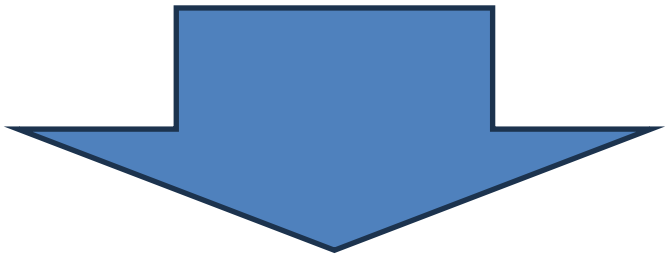
発生した事象

DNSレコード変更により、DNSクエリが急増しDNSの性能超過！！



原因はDNSレコードの増加によるDNSクエリ増

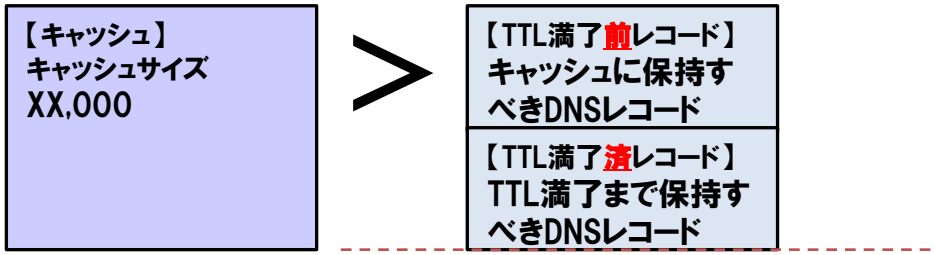
- DNSレコードが増加すると、なぜDNSクエリが増えたのか？



- モバイルコアのDNSクライアント(MME)のキャッシュあふれによりレコードのTTL満了前にクエリが発生

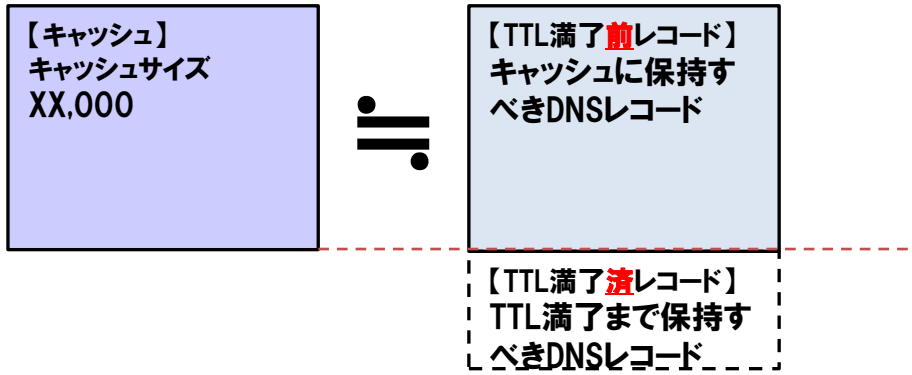
MMEのキャッシュ

【状態１】保持すべきレコードがキャッシュサイズより小さい



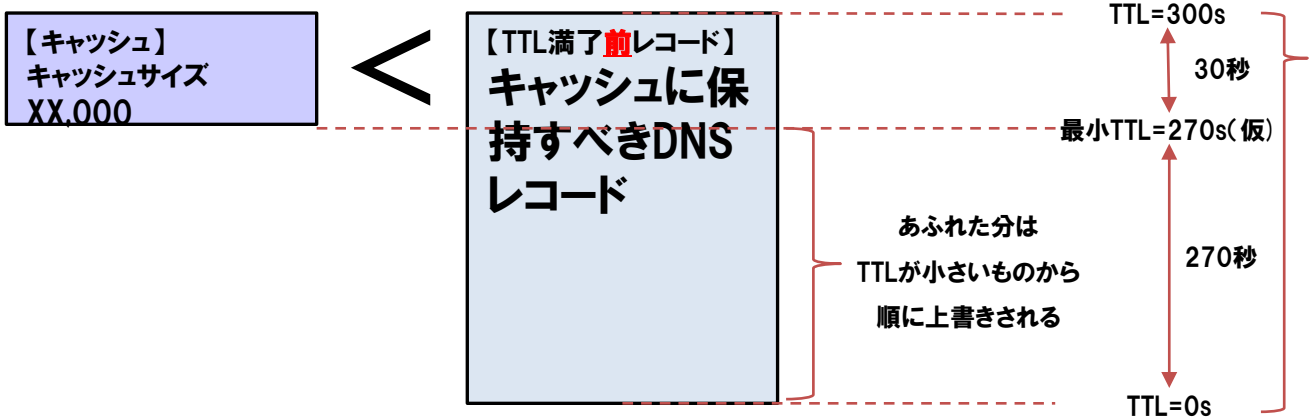
・TTL満了までキャッシュを使えている状態

【状態２】保持すべきレコードとキャッシュサイズがほぼ同等



・ほぼTTL満了までキャッシュを使えている状態

【状態３】保持すべきレコードがキャッシュサイズより大きい場合

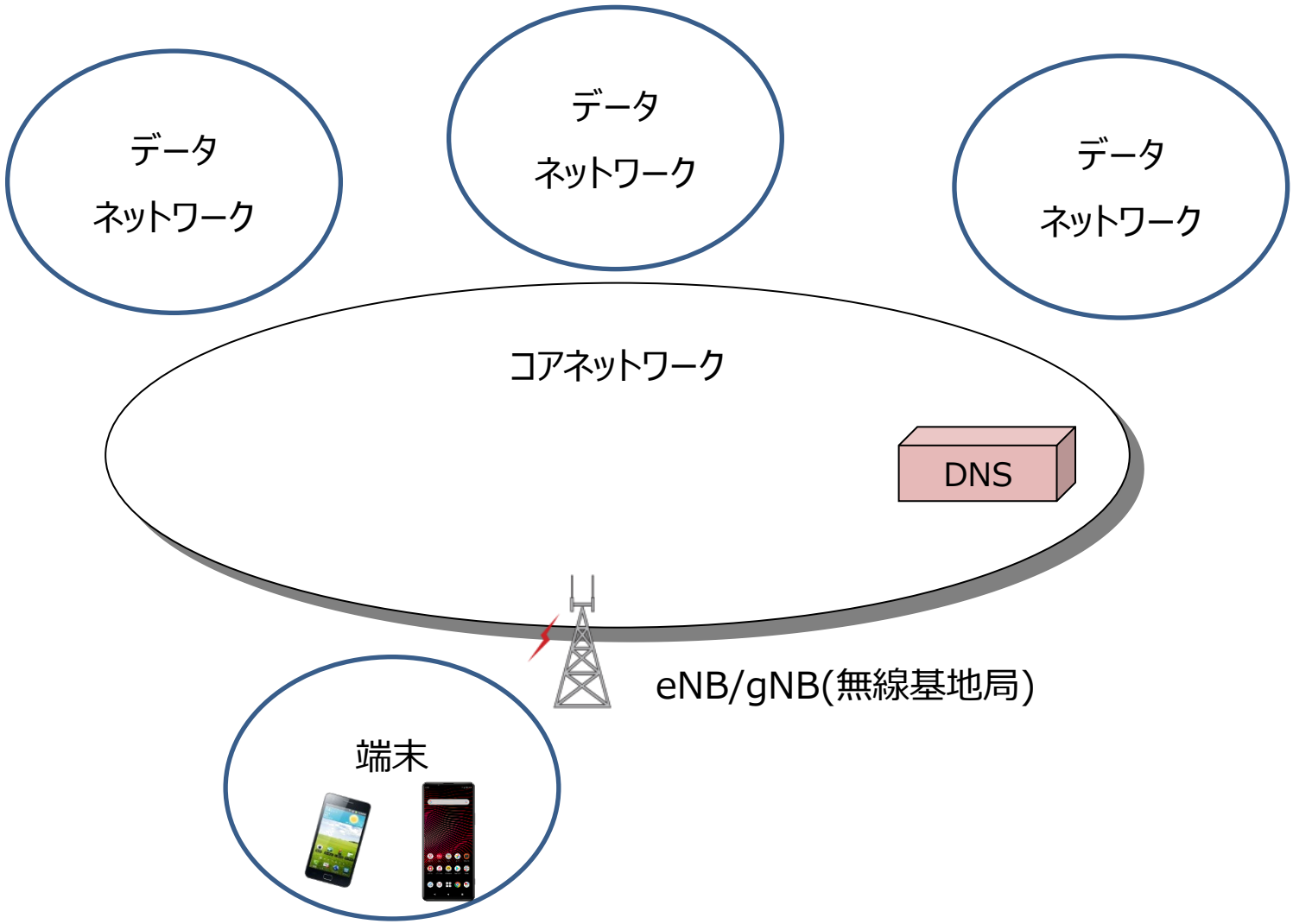


TTL=300から減算されていき、
TTL満了（TTL=0）までキャッシュに保持したいが、
キャッシュサイズより保持されるDNSレコードが多い場合、
TTLが小さいものから上書きされる（=キャッシュあふれ）

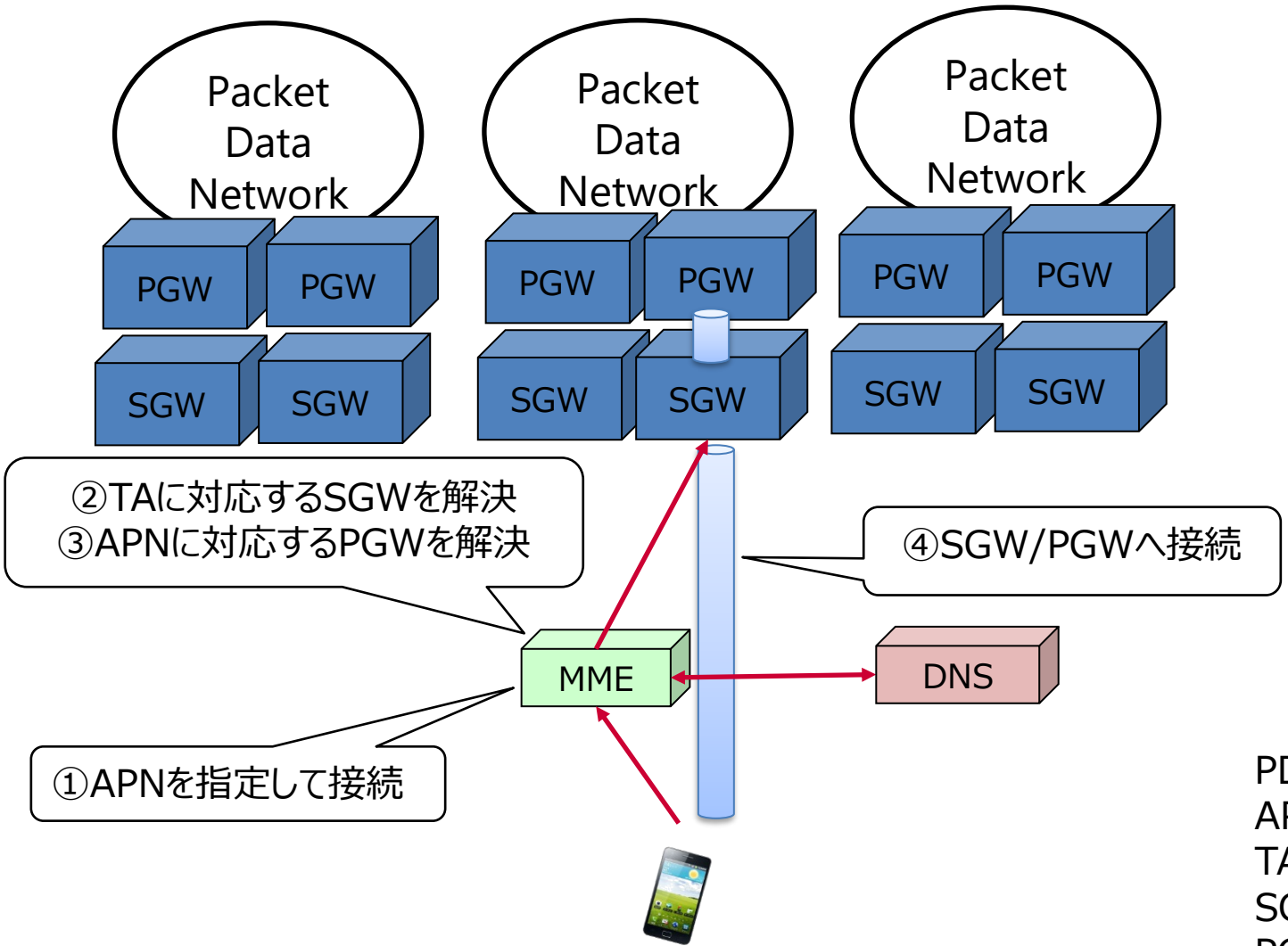
・TTL満了までキャッシュを使えていない状態

モバイルコアでのDNSの使い方

モバイルコア（4G）でもDNS使ってます！



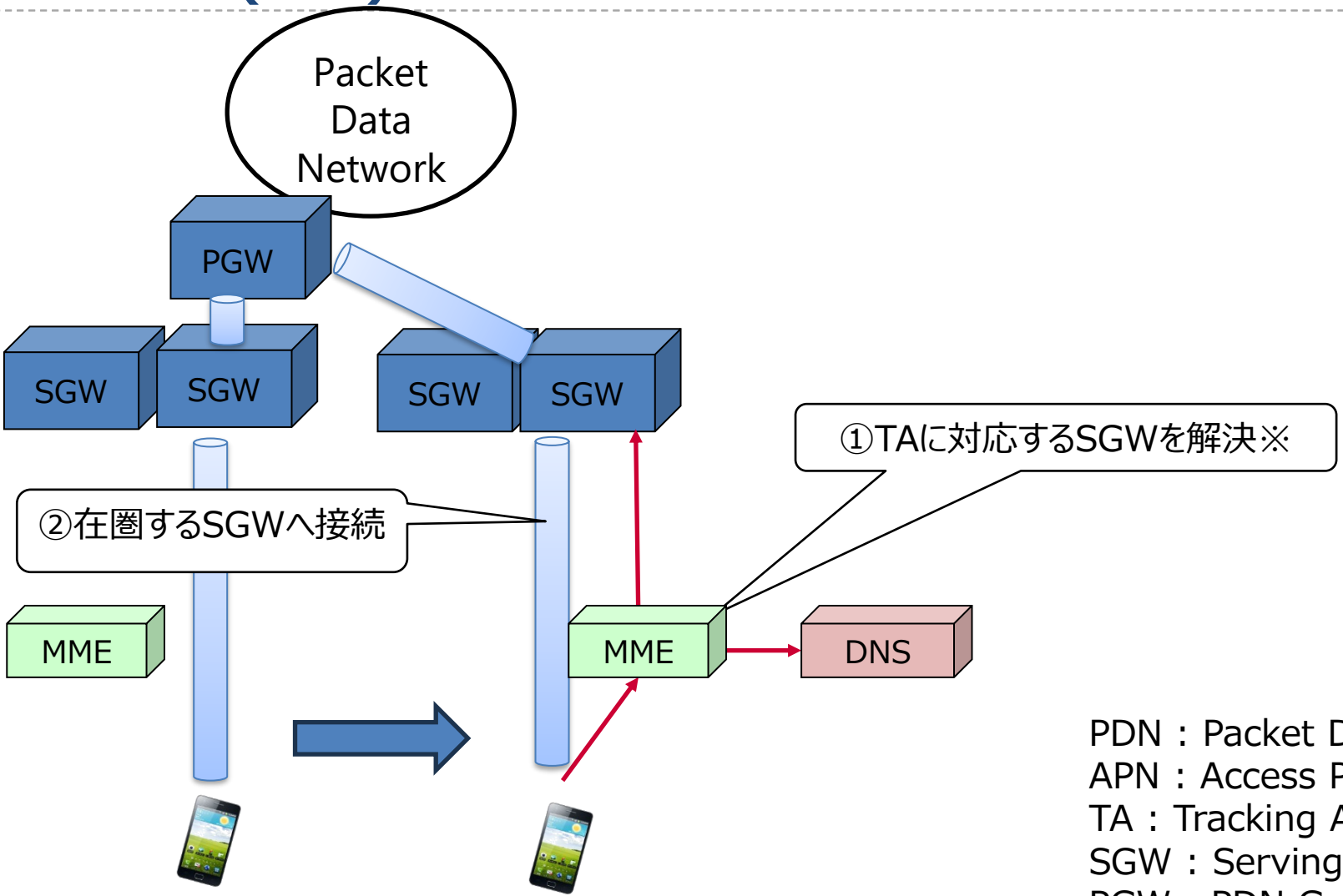
モバイル網(LTE)でのDNSの役割



PDN：Packet Data Network
APN：Access Point Name
TA：Tracking Area
SGW：Serving GateWay
PGW：PDN GateWay
MME：Mobility Management Entity

初回接続時　SGW/PGWの解決

モバイル網(LTE)でのDNSの役割

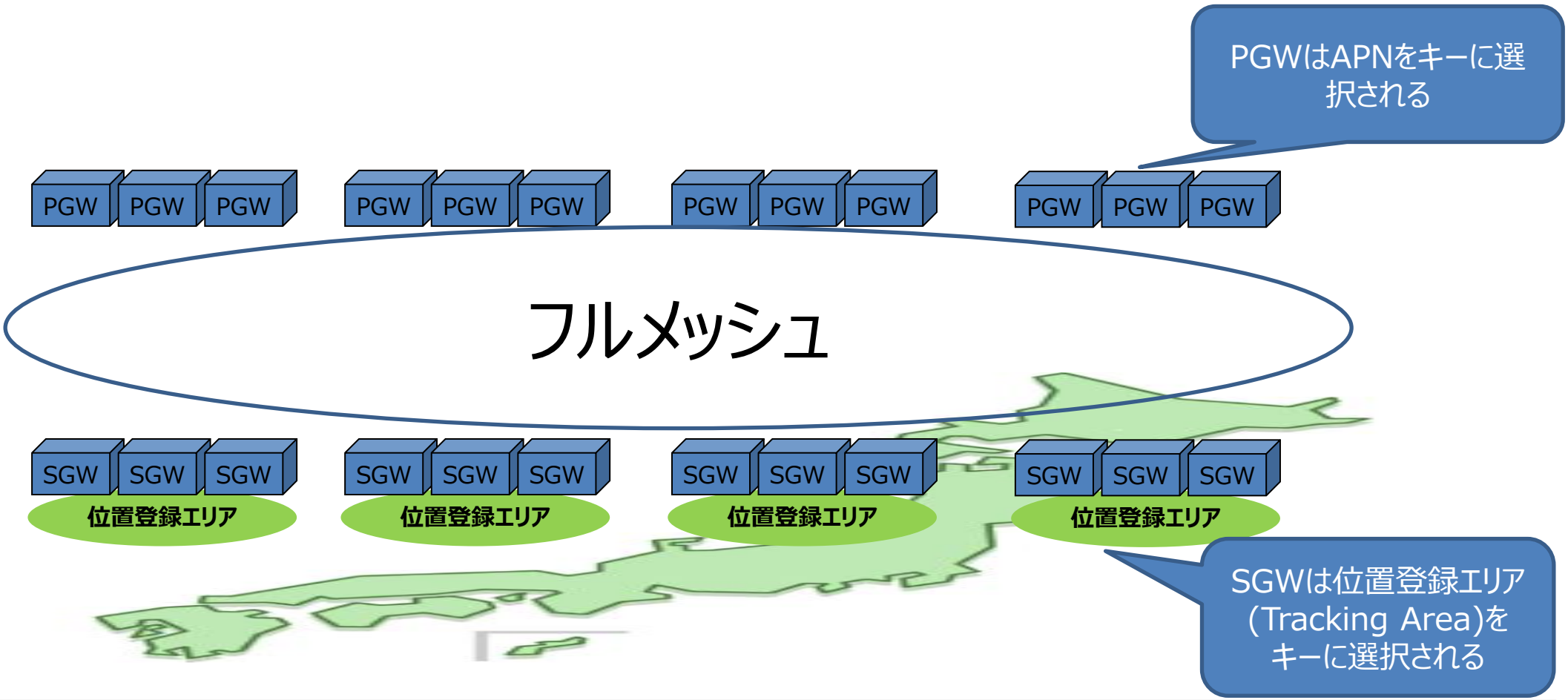


PDN：Packet Data Network
APN：Access Point Name
TA：Tracking Area
SGW：Serving GateWay
PGW：PDN GateWay
MME：Mobility Management Entity

移動時　SGWの解決

※　接続しているPGWのチェックを行うケースも。。

SGW/PGWは何を目的に選択するか？

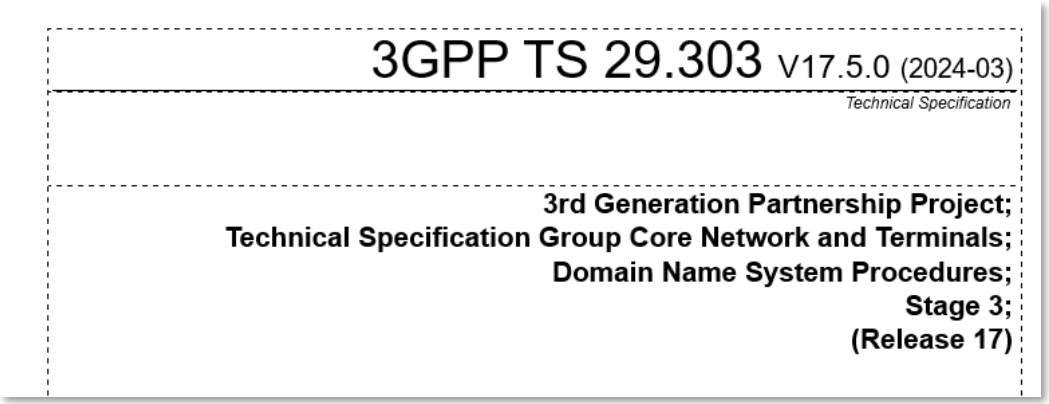


SGW：Serving GateWay

どうやって選択するのか？

- LTEのDNSに関する3GPP標準

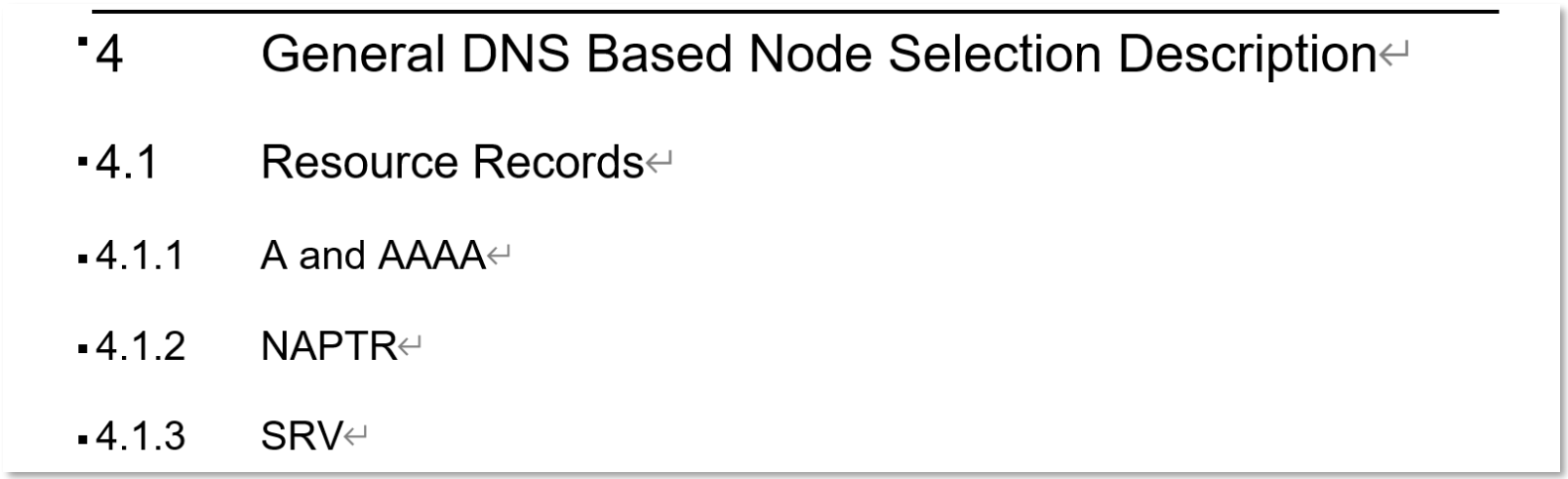
  →**3GPP TS 29.303**


3GPP TS 29.303 V17.5.0 (2024-03)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Core Network and Terminals;
Domain Name System Procedures;
Stage 3;
(Release 17)

登場するレコードは



SRV…… NAPTR ……

どうやって選択するのか？

- NAPTRレコードとは？
  - *IETF RFC 3403: “ Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database”.* で定義されているレコード
    - DDDS自体は RFC3401, 3402, 3403, 3404

レコード例

```
$ORIGIN example.com.
;              order preference  flags  service       regexp                                                  replacement
 IN NAPTR 100    10          “S”   “SIP+D2U”    “!^.*$!sip:customer-service@example.com!”   _sip._udp.example.com.
 IN NAPTR 102    10          “S”   “SIP+D2T”    “!^.*$!sip:customer-service@example.com!”   _sip._tcp.example.com.
```

order 16bit符号なし整数 （preferenceより優先）

preference 16bit符号なし整数 小さいもの優先

flags 文字 “S” “A” “U” “P” 置換・解釈の制御

S:次はSRV引き

A:次はA,AAAA引き

U:最終結果 URIを出力

P:プロトコル依存

なし:得られた結果についてさらにNAPTRを引く

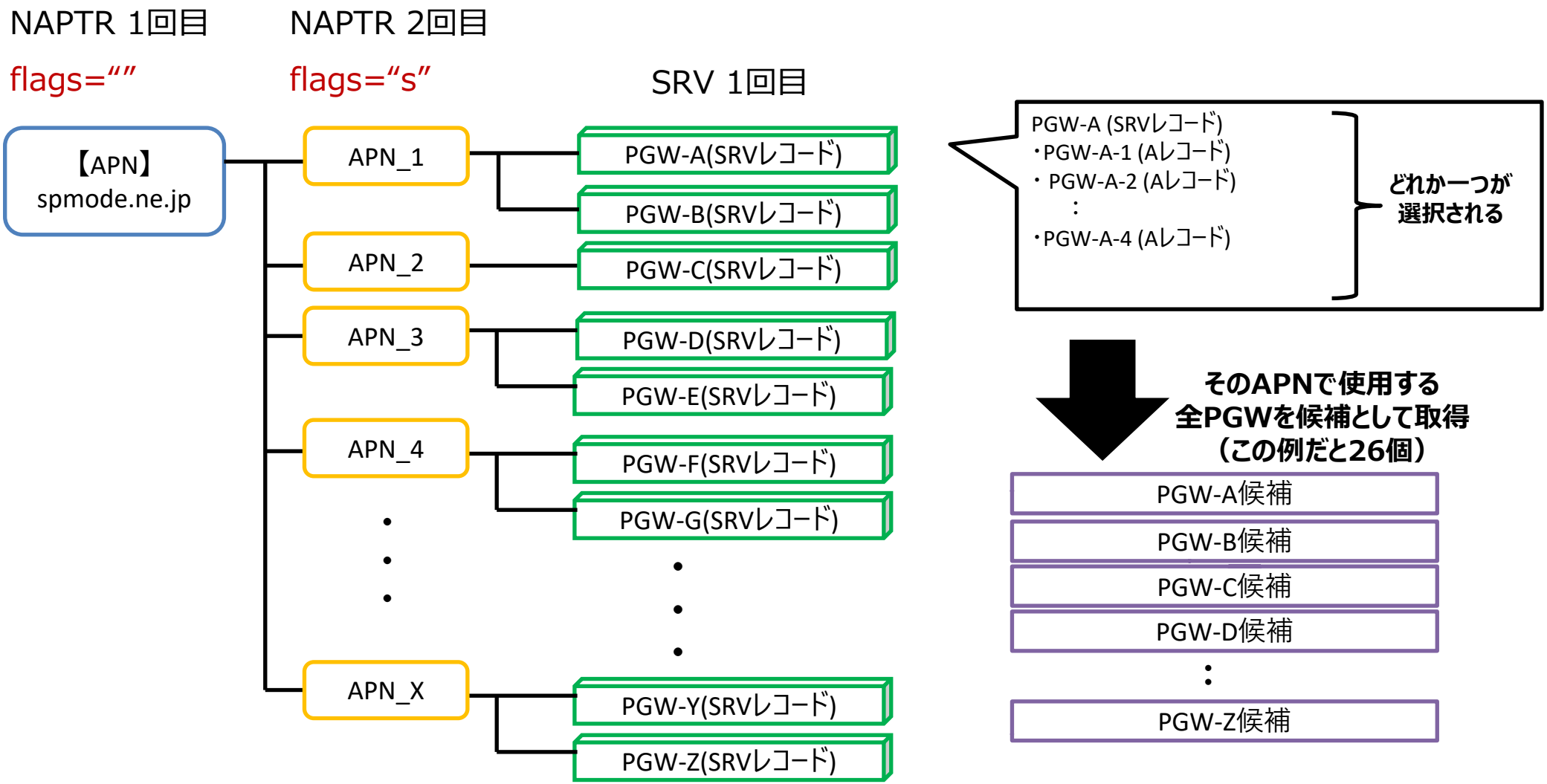
service 文字列 このエントリが適用されるサービスを指定

regexp 置換文字列

replacement 置換が不要でドメイン名を出力すればよい場合、regexpのかわりに記述 (regexpがあるとき

NAPTRとSRVを使ったSGW/PGWの選択

- PGW候補導出の例

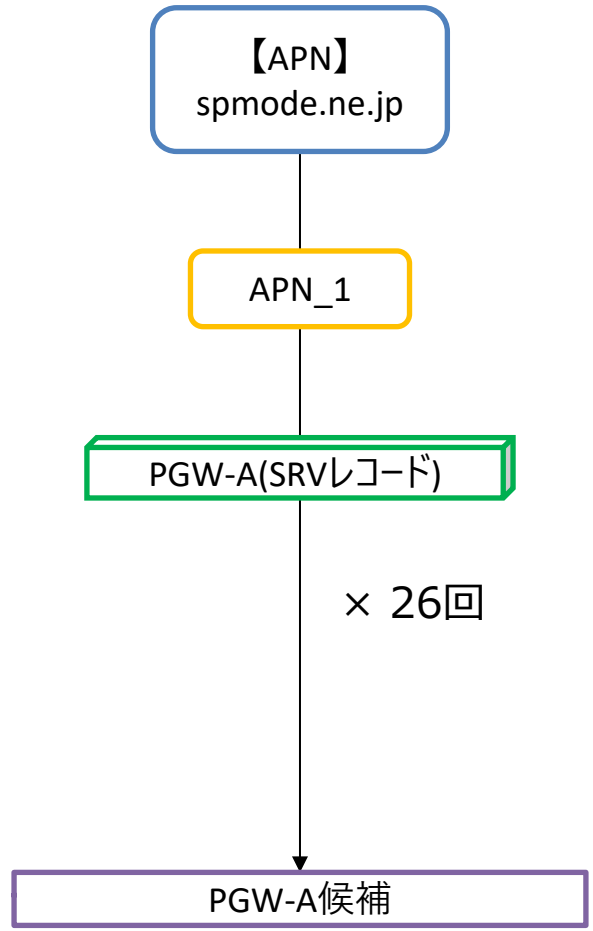


実際どんなレコードになる？

- PGW-A IPアドレス導出までの具体例

```
spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 63700 "" "x-3gpp-pgw:x-s5-gtp" "" a.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65510 "" "x-3gpp-pgw:x-s5-gtp" "" b.v2.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65400 "" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" 5g.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65513 "" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" 5g.v2.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
・・・

a.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp" "" _PGW.000
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp" "" _PGW.001
・・・

_PGW.000.vspgw (
  IN SRV 100  100 2123 topon.S5.pgw.00.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN SRV 100  100 2123 topon.S5.pgw.01.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
・・・

5g.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" _PGW.001.TAC10.5g
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" _PGW.001.TAC13.5g
・・・

_PGW.001.TAC10.5g (
  IN SRV 100  100 2123 topon.S5.pgw.5g00.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN SRV 100  100 2123 topon.S5.pgw.5g01.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
・・・

topon.S5.pgw.00.TAC10.node.EPC.MNC040.MCC440.3GPPNETWORK.ORG IN A 10.x.x.x
topon.S5.pgw.01.TAC10.node.EPC.MNC040.MCC440.3GPPNETWORK.ORG IN A 10.y.y.y
・・・
```

PGW候補のIPアドレス

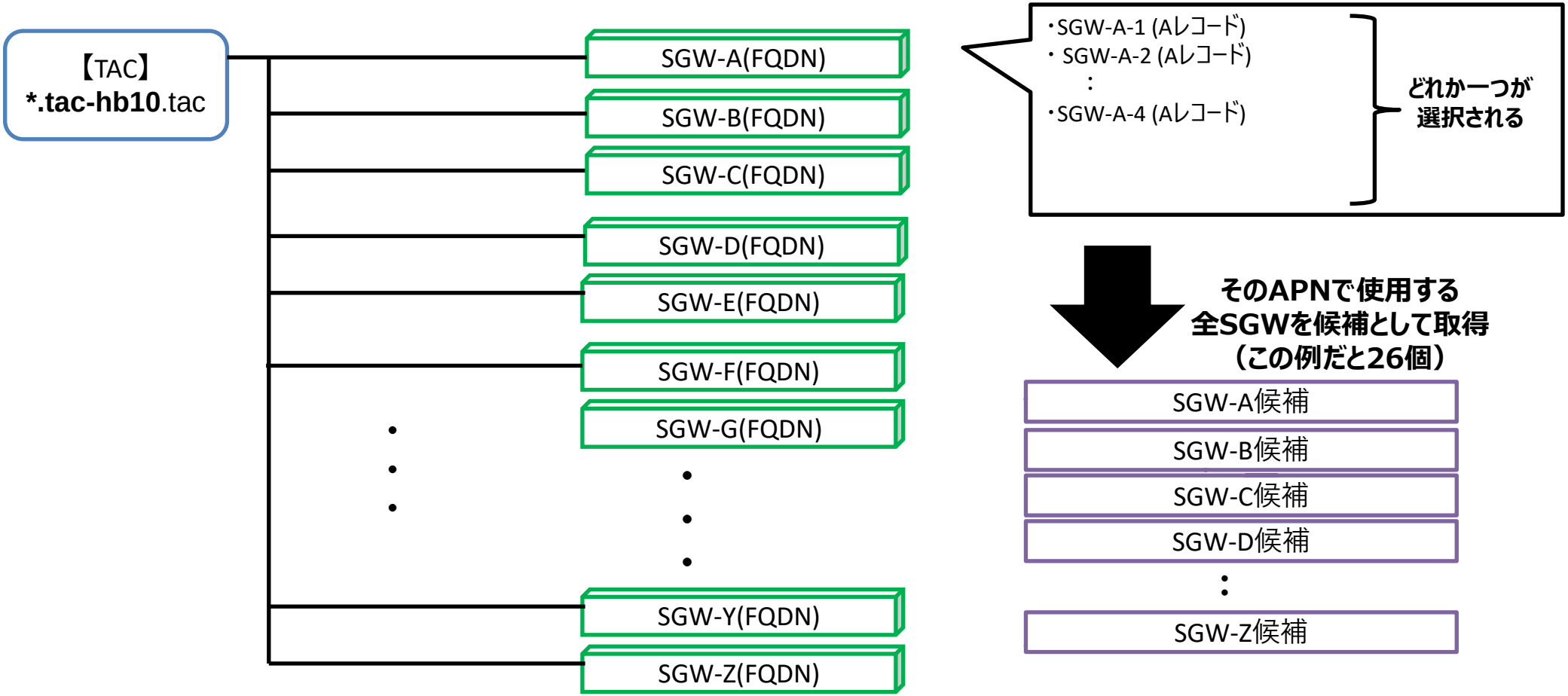


NAPTRとSRVを使ったSGW/PGWの選択

- SGW候補導出例

NAPTR

flags="a"



DNSレコードの解説

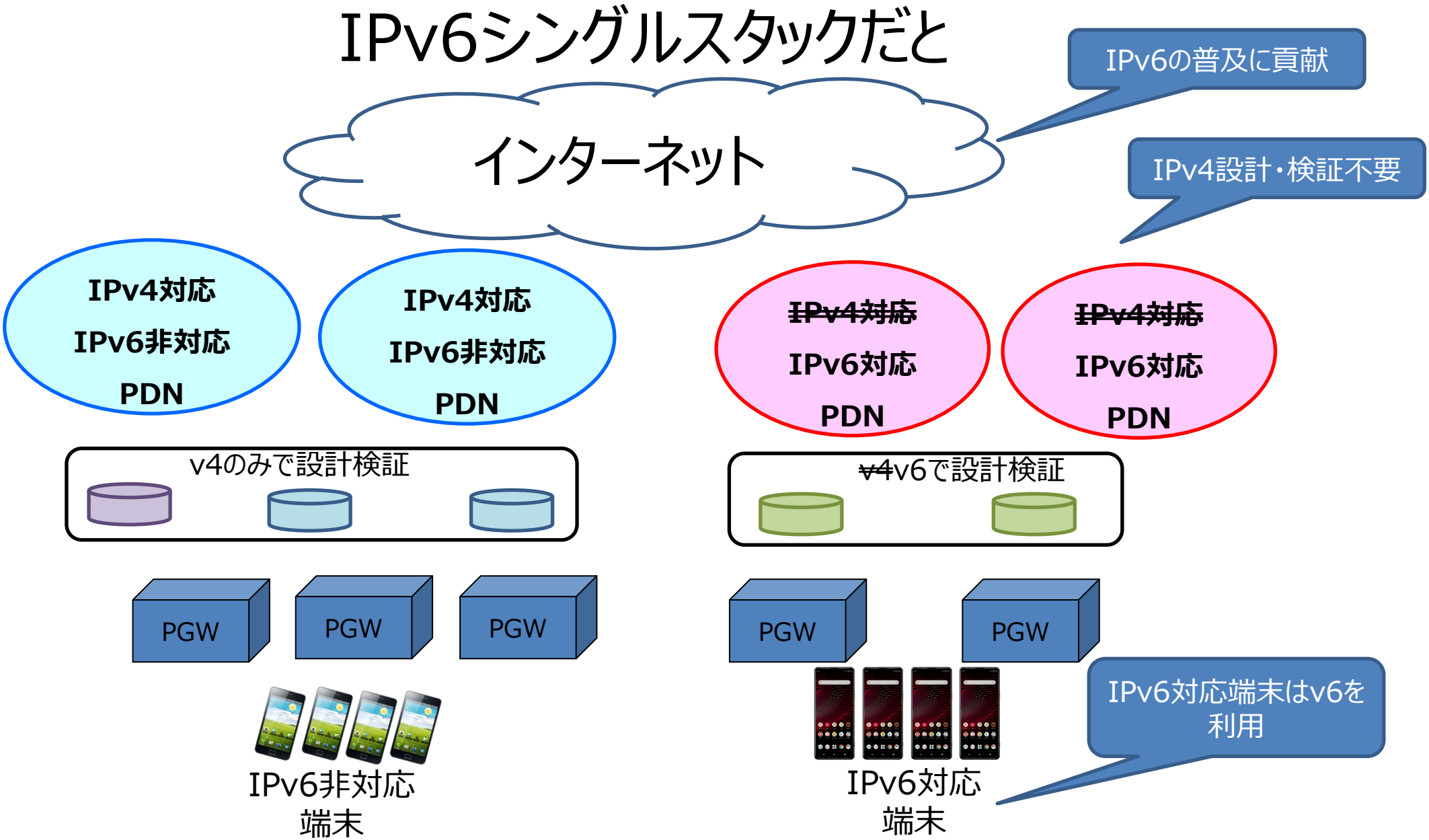
DNSレコード変更の中身は

↓

IPv6シングルスタックによる**PGW**のレコード増加

IPv6アドレス利用拡大に向けたドコモの取り組み – JANOG48 Meeting より一部抜粋

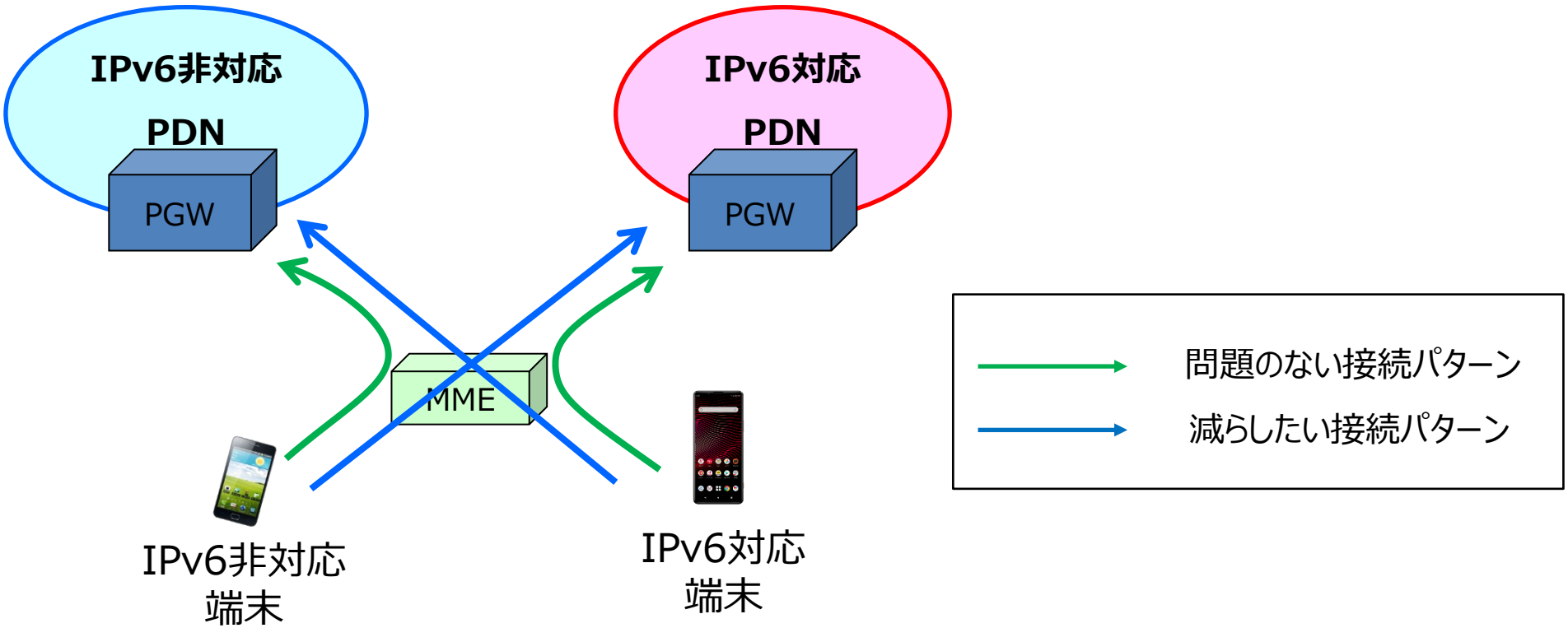
IPv6の利用拡大に向けて



利用率を最大化する方法

■ IPv6利用率を最大化する方法

IPv6端末はIPv6シングルスタックのネットワークへ、IPv4端末は、それ以外のネットワークへ接続させたい

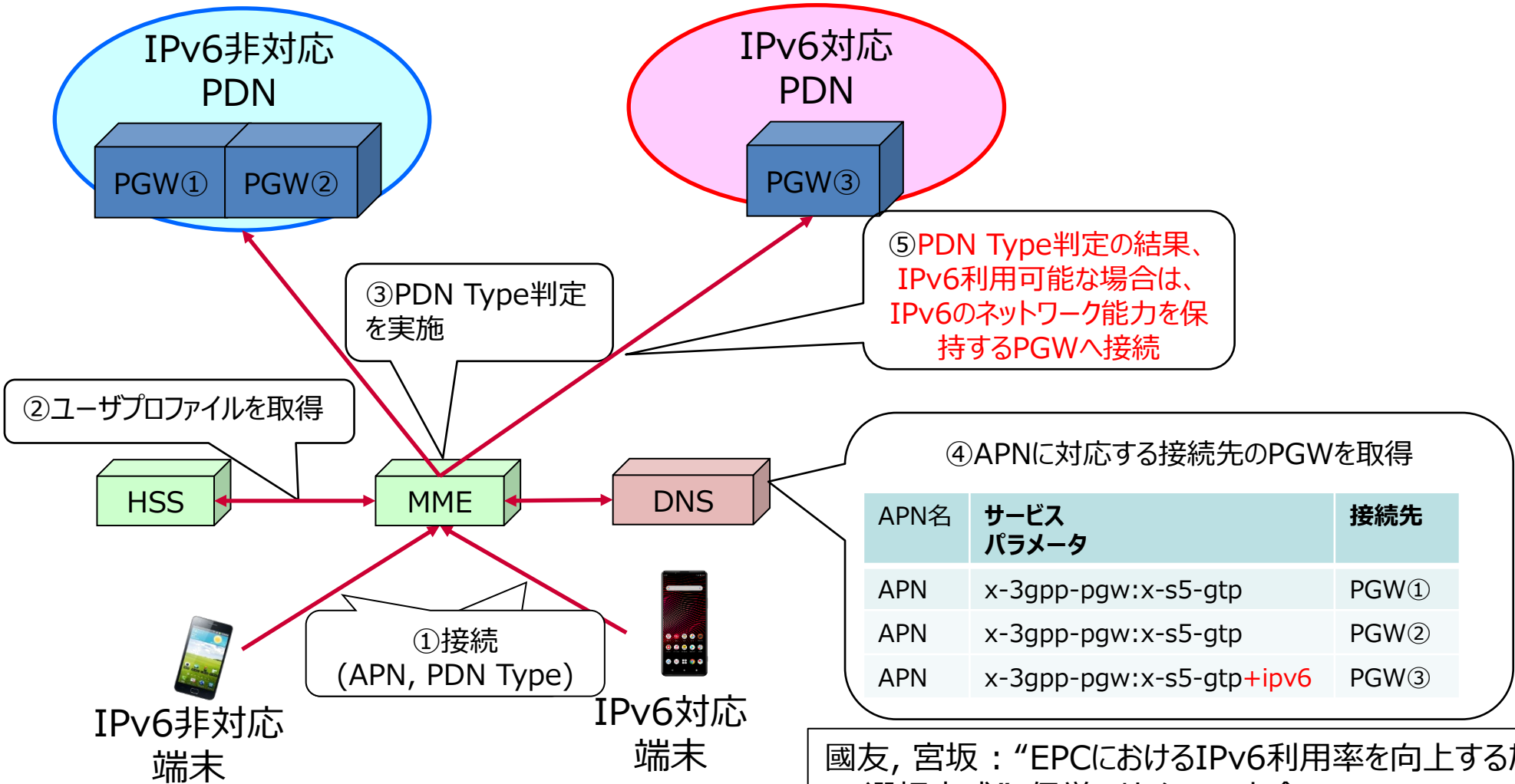


端末の対応するIPバージョンに応じてPDNを選択

利用率を最大化する方法

■ 案　MMEでの選択方式

DNS応答に、ネットワークの能力を示す値(+nc-<network capability>)を追加し、その値に応じてPGWを選択する。



國友, 宮坂：“EPCにおけるIPv6利用率を向上するためのPDN選択方式”, 信学ソサイエティ大会 B-6-38, 2019年9月

結果として詳細な　PGWのDNSレコード

SPモードのDNSレコード例 bindのzoneファイル

```
spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 63700 "" "x-3gpp-pgw:x-s5-gtp" "" a.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65510 "" "x-3gpp-pgw:x-s5-gtp" "" b.v2.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65400 "" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" 5g.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65513 "" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" 5g.v2.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65400 "" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" 5gv6.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65400 "" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" 5gv6.v2.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG


a.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp" "" _PGW.000
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp" "" _PGW.001
・・・

_PGW.000.vspgw (
  IN SRV 100  100 2123 topon.S5.pgw.00.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN SRV 100  100 2123 topon.S5.pgw.01.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
・・・

5g.v1.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" _PGW.001.TAC10.5g
  IN NAPTR 100 65435 "s" "x-3gpp-pgw:x-s5-gtp+nc-nr" "" _PGW.001.TAC13.5g
・・・

_PGW.001.TAC10.5g (
  IN SRV 100  100 2123 topon.S5.pgw.5g00.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN SRV 100  100 2123 topon.S5.pgw.5g01.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
・・・

5gv6.v1.n.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 "a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.nr.00.01.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65435 "a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.nr.01.02.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG

topon.S5.pgw.nr.00.01.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG IN A 10.a.a.a
topon.S5.pgw.nr.01.02.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG IN A 10.b.b.b

5gv6.v2.n.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
  IN NAPTR 100 65435 “a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.nr.00.03.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65435 “a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.nr.01.04.TAC13.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG


topon.S5.pgw.nr.00.03.TAC10.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG IN A 10.c.c.c.
topon.S5.pgw.nr.01.04.TAC13.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG IN A 10.d.d.d.
```

NAPTRレコードのServiceParameterにより、v6のレコードが増加

NAPTRレコードのServiceParameterにより、v6のレコードが増加

NAPTRレコードのServiceParameterにより、v6のレコードが増加

IPv6シングルスタックによる**SGW**のレコード増加

PGWレコードが増えるとなぜSGWレコードが増えるのか

LTEのDNSに関する3GPP標準

→3GPP TS 29.303

3GPP TS 29.303 Domain Name System Procedures

4.3.2 Identification of canonical node names

The host names shall have form:

<"topon" | "topoff"> . <single-label-interface-name> . <canonical-node-name>

Where the first label is "topon" or "topoff" to indicate whether or not collocated and **topologically close node selection shall be preferred,**

5.2.3 SGW Selection during TAU or RAU with SGW change - non-roaming case

Collocation of PGW and SGW and topological ordering rules both apply in this case.If the existing PGW hostname for the PDN has "topoff" then the "candidate" list of SGW would be used in the order given to try to contact a SGW after moving the PGW with the same SGW node name to the front of the list keeping relative order..

If the existing PGW hostname has "topon" the two candidate lists shall be used in the procedure in Annex C.4 with the SGW as "A" and the PGW as "B". Annex C.4 results in a list of SGW to try in order.

C.4 S-NAPTR procedure pseudo-code with topon

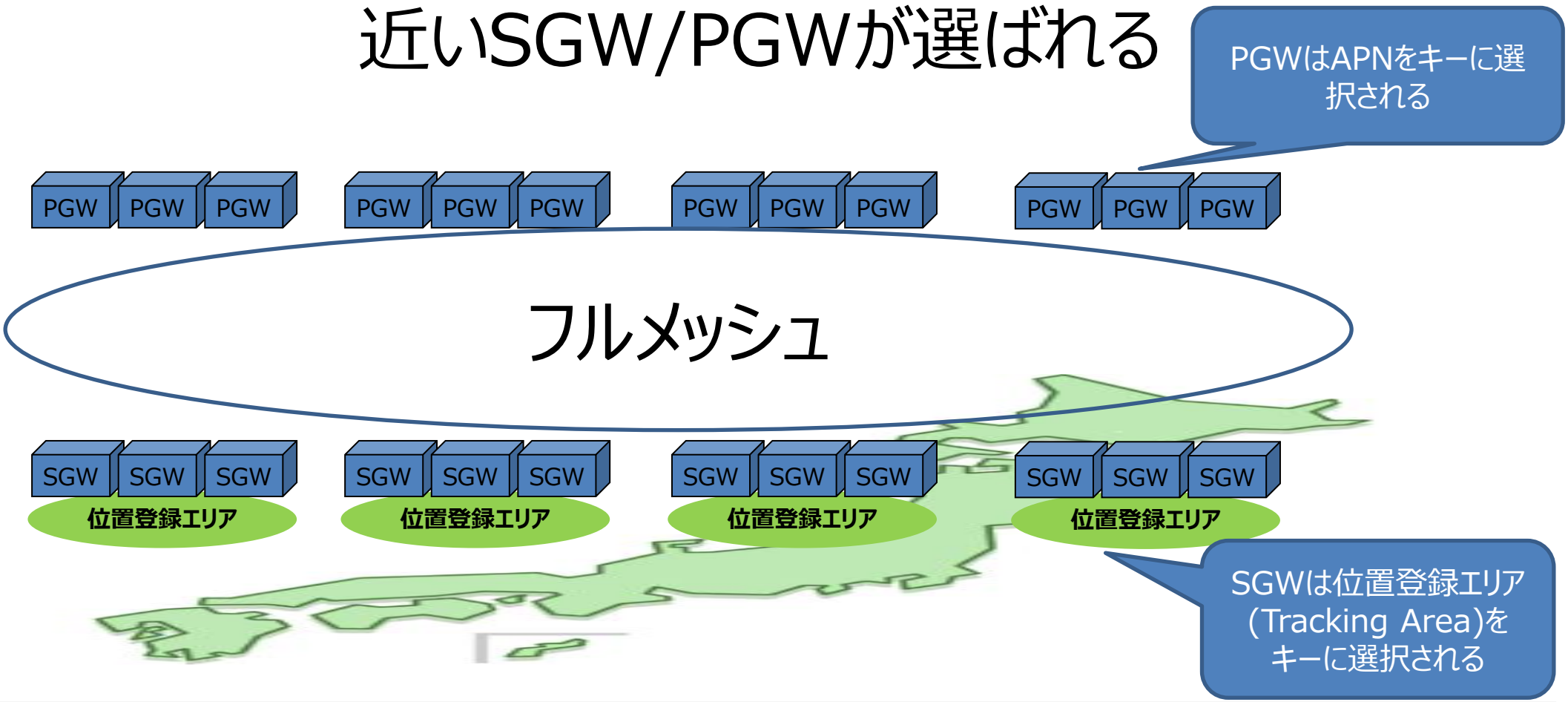
V13.6.0より抜粋

SGW/PGW選択

位置的に(topologically)近い選択とは

→SGW/PGWの選択はそれぞれ個別に行われるが

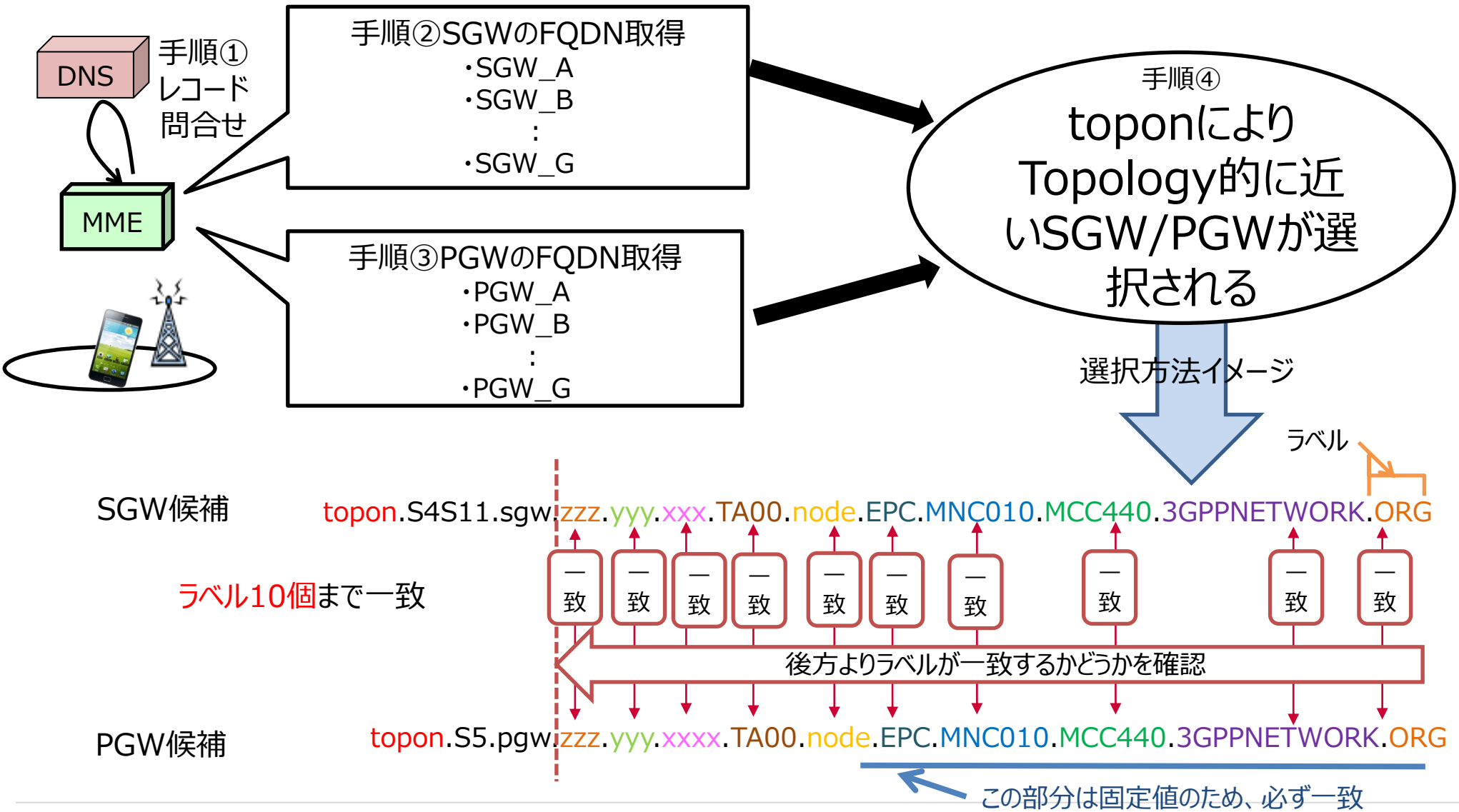
近いSGW/PGWが選ばれる



SGW：Serving GateWay

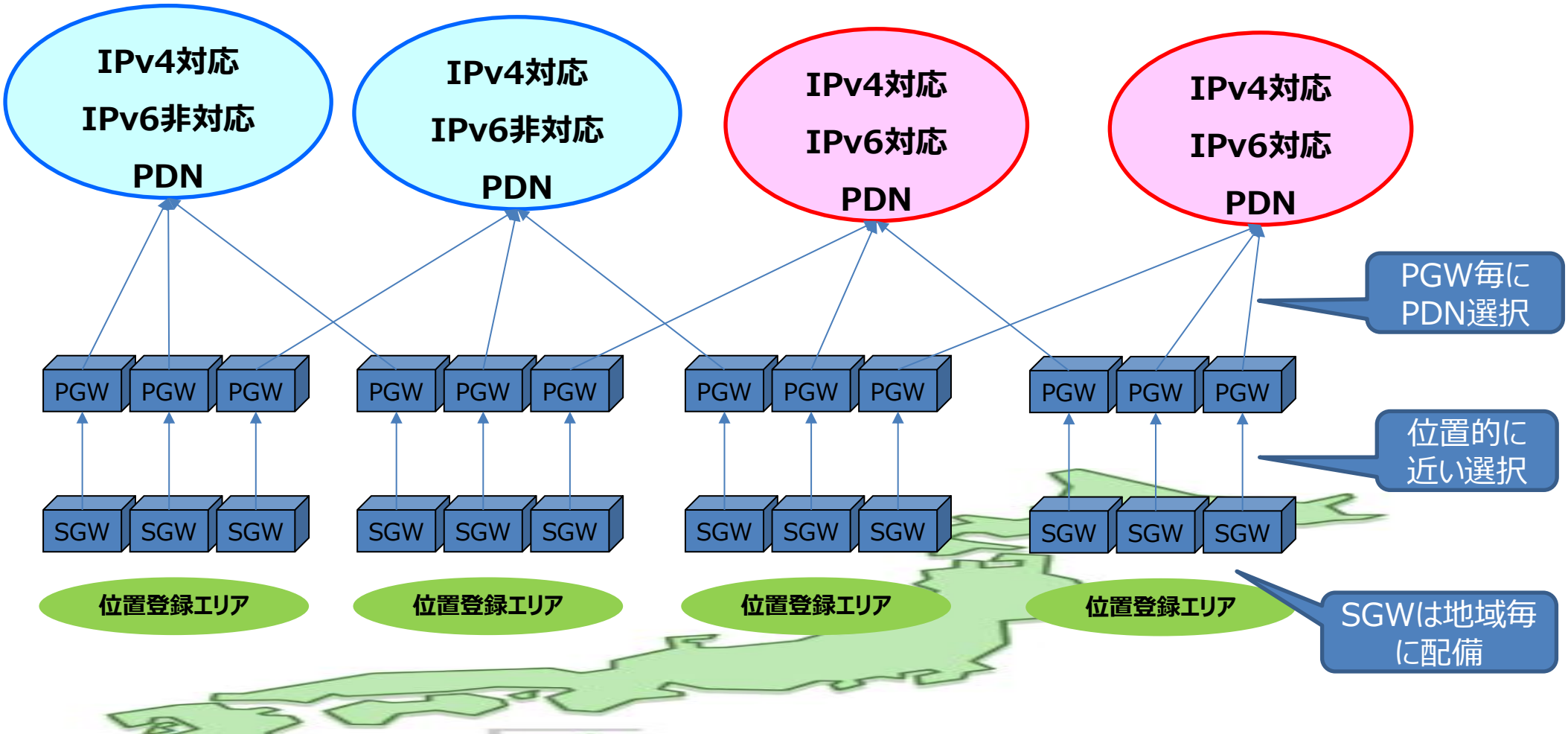
SGW/PGW選択

位置的に近いとの判断はどのように行われるか？



SGW/PGW選択

結局どうなる？

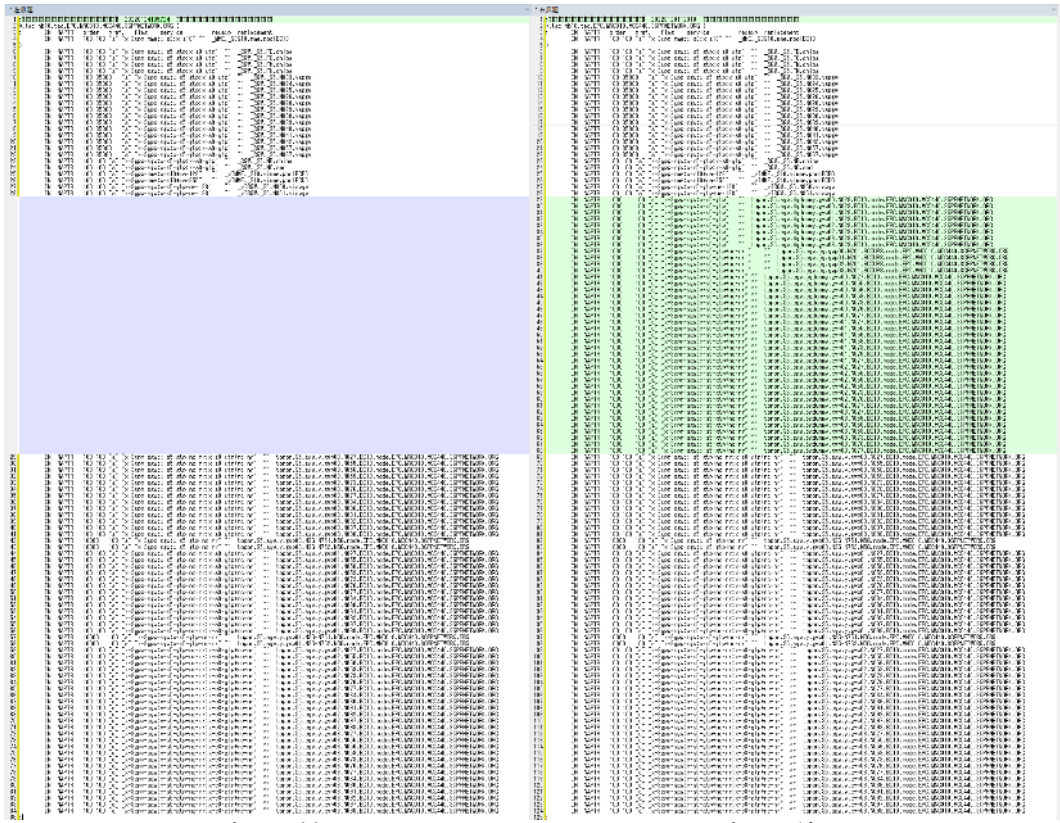


IPv6対応PGWレコードと同じ数だけSGWレコードが必要！！

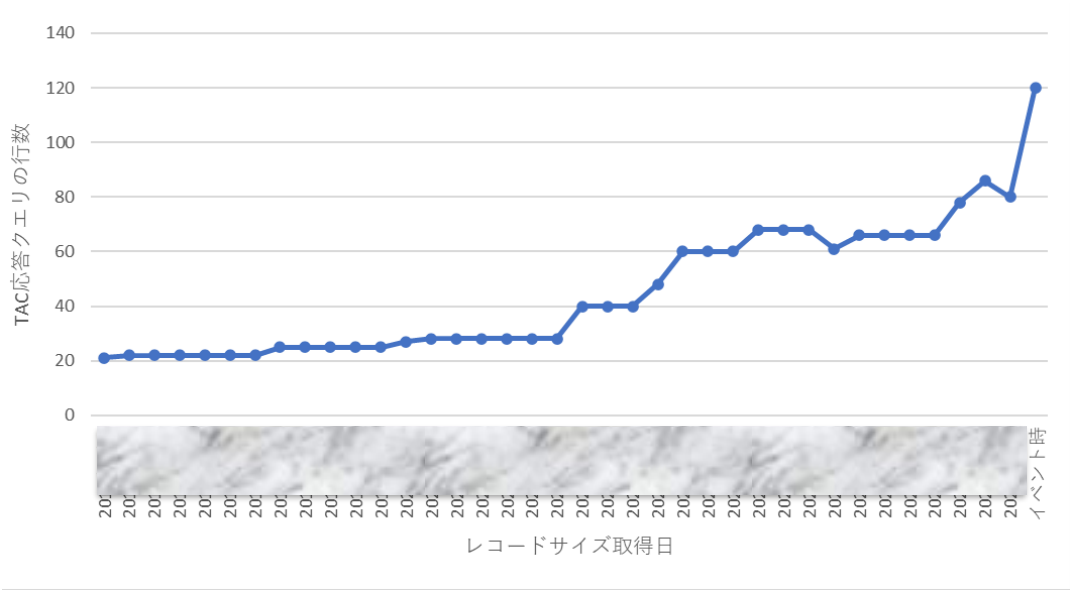
SGWの応答レコードサイズも急激に増大

- いい感じにSGWとPGWを選択しようとした結果、
  レスポンスの行数が急増

1つのNAPTRレコードの応答サイズが1.5倍に増加



変更 前　　変更 後



増えすぎやろ…どうしてこうなった

そもそもレスポンスサイズ(kb)は？

- 代表的かつ比較的大きいレスポンスになるクエリを引いてみると・・・
  23KB
- まぁフォールバックしますよね。ではその割合は？

```
# dig @10.xx.xx.xx test.tac-hb10.tac.EPC.MNC010.MCC440.3GPPNETWORK.ORG naptr
;; Truncated, retrying in TCP mode.
; <<>> DiG 9.9.4-RedHat-9.9.4-74.el7_6.2 <<>> @10.xx.xx.xx test.tac-hb10.tac.EPC.MNC010.MCC440.3GPPNETWORK.ORG naptr
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62051
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 93, AUTHORITY: 14, ADDITIONAL: 219
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;test.tac-hb10.tac.EPC.MNC010.MCC440.3GPPNETWORK.ORG. IN       NAPTR
;; ANSWER SECTION:
～略～
;; Query time: 22 msec
;; SERVER: 100.xx.xx.165#53(100.64.72.165)
;; WHEN:   1 08 15:11:18 JST 2025
;; MSG SIZE  rcvd: 23069
```

モバイル網の（ドコモ国内網の）DNSにおけるTCP fallback割合

- クエリ全体のうち**1/4近くがTCP fallbackの対象**でした
- ちなみに、下記のように常時測定しています(原始的)

毎秒statsを実行して、クエリと問い合わせ種別の数を取得する

```
+++ Statistics Dump +++ (1643669740)
++ Incoming Requests ++
        82080557698 QUERY
            1038 NOTIFY
++ Incoming Queries ++
         132456468 A
            1293 PTR
         8093594258 SRV
        73854505655 NAPTR
              24 ANY
++ Outgoing Queries ++
[View: default]
[View: _bind]
++ Name Server Statistics ++
        82080558736 IPv4 requests received
        62948091805 requests with EDNS(0) received
        19132464597 TCP requests received
        82080558729 responses sent
        19134194665 truncated responses sent
        62948091799 responses with EDNS(0) sent
        81951230132 queries resulted in successful answer
        82080557692 queries resulted in authoritative answer
           1432373 queries resulted in nxrrset
          127895187 queries resulted in NXDOMAIN
++ Zone Maintenance Statistics ++
              1038 IPv4 notifies received
             34060 IPv4 SOA queries sent
                 1 IPv4 AXFR requested
               358 IPv4 IXFR requested
               358 transfer requests succeeded
                 1 transfer requests failed
...
```

秒毎に引き算して、qpsもろもろを算出する

最大qps	15767		
最大qpsのときの	IN_A	IN_SRV	IN_NAPTR
	17	222	15527
クエリ種別の割合	0.1%	1.4%	**98%**
最大qpsのときの	全体	UDP	TCP
	82080558736	62948094139	19132464597
UDP・TCPの割合	–	77%	**23%**

TCP fallbackするレコードは？

- TACレコード(TAC数)とAPNレコード(1つ)がTCP fallback

TAC番号 **＊必ずTCP fallbackする**

```
*.tac-hb10.tac.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
;  IN NAPTR order pref. flag service   regexp replacement
  IN NAPTR 100 100 "s" "x-3gpp-mme:x-s3:x-s10" "" _MME.～～
)
  IN NAPTR 100 35000 "s" "x-3gpp-sgw:x-s5-gtp:x-s8-gtp" "" _SGW.～～
  IN NAPTR 100 35000 "s" "x-3gpp-sgw:x-s5-gtp:x-s8-gtp" "" _SGW.～～
  ...
  IN NAPTR 100 65534 "s" "x-3gpp-sgw:x-s5-gtp" "" _SGW.～～
  IN NAPTR 100 65534 "s" "x-3gpp-sgw:x-s5-gtp" "" _SGW.～～
  ...
  IN NAPTR 100 65534 "a" "x-3gpp-sgw:x-s5-gtp+nc-nr" "" topon.S5.sgw.5g.00.201.～.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65534 "a" "x-3gpp-sgw:x-s5-gtp+nc-nr" "" topon.S5.sgw.5g.00.213.～.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65534 "a" "x-3gpp-sgw:x-s5-gtp+nc-nr" "" topon.S5.sgw.5g.00.231.～.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  ...
```

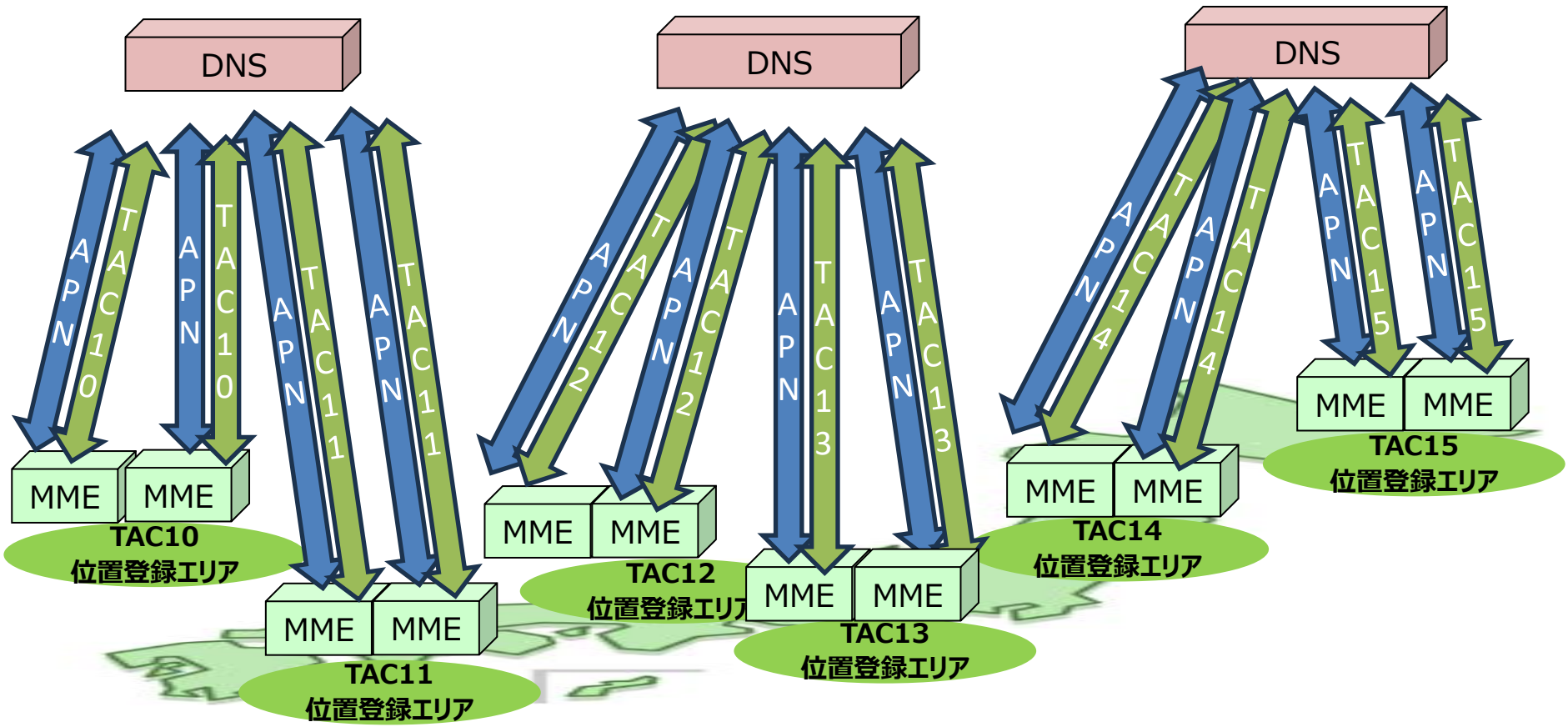
マス向けAPN **＊必ずTCP fallbackする**

```
spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
;  IN NAPTR order pref. flag service   regexp replacement
  IN NAPTR 100 64205 "" "x-3gpp-pgw:x-s5-gtp" "" ～～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
)
  IN NAPTR 100 62535 "" "x-3gpp-pgw:x-s5-gtp" "" ～～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65403 "" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" 5gv6.～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65403 "" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" 5gv6.～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65403 "" "x-3gpp-pgw:x-s5-gtp+nc-v6" "" v6.～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  IN NAPTR 100 65403 "" "x-3gpp-pgw:x-s5-gtp+nc-v6" "" v6.～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG
```

```
5gv6.～.spmode.ne.jp.apn.EPC.MNC010.MCC440.3GPPNETWORK.ORG (
;  IN NAPTR order pref. flag service   regexp replacement
  IN NAPTR 100 65475 "a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.00.201.～.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
)
  IN NAPTR 100 65435 "a" "x-3gpp-pgw:x-s5-gtp+nc-nr.v6" "" topon.S5.pgw.00.202.～.node.EPC.MNC010.MCC440.3GPPNETWORK.ORG
  ...
```

TCP Fallbackするレコードの問い合わせイメージ

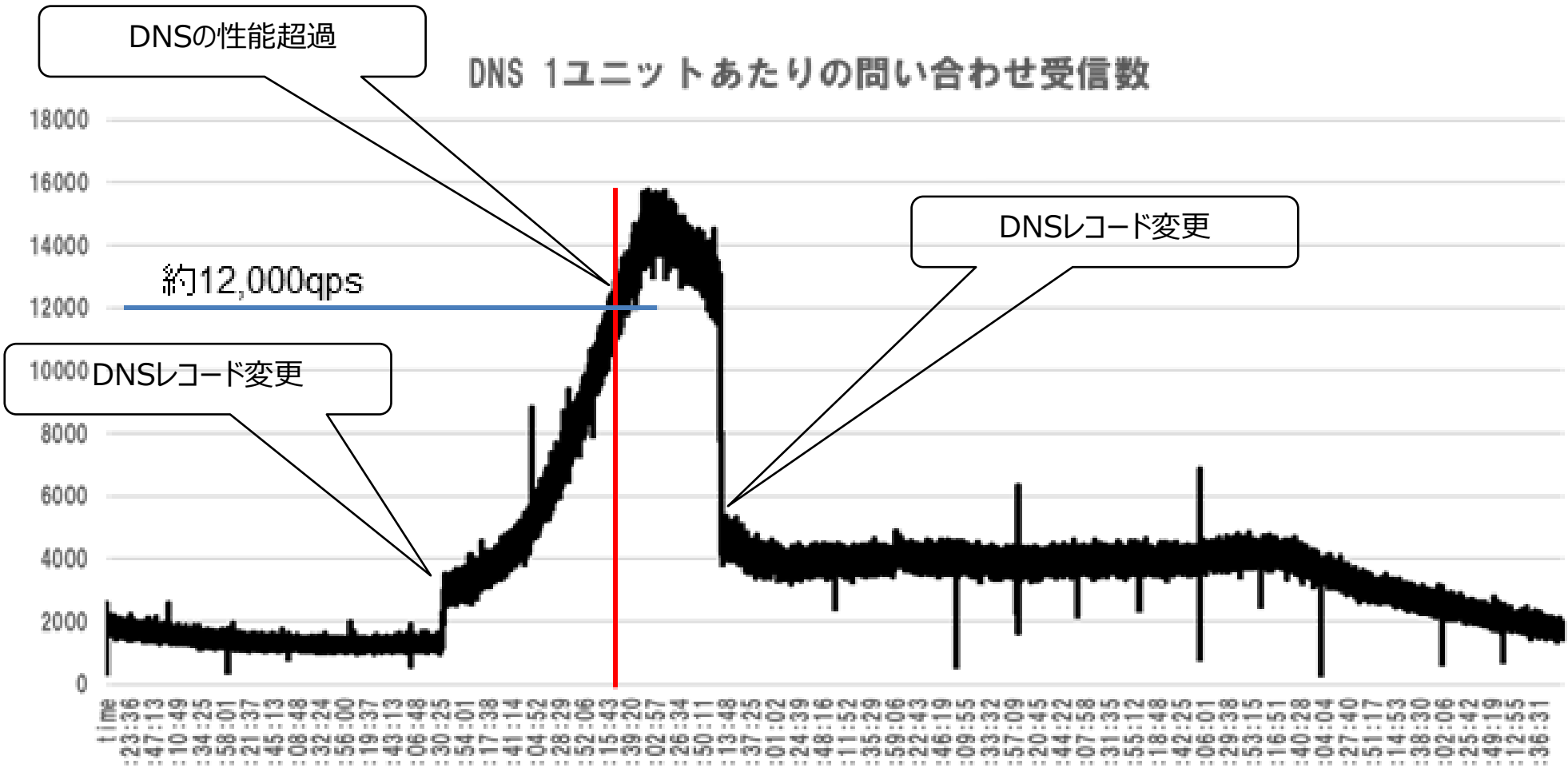
実際にはMMEとDNSの括り付けは、もう少し混在していましたがMMEは自分のエリアのTACとAPNを問い合わせる
DNSはAPNの問い合わせをMME数分受ける＋各MMEからMMEが所属するTACの問い合わせを受けるという形



発生した事象

再掲

DNSレコード変更により、DNSクエリが急増しDNSの性能超過！！



対応するために！

- MMEのキャッシュは増やせない（時間がかかる、メモリ影響）
- DNSレコードも減らせない（v6シングルスタックのため）
- DNSサーバの性能も12,000qpsまでだった　★改善点

検証結果

商用同条件で検証を行った結果、DNSレコード数を増加させると応答遅延値が増加することを確認。また、IPv6分のレコードを追加した状況で14,000qpsにおいては信号交換機のタイムアウトとなる応答遅延値の発生を確認。CPU及びメモリの使用量は正常範囲のため処理バッファ等のチューニングが必要。

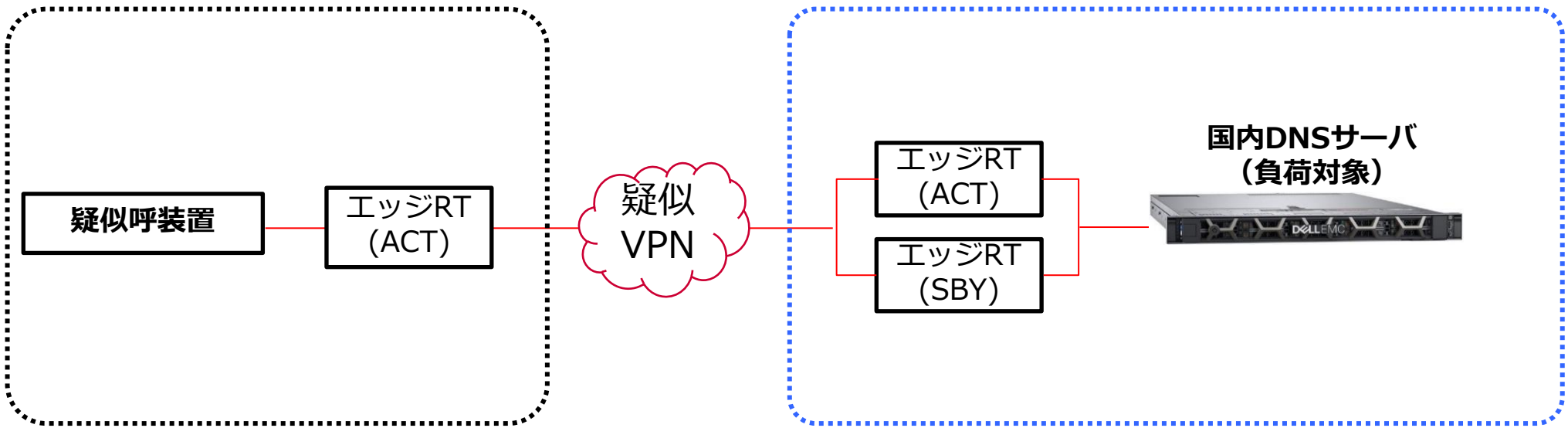
	負荷[qps]		
	10,000	12,000	14,000
DNSレコード	応答遅延値（最悪値）		
IPv6追加前	485ms	419ms	832ms
IPv6追加後	423ms	815ms	3241ms

- DNSの性能をチューニングすることに。。。

DNSの性能

よく使われるDNSプロダクトのTCP fallback性能

- 測定条件（環境）
  - 各種情報取得・監視用シェルスクリプトが動作している状態で検証を実施する
  - 国内DNSスレーブサーバ～疑似呼装置間は商用と同じホップ数とし、疑似呼装置をMMEと見立てる



よく使われるDNSプロダクトのTCP fallback性能

- 測定条件（内容）
  - Stair負荷検証
    - 2,000qps→25,000qpsまで段階的に負荷を上げて（1分ごとに1,000qps上げて）、DNSエラーが継続的に出ない、かつ、クエリ要求送信～応答受信のTATが1sを越えないqpsを判断する

      ＊新レコードでは応答遅延が爆増し信号交換機のタイムアウトを超えるため、
      測定条件にTAT制限も必要
  - Burst負荷検証
    - Stair負荷で求めたqpsを10分間かけ続け、 DNSエラーが継続的に出ない、かつ、クエリ要求送信～応答受信のTATが1sを越えないことを確認する
  - 機能検証
    - Stair負荷・Burst負荷を満たすqpsをかけた状態で、レコード追加削除（マスタ・スレーブ同期）が正常に行えることを確認する
  - 長期安定検証
    - Stair負荷・Burst負荷を満たすを満たすqpsを48時間かけ続け、 DNSエラーが継続的に出ない、かつ、クエリ要求送信～応答受信のTATが1sを越えないことを確認する

よく使われるDNSプロダクトのTCP fallback性能

- 4つの試験でエラーとならなかった負荷量を性能諸元とした

	障害前の想定諸元 (旧レコード)	チューニング前の再測定 (新レコード)
QPS性能	24000 qps	12000 qps

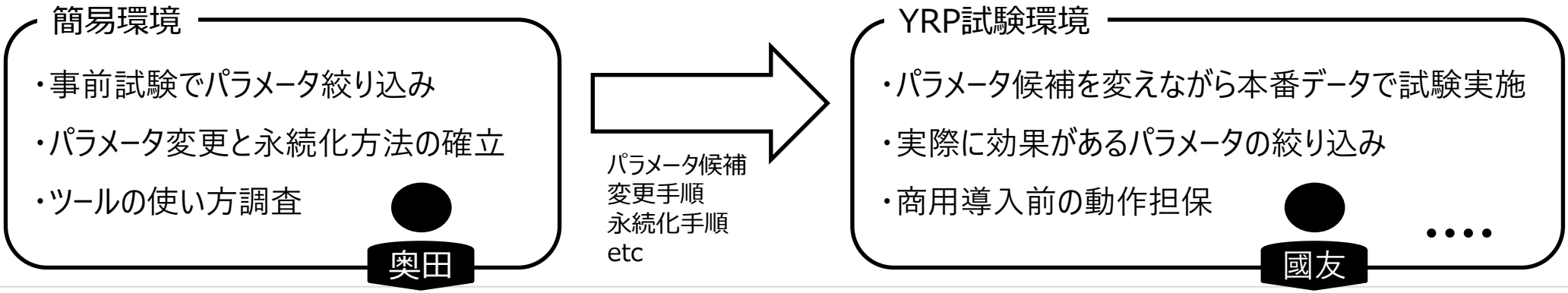
DNSで行った対策

DNSのTCP性能を上げよう

- 時間がないのであたりを付けて、性能検証をぶん回す作戦
  - CPU性能を使い切っていないところがポイント
- 3つのあたり
  - kernelパラメータ
    - TIME_WAITとかsysctlのnet.core…とかそういうやつ
  - NICパラメータ
    - ethtoolで設定する系、tcp-segmentation-offloadとか
  - bind9パラメータ
    - バッファサイズやクライアント数
- テスター
  - 最新のBreakingPointを手に入れるまでは
    適当なLinux PCを並べて適当なオープンソースで負荷を掛けて使えるか試す

手元で試してパラメータを絞り込み

- とりあえず自○サーバとかすぐ触れるサーバにCentOS 7とbind9 9.9.4-61.el7を片っ端から建てる
- ググってDNS負荷系ツールを探して全部試す
  - queryperf → UDPのみ
  - dnsperf → TCPいけるけど、負荷を上げづらい？
  - resperf → TCPいけるけど、試験データを作らないといけない
- 商用模擬のDNSレコードを作成して、手元で測定できる環境を構築
- dnsperfとresperfで負荷を掛け、効果があるパラメータを絞り込む



チューニング候補となったパラメータ

- 今回の事象に関係ありそうなパラメータをとにかく列挙
  - Kernel
    - ソケットバッファサイズ
    - キュー長
    - 割り込み
    - TCPパラメータ
  - NIC
    - キュー長
    - オフロード
  - Bind9
    - バッファサイズ、メモリサイズ
    - TCP関係

項番	カテゴリ	対象	項目	パラメータ
k-1	kernel	TCP/UDP	受信バッファサイズ	net.core.rmem_default
k-2	kernel	TCP/UDP	受信バッファサイズ	net.core.rmem_max
k-3	kernel	TCP/UDP	送信バッファサイズ	net.core.wmem_default
k-4	kernel	TCP/UDP	送信バッファサイズ	net.core.wmem_max
k-5	kernel	UDP	受信バッファサイズ	net.ipv4.udp_rmem_min
k-6	kernel	UDP	送信バッファサイズ	net.ipv4.udp_wmem_min
k-7	kernel	UDP	メモリ使用量	net.ipv4.udp_mem
k-8	kernel		SoftIRQの上限パケット数	net.core.netdev_budget
k-9	kernel	TCP	バックログキュー長	net.core.netdev_max_backlog
k-10	kernel	TCP	キュー長	net.core.somaxconn
k-11	kernel	TCP	TCP処理	net.core.optmem_max
k-12	kernel	TCP	受信バッファサイズ	net.ipv4.tcp_rmem
k-13	kernel	TCP	送信バッファサイズ	net.ipv4.tcp_wmem
k-14	kernel	TCP	メモリ使用量	net.ipv4.tcp_mem
k-15	kernel	TCP	TCP処理	net.ipv4.tcp_fin_timeout
k-16	kernel	TCP	TCP処理	net.ipv4.tcp_tw_reuse
k-17	kernel	TCP	TCP処理	net.ipv4.tcp_tw_recycle
k-18	kernel	TCP		net.ipv4.tcp_max_syn_backlog
k-19	kernel	TCP		net.ipv4.tcp_slow_start_after_idle
k-20	kernel	TCP		net.ipv4.tcp_frto
k-21	kernel	システム	ソケット数関連	fs.file-max
n-1	NIC	ip/ifconfig	送信キュー長	txqueuelen
n-2	NIC	ip/ifconfig	受信キュー長	rxqueuelen
n-3	NIC	ethtool	受信割り込み	rx-usecs
n-4	NIC	ethtool	送信割り込み	tx-usecs
n-5	NIC	ethtool	Current HW Setting Tx	ethtool -G em1/p1p1 tx
n-6	NIC	ethtool	Current HW Setting Rx	ethtool -G em1/p1p1 rx
n-7	NIC	ethtool	送信割り込み	adaptive-tx
n-8	NIC	ethtool	受信割り込み	adaptive-rx
n-9	NIC	ethtool	受信チェックサム	rx-checksumming
n-10	NIC	ethtool	送信チェックサム	tx-checksumming
n-11	NIC	ethtool	送信パケット分割	generic-segmentation-offload
n-12	NIC	ethtool	TCP送信セグメンテーション	tcp-segmentation-offload
n-13	NIC	ethtool	UDP送信フラグメント	udp-fragmentation-offload
n-14	NIC	ethtool	受信パケット結合	generic-receive-offload
n-15	NIC	ethtool	受信TCPパケット結合	large-receive-offload
b-1	bind9	バージョン	バージョン	バージョン
b-2	bind9	クエリ数		clients-per-query
b-3	bind9	メモリ		datasize
b-4	bind9	TCP	TCPクライアント数	tcp-clients
b-5	bind9	TCP	受信バッファサイズ	tcp-receive-buffer
b-6	bind9	TCP	送信バッファサイズ	tcp-send-buffer
b-7	bind9	UDP	UDPパケットサイズ	max-udp-size
b-8	bind9	UDP	UDPパケットサイズ	edns-udp-size
b-9	bind9	UDP	受信バッファサイズ	udp-receive-buffer
b-10	bind9	UDP	送信バッファサイズ	udp-send-buffer
b-11	bind9	DNS	レスポンスサイズ	minimal-responses
b-12	bind9	TCP	タイムアウト	tcp-idle-timeout
b-13	bind9	TCP		tcp-initial-timeout
b-14	bind9	TCP		tcp-listen-queue
b-15	bind9			clients-per-query
b-16	bind9			max-clients-per-query

確認手法

- チューニングにおいては厳密性に欠けるけど簡易に調べる手段を用意
  - UDPバッファサイズ
    - コマンド「netstat –su」を1分ほど空けて2度以上実行し、receive buffer errorsのカウンタが増加していれば、バッファ溢れ発生。
  - TCPソケット数
    - コマンド「sar -n SOCK 1」を実行し、tcpsckのカウンタが特定の値に張り付いていればソケット数不足の懸念あり。
  - TCP SYNキュー
    - コマンド「netstat –st」を1分ほど空けて2度以上実行し、下記のカウンタが増加していれば、TCP SYNキュー溢れ発生。
      - times the listen queue of a socket overflowed
      - SYNs to LISTEN sockets dropped
- その後、秘伝のタレとして受け継がれてしまった…

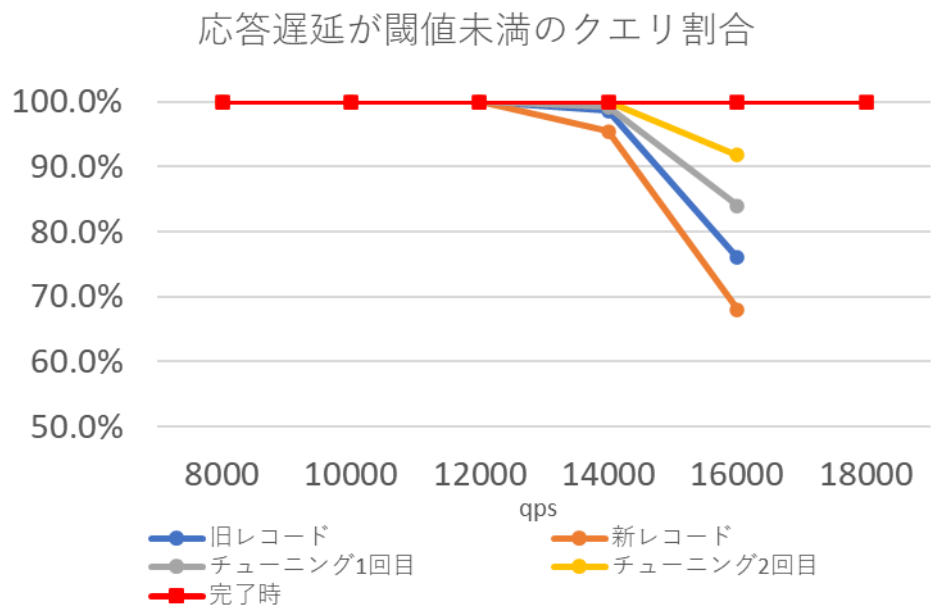
チューニングしたパラメータ

- チューニング効果があったパラメータ
  - Kernel
    - バッファサイズ
    - キュー長
    - TW_REUSE
  - NIC
    - ring buffer
  - Bind9
    - バッファサイズ、メモリサイズ
    - TCP関係
- minimal-responses
  - 元のレコードがデカすぎて効果が薄かった
  - MME動作影響が怖いので無効とした
- いいわけ
  - 他にもいじったほうがいいものもあるはず
  - 効果がないものも含まれます
    - 時短のため複数同時変更したため

項番	カテゴリ	対象	項目	パラメータ	どう変更したか
k-1	kernel	TCP/UDP	受信バッファサイズ	net.core.rmem_default	増やす
k-2	kernel	TCP/UDP	受信バッファサイズ	net.core.rmem_max	増やす
k-3	kernel	TCP/UDP	送信バッファサイズ	net.core.wmem_default	増やす
k-4	kernel	TCP/UDP	送信バッファサイズ	net.core.wmem_max	増やす
k-5	kernel	UDP	受信バッファサイズ	net.ipv4.udp_rmem_min	
k-6	kernel	UDP	送信バッファサイズ	net.ipv4.udp_wmem_min	
k-7	kernel	UDP	メモリ使用量	net.ipv4.udp_mem	
k-8	kernel		SoftIRQの上限パケット数	net.core.netdev_budget	増やす
k-9	kernel		バックログキュー長	net.core.netdev_max_backlog	増やす
k-10	kernel	TCP	キュー長	net.core.somaxconn	増やす
k-11	kernel	TCP	TCP処理	net.core.optmem_max	
k-12	kernel	TCP	受信バッファサイズ	net.ipv4.tcp_rmem	増やす
k-13	kernel	TCP	送信バッファサイズ	net.ipv4.tcp_wmem	増やす
k-14	kernel	TCP	メモリ使用量	net.ipv4.tcp_mem	増やす
k-15	kernel	TCP	TCP処理	net.ipv4.tcp_fin_timeout	
k-16	kernel	TCP	TCP処理	net.ipv4.tcp_tw_reuse	1
k-17	kernel	TCP	TCP処理	net.ipv4.tcp_tw_recycle	0
k-18	kernel	TCP		net.ipv4.tcp_max_syn_backlog	増やす
k-19	kernel	TCP		net.ipv4.tcp_slow_start_after_idle	1
k-20	kernel	TCP		net.ipv4.tcp_frto	2
k-21	kernel	システム	ソケット数関連	fs.file-max	
n-1	NIC	ip/ifconfig	送信キュー長	txqueuelen	
n-2	NIC	ip/ifconfig	受信キュー長	rxqueuelen	
n-3	NIC	ethtool	受信割り込み	rx-usecs	
n-4	NIC	ethtool	送信割り込み	tx-usecs	
n-5	NIC	ethtool	Current HW Setting Tx	ethtool -G em1/p1p1 tx	増やす
n-6	NIC	ethtool	Current HW Setting Rx	ethtool -G em1/p1p1 rx	増やす
n-7	NIC	ethtool	送信割り込み	adaptive-tx	
n-8	NIC	ethtool	受信割り込み	adaptive-rx	
n-9	NIC	ethtool	受信チェックサム	rx-checksumming	
n-10	NIC	ethtool	送信チェックサム	tx-checksumming	
n-11	NIC	ethtool	送信パケット分割	generic-segmentation-offload	
n-12	NIC	ethtool	TCP送信セグメンテーション	tcp-segmentation-offload	
n-13	NIC	ethtool	UDP送信フラグメント	udp-fragmentation-offload	
n-14	NIC	ethtool	受信パケット結合	generic-receive-offload	
n-15	NIC	ethtool	受信TCPパケット結合	large-receive-offload	
b-1	bind9	バージョン	バージョン	バージョン	
b-2	bind9	クエリ数		clients-per-query	
b-3	bind9	メモリ		datasize	
b-4	bind9	TCP	TCPクライアント数	tcp-clients	増やす
b-5	bind9	TCP	受信バッファサイズ	tcp-receive-buffer	
b-6	bind9	TCP	送信バッファサイズ	tcp-send-buffer	
b-7	bind9	UDP	UDPパケットサイズ	max-udp-size	
b-8	bind9	UDP	UDPパケットサイズ	edns-udp-size	
b-9	bind9	UDP	受信バッファサイズ	udp-receive-buffer	
b-10	bind9	UDP	送信バッファサイズ	udp-send-buffer	
b-11	bind9	DNS	レスポンスサイズ	minimal-responses	無効
b-12	bind9	TCP	タイムアウト	tcp-idle-timeout	
b-13	bind9	TCP		tcp-initial-timeout	
b-14	bind9	TCP		tcp-listen-queue	最大
b-15	bind9			clients-per-query	
b-16	bind9			max-clients-per-query	

チューニング結果

- 色々チューニングを試した結果、6000qpsくらい性能向上！！
  - たった6000qpsのために…
- 負荷印加**18000qps**でロスなしを達成
  - NIC (1GbE) の性能的にもここらが上限っぽい感じ
    - 19000qpsだと 889,533,120bps
    - 20000qpsだと 906,478,720bps



チューニング結果

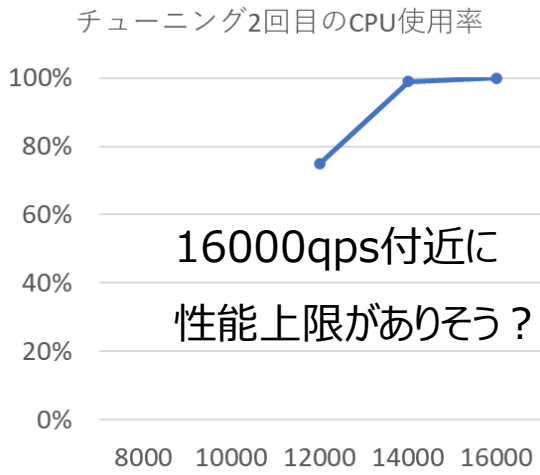
- CPU負荷上限を加えた再試験や安全率などを考慮し
  チューニング後諸元を **18000 qps** とした

	障害前の想定諸元 (旧レコード)	チューニング前の再測定 (新レコード)	チューニング後 (新レコード)
QPS性能	24000 qps	12000 qps	**18000 qps**

項目	負荷量（qps数）	qps数	備考	
1. Stair負荷検証	2,000qpsから25,000qpsまで 1,000qpsずつ上昇	**19,000qps**	20,000qpsでTAT閾値超過を確認	
2. Burst負荷検証	18,000qps	**18,000qps**	19,000qpsでTAT閾値超過を確認	
			CPU平均使用率	17%
			メモリ平均使用率	6%
3. 機能検証	18,000qps	**18,000qps**	問題なし、レコード追加・ゾーン転送ができることを確認	
4. 長期安定検証	18,000qps	**18,000qps**	総クエリ数	2,764,415,638
			成功数	2,764,415,638
			失敗数	0
			成功率	100.00%
			最大TAT	55 msec

その後

- CPUを100%使いきるところまでは達成
  - CPUを使い切った場合のだいたいの性能上限が判明
  - 後はチューニングでどれだけ引き上げられるか



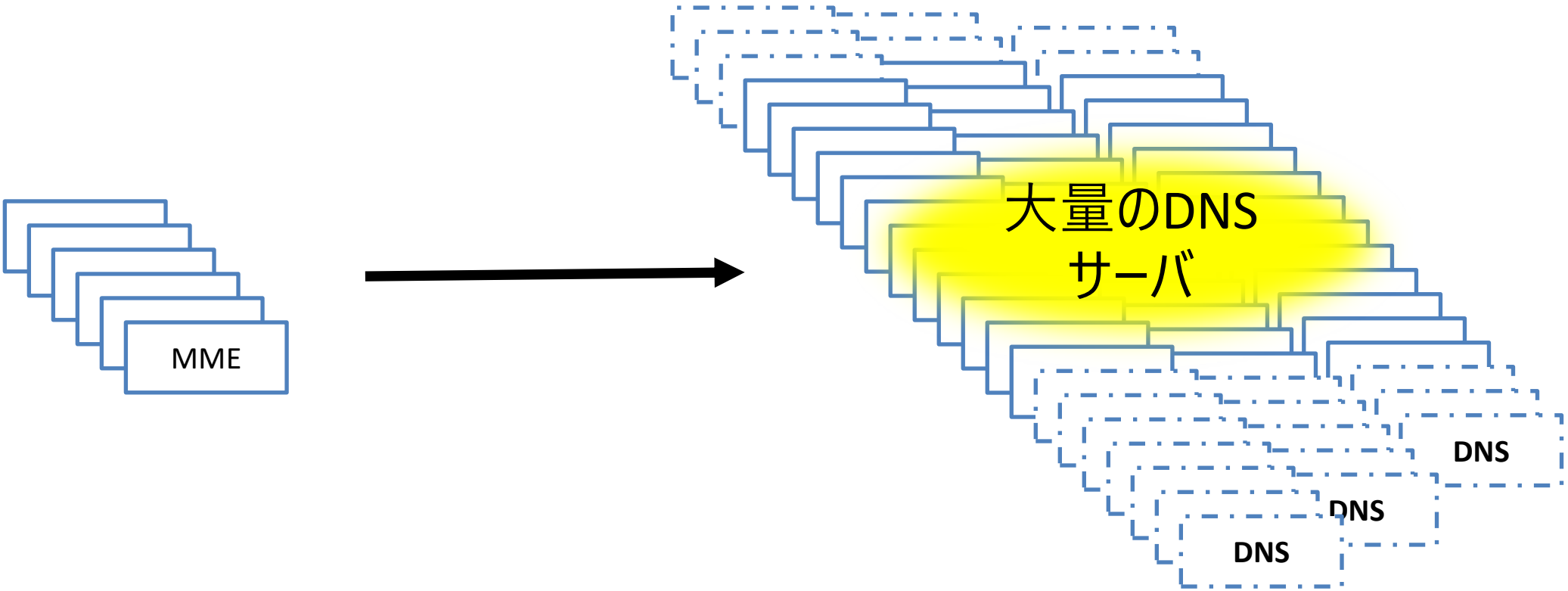
- その後、チューニング含め延々と繰り返される試験
  - X月XX日　再検証
  - X月XY日　TTL変更実験
  - X月YY日　リトライ検証
  - Y月XX日　実MME検証
  - Y月YY日　再々検証
  - Z月YY日　MME対向試験
  - Z月ZZ日　MME2対向試験　※運用中のMMEはいくつかのバージョンがありまして・・・
  - 　：
- 新レコードを適用

余談2

- bind9 9.11.4-26.P2.el7 にアップデートしてStair負荷検証すると、**22000 qps** を確認
  - UDPバッファ溢れ、TCP再送等が発生
  - スループットは 959.236 Mbps なのでNIC性能的にも上限
  - エラー無し諸元は **20000 qps** 程度と想定される

余談3

- そもそもどうなるはずだった？
- 大量の安価なDNSサーバで大量のクエリを処理するはずだった
  - でもその思想がうまく伝わらず、、、
  - 性能諸元に対して適切過ぎる量のDNSサーバしか存在しなかった

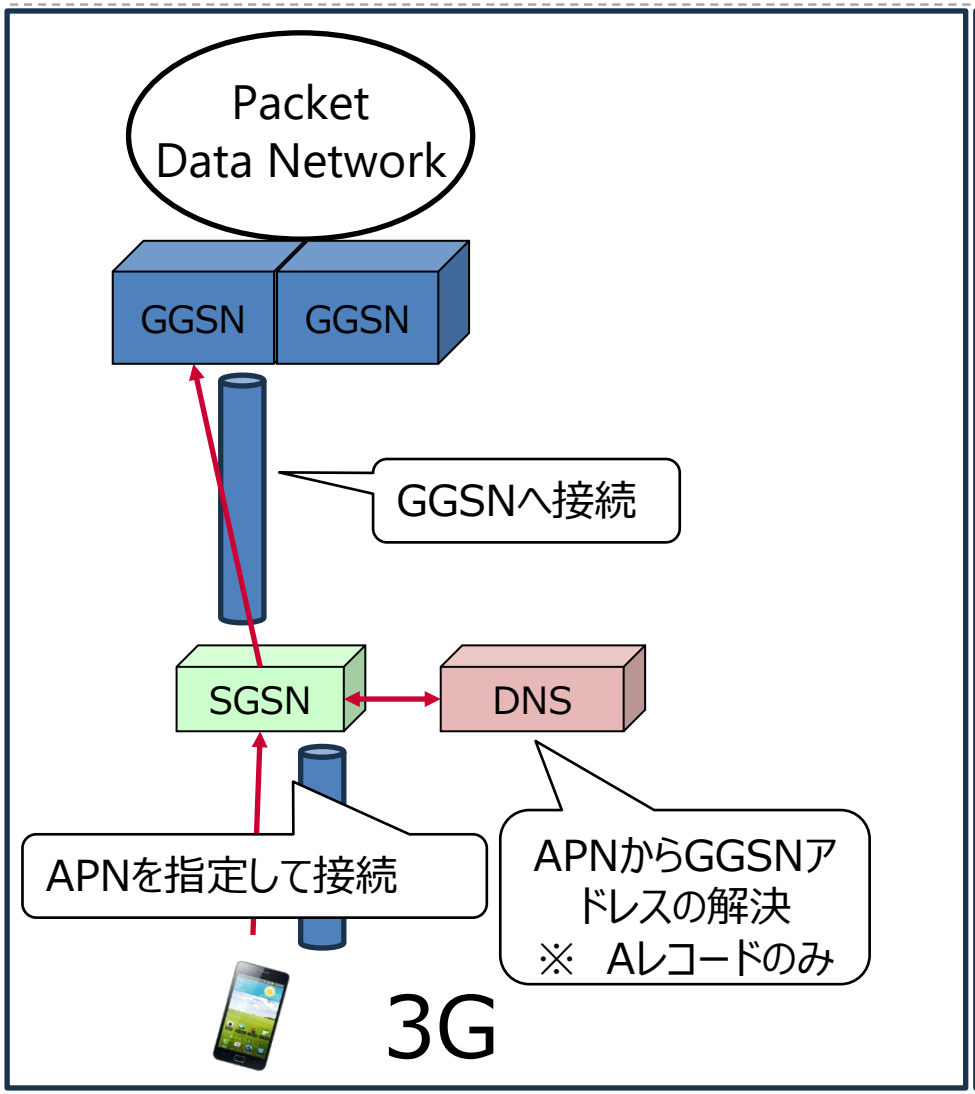


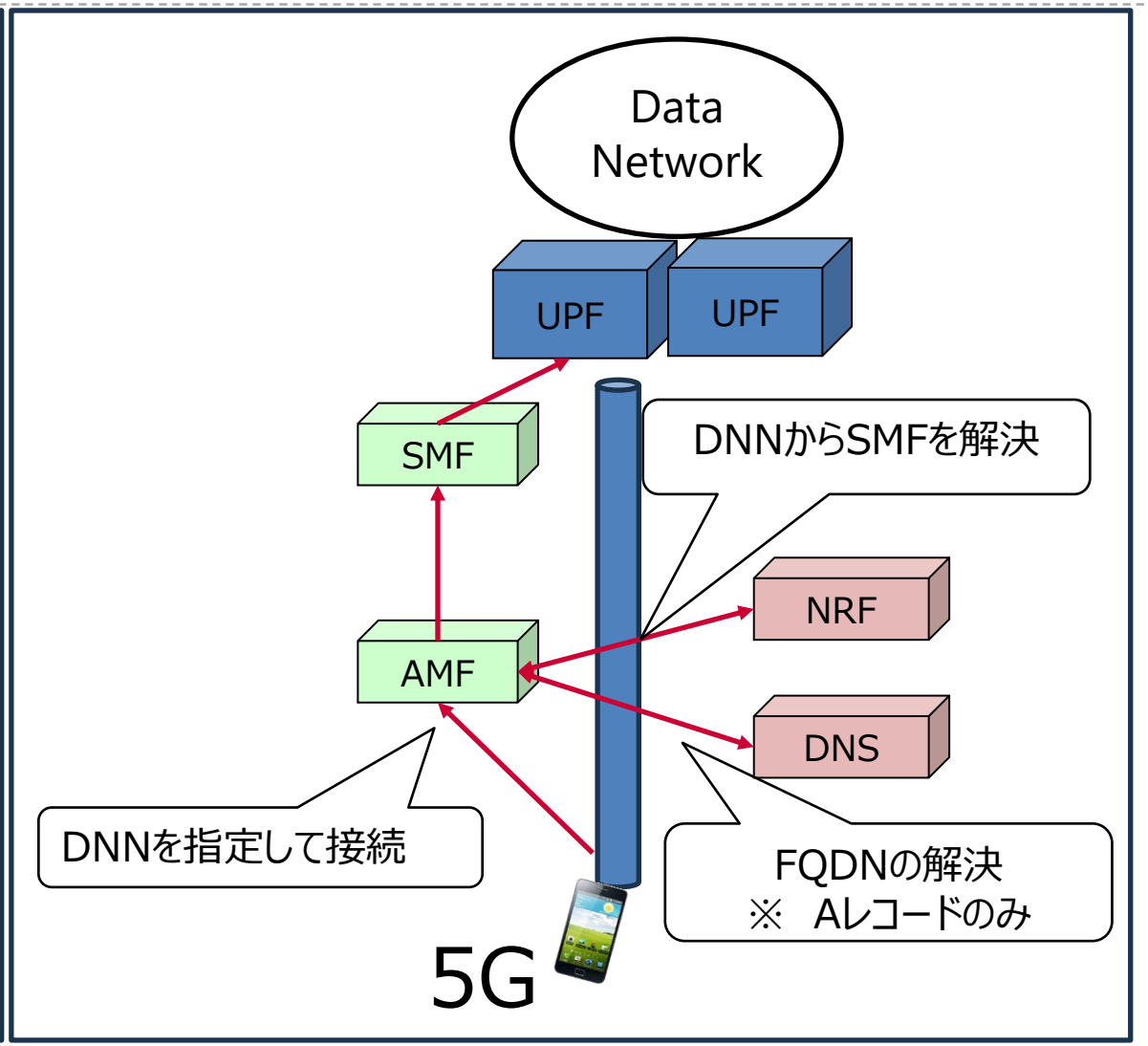
結論

まとめ

- DNS生誕からおよそ40年、枯れた枯れたと言われ続けたが、
  懲りずにDNSの話をしてみました
- 今なお辛いDNS
- DNSサーバは数をケチるな
- 皆さんはどうやってDNSのTCP性能を引き出してますか？

おまけ　3G/(4G)/5GでのDNSの役割





3GでもTCP 5GでもDNSは使っていますが、TCP fallbackは発生しません！

NRF:Network Repository Function

参考 過去ログから拾った確認コマンドたち

- rndc status
- tc -s qdisc
- sar -n DEV 1
- sar -n EDEV
- sar -n SOCK 1
- sar -n ETCP 1
- sar -n UDP 1
- /proc/net/softnet_stat
- /proc/softirqs
- ifconfig
- ss -m --info -npta
- dnsping
- …