Česká společnost uživatelů otevřených systémů EurOpen.CZ
Czech Open System Users' Group
www.europen.cz

47. konference

Sborník příspěvků

Hotel VZ Měřín, Slapy
4.–7. 10. 2015

Programový výbor:

Pavel Růžička
Martin Lávička
Jiří Sitera
Vladimír Rudolf

Sborník příspěvků z 47. konference EurOpen.CZ, 4.–7. 10. 2015

Plzeň 2015. První vydání.
Editor: Vladimír Rudolf
Sazba a grafická úprava: Ing. Miloš Brejcha – Vydavatelský servis, Plzeň
e-mail: servis@vydavatelskyservis.cz

Tisk: Typos, tiskařské závody, s. r. o.
Podnikatelská 1160/14, Plzeň

Příspěvky neprošly redakční ani jazykovou úpravou.

ISBN 978-80-86583-29-7

Obsah

Softwarově definovaná síťová infrastruktura – Testbed service framework

Michal Hažlinský

E-mail: hazlinsky@cesnet.cz

Abstrakt

Termín „Software defined networks“ (SDN) je ve světě sítí znám už několik let. Do širokého povědomí se dostal zejména spolu s protokolem OpenFlow a často se mezi těmito dvěma pojmy nesprávně kladlo rovnítko. Myšlenka softwarově definovaných sítí je nyní už řádově širší oblastí než „jen“ otázka řízení přepínání rámců na základě příslušnosti konkrétního datagramu k určitému datovému toku. Tento rychlý vývoj přinesl potřebu reálných síťových prostředí, kde by se nové přístupy k řízení počítačových sítí daly testovat v praxi. Ve světě tak vznikla různorodá experimentální prostředí – testbedy. Tento článek představí možnost využití různých virtualizačních nástrojů k vytvoření softwarově řízené infrastruktury pro dynamickou tvorbu testbedů nad reálnou fyzickou infrastrukturou.

1 Úvod

Je tomu již několik let, co oborem počítačových sítí začal rezonovat termín „Software defined networks“ (SDN). Ten představuje přesně takový koncept, jaký naznačuje jeho doslovný překlad, tedy přístup, kdy se pomocí různých nových protokolů umožňuje řízení počítačových sítí centrální inteligencí, kterou reprezentuje software, označovaný jako kontrolér. Ten bývá umístěn na výkonných serverech a lze ho chápat jako jakýsi operační systém sítě. Nad ním lze pak provozovat specifické řídící aplikace, které umožní přizpůsobit chování sítě na míru samotné aplikaci provozované nad sítí (např. distribuce dat mezi datacentry, streaming videa, privátní propojení pracovišť, ale může to být i nejznámější z aplikací, kterou je samotný world wide web).

Přesunutí řídící inteligence z jednotlivých směrovačů do centrálního kontroléru poskytuje řadu výhod, ale nese s sebou zároveň i spoustu nových problémů, které distribuovaný charakter řízení sítí řeší automaticky ze své podstaty (např.

odolnost proti výpadkům, nutnost zajistit spolehlivé a bezpečné spojení mezi síťovým prvkem a kontrolérem a různé další). Přesto se vývoj v této oblasti stále zrychluje a zabývá se jím stále více lidí. Nový prostor se také vytvořil pro uplatnění nezávislých a otevřených projektů v oboru, který byl dříve výsadou velkých výrobců síťového HW a jejich uzavřených systémů.

Tato nová situace vyvolala nikoliv novou, ale výrazně zvýšenou potřebu přístupu k síťovým prostředím, kde by bylo možné nová řešení a aplikace testovat v podmínkách co nejvíce podobných reálným sítím. Pro účely výzkumu a vývoje vznikla sice řada simulátorů sítí, ale bez testování v autentických podmínkách velké sítě se lze jen těžko obejít.

Některé akademické sítě ve světě dlouhodobě udržují části svých infrastruktur přístupné pro provádění experimentů (o takové vyhrazené části se mluví obecně jako o testbedu). Udržovat takto část infrastruktury mimo produkční provoz je však nákladné, a proto jich nevzniká mnoho. Pokročilé virtualizační nástroje spolu s konceptem SDN však přináší provozovatelům testbedů možnost flexibilně sdílet takovou infrastrukturu mezi více experimenty, a tak nabízet přístup k ní široké vědecko-výzkumné komunitě.

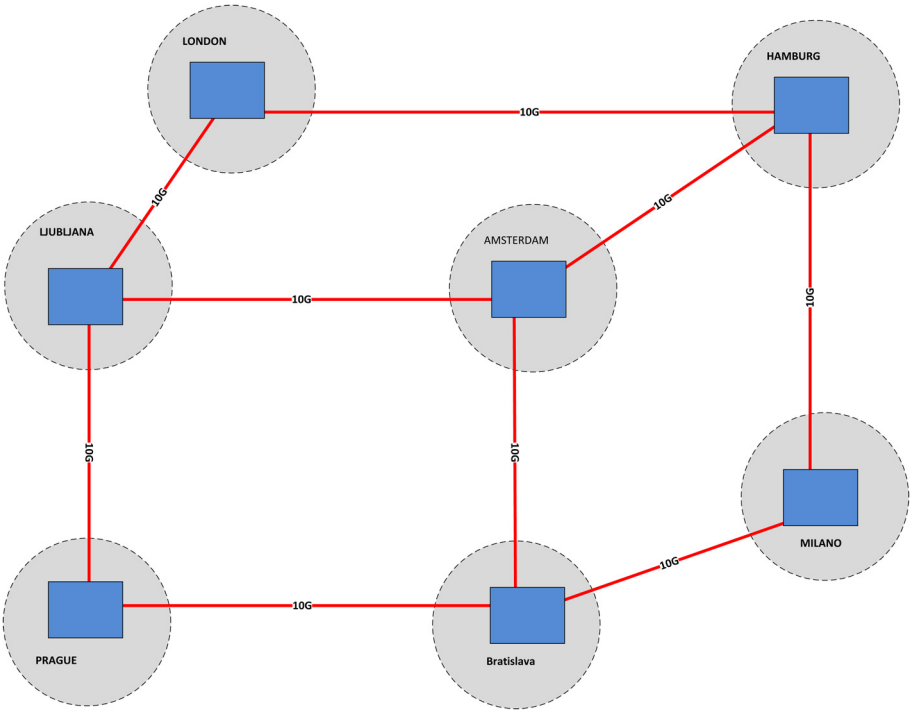
2 Geant testbed service Framework

V roce 2013 v rámci již několikátého cyklu panevropského projektu GEANT [1] (cyklus s označením GN3plus, aktuálně běží cyklus GN4), na němž se podílí i sdružení CESNET (provozovatel české sítě národního výzkumu), byla představena nová aktivita s cílem vybudovat v rámci evropské akademické sítě GEANT také infrastrukturu, která by nabízela možnost si nadefinovat a na čas rezervovat síťové prostředky a provádět tak bezpečně experimenty v reálném prostředí velké sítě. Vznikla tak nejen fyzická infrastruktura čítající aktuálně sedm uzlů v sedmi evropských městech, ale také vlastní řídící software, umožňující dynamické přidělování prostředků na základě požadavků definujících topologii a druhy prostředků. Tento řídící software je nadále aktivně vyvíjen v rámci aktivity GN4 SA2 – Geant Testbed Service (GTS) [2] jako open source projekt.

GTS má v současné době fyzická zařízení nasazena v uzlech sítě GEANT v sedmi zemích, a to ve městech Amsterdam, Bratislava, Hamburk, Ljubljana, Londýn, Milano a Praha.

2.1 Architektura

Infrastrukturu GTS tvoří hardware vyhrazený čistě pro účely služby, umístěný ve výše zmíněných uzlech propojených navzájem end-to-end optickými okruhy s kapacitou 10 Gb/s, kterou GTS nesdílí s provozní sítí. Všechny páteřní linky



Obr. 1: Páteřní topologie GTS

mezi uzly jsou tvořeny samostatnými DWDM kanály. Topologií je hyperkrychle a její aktuální podoba je na obrázku 1.

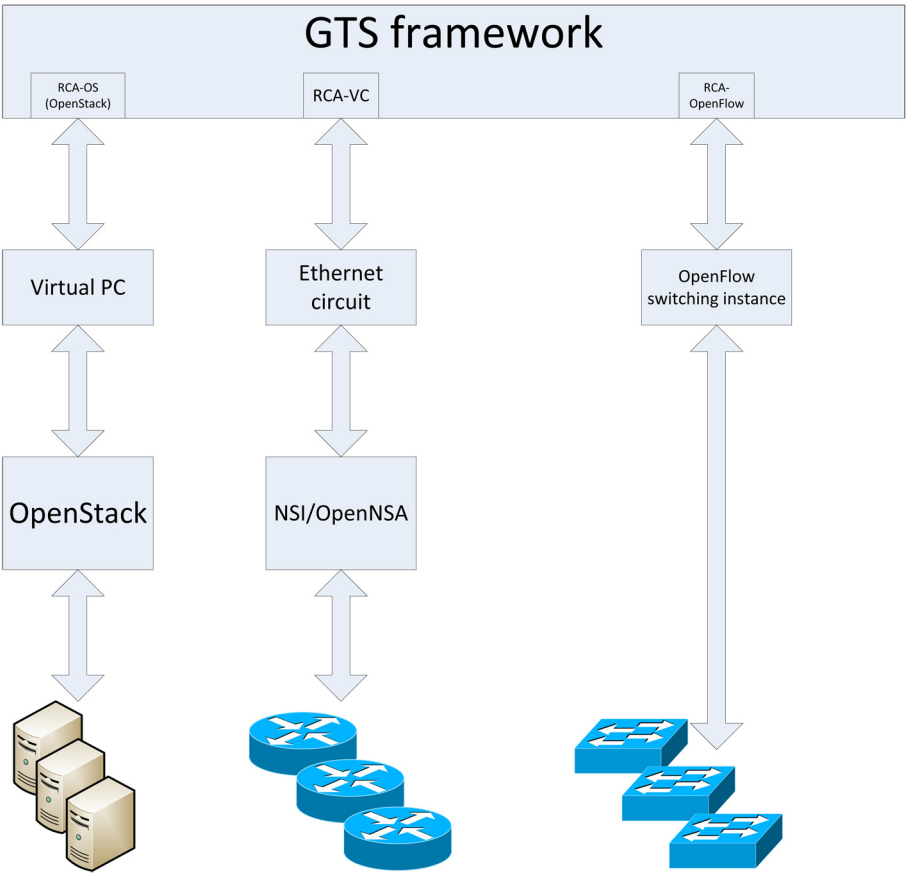
Každý uzel je shodně vybaven čtyřmi PC servery, jedním přepínačem s podporou protokolu OpenFlow [3], páteřním směrovačem a přepínačem pro kontrolní síť. V jednom z uzlů jsou navíc umístěny řídící servery.

Služba jako celek pak využívá různých nástrojů ke správě jednotlivých druhů prostředků a GTS Framework pak komunikuje s těmito nástroji a spravuje virtualizované infrastruktury poskytované uživatelům.

Framework v současné době umožnuje spravovat následující prostředky:

- Virtuální PC
- Ethernetové okruhy
- OpenFlow přepínače

Jako platforma pro správu virtuálních PC je používán „cloudový“ systém OpenStack. Jeho nasazení v GTS je svým způsobem nestandardní, jelikož Framework sám sice využívá OpenStack API, ale zároveň ho i obchází a při propojo-



Obr. 2: GTS architektura

vání virtuálních PC a ethernetových okruhů „sahá“ přímo do lokální konfigurace výpočetního serveru, kde je dané virtuální PC spuštěné. Tento přístup umožňuje zajistit prakticky fyzické oddělení více virtualizovaných infrastruktur spuštěných nad GTS současně.

Pro správu eternetových okruhů byl vybrán nástroj OpenNSA, který je implementací protokolu NSI [4, 5]. Ten umožňuje provozovat okruhy napříč více doménami a nezávisle na použitém síťovém HW.

Pro správu openflow přepínacích instancí bylo rozhodnuto nevyužívat žádnou virtualizační vrstvu mezi fyzickým přepínačem a uživatelským kontrolérem. Na místo toho byla přímo v „GTS Framework backendu“ (tzv. Resource Control Agent – RCA) pro správu openflow prostředků implementována správa openflow

instancí, které nabízí operační systém Comware přepínačů HP. Takto je možné na jednom fyzickém přepínači vytvořit více nezávislých logických přepínačů, a to až do vyčerpání volných portů. Takové přepínací instance mohou naplno využívat přepínací HW, a jediný omezením tak je sdílená paměť pro pravidla (flow specs).

Systém jednotlivých agentů pro správu dílčích druhů prostředků přináší možnost dodat do frameworku podporu pro libovolný druh prostředků, případně i pro různé nástroje zprostředkovávající stejné prostředky (např. zaměnit OpenStack za VMware znamená změny jen v rozhraní RCA).

2.2 Podporované funkce a ovládání

Nad každým druhem prostředku Framework provádí pět základních kontrolních operací:

- Reserve() – požadavek k rezervaci prostředku, rezervovaný prostředek znamená, že konkrétní fyzický zdroj je v daný čas vyhrazen pro konkrétního uživatele
- Activate() – v rámci rezervace se tímto pokynem prostředek uvede do funkčního stavu (VM nabootuje)
- Query() – dotaz na stav konkrétního prostředku
- Deactivate() – deaktivace prostředku se současným zachováním rezervace zdrojů
- Release() – uvolnění zdrojů

U různých druhů prostředků je možné specifikovat i další operace navíc k těmto pěti, které jsou povinné.

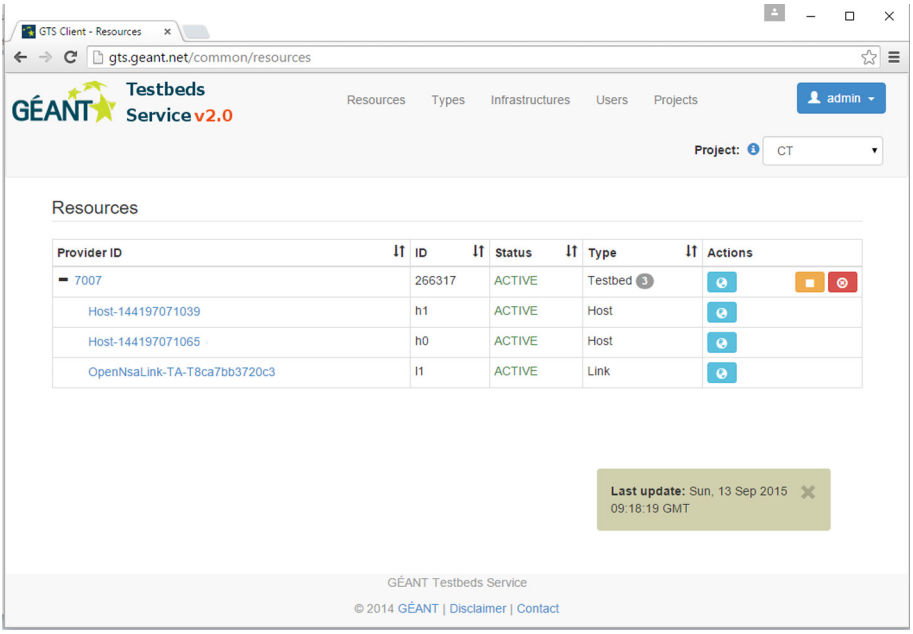
Příkladem takové operace může být například povel k vytvoření „snapshotu“ virtuálního PC. Taková operace však nedává smysl u ethernetového okruhu.

Po založení projektu v systému si uživatel rezervuje zdroje pro svůj testbed zavoláním reserve() nad jím popsanou topologií. Takový popis musí předem vytvořit a vložit do projektu. Popis se provádí prostřednictvím tzv. DSL (Domain Specific Language).

Níže je uveden příklad DSL, který popisuje testbed se dvěma virtuálními PC propojenými okruhem, kde jeden z nich má ještě druhý ethernetový port připojený pomocí druhého okruhu k externímu portu. Externí port je reprezentace fyzického ethernetového portu na páteřním routeru, jehož pomocí je možné testbed propojit s jakoukoliv externí infrastrukturou.

```
External {
        host {
                id = "h1"
                port { id = "eth1" }
                port { id = "eth2" }
        }

        host {
                id = "h2"
                port { id = "eth1" }
        }

        link {
                id = "l1"
                port { id = "src" }
                port { id = "dst" }
        }

        link {
                id = "l2"
                port { id = "src" }
                port { id = "dst" }
}

        ExternalDomain {
                id = "ex1"
                location = "prg"
                port { id = "ep1" }
        }

        adjacency h1.eth1, l1.src
        adjacency h2.eth1, l1.dst
        adjacency h1.eth2, l2.src
        adjacency l2.dst, ex1.ep1
}
```

Obr. 3: Příklad DSL popisující topologii testbedu



Obr. 4: GTS framework GUI

3 Závěr

Stručně byl představen software pro řízení a správu síťové infrastruktury pro experimenty. Software přináší inovativní přístup ke koncepci SDN a nezaměřuje se jen na řízení síťového provozu. GTS Framework ve své současné verzi umožňuje správu výpočetních prostředků, sítových prvků a uložišť. V rámci dalšího vývoje se počítá s rozšířením o podporu síťových prvků nižších vrstev, zejména optických přenosových systému a bezdrátových technologií. Systém pamatuje i na více-doménová prostředí. Do konce roku 2015 je plánováno vydání nové verze obohacené o podporu propojování infrastruktur řízených GTS frameworkem a tím umožnit vytváření experimentálních prostředí ze zdrojů spadajících do infrastruktur různých provozovatelů.

Literatura

[1] The GEANT project, 2015, http://www.geant.org/Pages/Home.aspx

[2] The Geant Testbed Service, 2015, http://services.geant.net/gts/Pages/Home.aspx

[3] The OpenFLow switch specification v1.4.0, 2015,
https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

[4] Network Service Framework v2.0,
https://www.ogf.org/documents/GFD.213.pdf

[5] NSI Connection Service v2.0, http://www.ogf.org/documents/GFD.212.pdf

Jak vybrat správný firewall

Martin Šimek

E-mail: simekm@civ.zcu.cz

Abstrakt

Pozice administrátora počítačové sítě s sebou v dnešní době přináší nové výzvy. Už nestačí pouze nastavovat svěřená zařízení podle požadavků, ale je vyžadován celkový přehled nejen v oblasti síťových prvků a managementu. Oblasti, kterými se musí zabývat, a podle jejichž indikátorů se má rozhodovat, spadají i do mezinárodní politiky nebo světové ekonomiky. Z postradatelného administrátora počítačové sítě se tak postupně stává nenahraditelný architekt sítě spojující zdánlivě nesouvisející kousky skládačky do funkčního celku. Každý jeho krok neodvratně směřuje k jedinému cíli: vytvořit perfektně fungující síťové prostředí a naplnit tak očekávání svých nadřízených. Pojďme společně prožít jeden takový složitý proces výběru nového síťového zařízení a zastavme se při důležitých bodech rozhodovacího procesu.

Úvod

Firewally se staly v posledních desetiletích ochránci nejen celých podnikových sítí, ale jejich funkce zasahují i do páteřních sítí a prostředí virtualizovaných serverů. Přitom prošly poměrně zásadním vývojem, kdy se jejich funkce postupně rozšiřovaly spolu s růstem výpočetní kapacity tradičních hardwarových platforem. Nejstarším typem firewallu jsou tzv. paketové filtry (Packet Filter). Jejich konfigurace obsahuje sadu pravidel, která specifikují pouze adresu nebo rozsah adres, protokol (např., TCP, UDP, ...) a port nebo rozsah portů. Na základě těchto pravidel pak firewall vyhodnotí každý příchozí paket a buď ho propustí, nebo zamítne. Paketové filtry jsou ze své podstaty bezestavové a to je v mnoha případech nasazení omezující. V dnešní době se už prakticky nepoužívají. Stavový firewall (Stateful Firewall) vznikl z výše zmíněného paketového filtru, doznal však významných rozšíření a vylepšení. Stavový firewall je ve srovnání s paketovým filtrem schopen měnit a zapamatovávat si vnitřní stavy jednotlivých spojení a realizovat tak vnitřní konečný automat. Rozhodnutí o možnosti průchodu paketu firewallem je závislé nejen na paketu samotném, ale také na historii zachycené ve stavových proměnných firewallu. Vznikají tak dynamická pravidla pro práci s pakety. Pro administrátora odpadá nutnost definovat

pravidla pro oba směry komunikace. Výhodou stavového firewallu je minimální procesorová náročnost pro pakety patřící k nějakému již vytvořenému a zapamatovaném spojení, protože největší část výpočetní kapacity je spotřebována při sestavování spojení.

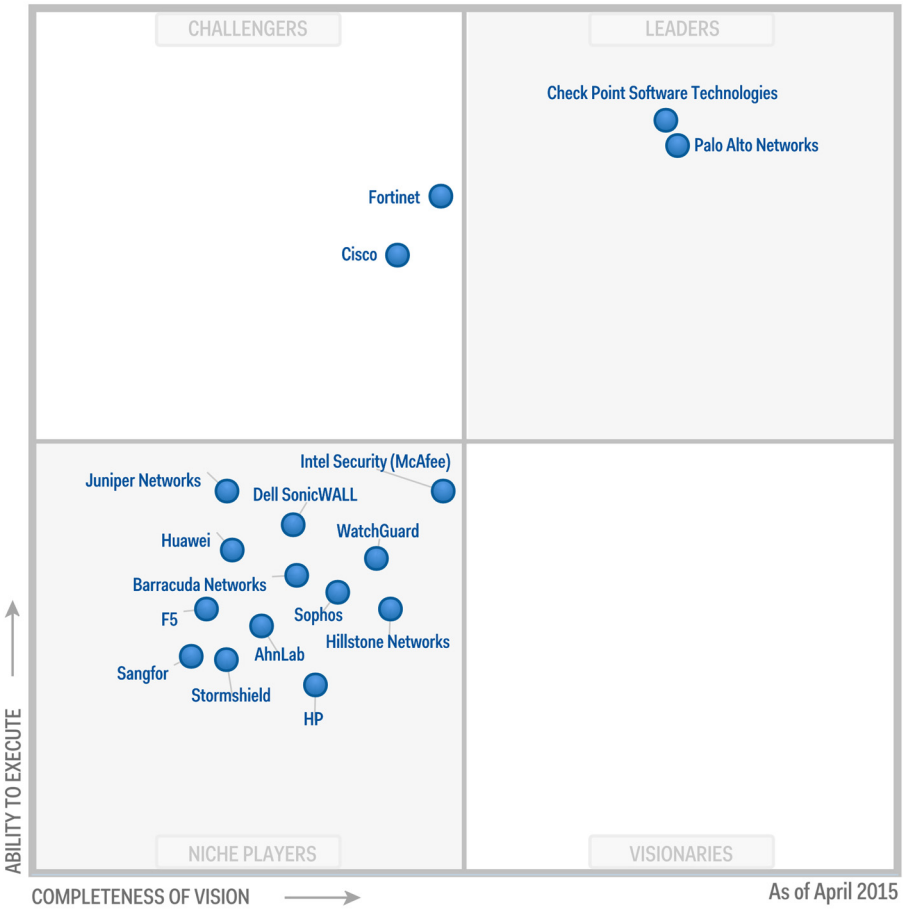
Tyto tradiční firewally se svými kontrolami založenými na číslech portů a protokolech mají omezenou možnost náhledu dovnitř moderní komunikace. Když v roce 1999 Darcy DiNucci ve svém článku „Fragmented future“ [1] definovala termín Web 2.0, pojmenovala tím zcela novou oblast, kde veškerá komunikace, útoky nevyjímaje, probíhá přes aplikační vrstvu. Stavové firewally ale nedokáží rozlišit, jaké aplikace jsou zabaleny v HTTP nebo HTTPS protokolech. A právě útočníci se stali přeborníky ve využívání aplikační vrstvy pro cílené útoky, které projdou přes standardní stavové firewally. To vše si vynutilo vývoj firewallů nové generace.

Co je to NGFW?

Abychom pochopili vnitřní logické členění firewallů nové generace, musíme si ještě objasnit pojmy detekce hrozeb (Intrusion Detection System, IDS) a prevence hrozeb (Intrusion Prevention System, IPS). V obou případech se jedná o systémy, které analyzují celé pakety, jak hlavičku, tak i uživatelská data, a hledají otisky (Signature) známých a popsaných hrozeb narušení. Zatímco systémy IDS jsou při rozpoznání otisku schopny pouze generovat události např. vygenerovat logovací záznam, systémy IPS mohou navíc zablokovat detekovaný nežádoucí datový provoz, resetovat spojení nebo blokovat veškerý provoz z nežádoucí adresy. Oba systémy tedy pracují podobně, v mnoha konkrétních implementacích se přechod od IDS k IPS provádí pouze konfigurační změnou. Aby mohli systémy IPS blokovat nežádoucí provoz, musí být zapojeny přímo v komunikační cestě. Celý systém tedy pracuje podobně jako stavový firewall, ale navíc provádí hloubkovou analýzu paketů a detekuje nebo blokuje známé hrozby.

Jedná se tedy o novou a ekonomicky velmi zajímavou oblast, která má velký potenciál růstu. Ale jak toto nové zařízení pojmenovat, aby bylo na první pohled zřejmé, že se jedná o něco jiného, než obyčejný stavový firewall? V roce 2009 analytici John Pescatore a Greg Young ze společnosti Gartner definovali pojem Next Generation Firewall (NGFW) [2] a zároveň zvýšili váhu hodnocení vlastností firewallů nové generace v oblasti Enterprise Network Firewall [3] při vytváření každoročních vysoce ceněných studií stejné společnosti označovaných jako Magické kvadranty (Magic Quadrant) viz obr. 1.

Tím se rozpoutaly doslova závody mezi výrobci tradičních stavových firewallů při implementaci vlastností firewallů nové generace do stávajících řešení klasických stavových firewallů. Zařízení typu firewall nové generace musí mít podle analytiků společnosti Gartner následující vlastnosti:



Obr. 1: Gartner Magic Quadrant for Enterprise Network Firewall, zdroj [2]

- Při zapojení místo stávajícího „drátu“ nesmí mít negativní dopad na ostatní prvky sítě – firewall musí podporovat nasazení v režimech L2 i L3 včetně jejich smysluplné kombinace.
- Podpora standardních funkcí firewallů první generace (klasických stavových firewallů) – překlad adres (NAT), stavovou kontrolu provozu, vytváření virtuálních privátních sítí (VPN) atd.
- Integrovaný IPS modul – místo integrace dalšího článku v cestě paketu je IPS přímo součástí architektury firewallů nové generace. Paket je tak zpracován paralelně v jednom průchodu firewallem, což výrazně zvyšuje výkon celku.

- Rozpoznávání aplikací a jejich detailní kontrola – definice aplikací a pravidel je oproštěna od znalosti konkrétního protokolu a čísla portu. Jednotlivé aplikace nebo skupiny aplikací je možné nad rámec definované politiky povolovat nebo zakazovat, omezovat datové toky nebo prohledávat, zda neobsahují důvěrná data.
- Integrace s informačními a ověřovacími službami mimo firewall – pravidla firewallu mohou obsahovat identifikace uživatelů místo jejich IP adres.
- Automatický update bezpečnostních zásad – online aktualizace všech bezpečnostních vlastností.
- Nahlížení do šifrovaného SSL provozu.

Spolehlivé odhady vývoje?

Společnost Gartner není samozřejmě jedinou společností zabývající se výzkumem a poradenstvím v oblasti IS/ICT technologií. Nicméně výstupy jejích studií jsou velmi názorné. Další vysoce ceněné studie v oblasti IS/ICT pocházejí od společnosti Frost & Sullivan [4]. Není náhoda, že obě společnosti působí v USA, kde je trh ICT vysoce náchylný na jakékoli zprávy z důvěryhodných zdrojů. Obě společnosti velmi doporučují přiklonit se při zvažování nákupu nového firewallu právě na stranu firewallů nové generace, protože právě do této oblasti investovala většina výrobců bezpečnostních zařízení nemalé částky. V odhadech vývoje trhu firewallů se lze orientovat podle společných studií společností Infiniti Research a TechNavio Insight [5]. Pokud si spojíme veškeré podklady dohromady, zjistíme, že zatímco v roce 2014 činilo nasazení firewallů nové generace něco mezi dvaceti a třiceti procenty celkového množství firewallových zařízení, lze v letech 2015–2019 očekávat zvýšení instalované základny firewallů na šedesát procent. A to už je odhad, který je vhodné brát v úvahu nejen na konci normálního cyklu obnovy bezpečnostních zařízení sítě.

Jeden je více než dva

Integrovat více funkcí do jednoho zařízení má nesporné výhody. Kontrola a zabezpečení založené přímo na aplikacích přináší finanční úspory a snížení náročnosti správy způsobené konsolidací několika bezpečnostních produktů do integrované platformy. Stejně významné je i to, že technologie lze použít nejen k monitorování, ale i vynucení shody s definovanými bezpečnostními zásadami společnosti. Aplikační kontrola neznamená jen možnost identifikovat tisíce jednotlivých aplikací a nastavit pravidla, ale také vybrat, které aplikace jsou povolené, za jakých okolností a komu. Například tak mohou být zakázány aplikace typu peer-to-peer, ale aplikace Skype může být povolena pro uživatele, kteří pro

něj mají legitimní firemní využití. Z perspektivy zabezpečení nabízejí firewally nové generace mnohem silnější schopnost filtrování a detekci hrozeb než kombinace tradičních firewallů a samostatných systémů IPS a dalších bezpečnostních kontrol, jako je filtrování URL.

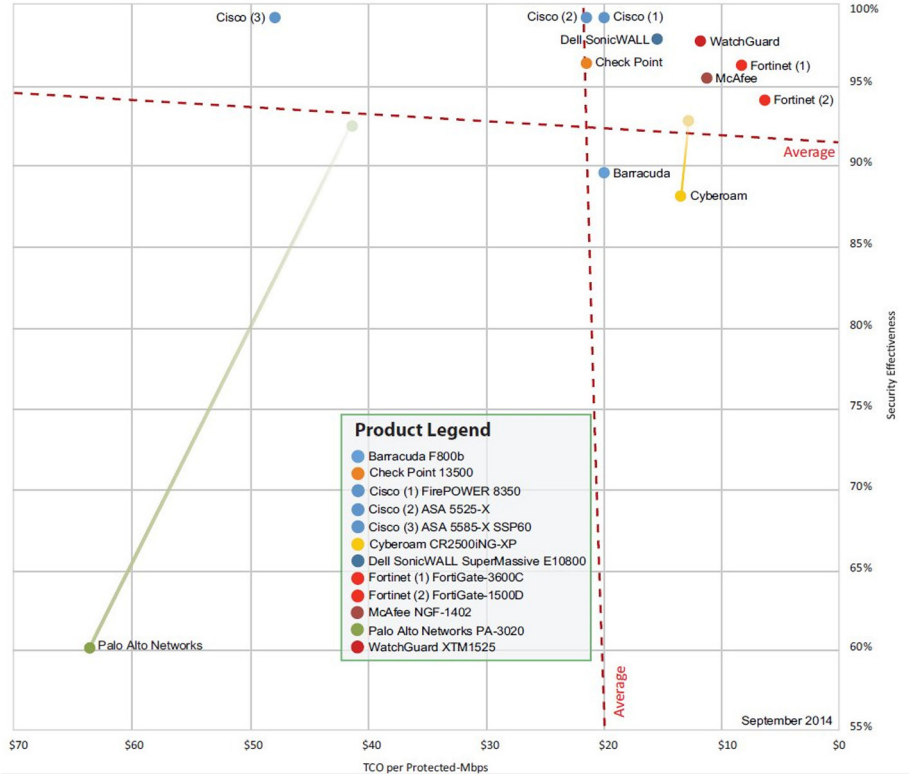
Výsledkem snažení výrobců jsou více či méně integrované systémy IPS mnohdy pouze přilepené k původním firewallům, někdy dokonce se samostatným managementem. Tento přístup integrace není zrovna ideální a mnohdy je to jeden z důvodů, proč takové řešení nekoupit.

Výkon

Firewally nové generace jsou velmi komplexní produkty. Zjištění toho, jak dobře splňuje konkrétní řešení vaše potřeby, vyžaduje zkušenosti a hodně výzkumu a testování. Všichni výrobci budou prohlašovat, že jejich řešení provádí všechny bezpečnostní funkce nejlépe. Ale ve skutečnosti firewall nové generace vyžaduje důmyslný hardwarový a softwarový návrh, který ještě před několika lety neexistoval. Nejjednodušší je nechat si vysvětlit od vybraných dodavatelů principy návrhu konkrétního řešení, architekturu softwaru a hardwaru a jak je dosaženo požadovaného zpracování a inspekce paketů, korelace a analýzy výsledků. Konkrétně lze položit dotazy týkající se následujících oblastí:

- Je paket komplexně zpracováván opravdu v jednom průchodu tak, aby mohli být získané informace využity dalšími bezpečnostními nástroji v produktu?
- Probíhá stavová inspekce paketu na firewallu, kde lze efektivně odfiltrovat nežádoucí přenosy a poskytovat kontext pro modul IPS a další nástroje?
- Je integrace firewallu a IPS opravdu bezešvá, nebo se jedná o samostatná řešení zabalená dohromady?
- Jedná se o produkt, který pracuje na standardním hardware nebo používá vyladěné účelově vyrobené vyhrazené zařízení?
- Jedná se o nový produkt firewallu nové generace nebo je řešení vnitřně poskládáno ze samostatných celků tradičního firewallu a IPS, ke kterým je např. doděláno řízení aplikací?

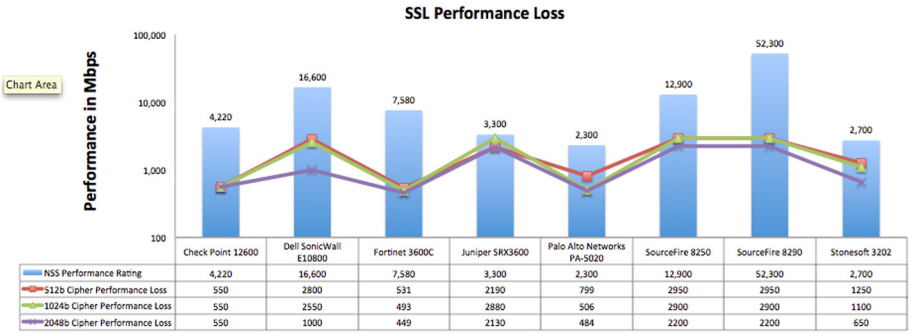
Všechny tyto funkce něco stojí. Na rozdíl od tradičních stavových firewallů je firewall nové generace úzkým místem, které může zpomalit produkční přenosy nebo zvýšit latenci spojení nad únosné meze. Počet spojení za sekundu tj. propustnost celého systému se zapnutím všech požadovaných bezpečnostních funkcí, je nutné otestovat v prostředí, které se co nejvíce blíží tomu reálnému produkčnímu. Dalším z potenciálních výkonnostních problémů, které je potřeba otestovat, je schopnost firewallu nové generace zpracovávat šifrované přenosy.



Obr. 2: Next Generation Firewall Security Value Map, zdroj [6]

Dokáže firewall šifrované spojení zachytit, dešifrovat a znovu zašifrovat, a pokud ano, jaké jsou dopady na jeho výkon? Jedná se o operace, které jsou náročné na výkon, a dá se spolehnout na poměrně značnou degradaci propustnosti celého firewallu. Naštěstí existují nezávislé společnosti, které se specializují na testování konkrétních produktů. Výsledkem jejich reálných testů jsou analýzy, které lze zohlednit při výběru konkrétního řešení. Jednou z nejrespektovanějších testovacích společností je NSS Labs. Výsledky jejích testů jsou přehledně zpracovány ve formě tzv. Value Map [6], čím výhodnější je konkrétní řešení, tím je v matici výše a více vpravo viz obr. 2.

Z výsledků analýz vyplývá, že většina výrobců dokonce podceňuje své produkty a dosažená reálná propustnost je mírně vyšší než udávaná. Co se týká propadu propustnosti firewallu při potřebě nahlížení do šifrovaných spojení, ten se podle testů stejné společnosti pohybuje okolo 70 procent [7] viz obr. 3.



Obr. 3: SSL Performance Impacts on Bandwidth, zdroj [7]

Certifikace řešení

Reálná propustnost firewallu je samozřejmě důležitá, ale řešení jako celek je vhodné certifikovat. Na trhu s firewally působí několik společností poskytujících certifikaci, z nichž nejznámější je ICSA Labs, což je nezávislá divize společnosti Verizon. Při jejích testech se postupuje podle předem daného neměnného scénáře pro měření shody výrobku, jeho spolehlivosti a výkonu [8]. Lze tedy předpokládat, že získání certifikátu ICSA poskytuje určitou záruku splnění celosvětově přijatých standardů v dané oblasti.

Virtualizované firewally

Technologie firewallu a IPS se ukázaly jako všestranné, nejsou omezené pouze na hardwarová zařízení, ale jsou k dispozici i jako software speciálně navržený tak, aby jej bylo možné integrovat do virtuálních desktopových a serverových řešení, založených na platformách VMware, Hyper-V, KVM nebo XEN. Ve virtualizovaném prostředí pak administrátor získá možnost ovládat a filtrovat datové toky mezi jednotlivými virtuálními instancemi (virtuálními stroji v různých zónách) a zabezpečit komunikaci zvenčí fyzické sítě. Pokud se k této funkcionalitě přidají vlastnosti firewallů nové generace, usnadní se správa pravidel a zároveň se přidá možnost definice nových vlastností filtrace provozu. Zdá se, že virtualizované firewally v současnosti představují jenom minimum celkového trhu firewallů. Hlavním problémem jsou totiž spory při rozdělování kompetencí správy virtualizovaných firewallů mezi týmem provozu sítě a týmem provozu serverů. O tom, že virtualizovaný firewall nové generace je zcela nový trend svědčí i to, že společnost NSS Labs prozatím neprovedla žádný srovnávací test, přestože má již zpracovanou metodiku testování [9]. Snad se dočkáme v letošním roce.

Softwarově definované firewally

V SDN sítích je logika řízení politiky zcela oddělena od hardwarových nebo softwarových přepínačů. Komunikace mezi centrálním prvkem (řadičem) a přepínačem probíhá pomocí standardizovaných protokolů např. OpenFlow. Toto oddělení rolí umožňuje velkou flexibilitu řízení sítě resp. provozu v síti. Lze jednotně nastavovat bezpečnostní zásady, směrovat toky v síti a blokovat nechtěný datový tok už v místech jeho vzniku. Potíže však přináší nemožnost rozpoznat stav přenosu přímo na přepínačích. Tuto logiku může obsahovat až centrální řadič. A právě dlouhá latence při výměně zpráv mezi přepínači a centrálním řadičem může mít negativní dopad na propustnost a latenci softwarově definovaného firewallu. Řešením tohoto problému je vytvoření další logiky na straně centrálního řadiče ve formě stavové tabulky, kde budou uložené pravidelně aktualizované informace o každém datovém toku. Aby rozhodnutí o blokování nebo povolení konkrétního provozu nezpůsobilo přílišné zdržení, používá se převážně metoda „first deny last allow“, kdy je vnitřně upřednostněno zpracování a odeslání pravidla nakazujícího provoz. Vytvoření stavové tabulky je hlavním tématem řady odborných publikací [10, 11] i praktických implementací [12, 13] softwarově definovaných firewallů. Většina výrobců se je v přístupu k SDN jednotná a nabízí buď tradiční firewall nové generace nebo virtualizovaný firewall, v obou případech s možností ovládat jej přes REST API.

Závěr

Vybrat vhodný firewall pro konkrétní nasazení není zrovna lehký úkol. Při rozhodovacím procesu je potřeba brát v úvahu mnoho faktorů. V první řadě je potřeba se oprostit od vazby na dodavatele stávající síťové infrastruktury. Firewall bude pracovat zcela samostatně, není nutné používat stejný přístup jako na ostatní síťové prvky. Nová generace firewallů je natolik propracovaná, že nemá smysl se při výběru omezovat pouze na stavové firewally. Tabulkové hodnoty propustnosti vždy neodpovídají reálným hodnotám a při zapnutí dalších funkcí jako IPS, kontrola aplikací nebo filtrování URL výrazně klesají. Při implementaci je potřeba zvážit, které datové toky v jakém směru vyžadují tyto další bezpečnostní kontroly. Naštěstí existují srovnávací testy, které mají větší vypovídací hodnotu, než hodnoty definované výrobci. Pokud je v plánu nasazení firewallu na perimetru sítě, je důležitá zvážit také vzrůst latence. V neposlední řadě je potřeba si dát pozor na to, zda firewall používá standardní nebo účelově vyrobený hardware. V případě standardního hardware nelze překročit limity dané použitou platformou, i když tabulkové hodnoty mohou vypadat zajímavě. Ve virtualizovaném prostředí je nutné zvážit potřebu filtrace provozu mezi jednotlivými virtuálními instancemi, kde může dojít ke snížení propustnosti. Softwarově definované firewally jsou prozatím ve stádiu hledání vhodné cesty.

Literatura

[1] Darcy DiNucci, Fragmented Future, 1999, Print, Volume 53, Issue 4, Page 32. http://darcyd.com/fragmented_future.pdf

[2] John Pescatore, Greg Young, Defining the Next-Generation Firewall, Gartner RAS Core Research Note G00171540, 2009, R3210 04102010. http://www.bradreese.com/blog/palo-alto-gartner.pdf

[3] Greg Young, Defining The Next Generation Firewall Research Note: The Liner Notes, Blog Gartner. http://blogs.gartner.com/greg_young/2009/10/15/defining-the-next-generation-firewall-research-note-the-liner-notes/

[4] Frost & Sullivan, Market Engineering, Analysis of the Global Unified Threat Management (UTM) Market, Enterprise Features and Product Value Propel Market Growth, NBAF-74. http://www.fortinet.com/sites/default/files/analystreports/Global_UTM_Market_Nov.pdf

[5] Infiniti Research and TechNavio Insight, Global Next Generation Firewall (NGFW) Market 2015–2019. http://www.technavio.com/report/global-next-generation-firewall-ngfw-market-2015-2019

[6] NSS Labs, Next Generation Firewall Comparative Analysis, Security Value Map. http://www.fortinet.com/sites/default/files/whitepapers/Next-Generation-Firewall-Comparative-Analysis-SVM.pdf

[7] NSS Labs, SSL Performance Problem. https://www.nsslabs.com/sites/default/files/public-report/files/SSL%20Performance%20Problems.pdf

[8] ICSA Labs, Network Firewalls Document Library. https://www.icsalabs.com/technology-program/firewalls/network-firewalls-document-library

[9] NSS Labs, Test Methodology, Virtual Firewall. https://www.nsslabs.com/sites/default/files/public-report/files/Virtual%20Firewall%20Test%20Methodology%20v1.0.pdf

[10] Jake Collings, Jun Liu, An OpenFlow-Based Prototype of SDN-Oriented Stateful Hardware Firewalls, 2014, The 22nd IEEE International Conference on Network Protocols (ICNP 2014). http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6980422

[11] Hongxin Hu, Gail-Joon Ahn, Wonkyu Han, Ziming Zhao, Towards a Reliable SDN Firewall. https://www.usenix.org/system/files/conference/ons2014/ons2014-paper-hu_hongxin.pdf

[12] FSFW: FlowSpace Firewall. http://globalnoc.iu.edu/sdn/fsfw.html

[13] Hongxin Hu; Wonkyu Han; Gail Joon Ahn; Ziming Zhao, FLOWGUARD: Building robust firewalls for software-defined networks, HotSDN 2014 – Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking.
http://www.public.asu.edu/∼zzhao30/publication/HongxinHotSDN2014.pdf

4G

Libor Dostálek

E-mail: dostalek@prf.jcu.cz

Od počátku používání telefonu až po digitální síť GSM (2. generace mobilních sítí) se na dobu hovoru sestavoval okruh mezi volajícím a volaným (3. generace mobilních sítí tzv. UMTS sítě se alespoň u nás prakticky k sestavování hovorů nepoužívaly). Okruh byl nejprve sestavován manuálně operátory telefonních ústředen, později byl sestavován mechanicky a nakonec elektronicky.

Zpočátku se hlasový signál moduloval analogově do sestaveného okruhu, později byl hlasový signál vzorkován a vzorky okruhem přenášeny datovými pakety. Pokud bylo třeba v těchto sítích přenášet data, pak se modulovala/demodulovala pomocí zařízení nazývaného modem do sestaveného okruhu, který byl prvotně určen pro přenos hlasu.

S příchodem 4. generace mobilních sítí se situace obrátila. Od sestavování okruhů se upustilo a vše se přenáší pomocí datových paketů rodiny protokolů TCP/IP. Zatímco 3. generace se víceméně využívala pro připojení k Internetu a pro hlasové služby se souběžně využíval GSM. 4. generace přináší již takové přenosové rychlosti, že i hlasové služby je možné realizovat pomocí paketů protokolů TCP/IP, tj. bez využití přenosových okruhů.

V rodině protokolů TCP/IP se nejenom hlasové služby, ale obecně multimediální služby, implementují pomocí skupiny protokolů, z nichž nejtypičtějšími reprezentanty jsou SIP a RTP. Hlasové (multimediální) služby na internetu realizované mj. těmito protokoly se označují jako *Voice over IP* nebo zkráceně VoIP.

Čtvrtá generace mobilních sítí používá pro poslední míli mezi mobilem a sítí standard LTE, který zprostředkovává spojení do internetu. Hlasové (multimediální) služby jsou pak implementovány rovněž za využití téže skupiny protokolů TCP/IP (SIP, RTP apod.). Tuto komunikaci pak označujeme jako *Voice over LTE* nebo zkráceně VoLTE.

VoLTE je tedy implantováno stejnými protokoly jako VoIP. Přesto se bráním tvrzením typu: „VoLTE je v podstatě zvláštním případem VoIP“. Obojí sice vychází ze standardů RFC, ale VoLTE tyto standardy často rozšiřuje o svá specifika nebo některé věci řeší standardy 3GPP. Rozdíl je také z pohledu bezpečnosti. Zatímco uživatelé VoIP jsou často obecnými uživateli internetu, kteří

se zpravidla autentizují jménem a heslem, tak uživatelé LTE jsou v rámci konkrétního poskytovatele chráněnou skupinou a pro autentizaci zpravidla používají čipovou kartu USIM/ISIM.

Místo „hlasové služby“ je dnes moderní používat termín „multimediální služby“. Je tím míněno, že kromě klasických „hovoru“ je běžné implementovat např. „video hovory“, „videokonference“ apod. Je třeba ale poznamenat, že toto je jen malý výsek multimediálních možností internetu.

I pro znalce TCP/IP bude v následujícím textu novým termínem pojem „referenční bod“. Je to obdoba dříve používaného termínu rozhraní (*interface*). Jako referenční bod si můžeme představit sondu, kterou do síťové komunikace vložil muž uprostřed sítě (*man in the middle*), aby komunikaci sledoval. Na rozdíl od síťových protokolů, ale referenční bod může popisovat nejenom sítové protokoly, ale také aplikační data, protože z hlediska síťových protokolů, to co je nad aplikační vrstvou, už není síťový protokol, ale aplikace.

Celá problematika je pokryta několika systémy norem, které jsou vesměs veřejně dostupné. Základem jsou normy 3GPP (*The 3rd Generation Partnership Project*), zejména roaming je specifikován v normách GSMA (*GSM Association*) a vše co se týká TCP/IP lze pochopitelně nalézt v RFC (*Requests for Comments*). Normy ITU (*International Telecommunication Union*), kterými se standardizovaly telekomunikace v 19. a 20. století ustupují do pozadí.

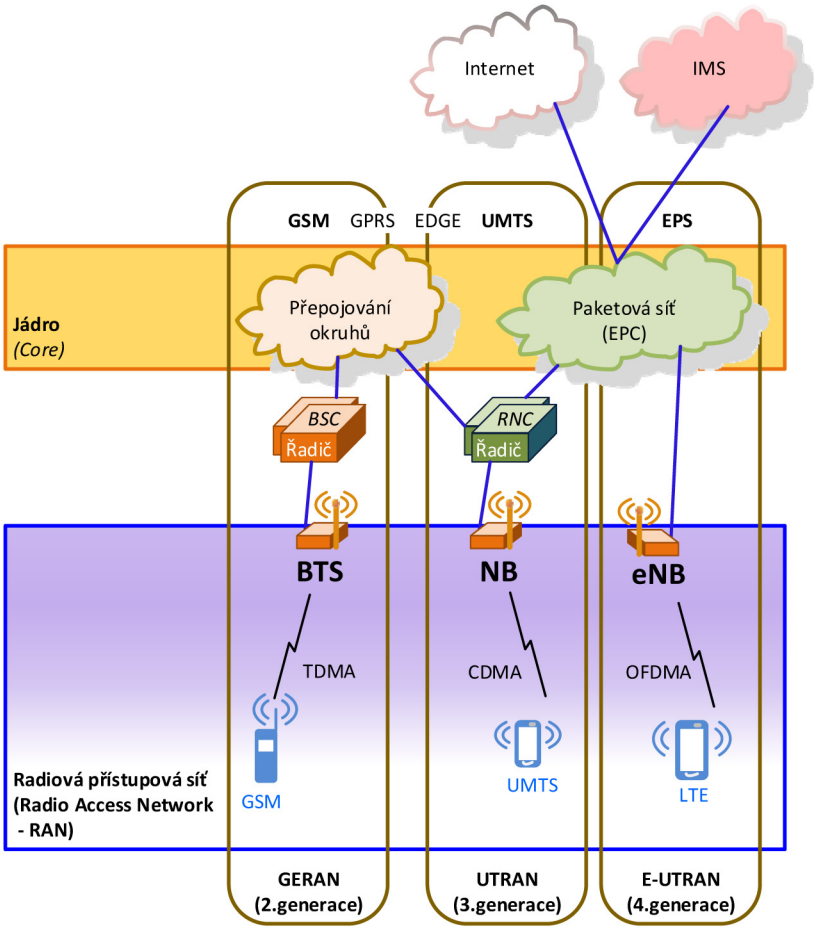
1 Buňky a generace mobilních sítí

Mobilní síť se snaží maximálně pokrýt obsluhované území. Pro pokrytí území využívá základnové stanice. V GSM sítích (2. generace) se základnová stanice nazývá BTS (*Base Transceiver Station*). V UMTS sítích (3. generace) se nazývá NB (*Node B*) v LTE (4. generace) se nazývá eNB (*evolved Node B*).

Základnová stanice pokrývá území, které se nazývá buňka (*cell*). Ze základnové stanice vede připojení směrem do jádra sítě (*core*) pomocí kabelů nebo mikrovlnného spoje. Zatímco v GSM a UMTS sítích byly skupiny základnových stanic vždy řízeny řadičem (*controller*), tak LTE již řadiče nepoužívá a základnové stanice (eNB) jsou propojeny přímo do jádra sítě, které se nazývá EPC (*Evolved Packet Core*).

Přesto i LTE má buňky v území logicky organizováno do vyšších územních celků nazývaných *Tracking Area*. LTE síť totiž sleduje lokalizaci mobilů pohybujících se sítí v nečinném stavu nikoliv po jednotlivých buňkách, ale právě po těchto vyšších územních celcích.

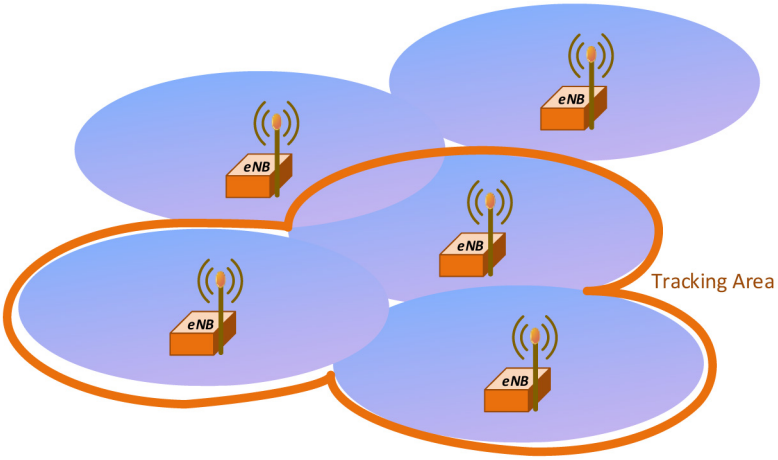
Nyní již zapomeneme na starší generace mobilních sítí (GSM a UMTS) a budeme doufat, že budou co nejdříve vypnuty, aby jejich frekvence mohly být dále využity. Jak?



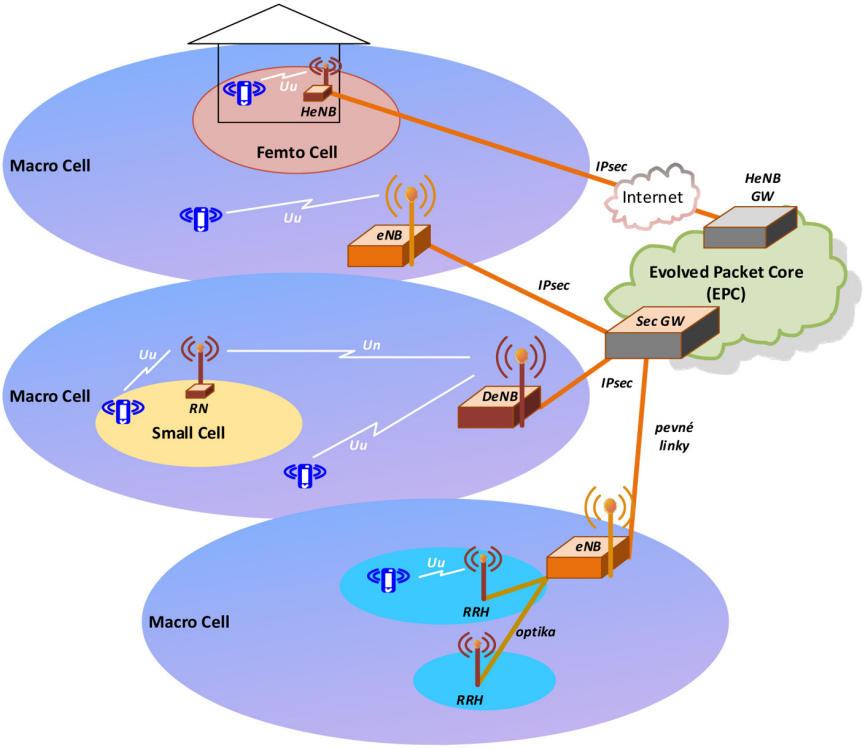
Obr. 1: Základnové stanice a jednotlivé buňky

Na obzoru je totiž další standard *LTE Advanced*. Rozdíl mezi LTE a *LTE Advanced* je v tom, že *LTE Advanced* využívá více frekvencí současně. Mobil má „více antén“. Maximální přenosovou rychlost pak můžeme teoreticky stanovit až jako součet přenosových rychlostí v jednotlivých pásmech (není to zcela přesné, ale zato snadno pochopitelné).

Buňky mohou být různé velikosti. Hovoříme pak o různých buňkách od makro-buněk (*macro cell*) rozlehlých až 35 km až po mikro-buňky (*micro cell*) do 200 m. Velice zajímavé jsou ještě menší buňky, které slouží k dokrývání území.



Obr. 2: Generace mobilních technologií



Obr. 3: Velké, malé a nejmenší buňky LTE

Makro-buňka sice může teoreticky pokrýt až 35 km, ale v tomto území s největší pravděpodobností díky členitému terénu vznikne řada hluchý míst. Otázka je dokrýt toto území. Pro dokrytí máme v LTE několik možností:

- Využít tzv. *Femtocell*, tj. buněk o velikost maximálně okolo 10 m.
- Využití vykrývací základnové stanice (*Relay Node* – RN).
- Využití *Remote Radio Head* (RRH).

Asi nejzajímavějším řešením je *Femtocell*, který je obsluhován nízko výkonovou základnovou stanicí HeNB (*Home* eNB). HeNB je často v soukromém držení uživatele. HeNB je propojení zpravidla přes Internet s jádrem sítě. Propojení je zabezpečeno IPsec tunelem, který na straně jádra je zakončen v entitě *HeNB Gateway*, za kterou již je jádro sítě (EPC). *HeNB Gateway* zajišťuje též bezpečnostní oddělení jádra sítě od Internetu.

HeNB je tedy ideálním řešením např. pro chalupy, kde není pokrytí mobilní sítí, ale kde je dobré připojení k Internetu. Pomocí HeNB je též možné vykrývat různé suterénní místnosti, velíny atp.

Další variantou je Vykrývací základnové stanice (RN), která je obdobou televizních vykrývacích vysílačů pro vykrytí v členitém terénu. Základnová stanice s RN komunikuje pomocí referenčního bodu (rozhraní) Un. RN pak již s mobily ve své malé buňce komunikuje přes standardní referenční bod LTE sítí, kterým je Uu. Základnová stanice musí podporovat nový referenční bod Un. Takové základnové stanice se nazývají DeNB (*Donor eNB*).

Jinou možností je *Remote Radio Head* (RRH). Klasickou představou eNB je domeček se stožárem. V domečku je veškerá elektronika včetně vysílačů. Z domečku pak vedou koaxiální kabely do antén na stožáru. V současné době se využívají tzv. distribuované eNB, kdy v domečku je síťová elektronika a z ní na stožár vede optický kabel do vlastních vysílačů/přijímačů, tzv. *Remote Radio Head* (RRH). Je to obdoba systému v letadle nebo na lodi, kdy v kokpitu je ovládání radia, které je umístěno jinde. Jestliže z domečku už vede optika, tak některé RRH mohou být umístěny o něco dále, na druhé budově, za rohem apod.

Základnové stanice (eNB) jsou s jádrem sítě propojeny pevnými linkami zabezpečenými IPsec tunelem. Na straně jádra sítě je pak IPsec tunel zakončen na entitě *Security Gateway*, která se specializuje právě na ukončování IPsec tunelů.

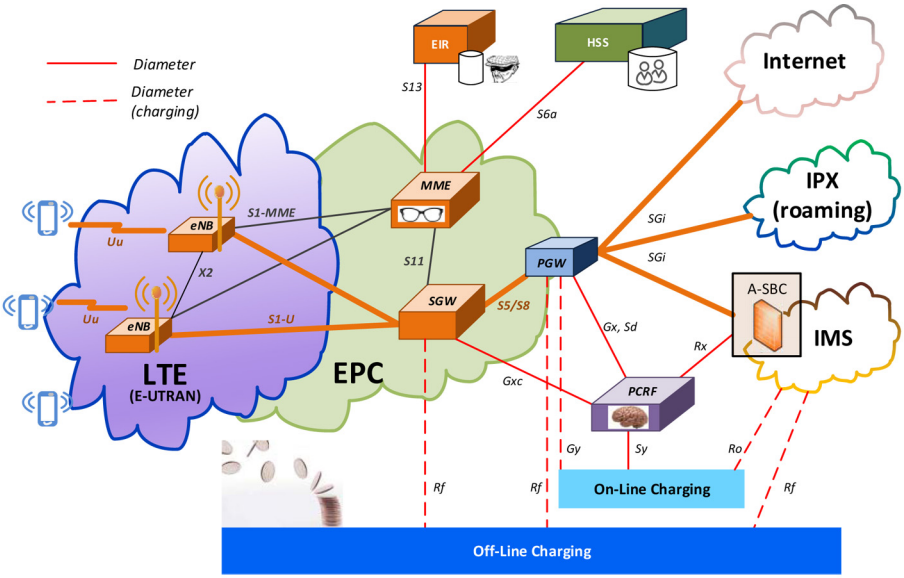
Z hlediska zabezpečení cesty z mobilu do jádra sítě se cesta skládá ze dvou částí:

- Mobil – základnová stanice, kde LTE využívá pro generování kryptografického materiálu mechanismus AKA se sdíleným tajemstvím, které je sdíleno mezi čipovou kartou USIM a domovským účastnickým serverem (HSS), který bude vysvětlen dále v textu.
- Základnová stanice – jádro sítě, kde je využit tunel IPsec.

Uprostřed základnové stanice je tedy komunikace nezabezpečena, pokud se nepoužije jiný mechanismus, což využívá např. VoLTE.

2 EPC

LTE síť je síť poslední míle (mezi mobilem a základnovou stanicí). Síť operátora je v podstatě stavebnicí, která se skládá ze systému LTE a systému EPS (*Evolved Packet System*), který byl zaveden už s 3G sítěmi. EPS řídí přístupovou síť. Tato síť má jádro (*core*), které koordinuje komunikaci – tzv. *Evolved Packet Core* (EPC). LTE s EPC uživateli poskytuje připojení na IP vrstvě. Tj. např. připojení do Internetu. Později si ukážeme, že pro zajištění „telefonování" budeme potřebovat ještě další jádro, které se nazývá IMS. IMS je ale jádro na aplikační vrstvě, které může provozovat jak týž operátor, který provozuje EPC, tak i jej může provozovat jiný (na operátoru EPC nezávislý) operátor.



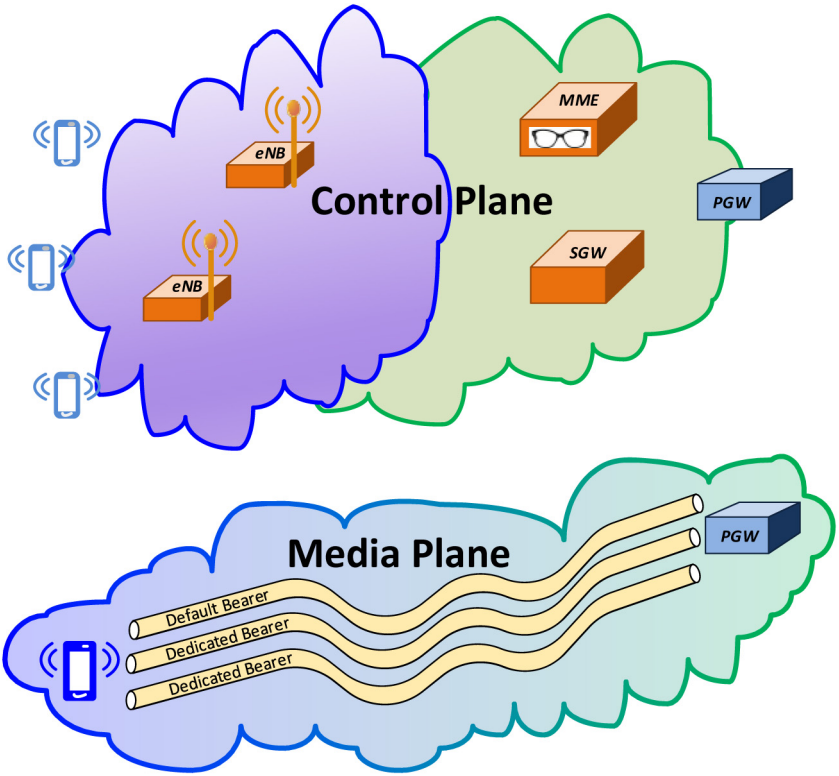
Obr. 4: Evolved Packet Core (EPC) a jeho struktura

Nyní se vraťme k EPC. Budeme-li abstrahovat od entit sloužících pro 3G sítě a od *Security Gateway* a HeNB GW z obr. 1, pak EPC obsahuje tři typy entit:

- MME (*Mobility Management Entity*) je klíčovou řídící entitou EPC. Zjišťuje ověření uživatele při přihlášení do sítě, sledování pohybu nečinných uživatelů, určení SGW přes kterou poběží uživatelův datový tok atd.

- SGW (*Serving Gateway*) je zodpovědná správu uživatelských datových toků, tzv. *data bearers*. Přes tuto entitu budou procházet uživatelská data.

- PGW (PDN GW, *Public data network Gateway*) je bránou uživatelových datových paketů do externích sítí. Externí sítí může být Internet, IMS (viz kap. 3) nebo i IPX (viz kap. 6) v případě roamingu. Uživatelé spíše znají termín APN (*Access Point Name*) což si zjednodušeně můžeme představit jako identifikaci vnějšího rozhraní PGW.



Obr. 5: Dva pohledy na EPC

Kromě zmíněných entit, které slouží výhradně pro EPC, tak případný operátor musí mít k dispozici ještě minimálně tři další entity a pochopitelně zpoplatnění (*charging*):

- EIR. Součástí přihlašovací procedury je předání hardwarové identifikace mobilu (*Mobile station Equipment Identities* – IMEI). MME pak v rámci přihlašování zkontroluje v registru zcizených zařízení (*Equipment Identity Register* – EIR), jestli mobil není kradený.

- HSS. Podstatně důležitější entitou je Domovský účastnický server (*Home Subscriber Server* – HSS), který obsahuje údaje o účastnících sítě. U každého účastníka vedeného v HSS pod interní identitou tzv. IMSI (*International Mobile Subscriber Identity*) se vedou údaje o něm, o jím nasmlouvaných službách a pak také sdílené tajemství K, které HSS sdílí s účastníkovou čipovou kartou USIM. Sdílené tajemství K slouží pro autentizaci uživatele do sítě a pro generování kryptografického materiálu pro zabezpečení jeho komunikace s mobilní sítí.

- Neméně důležitou entitou je PCRF (*Policy and Charging Rules Function*), která v reálném čase určuje pravidla (politiky) sítě. Automaticky vytváří pravidla pro každého účastníka přihlašujícího se do sítě. Např. skrze referenční bod Sd (obr. 4) může PCRF monitorovat provoz účastníků a na základě monitorování změnit jeho síťová pravidla (politiky). V případě příchozích hovorů VoLTE (příchozích z IMS) může PCRF skrze referenční body Rx a Gxc požádá SGW o alokaci příslušného datového nosiče (*data bearer*) pro telefonní hovor.

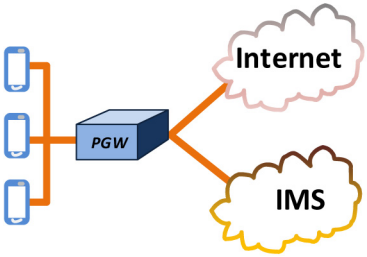
Na celou problematiku se můžeme podívat i z většího odstupu (obr. 5) a uvidíme, že se skládá ze dvou vrstev (*plane*):

- Vrstvy řízení (*Control Plane*), která zajišťuje, aby to celé fungovalo. Z hlediska účastníka sítě tato vrstva není vidět.

- Vrstvy přenosu medií (*Media Plane* nebo také *User Plane*), která zajišťuje přenos uživatelových paketů z mobilu do externích paketových sítí (Internet, IPX, IMS atp.). Tato vrstva pro každého účastníka může vytvořit jeden implicitní datový nosič (*Default Bearer*) a případně jeden nebo několik vyhrazených datových nosičů (*Dedicated Bearer*) určených pro konkrétní službu. Datové nosiče jsou různých kategorií. Např. pro přenos audia budeme požadovat nosič určité garantované šířky pásma. Pro přístup do Internetu zase garanci přenosového pásma nebudeme vyžadovat atd.

Nesmíme zapomenout ještě na zpoplatnění (*charging*). Pokud by účastníci měli jen smlouvy s operátorem a operátor jim na konci měsíce posílal fakturu, pak bychom vystačili s Off-line zpoplatňováním. Se zavedením předplatných kupónů (tj. kreditních kupónů) muselo být zavedeno i On-line zpoplatnění, aby bylo možno v reálném čase zjistit, jestli účastník má na požadovanou službu dostatečný kredit.

Z hlediska externího uživatele je výsledkem celého LTE a EPC je, že se všichni uživatelé LTE jeví, jakoby „seděli“ na lokální sítí za PGW. Podobně jako domácí uživatelé sedí za xDSL modemem (obr. 6).



Obr. 6: Pohled z externích sítí

3 IMS

Už jsme se zmínili o tom, že máme dva světy: jedním je LTE s EPC a druhým je IMS. Zatímco LTE s EPC nám umožní připojit mobil do sítě, tak IMS (*Internet Multimedia Subsystem*) nám umožní, aby to „telefonovalo“, tj. implementuje VoLTE (*Voice over LTE*).

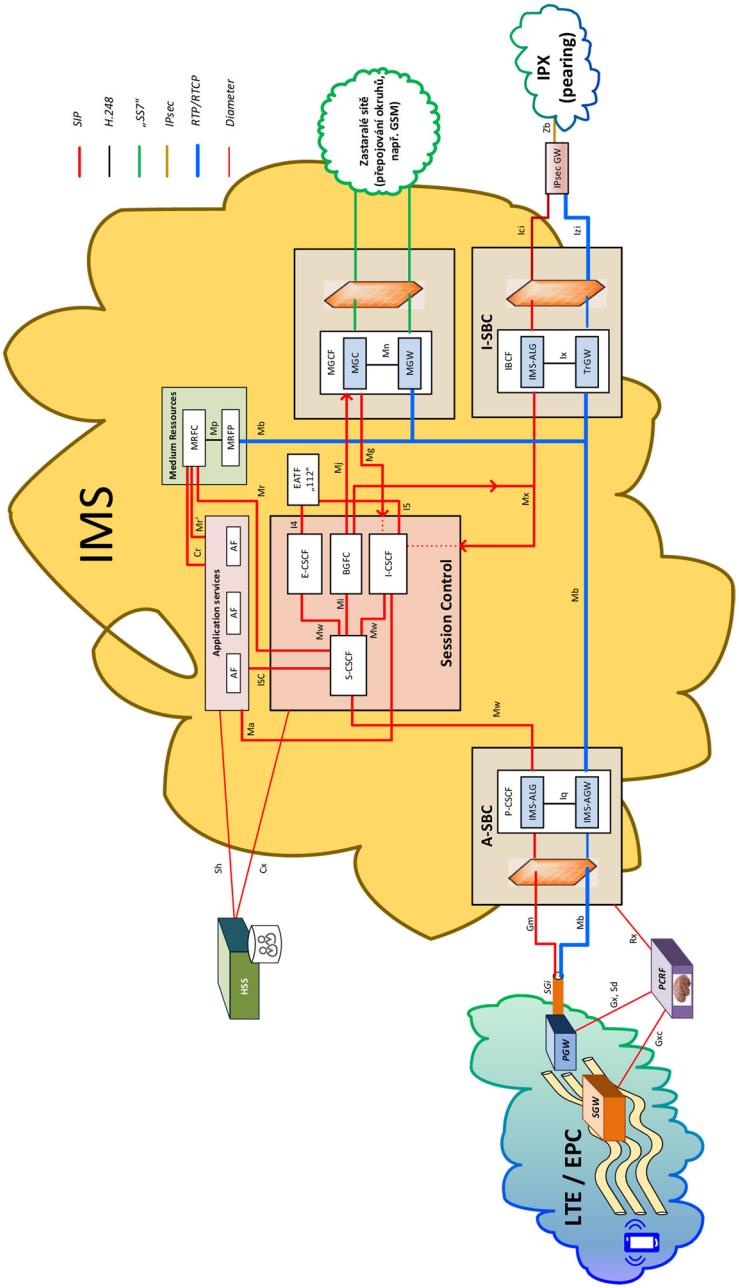
IMS (obr. 7) se skládá ze čtyř typů entit:

- SBC, které bezpečně oddělují IMS od ostatních sítí.
- *Session control*, který je zodpovědný za zpracování požadavků protokolu SIP. *Session control* obsahuje řadu proxy, které jsou popsány dále.
- *Aplikační služby*, které jsou nadstavbovými službami nad IMS. Jednotlivé aplikační funkce (AF) mohou být realizovány i servery třetích stran. Příklady aplikačních funkcí jsou Prezentační služby, konferenční servery, servery pro Rychlé zasílání zpráv (*Instant Messaging*), severy zajišťující služby „zmáčkni a mluv“, konverze zpráv do SMS a předání SMS centru atd.
- *Media resource*, který zajišťuje zvuková hlášení, případně mixáž multimediálních toků. Např. pro *Session control* může např. poskytovat audio: „Volaný účastník je dočasně nedostupný“.

Vedle těchto entit IMS vyžívá již zmíněné entity PCRF a HSS (viz též odstavec 2). HSS opět obsahuje informace o účastnících, jejich službách a též sdílené tajemství K, které sdílí s uživatelovou čipovou kartou USIM/ISIM. Na základě tohoto tajemství se účastník autentizuje vůči P-CSCF, která tento požadavek předá S-CSCF, která je vyřídí za pomocí HSS.

Jádrem IMS je entita S-CSCF, která vyřizuje požadavky, ale neobsahuje žádné bezpečností funkce. Předpokládá se, že je chráněna SBC. Pokud by útočník dostal pod svou moc S-CSCF, pak si se sítí může dělat cokoliv.

Pokud se volaný nenachází v tomto IMS, tak je požadavek předá dále na entitu BGFC, která zjistí, jestli je volaný účastníkem jiného poskytovatele IMS, pak požadavek předá entitě IBCF. Pokud je účastníkem telefonní sítě, pak požadavek předá entitě MGFC.



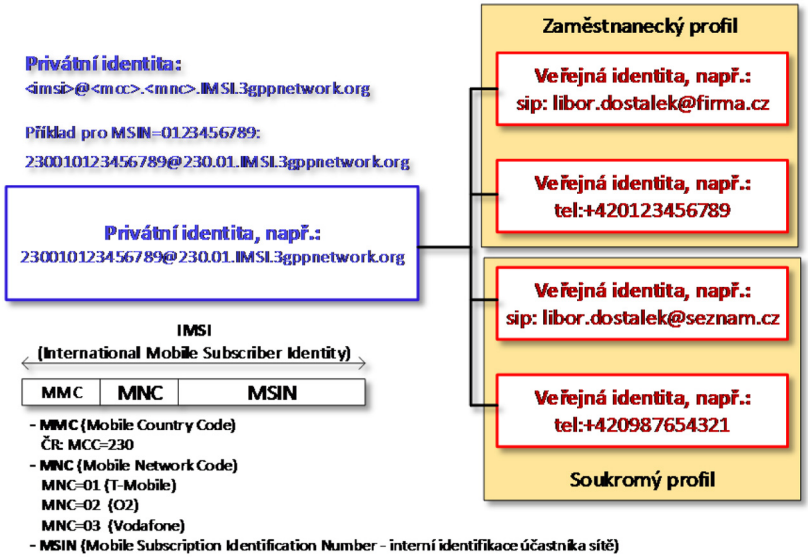
Obr. 7: IMS (Internet Multimedia Subsystem)

Na obr. 7 není znázorněno zpoplatnění a DNS. DNS je důležité pro nalezení SIP serveru domény do které se volá. Pokud se používají telefonní čísla, pak je nutné využít DNS ENUM pro převod telefonního čísla na SIP URI volaného (DNS ENUM je databáze ve které operátoři zveřejňují způsob, jak je možné se jejich účastníkům dovolat). Pomocí záznamů v DNS ENUM se rovněž řeší přenositelnost telefonních čísel. K DNS ENUM je třeba dodat, že DNS v IMS bude používat kořenové servery sítě IPX (viz kap. 6) nikoliv Internetu.

Ještě existuje DNS ENUM v Česku spravovaný NIC.CZ pro VoIP, který ale používá kořenové servery internetu (tento ENUM je internetová databáze, v níž majitelé telefonních čísel zveřejňují způsob, jak jim ostatní mohou na toto číslo volat přes internet). Jelikož IPX a Internet jsou dva nezávislé světy, tak i prostory telefonních čísel v obou světech jsou nezávislé.

4 Veřejná a privátní identita, IMPI a IMPU

Uživatel (*subscriber*) je veden v databázi operátora pod jeho interní identitou, která se označuje MSIN (*Mobile Subscription Identification Number*). Přidáním identifikace operátora a identifikace země, kde operátor operuje, vznikne IMSI (obr. 8). Ve VoLTE se od této identity odvozuje, tzv. Privátní identita účastníka (*IP Multimedia Private Identity* – IMPI) tak, že převede do „doménového tvaru“ domény sítě IPX (obr. 8). Privátní identita nesmí opustit síť operátora – je určena jen pro komunikaci v rámci jeho sítě.



Obr. 8: Veřejná a privátní identita

CSCF	*Call Session Control Function* (CSCF) entity máme: Proxy CSCF (P-CSCF), *Serving* CSCF (S-CSCF), *Emergency* CSCF (E-CSCF) a *Interrogating* CSCF (I-CSCF)
P-CSCF	Je první entitou IMS, kterou kontaktuje účastníkův UA. I když má v názvu „Proxy“, tak se jedná o B2BUA.
S-CSCF	Zpracovává požadavky a udržuje stav relací. Zprostředkovává autentizaci uživatelů (jménem P-CSCF) za využití HSS. V HSS rovněž vyhledává účastníky, kterým je voláno atd.
E-CSCF	Zpracovává tísňová volání, která předává skrze EATF do integrovaného záchranného systému („linka 112“).
I-CSCF	Je kontaktním bodem pro příchozí volání z cizích sítí.
MGCF	*Media Gateway Control Function* je branou do sítí založených na jiných protokolech než SIP (GSM apod.). Zajišťuje zejména konverzi formátu komunikace a konverzi formátu media (*transcoding*).
BGCF	*Breakout Gateway Control Function* (BGCF) je dispečerem odchozích požadavků. Rozhoduje, do jaké externí sítě má být požadavek předán: jestli do sítí na bázi protokolu SIP nebo do zastaralých sítí (přepínání okruhů) na bázi SS7.
IBCF	*An Interconnection Border Control Function* provádí případné modifikace SIP/SDP požadavků mezi sítěmi různých operátorů. Může se jednat o převod IPv4 a IPv6, skrytí topologie operátora, generování zpoplatňovacích údajů (*charging*) atd.
TrGW	*Transition Gateway* provádí překlad IP adres/portů, převod mezi IPv4 a IPv6 atd. Za jistých okolností může též provádět konverzi formátu media (*transcoding*).
EATF	*Emergency Access Transfer Function* je branou do integrovaného záchranného systému. Požadavky přijímá jak od vlastní sítě (z E-CSCF), tak i z cizích sítí (z I-CSCF).
IMS-ALG	IMS *Application Level Gateway* provádí úpravy na úrovni SIP/SDP. Např. uživatel se autentizuje vůči P-CSCF. Takže pro komunikaci s dalšími entitami už může být označen, jako důvěryhodný což se odrazí v příslušné hlavičce protokolu SIP. Provádí překlad IPv4 adres, konverzi IPv4 a IPv6 atd.
IMS-AGW	IMS *Access Gateway* provádí překlad IP adres/portů, převod mezi IPv4 a IPv6 atd. Za jistých okolností může též provádět konverzi formátu media (*transcoding*).
SEG (IPsec GW)	*Security Gateway* ukončuje IPsec tunely.
AF	Aplikační funkce – entita, která umožňuje vytvářet konkrétní aplikace na protokoly SIP/RTP, které přinášejí další přidanou hodnotu.
MRFC, MRFP	

Pro veřejnou komunikaci se používá tzv. veřejná identita (*IP Multimedia Public Identity* – IMPU).

Veřejnou identitou ve VoLTE je SIP URI nebo TEL URI. Přičemž účastník si může vytvořit řadu profilů, které obsahují jednu nebo více veřejných identit a nechat si např. v HSS nastavit, že v pracovní době má být kontaktován na jeden profil a mimo pracovní dobu na jiný profil. Na obr. 8 jsou tyto profily demonstrativně pojmenovány „Zaměstnanecký“ a „Soukromý“.

5 ISIM/USIM

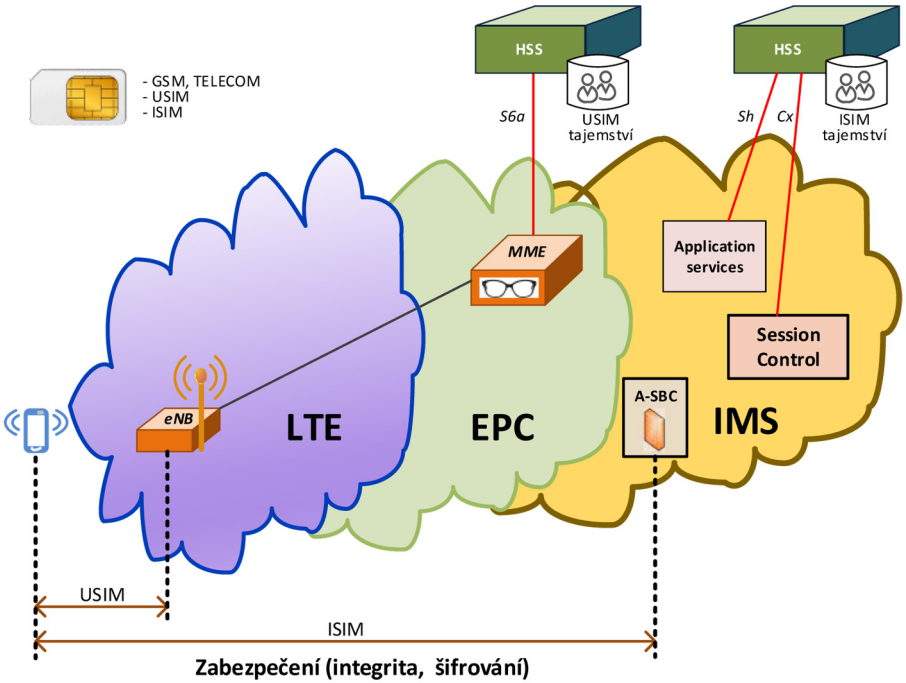
Každý uživatel má jednu privátní identitu (IMPI) a jednu nebo více veřejných identit (IMPU). V USIM je uložena pouze privátní identita (IMPI). Veřejné soukromé a veřejné identity umožňuje udržovat až ISIM. Přičemž veřejné identity mohou být v ISIM spravovány i přes mechanismus OTA (*Over The Air*). V případě, že není implementován ISIM, pak je možné využít pouze TEL URI jako veřejnou identitu, protože 3GPP sítích je zakázáno přenášet privátní identitu do cizí sítě. SIP URI odvozené od privátní identity tak není možné přenášet do cizích sítí, tj. není pomocí něj možné komunikovat mimo domovskou síť.

Jak jsme se již zmínili u mechanismu AKA, že účastníkova čipová karta obsahuje sdílené tajemství K. Čipová karta se oficiálně označuje UICC (*Universal Integrated Circuit Card*) a byla zavedena v GSM síti pod označením SIM (*Subscriber Identification Module*). Byl to velký přínos k bezpečnosti, protože se oddělil hardware mobilu od identifikačních údajů účastníka sítě. Pro potřeby GSM jsou na SIM kartě dvě aplikace (dva adresáře) GSM a TELECOM. Autentizační mechanismy GSM jsou dnes považovány za slabé, ale objektivně se musí říci, že před 20 lety čipové karty ani hardware mobilu náročnější algoritmy ani neumožňovaly. I tak to byl pokrok, protože např. dodnes používané pevné analogové linky nemají zabezpečení vůbec žádné.

S příchodem UMTS (3. generace sítí) byl zaveden mechanismus AKA a na čipové kartě („SIMce“) se objevila další aplikace (adresář) USIM a sdílené tajemství K. Tyto čipové karty se již nenazývají SIM karty, ale USIM karty.

A s příchodem IMS se může objevit na kartě jedna nebo více aplikací (adresářů) ISIM.

Důležité je, že sdílených tajemství K můžeme mít na kartě více. Jedno pro aplikaci USIM a další pro aplikaci ISIM. V případě, že je na kartě více aplikací ISIM, pak každá má své K. K nejsou umístěny přímo v aplikaci (tj. v souboru v adresáři USIM/ISIM), ale v oblasti (může být realizována i samostatným mikročipem v kartě), která se nazývá bezpečný prvek (*secure element*) a slouží právě k úschově soukromých klíčů, tajných klíčů a dalšího citlivého kryptografického materiálu účastníka účastníka. Bezpečný prvek byl zaveden pro NFC (platba mobilem místo platební kartou), aby uspokojil bezpečnostní nároky společností vydávajících platební karty.



Obr. 9: USIM/ISIM

ISIM kromě sdíleného tajemství K obsahuje zejména veřejné identity účastníka.

Na obr. 9 je znázorněno, že aplikace USIM slouží k autentizaci v LTE síti a k zabezpečení komunikace mezi mobilem a základnovou stanicí eNB (zabezpečení na IP vrstvě). Zatímco aplikace ISIM složí pro zabezpečení k autentizaci do IMS (zabezpečení na aplikační vrstvě) a zabezpečení komunikace mezi mobilem a hranou sítě IMS (A-SBC). Pozor ale, nezabezpečuje médium (RTP). Pokud chceme zabezpečit i médium, tj. použít protokol SRTP (*Secure RTP*), pak se příslušný kryptografický materiál pro toto zabezpečení přenese v SIP zprávě (protokol SDP). Na obr. 9 si ještě všimněte dvou HSS. Jedno pro EPC a druhé pro IMS. To je z toho důvodu, že obecně EPC a IMS mohou provozovat dva různé subjekty. Pokud obojí provozuje týž subjekt, pak může použít jedno HSS (nebo je replikovat).

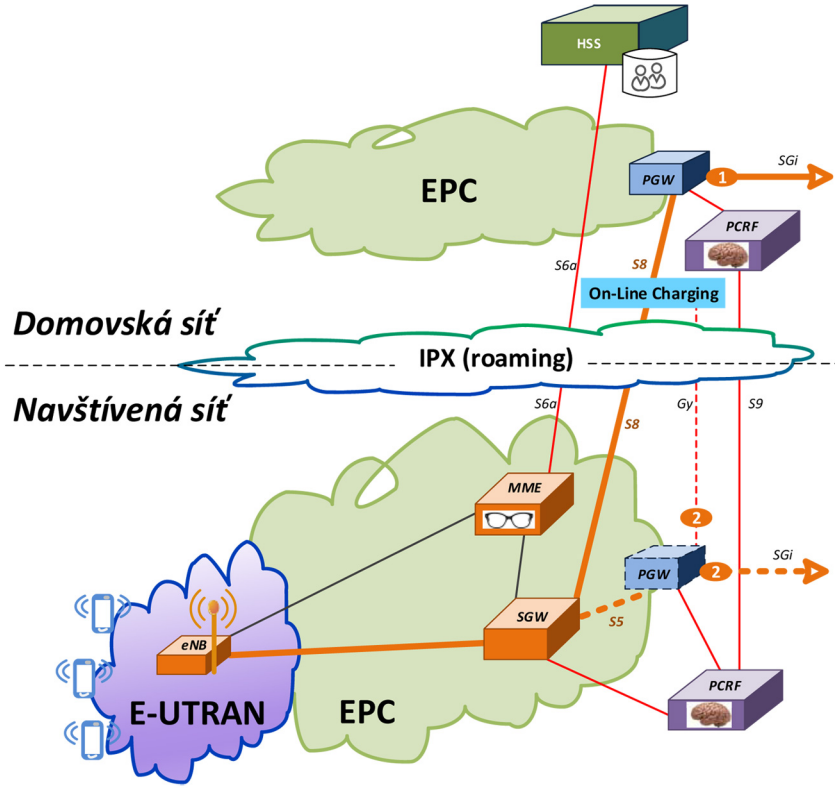
Otázkou je, jestli se můžete provozovat VoLTE i bez ISIM. Ano, ale musíte být operátorem jak LTE (včetně EPC), tak i IMS. Pak mi nevadí, když se využije stejný (nebo replikovaný) HSS. Další nevýhodou je, že na čipové kartě v aplikaci USIM nejsou uloženy veřejné identity. Důsledkem je, že se SIP URI uloží do konfigurace mobilu. Pak si jej ale uživatel může libovolně měnit. Nebo

se může uživatelům sdělit, že budou používat jen TEL URI. Tím se ale ochudí o možnost mít telefonní kontakt ve tvaru „mailové adresy“.

6 Roaming

Roamingem se myslí situace, kdy nemáme dostupnou svou domovskou síť, ale jsou dostupné cizí sítě, které nám umožňují se do nich přihlásit. V případě, že se připojíme do cizí sítě, pak tato síť se označuje jako „navštívená síť“. Účastníci českých mobilních sítí mohou t. č. navštívit cizí sítě jen v zahraničí.

Zajímavé je, že roaming máme jak na IP vrstvě, tzv. LTE roaming, tak roaming můžeme mít i na úrovni SIP protokolu, tzv. VoLTE roaming. Na úrovni VoLTE, máme kromě roamingu ještě volání do cizích sítí. Tím se míní, že volaný účastník je domovským účastníkem cizí sítě.



Obr. 10: LTE roaming

LTE roaming

Pro potřeby roamingu, ale pro volání do cizích sítí byla vytvořena síť IPX (*IP eXchange*). Jedná se o celosvětovou síť, která je paralelní sítí k Internetu a nemá s ním přímé propojení (doufejme). Obdobně jako Internet je i IPX postavena na protokolech TCP/IP, má vlastní kořenové DNS servery a tvorbu doménových jmen, vlastní autonomní systémy atd. Obdobně jako Internet je poskytována poskytovateli, se kterými operátoři uzavírají smlouvy o poskytování IPX.

Cizí uživatel, který navštíví LTE síť se nejprve potřebuje přihlásit. Přihlašovací proceduru zahájí s entitou MME navštívené sítě, která musí získat údaje o uživateli včetně kryptografického materiálu z HSS. Avšak nikoliv z HSS navštívené sítě, ale skrze IPX se dotáže HSS v domovské síti uživatele a pokračuje v přihlašovací proceduře, až umožní uživateli přístup do své sítě. Nyní máme dvě možnosti: *Home Routed LTE Roaming* a *Local Break Out*.

Home Routed LTE Roaming

SGW vytvoří uživateli datový nosič (*bearer*), ale pozor ten je tunelován od uživatele skrze síť IPX na PGW domovské sítě! A teprve pak do Internetu. Tento typ LTE roamingu se nazývá *Home Routed LTE roaming*. Výsledkem je, že si operátoři nechají tučně zaplatit za tunelované pakety skrze IPX.

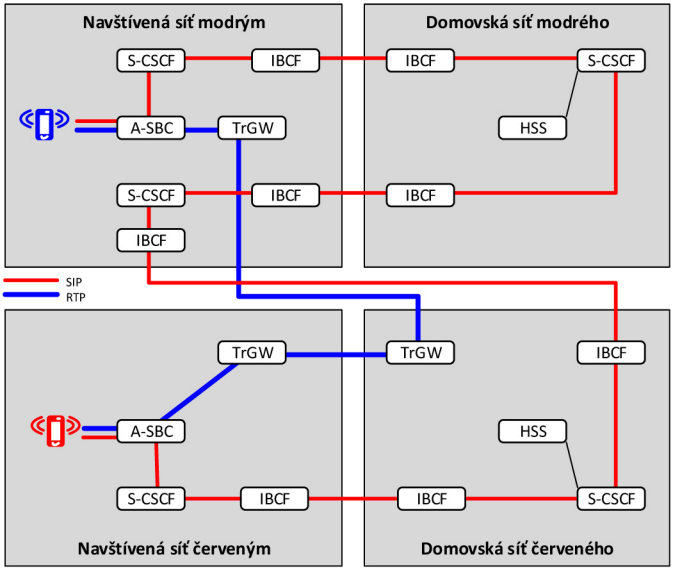
Local Break Out LTE

Druhou možností, je že by operátor navštívené sítě uživatelovy pakety přímo pouštěl do Internetu skrze jeho vlastní PGW. Tento typ LTE roamingu se nazývá *Local Break Out*. Běží ale o zpoplatnění. Pokud jsou pakety tunelovány do domovské sítě, pak domovský operátor má zpoplatnění úplně pod svou kontrolou. Jenže pokud by byly pakety přímo vypouštěny do Internetu navštívenou sítí tak, by navštívená síť posílala do domovské sítě jen účtovací údaje (skrze referenční bod Gy). Lze však považovat tyto účtovací údaje za důvěryhodné opravdu od každé sítě? To je pro každého operátora otázkou, na kterou je jasná odpověď: jistota je si měřit účtovací údaje sám.

Problém v EU spočívá v tom, že drahý (rozuměj skrze IPX tunelovaný) roaming ztěžuje a prodražuje spolupráci na společném trhu EU. Cílem je tedy regulovat (snížit) cenu. Jenže cílem regulace bude muset být zakotvení důvěry mezi evropskými operátory, aby používali *Local Break Out*. Potom se můžeme dočkat LTE roamingu za cenu, kdy nebudeme muset datové služby v zahraničí vypínat.

VoLTE Roaming

VoLTE roaming je roaming na úrovni protokolu SIP (tj. IMS).



Obr. 11: VoLTE roaming

Roaming je v IMS možné řešit dvěma způsoby:

1. Vyžije se *Home Routed LTE roaming*. Účastník ač v navštívené síti přímo kontaktuje P-CSCF (A-SBC) domovské sítě. Tím se přenese IP komunikace z navštívené sítě do domovské sítě a nic se na úrovni protokolu SIP nemusí řešit.

2. Účastník kontaktuje P-CSCF (A-SBC) navštívené sítě a komunikace s domovskou sítí se provede skrze Ici/Izi rozhraní do IPX (obr. 7).

Opět je ve vzduchu otázka, jestli by to nešlo nějak zlevnit. Při představě, že jak volaný, tak volající, jsou v roamingu, tak datový tok půjde skrze IPX třikrát, což volání prodraží. Na obr. 11 je znázorněná představa VoLTE roamingu podle normy GSMA IR.65. V tomto případě by médium neprocházelo domovskou sítí volajícího, tj. sítí IPX by se procházelo jen dvakrát.

7 Autentizace na web

V odstavci o autentizaci protokolu SIP jsme uvedli, že autentizaci SIP převzal z protokolu HTTP. Není ale možné opačně autentizační mechanismus pro VoLTE využít pro web? Ano, to je další z přínosů IMS. Původně byl navržen referenční

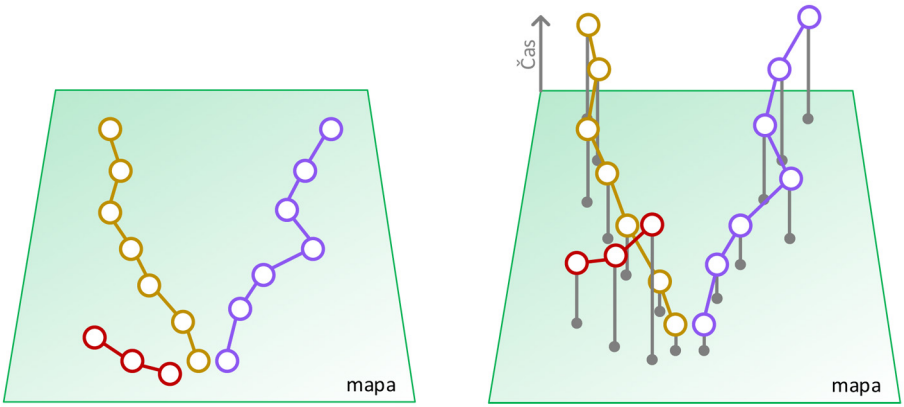
bod Ut, který měl sloužit k autentizaci uživatelů, aby si mohli na webu operátora spravovat své služby. Uživatel se vůči tomuto webu autentizuje AKA mechanismem.

Později tato myšlenka byla zobecněna pro obecnou webovou autentizaci jako GAA (*Generic Authentication Architecture*), která pro tyto účely zavádí referenční bod Ub (protokol HTTP/HTTPS).

VoLTE mobil musí z principu být neustále připojený do sítě a ještě jeho uživatel (na rozdíl od uživatele PC) je stále autentizován. Podle našeho názoru to otevírá ohromné možnosti pro tvůrce aplikací, které na PC jsou jen těžko představitelné. Např. SMS jízdenka má nevýhodu, že může být duplikována, protože revizor nemůže ověřit, že byla zakoupena z konkrétního mobilu. Jenže jízdenku zakoupenou konkrétní veřejnou identitou (neustále připojenou na síť) může jednoduše ověřit tak, že do aplikace na mobilu této veřejnou identity pošle náhodnou zprávu (např. smajlíka).

8 Odposlech

Odposlech může být legální nebo nelegální. Nelegální odposlech může provést libovolný „muž uprostřed“, který se dostane k nešifrované komunikaci SIP/RTP. Stačí k tomu např. obecně dostupný program Wireshark, který umí hovor nalézt a dokonce i uložit do souboru pro následné přehrání.



Obr. 12: Lokalizační údaje zanesené do mapy

Legální odposlech je vymezen §88 a 88a zákona 141/161 Sb., trestní řád. §88 specifikuje, za jakých okolností se může provádět odposlech a §88a pak specifikuje, za jakých okolností se mohou zjistit „údaje o telekomunikačním provozu,

které jsou předmětem telekomunikačního tajemství anebo na něž se vztahuje ochrana osobních a zprostředkovacích dat".

Zatímco zajistit klasický odposlech bude čím dál obtížnější, tak „údaje o telekomunikačním provozu" budou stále pro vyšetřování důležité, byť třeba nebudou nakonec využity v řízení před soudem. Např. lokalizační údaje zakreslené do mapy (obr. 12 vlevo) nám dají cesty, kudy se pohybovali jednotliví účastníci. Avšak pokud nad mapu vztyčíme časovou osu a výsledek zobrazíme v 3D, pak je výsledek opravdu zajímavý (obr. 12 vpravo).

Obtížnost zajištění odposlechu bude dána několika faktory. Jednak nic nebrání, aby uživatelé nešifrovali RTP komunikaci (médium) mezi sebou navzájem. Jiným problémem je, že volající může volaného jen prozvonit, tím získá IP adresu jeho mobilu a následně s ním může navázat komunikaci např. přes Internet.

9 Zmáčkni a mluv

Již zmíněná služba „Zmáčkni a mluv" (*Push to talk over Cellular*) umožňuje polo-duplexní komunikaci v rámci skupiny účastníků. Hodí se např. pro záchranný tým, kdy je nutné, aby celá skupina byla na příjmu a pouze jeden po zmáčknutí tlačítka mohl mluvit, a to k celé skupině (tzv. skupinové volání).

Pro profesionální záchranný systém se obecně používají specializované mobilní sítě TETRA (*Terrestrial Trunked Radio*) nebo Tetrapol. Oba systémy vycházejí z dnes již zastaralého systému GSM. Závisí na volbě konkrétního státu, jestli zvolí TETRA nebo Tetrapol. Jedná se o mobilní sítě, které zpravidla platí daňový poplatník. Pokrytí území státu základnovými stanicemi a vybudování příslušné infrastruktury je velice nákladná záležitost, takže ne vždy se dostatečně pokryje území. Uživatelské stanice (slovo mobil asi není to pravé) jsou proto často kombinovány s vysílačkami, aby se eliminoval problém s pokrytím.

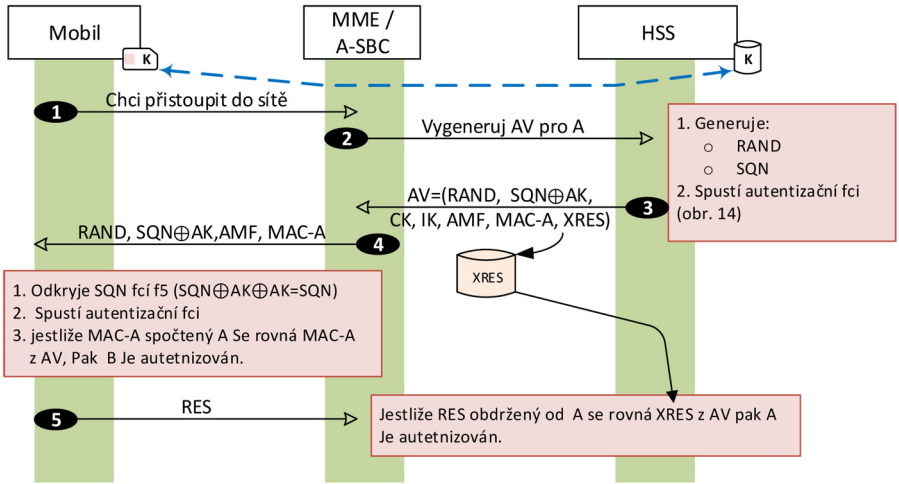
Obdobného efektu lze dosáhnout i aplikační funkcí IMS, která se nazývá „Zmáčkni a mluv" (*Push to talk over Cellular*). Stačí jeden aplikační server, který realizuje příslušnou aplikační funkci (AF) ...

10 AKA algoritmus

Kryptografický mechanismus AKA (*Authentication and Key Agreement*) je využíván jak LTE, tak i VoLTE. AKA mechanismus slouží k oboustranné autentizaci a ke generování kryptografického materiálu pro zabezpečení komunikace.

AKA mechanismus (obr. 13) zajišťuje:

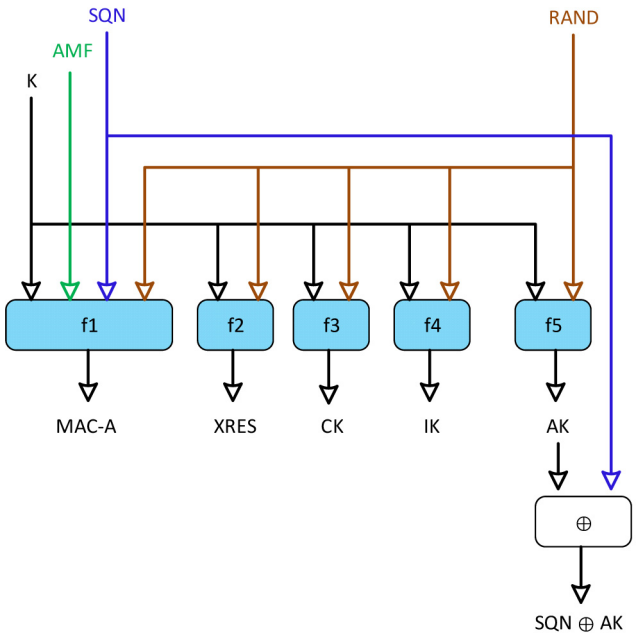
- Autentizaci sítě vůči účastníkovi pomocí jednorázového hesla MAC-A, které vygeneruje síť a účastník si jej nezávisle na tom vypočte. Pokud se rovnají, pak je **síť** autentizována.



Obr. 13: AKA (Authentication and Key Agreement)

- Autentizace účastníka vůči síti pomocí jednorázového hesla XRES, které vygeneruje síť a předá jej MME (resp. A-SBC). Uživatel vygeneruje RES a zašle jej MME (resp. A-SBC). Pokud RES je shodné s XRES, pak je **uživatel** autentizován.
- Šifrovací klíč CK a sdílené tajemství IK, které slouží pro zajištění integrity přenášených dat

f1 až f5	Jednocestné funkce
K	Sdílené tajemství mezi účastníkovou čipovou kartou USIM/ISIM a sítí (HSS)
RAND	Náhodné číslo generované HSS pro každou autentizaci
SQN	Pořadové číslo autentizace (udržuje síť i klient)
AK	Anonymizační klíč k SEQ (skryje SEQ během přenosu sítí)
AMF	*Authentication Management Field* (předem známý řetězec)
MAC-A	Jednorázové heslo pro autentizaci sítě
XRES	Očekávané jednorázové heslo pro autentizaci klienta
CK	*Cypher Key* – šifrovací klíč
IK	*Integrity Key* – sdílené tajemství pro zajištění integrity přenášených dat)



Obr. 14: Jednocestné funkce f1 až f5

Mechanismus je jednoduchý:

AKA 1 Mobil se chce autentizovat, tak zašle MME (resp. A-SBC) svou privátní identitu.

AKA 2 MME (resp. A-SBC) požádá HSS o vygenerování tzv. autentizačního vektoru AV pro danou privátní identitu.

AKA 3 HSS následně:

- Generuje náhodné číslo RAND.
- Uložené číslo SEQ zvětší o jedna.
- Spustí jednocestné funkce f1 až f5 (obr. 14). Do výpočtu vstupuje sdílené tajemství K a známý řetězec AMF.
- Výsledkem je autentizační vektor AV = (RAND, SQN⊕AK, AMF, MAC-A, XRES), který předá MME (resp. A-SBC).

AKA 4 MME (resp. A-SBC) vyzobne z AV jednorázové heslo XRES, které si uloží a zbytek pošle do mobilu.

AKA 5 Mobil nejprve spustí funkci f5, aby získal SEQ, které porovná s jim udržovaným SEQ. Poté spustí zbylé jednocestné funkce a získá MAC-A, XRES a kryptografický materiál IK, CK.

AKA 6 Mobil porovná jim spočtené MAC-A s MAC-A z jim přijatého autentizačního vektoru, pokud se rovnají, pak je síť autentizována. Mobil odešle spočtené XRES jako RES.

AKA 7 MME (resp. A-SBC) porovná XRES s RES...

Sociální sítě hýbou světem

Jiří Kolařík

E-mail: jiri.kolarik@socialbakers.com

Abstrakt

Sociální sítě nejsou jenom Facebook, ale je jich mnohem více a jsou všude kolem nás. Twitter, YouTube, Pinterest, LinkedIn, Instagram, Tinder, Spotify, Swarm, Vine, a další nás ovlivňují každým dnem. Do našich životů vstoupily neskutečně rychle a lidé si je oblíbili. A nebo si nás sociální sítě zotročily? Umíme s nimi pracovat? První blok představí jednotlivé sítě, jejich typické využití i specifika. Zabrousíme do vod tajemna, soukromí i hrozeb sociálních sítí. Podíváme se na trendy a budoucnost sociálních sítí.

V druhém bloku se zaměříme na vztahy mezi obchodníky a zákazníky, kdy jsou firmy nucené reagovat a fungovat na sociálních sítích. Firmy nyní vynakládají nemalé prostředky pro to, aby byly na sítích úspěšnými, připravují zajímavý obsah a snaží se zákazníky nejen bavit. Zákazníci jsou firmám blíže, než kdy dříve. Co to v praxi znamená? Co se mění? Představíme si nástroje, které firmy používají pro správu sociálních sítí a jak vůbec taková správa funguje. Proč by firmy neměly brát sociální sítě na lehkou váhu?

V závěru se zaměříme na firmy, které ze sociálních sítí těží a mají na nich postavený svůj business, konkrétně na české Socialbakers, kteří jsou mezinárodně uznávanou společností v oblasti sociálních sítí. Co vlastně dělají? Povíme si o big data, o agilním vývoji a developmentu vůbec. Jaké technologie jsou k tomu potřeba a jaké jsou radosti a strasti analýzy dat ze sociálních sítí?

Příspěvek naleznete na adrese:
http://jiri-kolarik.cz/prednaska/socialni-site-hybou-svetem

Cíl zaměřen: uživatel

Karel Nykles, Mirka Jarošová

E-mail: knykles@civ.zcu.cz, j.mirka@hotmail.com

Úvodem

Za poslední dva roky proletěla e-mailovými schránkami a přes sociální sítě celá řádka phishingových kampaní. Již od dob proseb nigerijského prince, letadlových her z devadesátých let mají tyto události společného jmenovatele, lidský faktor. Současné technologie pokročily, po signatury rozpoznávajících antivirech, přes firewally jsme se dostali k IPS systémům, SIEM korelátorům, sandboxům a monitoringu v reálném čase. Přesto však ani tato armáda drahých sluhů není vždy dostatečná, když do hry vstoupí lidský faktor.

Naším cílem bylo prozkoumat možné příčiny, pokusit se o jejich vysvětlení za pomoci psychologie a přirozených lidských vlastností, posuzovat uživatele nikoli jako doplněk, ale jako součást IT infrastruktury.

Pojmy v souvislostech

Virus

Malý infekční patogen replikující se pouze v buňkách živého organismu. Viry mohou infikovat veškeré známé životní formy.

Mem

Myšlenka, chování, styl, či způsob použití, které se šíří z člověka na člověka v rámci specifické kultury.

Zranitelnost

Značí neschopnost/nemožnost odolat působení nepřátelského vlivu. Může být trvalá – odolnost lidského těla vůči pádu meteoritu, nebo dočasná – doba mezi objevením zranitelnosti a instalací opravného balíčku.

Exploit

Exploit lze popsat jako sekvenci kroků, jež umožní zneužití vlastnosti cílového systému ve prospěch útočníka. Zde již pro účely přednášky nezáleží, zda jde o neošetřené vstupní textové pole, nebo o mem šířící se po sociálních sítích. Neošetřené vstupní pole může umožnit spuštění útočníkova kódu. Mem vykazuje virální potenciál – nutí uživatele ke svému šíření. Samotný mem ještě nebezpečný není, avšak ve spojení se závadným obsahem může být jeho dopad zničující. Exploit jako takový pouze odemyká přístup do systému.

Payload

Sada kroků, která útočníkovi umožňuje ovládat napadený systém.

Sociální inženýrství

Popisuje psychologickou manipulaci lidí, uživatelů, za účelem provedení vůle útočníka, či vyzrazení citlivých informací.

Asymetrická hrozba

Zatímco obránce, například společnost, musí pokrýt a zabezpečit svou infrastrukturu vůči všem možným způsobům exploatace, útočníkovi postačí zneužít pouze jediný článek této obrany, aby se dostal hlouběji k prostředkům společnosti.

Průběh útoku

Průběh útoku na IT infrastrukturu lze ve zjednodušené podobě popsat následovně:

Průzkum terénu

Exploatace

Ovládnutí

Záměrně jsou vynechány další (mezi)kroky.

Průzkum útočníkovi zajišťuje nutné informace k provedení útoku. Například specializace administrátorů (i bývalých), na LinkedIn, skenování IP rozsahu společnosti, samotný profil společnosti a čím se zabývá, seznam partnerů. S těmito informacemi se útočníkům již pracuje lépe. Nyní mohou útočníci zaútočit na zranitelnosti systémů společnosti. V případě IT infrastruktury to může být neaktualizovaný systém, či slabé heslo kam zamíří exploit. V případě že systémem je uživatel, dojde ke zneužití záludností lidské psychiky. Opět zde velmi dobře poslouží informace ze sociálních sítí, ale i osobní, či telefonický kontakt se zaměstnanci.

Cíl zaměřen

Po průzkumu terénu útočník vybírá nejslabší článek bezpečnostního řetězu. Velmi univerzální a slabý článek představuje uživatel. Proč? Protože uživatel dělá svou práci. A výjimkou IT personálu nemá o bezpečnosti ponětí, či je to pro něj otravná záležitost, která jej kupříkladu zdržuje od tisku výplatních pásek. Firewall či IPS systém lze za odpovídající částku upgradovat poměrně snadno, jak však upgradovat lidskou bytost, samotné uživatele? A ptal se jich vůbec někdo?

Situace z pohledu obětí BI

Za účelem získání uživatelského pohledu na bezpečnost v IT, bylo osloveno 76 respondentů z řad obětí BI napříč zaměstnanci organizace. Prostředkem sběru informací byl zvolen anonymní dotazník v Google Drive. Seznam otázek a možných odpovědí je dostupný zde:

https://docs.google.com/a/gapps.zcu.cz/forms/d/
113u7xzmmyql8_w3ZWPs7BDY6ZxZey5tgbXhRfsWKKsU/viewform

Od oslovených 80 uživatelů bylo získáno 28 vyplněných formulářů, což ukazuje na 35% výtěžnost ankety, a jednu obavu o ochranu osobních údajů z důvodu použití Google Drive.

Vyhodnocení dotazníku

Přestože rozsah dotazovaných nebyl velký, vypovídací hodnotu v podobě obecného uživatelského pohledu na IT bezpečnost přináší. Téměř 100 % respondentů odpovídá, že jsou v otázce bezpečnosti IT informováni. Z ohledem na skupinu z jaké byli respondenti vybráni, však lze soudit, že tomu tak není. Respektive respondenti žijí ve falešném pocitu porozumění problematice IT bezpečnosti. 64 % respondentů přichází do styku s e-maily mimo organizaci denně, zbytek pak minimálně jednou týdně. Polovina uživatelů kontroluje přílohu e-mailu na základě přípony, čtvrtina pomocí ikony, všichni nějaký způsob kontroly provádí. Pokud získá uživatel podezření, ve 46 % případů přílohu smaže, 21 % se zeptá kolegy(ně), 14 % zavolá uživatelskou podporu, 11 % soubor vůbec neotevírá a pouze 7 % soubor kontroluje antivirovým programem. Pouze 35 % respondentů sleduje aktivně stránky uživatelské podpory, může to znamenat selhání v komunikaci těchto informací uživatelů, ale také nedostatečnou motivaci uživatelů tyto informace aktivně vyhledávat.

Proč tedy uživatelé otevírají závadné přílohy? 26 % uživatelů přílohu otevřelo i přes mírné podezření, stejné množství podlehlo strachu z uvedené hrozby,

21 % neočekávalo takovouto možnost zneužití e-mailu, 10 % si nemůže vzpomenout, stejné množství přiznává otevření omylem. 8 % respondentů spoléhalo na výsledky antivirového skenu.

Zajímavý výsledek přináší uživatelský pohled na možnosti informování se o bezpečnostních hrozbách. 70% preferuje pravidelný e-mailový zpravodaj. Pro hromadné školení, či zasílání falešných podvodných e-mailů se staví shodně po 11 %. 4 % touží po online dokumentaci, či individuální konzultaci.

Rozhodovací proces v mozku

Příčiny současného stavu

Pro lepší porozumění současnému stavu bude následovat rozbor rozhodovacího procesu uživatele. Jaké faktory vedou uživatele k rozhodnutí podlehnout, či nepodlehnout útoku a uvědomuje si uživatel, že vůbec útoku čelí?

Pokud uživatel zvolí přístup vědecký, ověří, zdali je přijatá zpráva v mezích podobnosti k ostatním zprávám daného odesilatele. Zdali žádost v e-mailu uvedená odpovídá obvyklým žádostem v podobných situacích. Taková činnost však vyžaduje soustředěnou pozornost, které se ne vždy uživatelům dostává.

Uživatel tedy použije heuristiku a prahování pro rozhodnutí o legitimitě dané zprávy. Stačí mu k tomu odpovědi na následující otázky:

> Je daná informace dostatečně podobná těm, které již znám – rozhodoval jsem o nich dříve?
>
> Dokážu si podobný průběh událostí představit?
>
> Jaký mám z této informace pocit?
>
> Rozhodli by se ostatní podobně?

Z pohledu mozkových struktur jsou za rozhodovací proces odpovědné dvě struktury: Neokortex a amygdala.

Duální mysl

Uživatel má k dispozici dvě cesty rozhodovacího procesu.

Neokortex

Vývojově mladší část mozkové kůry zodpovědná za racionální uvažování. Rozhodování je přesné, ale pomalé. Zohledňuje veškerá dostupná fakta a využívá logiku.

Vědomé rozhodování – neokortex

Vědomě se uživatel rozhoduje pomalu, veden svým zájmem, pečlivě vyhodnocuje veškeré své kroky, až na základě kalkulací všech variant vybere nejlepší řešení. Tímto postupem soustředěně a s plným úsilím, zaměřen na jeden konkrétní problém, rozhodne. Chyby v tomto procesu vznikají při nedostatku informací, či zkušeností. Například výběr bytu ke koupi.

Amygdala

Jedná se o vývojově starší část mozku, zodpovědnou za rozhodování typu útok/útěk. Rozhodovací proces v amygdale je zaměřen na rychlost na úkor přesnosti, nejsou zohledněna veškerá fakta, velkou roli hrají v rozhodovacím procesu emoce.

Automatické, podvědomé rozhodování

Rozhodnutí jsou činěna v řádu milisekund na základě odhadů a heuristik. Lze takto rozhodnout mnoho jednoduchých úkonů současně. Rozhodnutí na základě „pocitu ze situace“. Například: „Mám žízeň, jdu se napít.“

Jak přimět uživatele k nepromyšlenému rozhodnutí?

Stačí jej zavalit prací, úkoly, nastavit nereálné termíny a dostat jej pod tlak, pohrozit mu a aktivovat tak emoční centra. Zároveň může jít i o případ nedostatečné motivace, anonymity jednání, kdy bez dostatečné odpovědnosti jedinec nepociťuje potřebu hlubšího zkoumání svých činů. Kupříkladu v davu, nebo kanceláři kdy kolega požádá o pomoc s přílohou, jež nejde otevřít. Dále také jednání automatizované rutinou – finanční a právní oddělení vyřizují pohledávky často.

Jak uživatele přimět k promyšlenému jednání?

Když dostane k dispozici fakta, kterým jednoduše porozumí. Bude mít jasně daná pravidla, na základě kterých se rozhodne. Bude znát, či mít k dispozici příběhy s podobnou situací. Bude za svá rozhodnutí odpovědný a výsledek jeho rozhodnutí bude veřejný.

Využití uvedených informací

Cílové rozhodování uživatelů v rámci IT bezpečnosti

Důležité je přimět bezohledné uživatele, kteří snadno a často způsobují incidenty, k promýšlení svých kroků a jejich důsledků. Zároveň motivovat zkušené a znalé uživatele ke spolupráci s těmi méně nadanými. Co však s uživateli, jejichž práce je důležitá, ale IT bezpečnost je pro ně nepochopitelnou záležitostí?

Klasifikace uživatelů

Průšvihář

Notorická oběť bezpečnostních incidentů, nereaguje na pokusy o vzdělávání, je mu to jedno. Je téměř jisté, že při příštím incidentu bude stanice tohoto uživatele mezi prvními kompromitovanými.

Zkušený matador

Umí rozlišit pokusy o podvod, odolává sociálnímu inženýrství, mívá autoritu mezi ostatními uživateli, tvoří „tajnou nultou linii uživatelské podpory“. Těmto uživatelům obvykle směřuje dotaz ještě dříve, než se dostane do rukou uživatelské podpory.

Ten „šedý průměr“

IT bezpečnost není jejich denním chlebem, ale učí se každým dalším dnem a incidentem, k sociálnímu inženýrství jsou různě odolní a to nejen individuálně, ale i s ohledem na pracovní a nepracovní události, což nelze předvídat. Tito uživatelé se zlepšují v čase a lze je motivovat k lepším výsledkům.

Jedna z cest k bezpečným zítřkům

Ideálním stavem by byla uživatelská IT bezpečnostní „domobrana“, lze ji však vytvořit?

Uživatele typu „Průšvihář“ lze využít jako „honey pot“. Na stanici tohoto uživatele lze nainstalovat rozšířený monitoring běžících procesů, nastavit restriktivnější politiky pro práci se soubory, omezit pravidla firewallu a sbírat informace o sociálním inženýrství, či phishingu, kterému podlehli.

Sebrané informace lze použít k tvorbě falešných podvodných e-mailů, rozesílaných supportem, popřípadě vystavených na webu, tak aby měli uživatelé referenci. Lze vytvořit i veřejnou tabulku nejlepších lovců phishingových kampaní, či pokusů o sociální inženýrství. Taková tabulka v rámci organizace nenápadně vystaví na odiv schopnosti „Zkušených matadorů“, na které pak bude směřovat větší část dotazů z „šedého průměru“. Za tuto činnost můžou být odměněni.

Zároveň se vyplatí udržovat si pro support interní neveřejnou tabulku největších průšvihářů, která bude využita pro a při detekci anomálií.

Sdělení informací o incidentu uživatelům

Sdělení uživatelům může vyznít následovně:

IT oddělění zaznamenalo šíření phishingové zprávy, prosím neotevírejte přílohu!

Nebo:

Díky včasné informaci pana Nováka o podezřelém e-mailu uživatelské podpoře, se podařilo IT oddělení zabránit úniku přihlašovacích údajů k internetovému bankovnictví firmy. Naše prostředky i vaše výplaty jsou v bezpečí. Děkujeme!

Krátký příběh je snažší na zapamatování, než suché sdělení. Personalizace sdělění také pomáhá (se souhlasem dotyčného).

Shrnutí

„Průšviháři“ slouží jako systém včasného varování. Díky příběhům ze světa IT bezpečnosti se mezi uživateli snadno rozšíří povědomí o problematice a vůči útokům typu sociálního inženýrství budou odolnější. S přidáním herního faktoru, je možné problematiku ještě více zatraktivnit, za cenu rizika poklesu pozornosti na vlastní práci. Tam kde pro čisté IT systémy slouží korelátory, firewally, a pokročilá heuristika, je u lidských uživatelů třeba nasadit psychologii a vyvinout adekvátní úsilí k analýze problémů. Upgradovat uživatele tedy odpovídajícími prostředky lze.

Technologie softwarově definovaného rádia

Jan Hrach

E-mail: jenda@hrach.eu

Technologie softwarově definovaného rádia (SDR) nám umožňuje snadno stavět a ladit přijímače i poměrně komplikovaných signálů. Poslední dobou se navíc začaly objevovat cenově dostupné univerzální SDR přijímače a toto odvětví tak zažívá boom.

Co je to SDR

Cílem technologie SDR je co nejdříve dostat rádiový signál do počítače. Díky tomu je pak možné operace s ním programovat, zatímco u „klasického“ přístupu byste při každé změně museli měnit zapojení přijímače, které by navíc u některých softwarově snadno realizovatelných aplikací bylo extrémně složité. To má samozřejmě spoustu různých výhod – dá se použít váš oblíbený programovací jazyk, debugger a verzovací systém, signály lze uložit do souboru a hrát si s nimi, i když zajímavé vysílání už skončilo, a vytvořené „rádio“ lze distribuovat jako běžný program po Internetu. Na druhou stranu je potřeba zmínit i některé nevýhody. Takové zpracování bývá náročné na výpočetní výkon, kvalitní hardware je spíš dražší a levný hardware je nekvalitní.

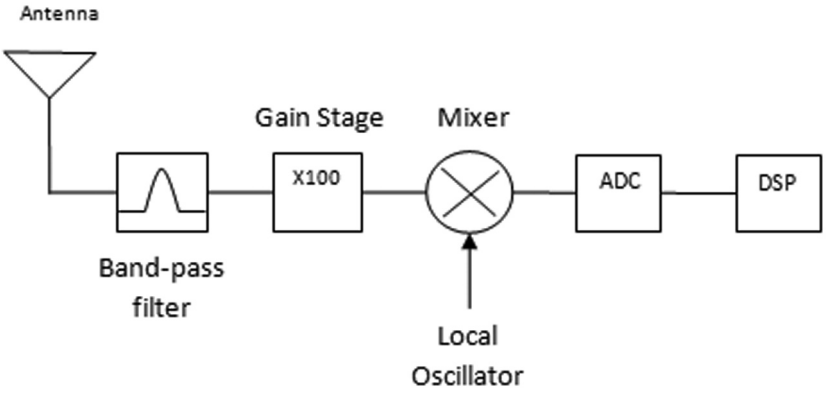
Úplně nejblíž tomuto cíli by bylo připojit přímo k anténě pekelně rychlý analogově-digitální převodník (ADC) a mít rovnou digitální data „ze vzduchu“. Bohužel často chceme přijímat signál s frekvencí ve stovkách MHz až jednotkách GHz a takto rychlé převodníky nejsou úplně běžnou záležitostí. Proto se budeme muset spokojit s nějakým kompromisem. S přístupem „anténa Š dolní propust (aby nedocházelo k aliasingu[1] vyšších frekvencí) → ADC → počítač“ se však někdy můžeme setkat při přijímání nižších frekvencí. Například projekt http://websdr.ewi.utwente.nl:8901/ používá AD převodník na frekvenci 78 MHz k samplování celého rádiového spektra od 0 do 30 MHz.

Řekněme tedy, že máme ADC o rychlosti pár MHz a chceme přijímat signál na 100 MHz. Co s tím? Vzpomeneme si, jak funguje přijímač typu superhet[2].

[1] https://en.wikipedia.org/wiki/Aliasing

[2] https://en.wikipedia.org/wiki/Superheterodyne_receiver#Design_and_principle_of_operation

Vygeneroval si frekvenci o trochu nižší, než na které chtěl poslouchat, a následně s ní signál z antény smíchal. Tím mu vznikl součet a rozdíl těchto dvou frekvencí[3]. Součet má ještě vyšší frekvenci, než s jakou jsme začali, ten nám nepomůže, ten můžeme zahodit. Ale rozdíl už má frekvenci stravitelnou pro náš AD převodník a počítač.



Obr. 1: Softwarově definované rádio
(zdroj: sdrjove – http://sdrjove.wikispaces.com/Background)
Pásmová propust vysekne tu část spektra, která nás zajímá (např. 90–100 MHz pro FM rozhlas), zesilovač slabý signál zesílí, mixér s lokálním oscilátorem ho posune na nižší frekvenci a ADC ho nasampluje. DSP (Digital Signal Processing) je potom samotné zpracování digitálního signálu na počítači

Mimochodem stejný princip lze použít i pro vysílání – vytvoříme signál na nízké frekvenci, posuneme na cílovou frekvenci a přivedeme ho do antény.

Hardware pro SDR

Univerzální SDR byla dlouhou dobu pro amatéra velmi drahá na nějaké obyčejné experimentování (od desítek tisíc do milionů korun). Na začátku roku 2012 ale bylo objeveno[4], že některé levné DVB-T tunery realizují poslech rozhlasu pomocí SDR a že je lze přeladit i mimo rozhlasové frekvence. A tak se zrodilo rtl-sdr. Kompatibilní klíčenky lze sehnat v zahraničních e-shopech v přepočtu za 200 korun.

Jaké má rtl-sdr parametry? Nic moc. Umí přijímat pásmo široké 2 MHz (některé lze přetaktovat až na 3,2 MHz), což je pro mnoho použití dostačující,

[3] https://en.wikipedia.org/wiki/Frequency_mixer
[4] http://comments.gmane.org/gmane.linux.drivers.video-input-infrastructure/44461

Obr. 2: Rtl-sdr klíčenka

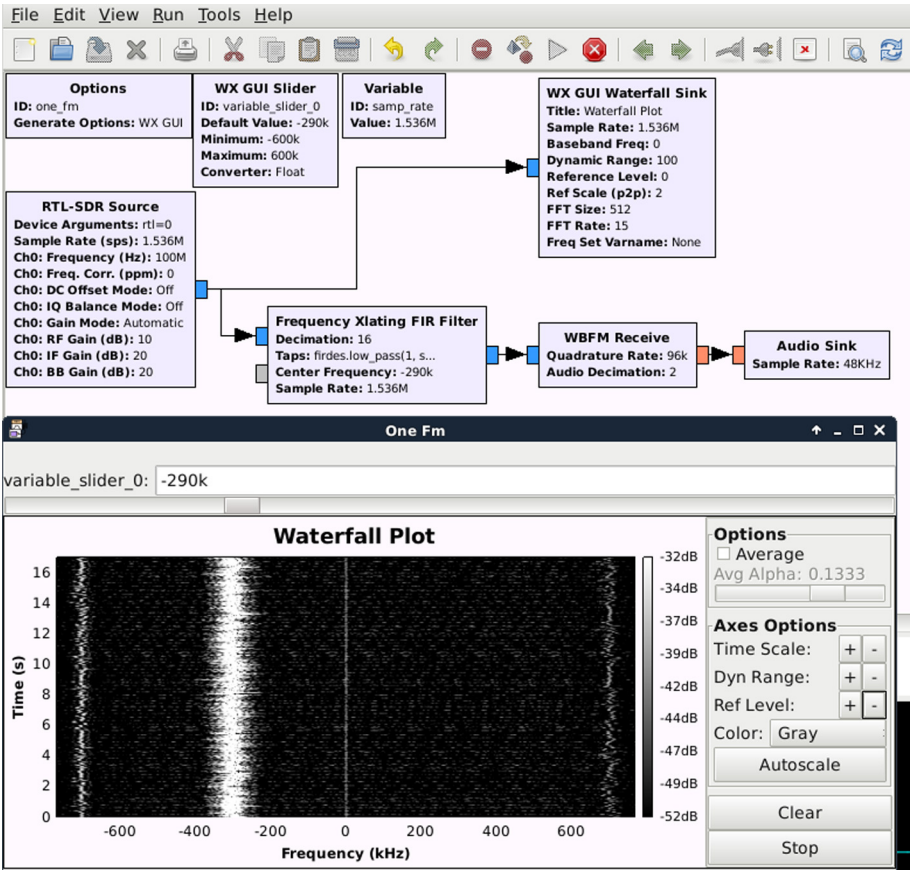
bohužel rozlišení je sotva 8 bitů. To také znamená, že pokud vedle sebe vysílá silná a slabá stanice, tu slabou prostě neuslyšíte. Mnohem větší problém je ale minimální odolnost proti rušení. Ve městě se od okolních vysílačů úplně zahltí a celé spektrum se pokryje souvislou vrstvou šumu, případně mnoha kopiemi nejsilnější stanice, která přehluší úplně všechno ostatní.

Jak se proti tomu bránit? Můžete si koupit pásmovou zádrž (zkušenosti mám s výrobci Alcad a Teroz), která rušivý signál potlačí, nebo pásmovou propust, která potlačí všechno kromě toho, co vás zajímá. Po instalaci tohoto filtru použitelnost rtl-sdr podstatně vzroste. Další užitečnou hardwarovou úpravou je připájet mezi anténu a zem tlumivku pro svod statické elektřiny.

Ostatní SDR hardware je například bladeRF, HackRF nebo USRP. Tato rádia umí i vysílat, mají větší šířku pásma, vyšší rozlišení a precizně provedenou vstupní část. Když jsem si po roce experimentování se zašuměným rtl-sdr poprvé připojil k počítači bladeRF, málem jsem nepoznal, co na obrazovce vlastně vidím. Jedinou nevýhodou těchto kvalitních výrobků je cena, která se pohybuje kolem 15 tisíc korun.

Software pro SDR

GNU Radio je oblíbená knihovna pro práci se signály. Obsahuje „bloky“, které čtou proud dat, dělají nad ním nějakou operaci (například násobení, filtrace, Fourierova transformace nebo FM demodulace) a výsledek zapisují na výstup. Tyto bloky lze potom propojovat – a to jak pomocí programu v C++ nebo v Pythonu, tak graficky v nástroji gnuradio-companion, který se výborně hodí na všelijaké experimentování.

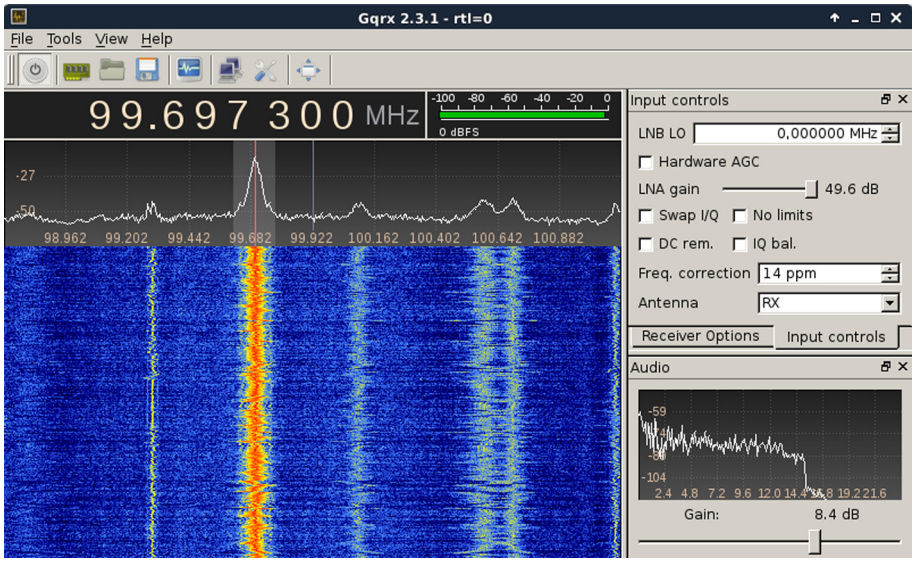


Obr. 3: FM rádio naklikané v gnuradio-companion

GQRX je „uživatelská“ aplikace, ve které si můžete ladit. Vidíte v něm spektrum, rozložení energie kolem zvolené frekvence. Můžete si pak vybrat některý z podporovaných demodulátorů (FM, AM, sideband, ...) a signál s ním zkusit přijímat.

Další uživatelský software je například HDSDR (pro Windows) nebo Sdr# (multiplatformní).

Nyní se podíváme na některá použití SDR aneb jaké signály kolem nás létají.



Obr. 4

Aplikace SDR

FM zvuk

Jednoduchá analogová modulace lidem nejspíš známá z FM rádia. Její na šířku pásma méně náročné varianty se však používají i pro všemožné vysílačkové systémy, například v kurýrních službách, řízení provozu metra a dráhy atd. To je možné většinou najít v pásmu 150–170 a 440–480 MHz. Ze své podstaty neumožňuje jakékoli zabezpečení.

Samostatnou kapitolou jsou potom bezdrátové mikrofony. Používají se na různých akcích a konferencích pro ozvučení většího sálu. Často lze signál přijímat i na ulici před budovou, což je výhodné, pokud jde třeba o nějakou uzavřenou konferenci, kam vás nechtějí pozvat.

GSM

„2G“ mobilní telefony používají šifru A5/1, kterou lze na silnějším počítači zlomit během několika sekund. Pro příjem pomocí SDR lze použít třeba program Airprobe, pro cracknutí šifry potom můj program Deka[5].

[5] http://brmlab.cz/project/gsm/deka/start

Tetra

Tetra je síť podobná GSM určená pro průmyslové využití. Používají ji například městské policie, pražský Dopravní podnik atd. Tetra volitelně podporuje šifrování, které ovšem zvyšuje licenční i servisní náklady na provoz sítě, proto naleznete mnoho sítí nešifrovaných. Malý cluster ze 2–3 běžných počítačů dokáže v reálném čase dekódovat a ukládat veškerý provoz na síti ve velkém městě. Jako software můžete použít projekt osmo-tetra, případně jeho brmlabí fork[6].

Mototrbo/DMR

Další digitální síť pro přenos hlasu a dat. Je populární mezi obecními a městskými policiemi, které ovšem postupně zapínají šifrování. Z dostupné dokumentace lze vyčíst, že šifra je mizerná („basic encryption“ je 16bitový LFSR, „enhanced“ je 40bitová RC-4), nejsou ovšem známy parametry nastavení šifrovacího algoritmu jako inicializační vektor a nevím o tom, že by někdo publikoval louskáček. Nemělo by to být ale zas tak složité… Síť je zajímavá také tím, že lze najít informace, že ji ČEZ používá pro vzdálené ovládání distribuční soustavy (to je tedy šifrované).

Příjem: „dsd“. Bylo trochu netriviální vychytat parametry. Informace, že chcete 13 kHz široký NFM přijímač s jednokanálovým výstupem na 48 kHz, snad ušetří lidem, co by si to chtěli rozchodit, nějakou tu chvíli.

Tetrapol

Aby toho nebylo málo, další datová síť. Tentokrát používaná (státní) policií a dalšími složkami. Existuje experimentální dekodér tetrapol-kit, který umí číst některá metadata ze základnových stanic. Síť používá šifrování, použitý algoritmus ani jeho síla nejsou známy.

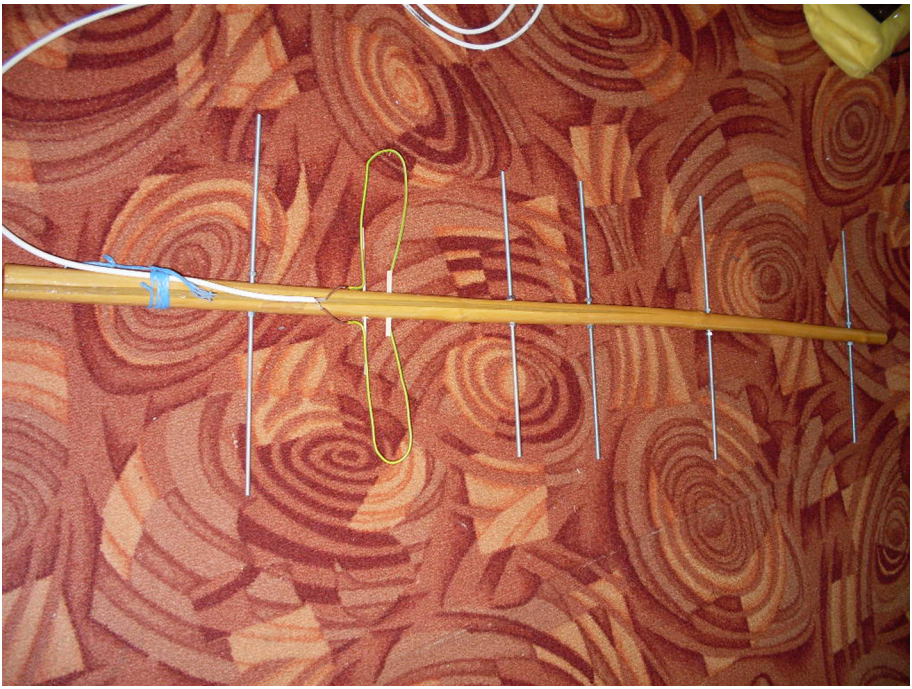
Paging

Programem multimon-ng lze přijímat provoz pagerů, které byly populární tak v půlce devadesátých let. Nyní lze slyšet snad už jen sanitky, které si posílají krátký popis incidentu („POROD“, „KOLAPS“) a pár ne-ascii bajtů, ve kterých jsem se nerýpal.

Data modulovaná AFSK modulovaná FM

(… což nemá moc dobré technické parametry, ale zase je to velmi levné). Přes FM na sebe pískají vláčky, různé senzory a meteorologické sondy. S těmi sondami

[6] http://brmlab.cz/gitweb/?p=tetra-listener.git;a=summary



Obr. 5: Směrová anténa z bambusu a závitové tyče používaná k dohledávání sond (a dalších vysílačů v pásmu 70 cm). Pro výpočet délky prvků a jejich rozteče lze použít mnoho programů i javascriptových stránek

je docela zábava, vypouští se třikrát denně z Hydrometeorologického ústavu, chvíli letí, a pak spadnou. A jako někdo provozuje goecaching, tak se dá provozovat sondecaching. Sonda po dopadu stále vysílá, takže se dá zaměřit směrovou anténou. Software: SondeMonitor (shareware pro Windows, funguje ve Wine).

Letadla

Letadla používají radarové odpovídače, které po ozáření svazkem letištního radaru vyšlou informace o letadle, aby na se na radaru kromě tečky zobrazilo i co za letadlo to je. Obsahují i další telemetrii. Hledáte dekodér systémů ACARS a ADS-B, například program dump1090. Známá stránka Flightradar24 agreguje tato data od mnoha dobrovolníků po celém světě.

Ne vždy jsou ale v letadlem odeslaných datech souřadnice. Máme tedy letící vysílač a chceme určit jeho polohu. To je úloha, kterou dělaly československé systémy Kopáč, Ramona a Tamara. Mají v krajině několik přijímačů, které si mezi

sebou drží velmi přesné hodiny. Když potom chytí signál od letadla, porovnají, ke komu došel s jakým zpožděním (protože signál se šíří „pomalu“, „jenom“ rychlostí světla), a z toho vypočtou polohu. Flightradar tohle začal dělat také – a díky technologii SDR a tomu, že přesné hodiny lze levně získat příjmem GPS, k tomu není potřeba Tatra plná techniky.

Nejzajímavější je pak zjištění polohy letadla, které *nevysílá*. V době analogových televizí šlo v údolí při přeletu letadla vidět v obraze duchy: v údolí je slabý signál, a když přes údolí letí letadlo, část signálu se od něj odrazí a signál se posílí. Jenže signál, který šel přes letadlo, urazil delší cestu, a je proto zpožděný. Z doby zpoždění a rychlosti světla snadno spočítáme rozdíl vzdáleností, a když máme vysílačů a/nebo přijímačů více, můžeme určit polohu letadla.

Našel jsem paper[7], kde tohle umí s digitální televizí. A navíc měří dopplerovský posun signálu, takže vidí, jak rychle se letadlo vzdaluje, a u helikoptéry vidí rotující vrtule (!).

Odkazy na další zdroje

[1] DSP Workshop – spousta odkazů na matematické pozadí zpracování signálů.
http://brmlab.cz/event/dsp#zdroje

[2] SDR – spousta odkazů na další aplikace SDR.
http://brmlab.cz/project/sdr#links

[3] Záznam přednášky „Postavte si vlastní NSA“.
http://nat.brmlab.cz/talks/2014-09-13-postavte-si-vlastni-nsa.mkv

[7] http://people.duke.edu/~hah16/papers/passive-radar-processing-preprint.pdf

YodaQA: A Modular Question Answering System Pipeline

Petr Baudiš

E-mail: pasky@ucw.cz

Abstract

Question Answering as a sub-field of information retrieval and information extraction is recently enjoying renewed popularity, triggered by the publicized success of IBM Watson in the Jeopardy! competition. But Question Answering research is now proceeding in several semi-independent tiers depending on the precise task formulation and constraints on the knowledge base, and new researchers entering the field can focus only on various restricted sub-tasks as no modern full-scale software system for QA has been openly available until recently.

By our YodaQA system that we introduce here, we seek to reunite and boost research efforts in Question Answering, providing a modular, open source pipeline for this task — allowing integration of various knowledge base paradigms, answer production and analysis strategies and using a machine learned models to rank the answers. Within this pipeline, we also supply a baseline QA system inspired by DeepQA with solid performance and propose a reference experimental setup for easy future performance comparisons.

In this paper, we review the available open QA platforms, present the architecture of our pipeline, the components of the baseline QA system, and also analyze the system performance on the reference dataset.

1 Introduction

We consider the Question Answering problem — a function of unstructured user query that returns the information queried for. This is a harder problem than a linked data graph search (which requires a precisely structured user query) or a generic search engine (which returns whole documents or sets of passages instead of the specific information). The Question Answering task is however a natural extension of a search engine, as currently employed e.g. in Google Search [22] or personal assistants like Apple Siri, and with the high profile IBM Watson Jeopardy! matches [10] it has became a benchmark of progress in AI research. As we are interested in a general purpose QA system, we will consider an

"open domain" general factoid question answering, rather than domain-specific applications, though we keep flexibility in this direction as one of our goals.

Diverse QA system architectures have been proposed in the last 15 years, applying different approaches to information retrieval. A full survey is beyong the scope of this paper, but let us outline at least the most basic choices we faced when designing our system.

Perhaps the most popular approach in QA research has been restricting the task to querying structured knowledge bases, typically using the RDF paradigm and accessible via SPARQL. The QA problem can be then rephrased as learning a function translating free-text user query to a structured lambda expression or SPARQL query. [3, 5] We prefer to focus on unstructured datasets as the coverage of the system as well as domain versatility increases dramatically; building a combined portfolio of structured and unstructured knowledge bases is then again an obvious extension.

When relying on unstructured knowledge bases, a common strategy is to offload the information retrieval on an external high-quality web search engine like Google or Bing (see e.g. the **Mulder** system [15] or many others). We make a point of relying purely on local information sources. While the task becomes noticeably harder, we believe the result is a more universal system that could be readily refocused on a specific domain or proprietary knowledge base, and also a system more appropriate as a scientific platform as the results are fully reproducible over time.

Finally, a common restriction of the QA problem concerns only selecting the most relevant answer-bearing passage, given a tuple of input question and set of pre-selected candidate passages [23]. This Answer Sentence Selection task is certainly worthwhile as a component of a QA system but does not form a full-fledged system by itself. It may be argued that returning a whole passage is more useful for the user than a direct narrow answer, but this precludes any reasoning or other indirect answer synthesis on the part of the system, while the context and supporting evidence can be still provided by the user interface. Direct answer output may be also used in a more general AI reasoning engine.

In this paper, we present our open source Question Answering system **brmson**$^{\mathbf{TM}}$ **YodaQA**. This is not the only open source QA framework currently available, but we found our goals not entirely compatible with any of the other systems we investigated. Specifically, we aim to build a system that (A) provides an end-to-end, universal pipeline integrating different knowledge base paradigms in a modular fashion; (B) is domain flexible and could cater even to the long tail of rarer question subjects, therefore has minimum of fixed categories and hand-coded rules.

In contrast, the classic QA system **OpenEphyra** [21] operates on the basis of fixed question categories with hand-crafted rules, and puts emphasis on

querying web search engines. The **OAQA** initiative [12] has developed a basic QA framework, but does not provide an end-to-end pipeline and its usage of UIMA has in our opinion severe design limitations (see below). The **WatsonSim** system [13] has begun developing independently during the course of our own work and it works on Jeopardy! statements rather than questions.

Jacana [24, 25] is a promising set of loosely coupled QA-related methods and algorithms, focused on machine learning of textual entailment. It is not meant to be a full QA framework and using it as an end-to-end pipeline is not straightforward, but integration of the Jacana implementation as modules in YodaQA is our long-term plan.

OpenQA [1] is a recently introduced end-to-end QA pipeline platform also developed independently during the course of our work, and shares our goal of a common research platform in the field. However, the approach is very different, as OpenQA is more of a portfolio-style engine with mostly independent pipelines which have their candidate answers combined, while YodaQA emphasizes modularity on the pipeline stage level, with e.g. all answer producers sharing a common answer analysis stage.

The rest of the paper is structured as follows. We outline the general structure of our framework in Sec. 2. We then describe the current reference implementation of the pipeline components in Sec. 3. We propose a common experimental setup and analyze the system performance in Sec. 4. We conclude with a summary of our contributions and an outline of future extensions in Sec. 5.

2 YodaQA Pipeline Architecture

The goals for our system **brmson**$^{\mathbf{TM}}$ **YodaQA** are to provide an open source Question Answering platform that can serve both as scientific research testbed and a practical system. The pipeline is implemented mainly in Java, using the popular Apache UIMA framework [11]. Extensive support tooling is included within the package.

Unlike OAQA, in YodaQA each artifact (question, search result, passage, candidate answer) is represented as a separate UIMA CAS, allowing easy parallelization and easy leverage of pre-existing NLP UIMA components; we also put emphasis on aggressive compartmentalization of different tasks to interchangeable annotators rather than using UIMA just for high level flow and annotation storage.

The framework is split in several Java packages: **io** package takes care of retrieving questions and returning answers, **pipeline** contains classes of the general pipeline stages, **analysis** contains modules for various analysis steps, **provider** has interfaces to various external resources and **flow** carries UIMA helper classes.

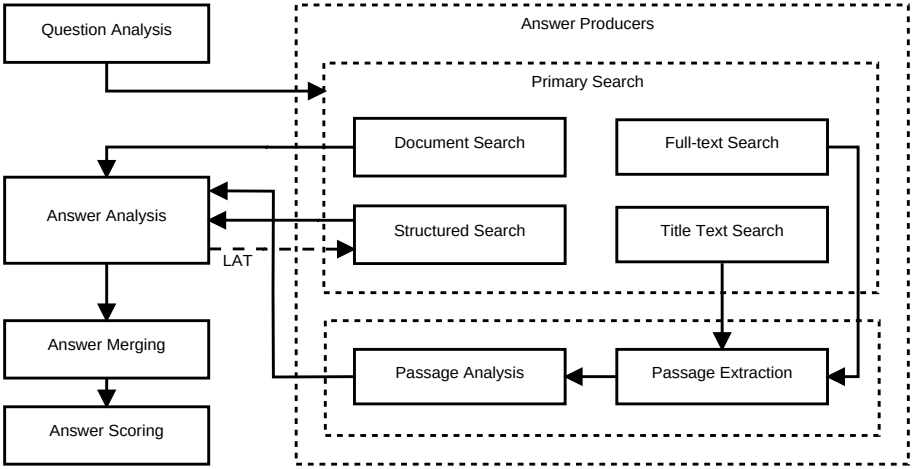


Fig. 1: The general architecture of the YodaQA pipeline. Present but unused final pipeline portions not shown

The system maps an input question to ordered list of answer candidates in a pipeline fashion, with the flow as in Fig. 1, encompassing the following stages:

- **Question Analysis** extracts natural language features from the question text and produces QA features based on them (clues, type, etc.).
- **Answer Production** generates a set of candidate answers based on the question. Typically, this happens by performing a **Primary Search** in the knowledge bases according to the question clues, and either directly using search results as candidate answers or filtering relevant passages from these (the **Passage Extraction**) and generating candidate answers from picked passages (the **Passage Analysis**).
- **Answer Analysis** generates various answer features based on detailed analysis (most importantly, type determination and coercion to question type).
- **Answer Merging and Scoring** consolidates the set of answers, removing duplicates and using a machine learned classifier to score answers by their features.
- **Successive Refining** (optional) prunes the set of questions in multiple phases, interjecting some extra tasks (evidence diffusion and gathering additional evidence).

The basic pipeline flow is much inspired by the DeepQA model of IBM Watson [9]. Throughout the flow, answer features are gradually accumulated and some results of early flow stages (especially the question analysis) are carried through the rest of the flow.

3 YodaQA Reference Pipeline

The particular Question Answering problem considered in the reference pipeline is finding a precise (narrowly phrased) answer to a naturally phrased English question, based on both unstructured (English Wikipedia, *enwiki*) and structured (DBpedia [17], Freebase [4]) knowledge bases.

In our pipeline, we build on existing third-party NLP analysis tools, in particular Stanford CoreNLP (Segmenter, POS-Tagger, Parser) [18, 6], OpenNLP (Segmenter, NER) [2] and LanguageTool (Segmenter, Lemmatizer).[1][2] NLP analysis backends are freely interchangeable thanks to the DKPro UIMA interface [8]. For semantic analysis, we also rely heavily on the WordNet lexicon [19].

For a practical illustration of the pipeline processing two particular example questions, please follow Fig. 2 and 3.

3.1 Question Analysis

During question analysis, produce a part-of-speech tagging and dependency parse of the question text, recognize named entities and, roughly inspired by the DeepQA system [16], heuristically generate several QA features: Clues, Focus, and LAT.

Clues represent keywords in the question that determine its content and are used to query for candidate answers. Clues based on different question components are assigned different weight (used in search retrieval and passage extraction, determined empirically) — in ascending other, all noun phrases, noun tokens and the selection verb (SV); the LAT (see below); named entities; the question sentence subject (determined by dependency parse). If the clue text corresponds to an *enwiki* article name or redirect alias, its weight is boosted and it is flagged as a **concept clue**.

Focus is the center point of the question sentence indicating the queried object. Six simple hand-crafted heuristics extract the focus based on the dependency parse. "name of —" constructions are traversed.

LAT (Lexical Answer Type) describes a type of the answer that would fit the question. This type is not of a pre-defined category but may be an arbitrary

[1] http://www.languagetool.org/

[2] Sometimes, different pipeline components default to different NLP backends to perform the same task, e.g. segmentation, based on empirically determined best fit.

Question Text	Who wrote Ender's Game?
Q. Analysis	**Focus:** who; **SV:** wrote; **LAT:** person
Question Clues	Ender's Game **(concept clue)**, wrote
Primary Search (DBpOnt.)	**author:** Orson Scott Card, **publisher:** Tor Books, ...
Primary Search (DBpProp.)	(ibid), **cover artist:** John Harris, **Caption:** 2005, ...
Primary Search (Freebase)	**Author** Orson Scott Card, **Characters** Valentine Wiggin, Hive Queen, ...
Primary Search (concepts)	*enwiki* Ender's Game
Primary Search (fulltext)	**Query:** `+("wrote" OR titleText: "wrote")^1.0 +("ender's Game" OR titleText:"ender's Game")^1.1 ("wrote ender's Game "~4)^2.1...`
	Found: Ender's Game (series), Ender's Game, Ender's Game (film), Ender's Game (comics), Jane (Ender's Game), List of Ender's Game series planets **Sample picked passages:** Elaborating on characters and plot lines depicted in the novel, Card later wrote additional books to form the Ender's Game series. ... Valentine wrote an essay here comparing the priestly class to the skeletons of small vertebrates some time before Speaker for The Dead.
Primary Search (titles)	Ender's Game, List of Ender's Game characters, Jane (Ender's Game), Ender's Game (short story), Ender's Game (film), Ender's Game (series) **Sample first passage:** "Ender's Game" is a 1985 military science fiction novel by American author Orson Scott Card.
Primary Search (document)	Ender's Game (series), Elisabeth Hirsch, Orson Scott Card, Ender in Exile, Worthing Inn, Jane (Ender's Game), ...
Orscon Scott Card	Structured primary search LAT *author* (Wordnet hypernyms *communicator*, *person*, *maker*, *creator*); DBpedia LAT *writer* (Wordnet hypernyms *communicator*, *person*); NER LAT *person* Successful type coercion match**!**, "sharp" (exact specific) match from NER LAT**!** **occurences:** 19**!**, **origins:** document title, concept**!**, first passage, noun phrase, named entity, multiple origins, **other:** adjecent to a concept clue mention, no clue text overlap**!**
Jane	Structured primary search LAT *character* (Wordnet hypernyms *imaginary being*, *creativity*, *person*, *message* and 36 others); NER LAT *person* Successful type coercion match**!**, "sharp" (exact specific) match from NER LAT**!** **occurences:** 4, **origins** document title, first passage, noun phrase, named entity, multiple origins, **other:** no clue text overlap**!**
Final Answers	**Orson Scott Card** (0.99), Neal Shusterman (0.96), Elisabeth Hirsch (0.96), American author Orson Scott Card (0.96), ..., List of Ender's Game series planets (0.94), Gavin Hood (0.94), Print (0.93), Jane (0.91), ...

Fig. 2: A sample of the pipeline process when (correctly) answering a training question. **!** indicates particularly distinguishing features

Question Text	What is the name of the famous dogsledding race held each year in Alaska?
Q. Analysis	**Focus:** name; **SV:** held; **LAT:** race (by Wordnet hypernym: *contest*, *event*, *biological group*, *canal* and 9 others)
Question Clues	name, Alaska **(concept clues)**, race, held, famous, dogsledding, race, year
Primary Search (DBpOnt.)	**area Total:** 1717854.0, **country:** United States, ...
Primary Search (DBpProp.)	(ibid), **West:** Chukotka, **Income Rank:** 4, ...
Primary Search (Freebase)	**Date Founded** 1959-01-03, **Capital** Juneau, ...
Primary Search (concepts)	*enwiki* Alaska, Name **Sample picked passages:** Various races are held around the state, but the best known is the Iditarod Trail Sled Dog Race, a 1150 mi trail from Anchorage to Nome (although the distance varies from year to year, the official distance is set at 1049 mi). ... Automobiles typically have a binomial name, a "make" (manufacturer) and a "model", in addition to a model year, such as a 2007 Chevrolet Corvette.
Primary Search (fulltext)	**Query:** `(''the name of the famous dogsledding race held each year in Alaska'' OR titleText:...)^2.7 + (''name'' OR titleText:''name'')^2.6...` **Found:** List of New Hampshire historical markers
Primary Search (titles)	Name of the Year, Danish Sports Name of the Year, List of organisms named after famous people, Alaska!, Alaska, Race of a Thousand Years **Sample first passage:** The 2000 Race of a Thousand Years was an endurance race and the final round of the 2000 American Le Mans Series season.
Primary Search (document)	List of New Hampshire historical markers
The 2000 Race of a Thousand Years	**Focus:** Race; DBpedia LAT *automobile race*, *auto race in australia*, *new year celebration*, *quantity* LAT Successful type coercion match**!**, "sharp" (exact specific) match**!** **occurences:** 1, **origins:** first passage, noun phrase, **other:** adjecent to an LAT clue mention**!**, containing clue text
Iditarod Trail Sled Dog Race	**Focus:** Race; DBpedia LAT *sport*, *sport in alaska*, *alaska*, *winter sport*, *attraction*; (N.B. no *race* LAT) Successful type coercion match, loose match by generalization of *attraction* to *social event***!** **occurences:** 1, **origins** passage by various clues, noun phrase, **other:** suffixed by clue text
Final Answers	The 2000 Race of a Thousand Years (0.97), −01–03 (0.94), List of New Hampshire historical markers (0.93), a binomial name, a "make" (manufacturer) and a "model", in addition to a model year, such as a 2007 Chevrolet Corvette (0.90), **the Iditarod Trail Sled Dog Race** (0.89), Various races (0.83), ...

Fig. 3: A sample of the pipeline process when (not quite correctly) answering a training question. **!** indicates particularly distinguishing features

English noun, like in the DeepQA system. [20] The LAT is derived from the focus, except question words are mapped to nouns ("where" to "location", etc.) and adverbs (like "hot") are nominalized (to "temperature") using WordNet relations.

3.2 Unstructured Answer Sources

The primary source of answers in our QA system is keyword search in free-text knowledge base, in our default setting the *enwiki*. While the information has no formal structure, we take advantage of the organization of the *enwiki* corpus where entity descriptions are stored in articles that bear the entity name as title and the first sentence is typically an informative short description of the entity. Our search strategies are analogous to basic DeepQA free-text information retrieval methods [7]. We use the Apache Solr[3] search engine (frontend to Apache Lucene).

Title-in-clue search [7] looks for the question clues in the article titles, essentially aiming to find articles that describe the concepts touched in the question. The first sentence of the top six articles (which we assume is its summary) is then used in passage analysis (see below).

Full-text search [7] runs a full-text clue search in the article texts and titles, considering the top six results. The document texts are split to sentences which are treated as separate passages and scored based on sum of weights of clues occuring in each passage[4][5]; the top three passages from each document are picked for passage analysis.

Document search [7] runs a full-text clue search in the article texts; top 20 article hits are then taken as potential responses, represented as candidate answers by their titles.

Concept search retrieves articles whose titles (or redirect aliases) exactly match a question clue. The first sentence and also passages extracted as in the full-text search are used for passage analysis.

Given a picked passage, the **passage analysis** process executes an NLP pipeline and generates candidate answers; currently, the answer extraction strategy entails simply converting all named entities and noun phrases to candidate answers. Also, object constituents in sentences where subject is the question LAT are converted to candidate answers.

[3] http://lucene.apache.org/solr/

[4] The *about-clues* which occur in the document title have their weight divided by four (as determined empirically).

[5] We also carry an infrastructure for machine learning models scoring candidate passages, but they have not been improving performance so far.

3.3 Structured Answer Sources

Aside of full-text search, we also employ structured knowledge bases organized in RDF triples; for each concept clue, we query for predicates with this clue as a subject and generate candidate answers for each object in such a triple, with the predicate label seeded as one LAT of the answer.

In particular, we query the DBpedia `ontology` (curated) and `property` (raw infobox) namespaces and the Freebase RDF dump. For performance reasons, we limit the number of queried Freebase topics to 5 and retrieve only 40 properties per each; due to this limitation, we have manually compiled a blacklist of skipped "spammy" properties based on past system behavior analysis (e.g. location's *people_born_here* or music artist's *track*).

3.4 Answer Analysis

In the answer analysis, the system takes a closer look at the answer snippet and generates numerous features for each answer. The dominant task here is type coercion, i.e. checking whether the answer type matches the question LAT.

The answer LAT is produced by multiple strategies:

- Answers generated by a named entity recognizer have LAT corresponding to the triggering model; we use stock OpenNLP NER models *date*, *location*, *money*, *organization*, *percentage*, *person* and *time*.
- Answers containing a number have a generic *quantity* LAT generated.
- Answer focuses (the parse tree roots) are looked up in WordNet and *instance-of* pairs are used to generate LATs (e.g. *Einstein* is *instance-of scientist*).
- Answer focuses are looked up in DBpedia and its ontology is used to generate LATs.
- Answers originating from a structured knowledge base carry the property name as an LAT.

Type coercion between question and answer LATs is performed using the WordNet hypernymy relation — i.e. *scientist* may be generalized to *person*, or *length* to *quantity*. We term the type coercion score **WordNet specificity** and exponentially decrease it with the number of hypernymy traversals required. Answer LATs coming from named entity recognizer and quantity are not generalized. We never generalize further once within the `noun.Tops` WordNet domain and based on past behavior analysis, we have manually compiled a further blacklist of WordNet synsets that are never accepted as coercing generalizations (e.g. *trait* or *social group*).

The generated features describe the origin of the answer (data source, search result score, clues of which type matched in the passage, distance-based score of adjacent clue occurences, etc.), syntactic overlaps with question clues and type coercion scores (what kind of LATs have been generated, if any type coercion succeeded, what is the WordNet specificity and whether either LAT had to be generalized).

3.5 Answer Merge-and-Score

The merging and scoring process also basically follows a simplified DeepQA approach [14]. Candidate answers of the same text (up to basic normalization, like *the-* removal) are merged; element-wise maximum is taken as the resulting answer feature vector (except for the `#occurences` feature, where a sum is taken). To reduce overfitting, too rare features are excluded (when they occur in less than 1% questions and 0.1% answers).

Supplementary features are produced for each logical feature — aside of the original value, a binary feature denoting whether a feature has *not* been generated and a value normalized over the full set of answers so that the distribution of the feature values over the answer has mean 0 and standard deviation 1. The extended feature vectors are converted to a score $s \in [0, 1]$ using a logistic regression classifier. The weight vector is trained on the gold standard of a training dataset, employing L2 regularization objective. To strike a good precision-recall balance, positive answers (which are about $p = 0.03$ portion of the total) are weighed by $0.5/p$.

3.6 Successive Refining

The pipeline contains support for additional refining and scoring phases. By default, after initial answer scoring, only the top 25 answers are kept with the intent of reducing noise for the next answer scoring classifier. Answers are compared and those overlapping syntactically (prefix, suffix, or substring aligned with sub-phrase boundaries) are subject to evidence diffusion where their scores are used as features of the overlapping answers. Another answer scoring would be then performed, and the answer with the highest score is then finally output by the system.[6]

However, while we have found these extra scoring steps beneficial with weaker pipelines (in particular without the clue overlap features), in the final pipeline configuration the re-scoring triggers significant overfitting on the training set and we therefore ignore the successive refining stage in the benchmarked pipeline.

[6] There is also experimental support for additional evidence gathering phase, where the top 5 answers are looked up using the full-text search together with the question clues, and the number and score of hits are used as additional answer features and final answer rescoring is performed. Nevertheless, we have not found this approach effective.

4 Performance Analysis

As we present performance analysis of our system, we shall first detail our experimental setup; this also includes discussion of our question dataset.

Then, we proceed with the actual results — we measure the *recall* of the system (whether a correct answer has been generated and considered, without regard to its score) and *accuracy-at-one* (whether the correct answer has been returned as the top answer by the system). We find this preferrable to typical information retrieval measures like MRR or MAP since in many applications, eventually only the single top answer output by the system matters; however, we also show the *mean reciprocial rank* for each configuration and discuss the rank distribution of correct answers.

Aside of the performance of the default configuration, we also discuss scaling of the system (extending the alotted answer time) and performance impact of its various components (hold-out testing).

4.1 Experimental Setup

Our code is version tracked in a public GitHub repository[7], and the experiments presented here are based on commit `f6c0cf6` (tagged as `v1.0`). The quality of full-text search is co-determined by Solr version (we use 4.6.0) and models of the various NLP components which are brought in by DKPro version 1.7.0. As for the knowledge bases, we use enwiki-20150112, DBpedia 2014, Freebase RDF dump from Jan 11 2015, and WordNet 3.1. Detailed instructions on setting up the same state locally (including download of the particular dump versions and configuration files) are distributed along the source code.

As a benchmark of the system performance, we use a dataset of 430 training and 430 testing open domain factoid questions. (For system development, exclusively questions from the training set are used.) This dataset is based on the public question answering benchmark from the main tasks of the TREC 2001 and 2002 QA tracks with regular expression answer patterns[8] and extended by questions asked to a YodaQA predecessor by internet users via an IRC interface. This dataset was further manually reviewed by the author, ambiguous or outdated questions were removed and the regex patterns were updated based on current data. We refer to the resulting 867 question dataset as `curated` and randomly split it to the training and testing sets.[9]

An automatic benchmark evaluation system is distributed as part of the YodaQA software package. The system evaluates the training and test questions in parallel and re-trains the machine learning models before scoring the answers.

[7] https://github.com/brmson/yodaqa

[8] http://trec.nist.gov/data/qa/2001_qadata/main_task.html and 2002 analogically.

[9] The remaining 7 questions are left unused for now.

Therefore, in all the modified system versions considered below, a model trained specifically for that version is used for scoring answers.

Our benchmark is influenced by two sources of noise. First, the answer correctness is determined automatically by matching a predefined regex, but this may yield both false positives and false negatives.[10] Second, during training the models are randomly initialized and therefore their final performance on a testing set flutters a little.

4.2 Benchmark Results

Benchmark results over various pipeline configurations are laid out in Fig. 4. Aside of the general performance of the system, it is instructive to look at the histogram of answer ranks for the default pipeline, shown in Fig. 5. We can observe that while accuracy-at-one is 32.6 %, accuracy-at-five is already at 52.7 % of test questions.

Pipeline	Recall	Accuracy-at-1	MRR	time
default	79.3 %	32.6 %	0.420	28.8 s
full-text scaling (6 → 12 fetched results)	82.3 %	34.0 %	0.430	50.0 s
passage scaling (3 → 6 picked passages)	81.2 %	31.4 %	0.415	43.5 s
document search scaling (20 → 40)	80.0 %	31.6 %	0.418	34.9 s
freebase scaling (5 → 10 topics, 40 → 80 properties)	79.8 %	31.6 %	0.416	29.8 s
full-text hold-out	49.5 %	20.9 %	0.277	5.8 s
structured hold-out	73.5 %	29.1 %	0.376	23.8 s
type coercion hold-out	79.3 %	22.1 %	0.314	30.0 s
concept clues hold-out	67.9 %	23.0 %	0.314	25.6 s
clue overlap test hold-out	79.3 %	29.5 %	0.390	30.1 s

Fig. 4: Benchmark results of various pipeline variants on the *curated* test dataset. MRR is the Mean Reciprocal Rank $|Q| \cdot \sum_{q \in Q} 1/r_q$, time is the average time spent answering one question

The information retrieval parameters of the default pipeline are selected so that answering a question takes about 30 s on average on a single core of AMD FX-8350 with 24 GB RAM and SSD backed databases. (Note that no computation pre-processing was done on the knowledge bases or datasets; bulk of the time per question is therefore spent querying the search engine and parsing sentences, making it an accurate representation of time spent on previously unseen questions.) By raising the limiting parameters, we can observe further

[10] For example numerical quantities with varying formatting and units are notoriously tricky to match by a regular expression.

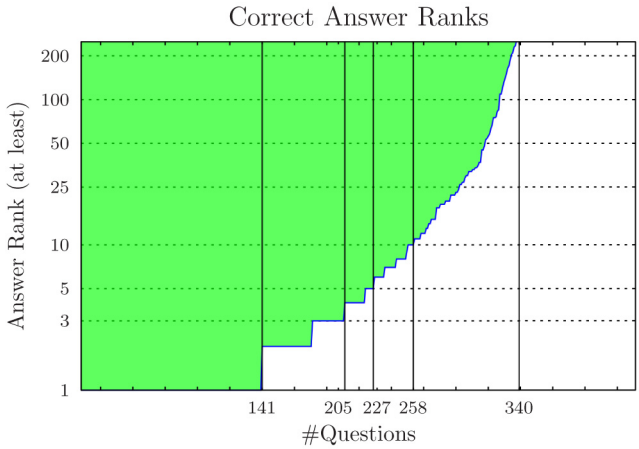


Fig. 5: Number of questions x that output the correct answer ranked at least y

recall increase, and in case of considering more full-text search results, also a solid accuracy improvement. Our system could therefore meaningfully make use of further computing resources.

We also benchmarked performance with various components of the pipeline disabled. We can see that the full-text and structured knowledge bases are complementary to a degree, but the full-text base is eventually a much stronger answer source for our system. Type coercion and detection of the concept clues in the question are both important heuristics for our QA system.

Comparison of performance across multiple systems is currently non-trivial, unfortunately, as there is no universally agreed experimental setup so far and not even published results on the TREC datasets from the years we use are readily available. OpenEphyra seemed to typically achieve accuracy-at-one of "above 25 %" on the TREC datasets including our years according to [21]. In the Answer Sentence Selection task [23], Jacana and similar textual entaliement systems are reported[11] to achieve MRR around 0.750 but this task represents a significant restriction upon the general end-to-end QA pipeline.

5 Conclusion

We have described a modular question answering system which can be used for effective mixing of both structured and unstructured knowledge bases, is domain-flexible and highly amenable to further extensions in various stages of

[11] See the **ACL Wiki** topic *Question Answering (State of the art)*.

its pipeline. We put emphasis on universal, machine learned methods and employ only a very limited amount of hand-crafted heuristics.

Meanwhile, the system is already demonstrating a reasonable open domain factoid question answering performance, being able to answer a third of the testing set questions correctly, over half of the questions in top five answers, and considering the correct answer for just about 80 % of the questions; we have also shown a head-room for further performance scaling by extending the available computational resources.

Our system is made available as open source under the highly permissive Apache Software Licence[12] and available for research collaboration on the GitHub social software hosting site. We hope for our system to become an universal research platform for testing of various question answering related strategies not only in isolation but also measuring their effect in a real-world high performance end-to-end pipeline. We also hope our work helps to clarify which of the numerous DeepQA contributions are most essential to a minimal working modern QA system.

5.1 Future Work

We present here just the first version of a system that could be improved in many desirable ways. The software platform itself would benefit in particular from a multi-threaded pipeline flow driver, a sophisticated user interface showing the generated answer in context and hypertext, and sped up benchmarking by caching information retrieval results (parsed picked passages) across runs.

Regarding algorithmic improvements, the most obvious candidates seem a more sophisticated answer extraction strategy (e.g. employing methods introduced in Jacana seems as a natural fit) and more reliable type coercion as a negative evidence source; we also hope that distributed representations might improve both areas. We feel that without further large effort in feature engineering, logistic regression is inadequate for scoring answers and we are seeing promising preliminary results from employing random forests instead.

Analysis and model training would be improved with larger benchmark datasets with more sophisticated correct answer verification. Some sub-tasks like type coercion would benefit from specialized datasets, and passage extraction scoring could be tuned on the Answer Sentence Selection dataset.

A robust heuristic for additional evidencing of most promising answers remains as an open problem in our system. While the natural idea of additional fulltext search combining the question and answer has been beneficial with a less sophisticated pipeline, it does not improve performance with our current featureful pipeline.

[12] The GPL licence applies in case Stanford CoreNLP components are employed.

Acknowledgements

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/194/OHK3/3T/13. The research was co-supervised by Dr. Jan Šedivý and Dr. Petr Pošík.

References

[1] openQA: Open source question answering framework.

[2] BALDRIDGE, J. The OpenNLP project. http://opennlp. apache.org/

[3] BERANT, J., CHOU, A., FROSTIG, R., LIANG, P. Semantic parsing on freebase from question-answer pairs. In EMNLP (2013), pp. 1 533–1 544.

[4] BOLLACKER, K., EVANS, C., PARITOSH, P., STURGE, T., TAYLOR, J. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (2008), ACM, pp. 1 247–1 250.

[5] BORDES, A., CHOPRA, S., WESTON, J. Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676 (2014).

[6] CHEN, D., MANNING, C. D. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014), pp. 740–750.

[7] CHU-CARROLL, J., FAN, J., BOGURAEV, B., CARMEL, D., SHEINWALD, D., WELTY, C. Finding needles in the haystack: Search and candidate generation. IBM Journal of Research and Development 56, 3.4 (2012), 6–1.

[8] DE CASTILHO, R. E., GUREVYCH, I. A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014 (Dublin, Ireland, Aug. 2014), N. Ide and J. Grivolla, Eds., Association for Computational Linguistics and Dublin City University, pp. 1–11.

[9] EPSTEIN, E. A., SCHOR, M. I., IYER, B., LALLY, A., BROWN, E. W., CWIKLIK, J. Making watson fast. IBM Journal of Research and Development 56, 3.4 (2012), 15–1.

[10] FERRUCCI, D., BROWN, E., CHU-CARROLL, J., FAN, J., GONDEK, D., KALYANPUR, A. A., LALLY, A., MURDOCK, J. W., NYBERG, E., PRAGER, J., ET AL. Building watson: An overview of the deepqa project. AI magazine 31, 3 (2010), 59–79.

[11] FERRUCCI, D., LALLY, A. UIMA: An architectural approach to unstructured information processing in the corporate research environment. Nat. Lang. Eng. 10, 3–4 (Sept. 2004), 327–348.

[12] FERRUCCI, D., NYBERG, E., ALLAN, J., BROWN, E., BARKER, K., CHU-CARROLL, J., CICCOLO, A., DUBOUE, P., FAN, J., GONDEK, D., ET AL. Towards the open advancement of question answer systems. Tech. rep., IBM Technical Report RC24789, Yorktown Heights, NY, 2009.

[13] GALLAGHER, S., ZADROZNY, W., SHALABY, W., AVADHANI, A. Watsonsim: Overview of a question answering engine. arXiv preprint arXiv:1412.0879 (2014).

[14] GONDEK, D., LALLY, A., KALYANPUR, A., MURDOCK, J. W., DUBOUÉ , P. A., ZHANG, L., PAN, Y., QIU, Z., WELTY, C. A framework for merging and ranking of answers in deepqa. IBM Journal of Research and Development 56, 3.4 (2012), 14–1.

[15] KWOK, C., ETZIONI, O., WELD, D. S. Scaling question answering to the web. ACM Transactions on Information Systems (TOIS) 19, 3 (2001), 242–262.

[16] LALLY, A., PRAGER, J. M., MCCORD, M. C., BOGURAEV, B., PATWARDHAN, S., FAN, J., FODOR, P., CHU-CARROLL, J. Question analysis: How watson reads a clue. IBM Journal of Research and Development 56, 3.4 (2012), 2–1.

[17] LEHMANN, J., ISELE, R., JAKOB, M., JENTZSCH, A., KONTOKOSTAS, D., MENDES, P. N., HELLMANN, S., MORSEY, M., VAN KLEEF, P., AUER, S., ET AL. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web (2014).

[18] MANNING, C. D., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S. J., MCCLOSKY, D. The stanford corenlp natural language processing toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2014), pp. 55–60.

[19] MILLER, G. A. WordNet: a lexical database for english. Communications of the ACM 38, 11 (1995), 39–41.

[20] MURDOCK, J. W., KALYANPUR, A., WELTY, C., FAN, J., FERRUCCI, D. A., GONDEK, D., ZHANG, L., KANAYAMA, H. Typing candidate answers using type coercion. IBM Journal of Research and Development 56, 3.4 (2012), 7–1.

[21] SCHLAEFER, N., GIESELMANN, P., SCHAAF, T., WAIBEL, A. A pattern learning approach to question answering within the ephyra framework. In Text, speech and dialogue (2006), Springer, pp. 687–694.

[22] SINGHAL, A. Introducing the knowledge graph: things, not strings. Official Google Blog, May (2012).

[23] WANG, M., SMITH, N. A., MITAMURA, T. What is the jeopardy model? a quasi-synchronous grammar for qa. In EMNLP-CoNLL (2007), vol. 7, pp. 22–32.

[24] YAO, X., VAN DURME, B., CALLISON-BURCH, C., CLARK, P. Answer extraction as sequence tagging with tree edit distance. In HLT-NAACL (2013), Citeseer, pp. 858–867.

[25] YAO, X., VAN DURME, B., CLARK, P. Automatic coupling of answer extraction and information retrieval. In ACL (2) (2013), Citeseer, pp. 159–165.

Prometheus – moderní monitorovací systém

Štefan Šafár

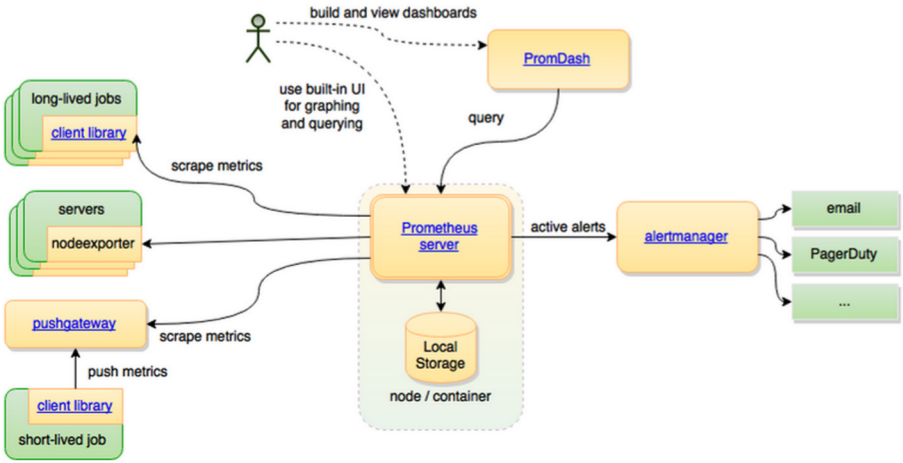
E-mail: stefan@safar.sk

1 Úvod

Historicky je většina správců zvyklá na systémy jako Nagios. V této oblasti to ale vypadá, že se vývoj zaseknul někde v 90. letech, a od té doby se toho moc nezměnilo. Existují sice modernější verze a forky, nicméně ty také trpí podobnými neduhy.

2 Co je Prometheus

Prometheus je v první řadě od základu jiný monitorovací systém, a tak je k němu nutné také přistupovat. Jedná se o několik komponent, které spolupracují. Architektura je na obrázku. Prometheus využívá pull model, což znamená, že si data aktivně stahuje ze všech monitorovaných serverů a komponent. Diskuse k tomu, proč byl vybrán tento model, se nachází na webu projektu.



Obr. 1

Data se stahují pomocí protokolu HTTP, přičemž se využívá buď obyčejný textový formát, nebo protobuf. Tento formát nám umožňuje velice snadno rozšiřovat software, který je možné pomocí promethea monitorovat.

Prometheus server

Ve svém jádru se jedná o systém na ukládání obrovského množství dat, kterým se v angličtině říká timeseries - kombinace timestampu a hodnoty. Prometheus využívá na skladování dat dvě části. Indexy jsou uloženy v LevelDB, samotná data pak ve vlastním formátu přímo na disk. Kromě démona na ukládání dat je součástí také webserver, který obsahuje jednoduchou webovku, kterou je možné použít na dotazování nad daty a jednoduchý přehled o monitorovaných systémech.

node_exporter

Tato komponenta zobrazuje typické hodnoty, které se povětšinou dají najít v /proc – využití CPU, paměti, disků, síťových karet a podobně.

push_gateway

Ne všechny systémy je možné scrapeovat přímo ze serveru. Typickým příkladem jsou krátko žijící procesy, například cron joby. Pro tyto je k dispozici push_gateway, která je můstkem mezi push a pull světem.

Další exportéry

Exportérů je k dispozici obrovské množství a to jak oficiálních, tak i neoficiálních. Jejich množství se pořád rozrůstá. Dnes už není problém monitorovat většinu běžného software pomocí promethea. Pro příklad uvádíme například haproxy, nginx, mysql, memcached, redis, rabbitmq, mongodb.

3 Instrumentace

Instrumentace je v oblasti IT relativně nový pojem. Jedná se o přidání jednoduchých metrik na důležitá místa kódu. Tyto metriky pak můžeme sledovat a odvozovat z nich informace o stavu služby.

Ukažme si to na příkladu RPC serveru. Typické cíle na instrumentaci jsou délky trvání jednotlivých metod, počty úspěšných a neúspěšných dotazů, počet aktuálních spojení a podobně. Obecně se dá doporučit, že při každé hlášce do aplikačního logu bychom měli inkrementovat nějaké počítadlo.

Prometheus výrazně podporuje instrumentaci. Poskytuje knihovny pro mnoho běžně používaných jazyků, které tuto činnost významně usnadňují. V knihovně pro jazyk python se jedná o jednoduché dekorátory kolem funkcí, případně counter.inc(). Kromě pythonu jsou k dispozici oficiální knihovny i pro go, javu a ruby, neoficiální pak i pro bash, haskell, node.js a C#.

Existuje také několik projektů, které už instrumentaci obsahují. Typicky to bývají projekty Googlu, jako etcd a kubernetes. Je to způsobeno tím, že prometheus vzniknul u bývalých zaměstnanců Google, kteří byli nespokojeni se současným stavem open source monitorovacích nástrojů.

4 Dotazování a zpracování dat

Jedním z důležitých aspektů promethea je používaní takzvaných labelů. Labely usnadňují práci s daty. Můžeme pomocí nich filtrovat pouze to, co nás zajímá, nebo seskupovat data pomocí některé z mnohých agregačních funkcí, které má dotazovací jazyk k dispozici.

Příklad surových dat:

```
node_cpu{cpu="cpu0",instance="there.org:9100",mode="idle"}    17914450.35
node_cpu{cpu="cpu0",instance="there.org:9100",mode="iowait"}     81386.28
node_cpu{cpu="cpu0",instance="there.org:9100",mode="system"}     47401.76
node_cpu{cpu="cpu0",instance="there.org:9100",mode="user"}      124549.65
node_cpu{cpu="cpu1",instance="there.org:9100",mode="idle"}    18005086.74
node_cpu{cpu="cpu1",instance="there.org:9100",mode="iowait"}     12934.74
node_cpu{cpu="cpu1",instance="there.org:9100",mode="system"}     44634.8
node_cpu{cpu="cpu1",instance="there.org:9100",mode="user"}       86765.05
```

Surová data nám ukazují počet sekund, které daný procesor strávil v daném módu od posledního bootu. Samy o sobě tyto data nemají moc vypovídající hodnotu. Můžeme ale data agregovat podle libovolného labelu, například instance a mode, a zároveň použít funkci rate, která počítá rychlost změny hodnoty za vteřinu.

sum by (instance, mode) (rate(node_cpu[1m]))

```
{instance="there.org:9100",mode="idle"}                          1.8464
{instance="there.org:9100",mode="iowait"}                        0.0217
{instance="there.org:9100",mode="system"}                        0.0211
{instance="there.org:9100",mode="user"}                          0.107
```

Tady vidíme už zajímavější data. Funkce *sum* zachovala námi žádané labely a agregovala všechny ostatní. Tím získáme přehled, kolik vteřin strávily všechny procesory na daném systému v poslední vteřině. Když už jsme spokojeni s výsledkem, stačí přepnout záložku na graph a uvidíme graf. Přehled funkcí a agregačních operátorů je zdokumentován na webových stránkách projektu.

Všechny takto vytvořené dotazy, které agregují data napříč servery, procesory, nebo kterýmkoliv labelem, můžeme zachovat a uložit pod vlastním názvem v konfiguraci. Tyto agregované hodnoty se pak ukládají stejně jako jakékoliv jiné hodnoty – můžeme si z nich nakreslit graf, použít je pro složitější výpočty nebo na jejich základě vytvořit alert. Důležité také je, že se ukládají přímo hodnoty, což nám nevadí díky úspornému ukládání na disk a umožňuje to rychlé dotazy na agregovaná data.

Další užitečnou součástí jsou takzvané konzole. Konzole jsou jednoduché HTML šablony, ve kterých můžeme používat výpočty hodnot a kreslení grafů přímo z promethea. Jsou uloženy jako soubory v podadresáři consoles. Můžeme si tak udělat přehledové weby s vzhledem a daty, které potřebujeme. Tyto weby nám pak zobrazuje přímo samotný prometheus server.

Promdash je separátní komponenta – jedná se o webovou aplikaci napsanou v node.js a rails, která umožňuje vytvořit přehledové dashboardy s grafy. Rozhrani je podobné systému grafana. Velikou výhodou promdash je možnost zobrazovat grafy z více serverů zároveň. Výsledné dashboardy pak obsahují pouze javascriptový kód, který stahuje data z jednotlivých instancí serveru a následně zobrazí grafy. Z tohoto důvodu nemusí mít promdash přímou viditelnost na servery, a ani rozsáhlý úložný prostor.

5 Alerty

Jazyk na psaní alertů je jenom mírným rozšířením dotazovacího jazyka. Vytvořit alert můžeme na všechno, na co se dokážeme dotázat nebo vytvořit graf. Můžeme porovnat hodnotu se statickou mezí, nebo s hodnotou před 1 dnem, nebo před týdnem, měsícem. Samozřejmě je možné nastavit i dobu trvání nechtěného stavu, po které se odešle alert. Dva další parametry jsou summary (typicky nadpis emailu) a description s detailním popisem závady. Můžeme použít také labely a hodnotu do obsahu alertu. Je možné přidat dodatečný label, například severity, dle kterého se pak orientuje Alertmanager.

Alertmanager je samostatná komponenta, které jednotlivé instance serveru posílají své alerty. Alertmanager je pak deduplikuje a dle nastavených pravidel odešle na některou z definovaných destinací. Podporuje kromě klasických komunikačních systémů jako email také modernější nástroje – Pushover, Hipchat, Slack, Flowdock a Pagerduty. Pokud nám to nestačí, je k dispozici generický HTTP post, kam můžeme napasovat jakýkoliv systém, který potřebujeme.

6 Nedostatky a možné problémy

Obecně je prometheus stavěný primárně pro rozsáhlejší a modernější nasazení. Nejvíc z něj budou těžit firmy, které používají model microservices. Prometheus

se nicméně dá nasadit i na běžnější infrastrukturu, čemuž vydatně napomáhají různé exportéry pro specifický software.

Škálování se řeší pomocí samostatných instancí, které jsou na sobě nezávislé. Můžeme mít jednu instanci pro každou službu, pro každou řadu racků, nebo jakékoliv jiné rozdělení, které dává smysl. Pomocí pravidel si pak agregujeme metriky a dáváme je k dispozici nadřazenému prometheovi, čímž se dá škálovat prakticky neomezeně.

Pro dynamické systémy jako Docker cloud je k dispozici možnost stahovat informace o cílech na monitorování z různých systémů, jako etcd, consul, marathon, kubernetes a zookeeper. Je také k dispozici možnost dynamicky načítat konfiguraci cílů ze složky na disku, případně pokaždé upravovat konfiguraci promethea a následně mu zaslat signál na znovunačtení konfigurace.

Největším současným nedostatkem je dle názoru autora nutnost ručně nastavit všechna agregační pravidla, alerty, konzole. Mnohé z nich jsou specifické pro každé konkrétní nasazení, ale repositář s příklady pro začátečníky by byl vhodným doplňkem.

7 Závěr

Prometheus je zatím ještě mladý, ale hodně ambiciózní projekt. Spojuje v sobě jak samotný monitorovací systém, tak i grafování a sledování trendů. Po vyřešení některých porodních bolestí by se mělo mnohé zlepšit, každopádně už teď je možné prometheus nasadit a s úspěchem používat. Autor by se po několika měsících s prometheem už nechtěl ke klasickým monitorovacím systémům nikdy vracet.