# T1

**1. Co si představíte pod pojmem mobilní zařízení?**

Mobilním zařízením rozumíme zařízení kapesních rozměrů, které má díky tomu typicky omezené rozměry obrazovky a miniaturní či jen softwarově emulovanou klávesnici. Typickými příklady jsou mobilní a chytré telefony, pagery, navigační zařízení, multimediální přehrávače, kamery a fotoaparáty, přenosné herní konzole, čtečky elektronických knih, tablety a phablety, netbooky, zařízení určená k nošení na těle a další specializovaná zařízení

**2. Jaká jsou specifika mobilních platforem?**

K typickým omezením mobilních zařízení patří rozlišení displaye dané jeho velikostí, omezené možnosti ovládání (často dotykovou obrazovkou s méně přesným zaměřením dotyku uživatele, čemuž musí být přizpůsobeno GUI), méně operační paměti, menší výkon procesoru (CPU), absence plnohodnotného grafického čipu (GPU) a souvisejících omezení (neplnohodnotné OpenGL ES, absence technologií jako je Nvidia CUDA, OpenCL, menší množina videoformátů s podporou hardwarové dekomprese na GPU), omezení instrukční sady procesoru, a podobně.

Dalším typickým problémem jsou omezení v síťové konektivitě, nutnost práce bez připojení k

externímu zdroji napájení a s tím související nutnost omezování spotřeby elektrické energie, menší knihovna systémových funkcí a volání (například knihovna Bionic místo plnohodnotné knihovny libc v rámci OS Android), a jiné způsoby tvorby GUI oproti aplikaci určené pro desktopové prostředí (obrazovky namísto oken).

**3. Jaké hlavní mobilní platformy znáte?**

V současnosti jsou 4 hlavní dominantní platformy chytrých telefonů podle zastoupení prodaných zařízení:

* Android
* iOS
* Windows (Phone, 8)
* BlackBerry

**4. Co je to fragmentace zařízení a fragmentace platforem a jaký je mezi nimi rozdíl?**

Namísto původního problému s fragmentací zařízení (různá zařízení různých výrobců se v rámci téže platformy chovala různým způsobem při provádění téhož kódu) se jako stále palčivější projevuje problém fragmentace platforem – různá zařízení mají různé platformy, lišící se podobou a možnostmi, poskytovanými API platformy, použitým programovacím

jazykem, mírou otevřenosti, vývojovými nástroji, způsobem distribuce aplikací a souvisejícím

ekosystémem.

**5. Je možný multiplatformní vývoj aplikací, nebo je nutné pro každou aplikaci vytvářet aplikaci zcela od začátku?**

**ANO!**

Je mozne vytvorit knihovny pro ruzne platformy a nasledne vyuzivat funkce jednotne

NEBO

vyuzit HTML5, CSS, JS

**6. Pro jakou z mobilních platforem jste už vyvíjeli aplikaci? Jaká jsou její specifika**

Android

Pros:

* The biggest market share
* Wider demographics
* Variety of devices
* Higher ad revenue
* Fewer requirements for development
* Faster review in Google Play
* Cheaper to feature on Google Play
* Easy publication
* More beta testers

Cons:

* Fragmentation
* More potential bugs
* Longer to develop (and more expensive)

1. **Popište mobilitu.**

Schopnost přesunu z místa A na místo B (např. studentská mobilita v rámci programu

Erasmus)

Přesun uzlu (node) mezi jednotlivými segmenty v rámci IP sítí, změna jeho umístění/lokace

(location)

V multiagentních systémech (dle definice FIPA – Foundation for Intelligent Physical

Agents) jako speciální služba mezi platformami (umístěními), které poskytují služby a

agenty (objekty), které jsou v daný okamžik umístěny v rámci dané platformy.

1. **Jak může být realizován přesun?**

Přesun je pak definován jako interakce mezi platformami/lokacemí a agenty/uzly. Samotný přesun je možné realizovat následujícími způsoby, které se liší způsobem synchronizace nutné pro migraci a předchozími podmínkami:

* Spontánní přesun (bez předchozí synchronizace)
* Přesun objektu
* Transport (doprava, transportation) – iniciováno umístěními
* Přesun po dohodě obou stran

V rámci přesunu je nutno vyřešit několik problémů:

* Zjištění pozice

◦ Reaktivní přístup – zjišťujeme v okamžiku, kdy je nutné s objektem komunikovat

◦ Proaktivní přístup – průběžně si udržujeme informace o umístění všech objektů

◦ Oba přístupy lze kombinovat – např. známe přibližnou pozici objektu a přesnou

dohledáváme v případě, že s ním potřebujeme komunikovat

* Detekce přesunu
* Určení nejlepší komunikační cesty
* Výběr servisní buňky v buňkových sítích
* Signalizace aktualizací – změny umístění, informace o dostupnosti zařízení, ...
* (Znovu)vytvoření cesty k zařízení, aby bylo dostupné na novém umístění.

Ideálem mobilního využití je AOAC – always on, always connected – zařízení je vždy v provozu a vždy připojeno k síti. Někdy se také ideál používání označuje zkratkou AAA:

* Anytime – kdykoliv dostupné
* Anywhere – použitelné kdekoliv
* Any Device – na jakémkoliv zařízení

1. **Jaké formy mobility znáte, popište rozdíly mezi nimi.**

Bezdrátová mobilita

Bezdrátová mobilita nám umožňuje pohyb v dané oblasti, real-time komunikaci kdykoliv a

kdekoliv je zařízení zapnuto a v dosahu sítě, a její využívání vedlo k rozvoji celé řady mobilních zařízení (PDA, chytré telefony, tablety, navigace s on-line aktualizacemi dopravní situace a podobně). Mezi její nejznámější podoby patří sítě standardu 802.11 a buňkové sítě.

Můžeme dosahovat různého stupně bezdrátové mobility – od komunikace s jediným přístupovým bodem, přes připojování se k různým přístupovým bodům během přesunu (bez zachování existujících spojení) až po nepřerušované poskytování služeb a zachovávání existujících spojení nebo hovorů.

U bezdrátových sítí se rovněž setkáváme s pojmeme mobility v rámci kampusu (campus mobility) a roamingem. Mobilita v rámci kampusu je typicky omezena jednou administrativní doménou. Může se jednat o kampus univerzity, skupinu budov hotelového komplexu, nemocnici či jinou podobnou oblast. Nedochází zde k předávání uživatele v rámci globálního Internetu, což usnadňuje implementaci celého řešení. Je tak možné řešit mobilitu na úrovni 2. vrstvy ISO OSI modelu a jednotlivé přístupové body (AP) poskytují pro dané ESSID stejnou jednotnou VLAN. Uživatel tak může přecházet mezi přístupovými body, aniž by musel měnit svou IP adresu a pokračovat v komunikaci.

Oproti tomu roaming spočívá ve využívání služeb navštívené sítě mimo domovskou síť, kde je uživatel registrován. Zařízení zůstává připojeno k síti díky správě mobility, autentizaci, autorizaci a zaznamenávání aktivit uživatele (AAA). Ty mohou být zajištěny pomocí SIM karty, na základě vydaných certifikátů, kombinací uživatelského jména a hesla, nebo kombinací těchto metod. Příkladem může být roaming v buňkových sítích a akademický projekt Eduroam.

Nomadicita

Nomadicita (kočující mobilita) spočívá v možnosti přesunu uživatele mezi lokalitami resp.

zařízeními, přičemž během přesunu typicky k samotné komunikaci nedochází. Komunikace je ukončena a znovunavázána jako důsledek přesunu. Nomadické prostředí by mělo být vůči uživateli transparentní bez ohledu na umístění, použitém zařízení resp. platformě (PC v kanceláři, tablet, sdílené PC na jiné pobočce, apod.), rychlosti připojení a tom, zda je uživatel na cestě. Na rozdíl od mobilního uživatele, který je k síti připojen stále a využívá nejlepší z dostupných připojení nomadický uživatel (tzv. technomád) typicky nevyžaduje služby během přesunu, využívá lokální infrastruktury v cílovém místě a používá potenciálně sdílená zařízení, na kterých má jedinečnou session.

Bezešvá mobilita

Bezešvá mobilita je jedním z ideálů, kdy uživatel nevnímá to, že došlo k migraci (i mezi

technologiemi). Může se jednat o přesun se zařízením, ale také o přesun mezi zařízeními.

Základními příklady může být přechod mezi typem připojení k Internetu (buňková síť vs. WiFi) bez přerušení komunikace (zachování IP adresy), přechod mezi 2G a 3G sítí bez přerušení probíhajícího hovoru a v budoucnu jsou zvažovány rovněž komplikovanější varianty:

* přechod mezi technologiemi (video)hovoru (bezdrátový telefon využívá pevná linku v

kanceláři, po odchodu se přepne na mobilní síť, po příchodu do vozidla je automaticky

převzat handsfree integrovaném ve vozidla a po příjezdu domů může pokračovat za využití

IP telefonie, to vše bez přrušení.

* Přehrávání multimediální obsah je přerušeno po opuštění místnosti a pokračuje v jiné

místnosti, kam se uživatel přesune.

Ubicomp

Ubicomp (ubiquitous computing), všudypřítomná zařízení je trend, při němž se předpokládá

rozšíření všudypřítomných zařízení, která obklopují uživatele, jenž s nimi interaguje běžným

způsobem. Tato zařízení mohou být součástí objektů denní potřeby (včetně komponent inteligentní domácnosti) a budou většinou tvořena malými, levnými vzájemně propojenými zařízeními s přirozenou interakcí. Podle rozměrů se tato zařízení dělí na:

* Prach (Dust) – žádný vizuální výstup, [nm→mm]
* Zařízení určená k nošení na těle (tabs), [cm]
* Příruční zařízení (Pads) [dm]
* Tabule (Boards) – interaktivní displeje [m]

Kromě toho se zvažují další dva typy zařízení:

* Kůže (Skin) – ohebné 2D displeje a plochy, které jsou součástí oblečení a bytových textilií, většinou založené na OLED technologiích.
* Hlína (Clay) – 3D objekty, vypadající jako běžné fyzické objekty, se kterými se interaguje prostřednictvím dotykového rozhraní.

Speciálními aplikacemi z této oblasti jsou zařízení s haptickou zpětnou vazbou nebo internet věci

1. **Čím se liší bezdrátová mobilita v rámci kampusu a roaming?**

viz vyse

1. **Jaký je rozdíl mezi mobilitou a nomadicitou?**

viz vyse

**Proč je nutné adaptovat aplikace pro mobilní prostředí?**

Tvůrci aplikací pro mobilní zařízení stojí před zásadním dilematem. Na jednu stranu je mobilní

zařízení považováno za méně spolehlivé (může být snadněji ukradeno, ztraceno či zničeno) a má méně prostředků než kancelářské PC, na druhou stranu se často můžeme pohybovat v oblasti, kde není dostupné žádné či pouze velmi pomalé a nespolehlivé připojení k Internetu. Nespolehlivost zařízení by bylo možné řešit tak, že mobilní zařízení reprezentuje pouze prezentační vrstvu aplikace, realizuje tenkého klienta a většina činností je řešena na serveru. Druhý přístup vede k tvorbě komplexních aplikací, které jsou schopny pracovat na mobilním zařízení zcela či z části nezávisle na serveru v případě slabé a přerušované konektivity, či v případě absence konektivity jako takové. V reálných aplikacích je často nutno volit kompromis mezi těmito přístupy.

**Vysvětlete pojmy silná, slabá a přerušovaná konektivita.**

Můžeme být připojeni v režimu plné konektivity, kdy nejsme jakkoliv omezováni, v režimu

slabé konektivity, kdy jsou možnosti komunikace zásadně omezeny (např. komunikace prostřednictvím technologie GPRS), v režimu přerušované konektivity, kdy dochází ke

krátkodobým výpadkům spojení (způsobeným např. přechodem mezi technologiemi WiFi a

datovými přenosy v buňkvých sítích), nebo pracovat zcela v odpojeném režimu.

**Je mobilní zařízení vhodné jako jediné úložiště dat? Proč?**

**Čím je specifická práce v odpojeném režimu?**

**Jak lze adaptovat aplikaci pro použití v mobilním prostředí?**

* Dynamická distribuce úkolů a úloh mezi prvky v pevné části sítě a mobilními uzly, např.

◦ Při silné konektivitě jsou šetřeny lokální prostředky a většina činností je realizována pevnou sítí

◦ V odpojeném (off-line) režimu pracují mobilní uzly autonomně

* Přizpůsobení kvality dat na mobilním uzlu dostupným prostředkům (fidelity).

**Jaké výhody a nevýhody má podpora na úrovni OS resp. samotné aplikaci?**

1. Transparentní z hlediska OS i aplikace

– zde spadají klasické aplikace bez úprav pro použití v mobilním prostředí, použit je klasický model klient-server

2. Podpora v OS, transparentní z hlediska aplikace

– je možná práce v odpojeném režimu,

použit je model klient-agent-server nebo klient-proxy-server

3. Podpora v aplikaci, transparentní z hlediska OS

– použit je dynamický klient-server nebo lze využít mobilních agentů

4. Podpora v aplikaci i v OS

nejlepší a nejhůře realizovatelná varianta. Je nutné koordinovat správu zdrojů z aplikace a využívat API OS. Vzhledem k náročnosti úprav se vyplatí spíše u tvorby nových aplikací. OS může poskytovat informace o kvalitě, úrovni a dostupnosti zdrojů (např. použité datové připojení, jeho typ, stav, přenosová rychlost), pomocí zasílání zpráv může aplikaci také informovat o případných změnách ve stavu těchto zdrojů. Agilní aplikace zvládají přijímat události asynchronně a reagují na ně odpovídajícím způsobem, přičemž příjem a zpracování události může být oddělen. Typicky existuje centrální bod, pro správu prostředků a autorizaci požadavků aplikací. Příkladem může být nastavení rozsahu požadované přenosové rychlosti. Pokud OS zjistí, že přenosová rychlost klesla pod dolní mez nebo naopak překročila mez horní, aplikace je notifikována a může přizpůsobit své chování (například automaticky změnit požadovanou kvalitu a rozlišení přehrávaného videa).

**Co je to věrnost dat a co lze s daty dělat v případě slabé konektivity?**

Věrnost dat (fidelity), tedy míra s jakou prezentovaná data odpovídají referenční kopii má více dimenzí:

◦ Konzistence dat (univerzální dimenze)

◦ Bezeztrátová vs. ztrátová komprese a její úroveň

◦ Kvalita obrazu (video, obrázky) – počet snímků za sekundu, kvalita snímků, další

nastavení MPEG/JPEG komprese, rozlišení, ...

◦ Vzorkovací frekvence a bitová hloubka vzorků u zvuku

**Čím se liší Push a Pull přístup při získávání dat?**

Pull přístup stahuje data ze serveru na základě požadavku klienta

Push přístup doručuje data klientům na základě akce serveru (callback, broadcasting),

typicky lze takto zaslat potřebná data celé skupině klientů

Klient-server

V tomto modelu je vidět mobilní uzel, který požaduje data od serveru v pevné síti. I u tohoto modelu může být už síť alespoň do určité míry optimalizována. Namísto jednoho serveru můžeme mít více serverů nebo CDN (content-delivery network), přičemž klient se připojuje k nejbližšímu/nejvhodnějšímu z nich. Server je vybrán klientskou aplikací nebo automaticky, například poskytnutím vhodné DNS odpovědi, využitím anycastové adresy či podobnou technikou.

Klient-agent-server

V tomto modelu je k dispozici agent, který zastupuje klienta vůči serveru (agent v pevné síti), server vůči agentovi) nebo dvojici agentů). Samotný agent může být implementován jako samostatný software, nebo komponenta klientské či serverové aplikace. Veškerá komunikace klienta se serverem pak prochází přes agenta resp. dvojici agentů.

Agent na straně serveru zajišťuje předávání zpráv a jejich kešování (ukládání do fronty) v případě nedostupného klienta. Je možné vytvořit obecného agenta, který bude nahrazovat celého klienta (proxy), nebo agenty pro jednotlivé typy služeb (databáze, souborový systém, prohlížení webu, ...). Pro každý typ služby pak většinou postačuje jeden agent. Agent může nahrazovat jednoho či více klientů.

Zároveň může agent na straně serveru optimalizovat data přenášená ke klientovi, zejména v režimu slabé konektivity, modifikovat je (přeuspořádat podle důležitosti, aplikovat ztrátovou kompresi pro vhodná data), řetězit více odpovědí do jediné a upozorňovat klienta na výskyt předdefinovaných událostí. V extrémním případě může agent poskytovat serveru některé služby, běžně poskytované klientem, spouštět/ukončovat služby na mobilním uzlu a nahrazovat funkcionalitu mobilního uzlu, tam, kde by jeho využití znamenalo velké množství přenesených dat.

Samotný agent může být umístěn co nejblíže serveru (zejména při velkém množství přenášených dat), co nejblíže nejvyšší koncentraci klientů, nebo přímo na okraji pevné sítě na přístupovém bodě nebo související části komunikační infrastruktury. V takovém případě může využít informace o bezdrátové síti, poskytovat personalizované informace klientovi, případně využívat speciální protokol na úrovni spojové vrstvy OSI referenčního modelu pro komunikaci s klientem (pak se ale musí přemisťovat spolu s klientem a musí být zajištěna vzájemná důvěryhodnost agentů a přístupových bodů, což je v praxi jen velmi obtížně realizovatelné. V případě přesunu agenta je navíc nutné predikovat budoucí umístění klienta.

Výhodou tohoto modelu je snížení vlivu slabé konektivity a horší spolehlivosti bezdrátového

připojení díky existenci agenta v pevné síti, a rozdělení interakce na část mezi klientem a agentem a komunikaci mezi agentem a serverem. Každý z těchto částí může mít vlastní, specializovaný, protokol a mohou fungovat do velké míry nezávisle. Nevýhodou je to, že stále není možná práce klienta v odpojeném režimu, data, zasílaná klientem nejsou optimalizována a pro interakci s agentem je nutné změnit implementaci klienta (resp. nastavení v případě transparentního agenta) V případě nahrazení celého mobilního uzlu náhradníkem (proxy), je možné nastavovat globální omezení a parametry (např. míru věrnosti dat) pro konkrétní uživatele a jejich aplikace. V takovém případě typicky proxy prochází veškerá komunikace z a do mobilního uzlu. Tento přístup je používán jako jeden z modelů služeb v mobilních Ad Hoc sítích a umožňuje návrh tenkých klientů pro mobilní uzly s omezenými zdroji.

V případě lokálního agenta na straně klienta je výhodou možnost práce v odpojeném

režimu, hromadění dat na mobilním uzlu, poloautomatizovaná reintegrace dat se serverem po obnovení konektivity, emulace zásadních funkcí serveru a ve výsledku pak také zvýšení dostupnosti a umožnění pokračovat v práci za ztížených podmínek. Agent na straně klienta zastupuje server po dobu jeho nedostupnosti. V odpojeném režimu je možné poskytnout:

* Tenkou verzi serveru, která zpřístupňuje data, replikovaná na mobilním uzlu, synchronizuje kopie dat po (opětovném) připojení na server, a řeší případné konflikty v datech, které vznikly v odpojeném režimu.
* Agenta, který bude zachytávat požadavky na (nedostupný) server, bude je kešovat a vrátí data až po opětovném připojení, poté co předá požadavky serveru a obdrží odpovědi.

Pokud data nejsou k dispozici v keši nebo dojde k timeoutu při čekání na nedostupný server, vrací agent většinou chybu klientské aplikaci.

V případě modelu s dvojicí agentů je nutné vytvářet relativně náročné řešení, provádět

změny na straně klienta a serveru a samotní agenti budou složití a pro každou aplikační doménu je potřeba jedné dvojice agentů. Na druhou stranu tento model přináší řadu výhod – lze provádět velké množství optimalizací, přesouvat výpočet/služby mezi agenty dle aktuální potřeby, jasně oddělit zodpovědnost klientského a serverového agenta a vytvořit vysoce optimalizovaný komunikační protokol mezi dvojicí agentů včetně automatizované redukce vernosti dat.

Dynamický model klient-server

V tomto modelu je možné server (nebo jeho tenkou variantu) dynamicky přesouvat mezi klienty. Pokud tedy klienti nemohou komunikovat s pevnou sítí, ale mezi sebou ano, jeden z nich převezme roli serveru. Je tak možné náhradníka serveru dynamicky vytvářet a přesouvat, a optimalizovat síť s ohledem na co nejvyšší dostupnost a výkon.

.

Architektura tohoto systému je distribuovaná, je nutno replikovat informace mezi uzly, systém je odolný vůči selhání a je možno optimalizovat komunikaci mezi uzly.

Peer-to-peer model

Peer-to-peer model je posledním z popisovaných modelů. Je založen na tom, že každý

mobilní uzel může zároveň vystupovat v roli klienta i serveru. Samotná implementace je díky tomu výrazně náročnější, úlohy a zátěž mohou být distribuovány mezi uzly sítě, kteří jsou rovnocennými partnery v komunikaci v rámci distribuované aplikace. Tímto způsobem je možné řešit napříkad distribuci velkých souboru.

1. **Jakými fázemi prochází systém, který podporuje práci v odpojeném režimu?**

Pro podporu práce v odpojeném režimu přechází operační systém mezi následujícími stavy:

1. Hromadění dat (hoarding) – probíhá typicky v režimu silné konektivity, položky potřebné pro další práci jsou (před)načteny do mobilního zařízení. Oproti přednaíčítání dat, která budeme brzy potřebovat (prefetching) je nadhodnocena potřeba dat, která bude klient potřebovat a je nutné je stáhnout.
2. Práce v odpojeném režimu

– pracuje se pouze s lokálně dostupnými daty. Požadavky na další data mohou být uloženy do fronty požadavků a asynchronně zpracovány po opětovném připojení

3. Reintegrace

– aktualizovaná data z mobilního zařízení jsou reintegrována na serveru

(přehrajeme na něm log operací z mobilního zařízení).

1. **Lze v odpojeném režimu modifikovat data a jak jsou řešena nedostupná data?**

odpojeném režimu musíme řešit co s požadavky na lokálně nedostupná data – aplikace v takovém případě musí buď přerušit své provádění, nebo vykonávat činnosti, které nejsou na těchto datech závislé. Dalším problémem práce v odpojeném režimu je aktualizace dat. Zde lze využít jeden z následujících přístupů:

1. Pesimistický přístup

– aktualizace je prováděna pouze na jednom místě (používáme

zamykání nebo check-in a check-out dat).

2. Optimistický přístup

– data lze aktualizovat na více místech (mohou vzniknout konflikty)

1. **Co je to hromadění dat?**

Při hromadění je možné data buď relokovat (v tom případě nejsou dostupná dalším uzlům) nebo replikovat či kešovat (vznikají samostatné kopie, ale je zde riziko konfliktů). Typ hromaděných objektů závisí na aplikační oblasti a jejím datovém modelu. Data lze hromadit buď periodicky, nebo před plánovaným přechodem do odpojeného režimu. Problémem je ovšem určení, jaká data budeme v budoucnu potřebovat. Nejjednodušší z hlediska aplikace je nechat výběr na uživateli, je možné ovšem rovněž analyzovat předchozí požadavky, jejich četnost a pořadí, definovat uživatelské či aplikační profily a v nich hledat vzory chování při práci s daty.

1. **Odhadněte, kdy se vyplatí uložit objekt do keše.**
2. **Jaké známe druhy chyb?**

Obecně může dojít ke dvěma základním typům selhání:

1. Závažná chyba

(hard failure) – řešena kontrolními body umístěnými v pevné síti (hard

checkpoints), typicky jako následek resetu do továrního nastavení, destrukce či ztráty

mobilního zařízení a podobně.

2. Méně závažná chyba

(soft failure) – lze řešit pomocí kontrolních bodů uložených lokálně na

mobilním zařízení (soft checkpoints), bez nutnosti komunikace (navíc lze tyto kontrolní

body využívat i během odpojeného režimu. Tento typ chyby typicky nastává při pádu

aplikace, samovolném restartu zařízení či nekorektním vypnutí zařízení z důvodu vybití

baterie.

1. **Co je to kontrolní bod (checkpoint)?**

Velký důraz je kladen na konzistentní globální kontrolní body. Důležité je umístění, kam lze

kontrolní body ukládat. Pokud jsou ukládány v pevné síti co nejblíže mobilnímu zařízení, je vhodné plánovat, kde bude uložen další kontrolní bod. Pokud má mobilní uzel dostatečně spolehlivé perzistentní úložiště, bude se s vyšší pravděpodobností účastnit procesu tvorby kontrolních bodů.

Granularita kontrolních bodů a frekvence jejich tvorby pak závisí také na režimu konektivity, ve

kterém se uzel nachází. Pro možnost obnovy dat na zařízení k danému kontrolnímu bodu musí být zachován obnovitelný konzistentní globální stav (recoverable consistent global state).

1. **Jaké jsou přístupy k tvorbě kontrolních bodů?**

Z hlediska typu se protokoly pro tvorbu kontrolních bodů dělí na:

* Koordinované protokoly

– účastnící musí koordinovat své lokální kontrolní body aby bylo

dosaženo obnovitelného konzistentního globálního stavu. Dosahují toho výměnou řídících

zpráv (může zde ale nastat problém s dohledáním konkrétního uzlu, jeho přesunem jinam, a

nepřístupností lokálního kontrolního bodu, pokud je uzel momentálně odpojený.

* Nekoordinované protokoly

– každý účastník si vytváří kontrolní body nezávisle, ale během

obnovy dat je nutno koordinovat jednotlivé kontrolní body, aby byl vytvořen globální

kontrolní bod. V průběhu tvorby není nutné předávat zprávy, ale na zařízeních musí být

uloženo více informací, a při obnově dat (soft failure) dochází k výměně většího množství

zpráv.

1. **Čím se liší koordinované a nekoordinované protokoly?**

Z hlediska typu se protokoly pro tvorbu kontrolních bodů dělí na:

* Koordinované protokoly

– účastnící musí koordinovat své lokální kontrolní body aby bylo

dosaženo obnovitelného konzistentního globálního stavu. Dosahují toho výměnou řídících

zpráv (může zde ale nastat problém s dohledáním konkrétního uzlu, jeho přesunem jinam, a

nepřístupností lokálního kontrolního bodu, pokud je uzel momentálně odpojený.

* Nekoordinované protokoly

– každý účastník si vytváří kontrolní body nezávisle, ale během

obnovy dat je nutno koordinovat jednotlivé kontrolní body, aby byl vytvořen globální

kontrolní bod. V průběhu tvorby není nutné předávat zprávy, ale na zařízeních musí být

uloženo více informací, a při obnově dat (soft failure) dochází k výměně většího množství

zpráv.

# T2

1. Popište, co jsou to transakce v odpojeném režimu (isolation-only transactions)
2. Jaké problémy přináší práce v odpojeném režimu u souborových systémů a jak je řešíme?
3. Jaké problémy přináší práce v režimu slabé konektivity u souborových systémů a jak je řešíme?
4. Co je to ASR? Je jeho použití vhodné?
5. Jaké problémy přináší práce v odpojeném režimu u databázových systémů a jak je řešíme?
6. Jaké problémy přináší práce v režimu slabé konektivity u databázových systémů a jak je řešíme?
7. Jaké problémy přináší práce v odpojeném režimu u webových stránek a jak je řešíme?
8. Jaké problémy přináší práce v režimu slabé konektivity u webových stránek a jak je řešíme?
9. Co je to destilace a zjemnění (Distillation & Refinement)?
10. Jak funguje rozdílový přístup u kešovaných objektů?
11. K čemu je vhodný broadcasting?
12. Co je to zpětný kanál a k čemu jej využíváme?
13. K čemu využijeme index?
14. Jaké synchronní a asynchronní metody zneplatňování obsahu keše znáte?
15. Čím je popsán systém, využívající mobilitu kódu?
16. Jaké mechanismy mobility znáte?
17. Jaká jsou návrhová paradigmata mobility kódu a jak se mezi sebou liší?
18. Uveďte příklady aplikačních domén mobility kódu a platforem pro ně.
19. Jaké jsou výhody a nevýhody použití mobilních agentů, můžete uvést příklady agentů?
20. Jak lze omezit spotřebu CPU, co je to DFS, DVS a governor a proč se používají?
21. Co nejvíce spotřebovává energii zařízení a jak lze spotřebu omezit?
22. Jaké typy akumulátorů se podle Vás v mobilních zařízeních používají a proč?

# T3

1. **Jaké metody určování polohy jsou možné na základě radiového signálu? Jaký je rozdíl mezi ToA a TDoA?**

K určení polohy lze použít následující techniky, resp. jejich kombinace:

1. Měření úhlů, měření azimutu
2. Měření vzdálenosti
3. Měření doby šíření signálu
4. Měření rozdílu v čase příchodu signálu

nebo speciální metody (např. měření vzdáleností od překážek pomocí ultrazvuku), mapování

prostoru a navigace v něm pomocí LIDARu.

Doba šíření signálu – Time of Arrival (ToA) - jak dlouho se signál šíří

z vysílače na přijímač (doba šíření elektromagnetického signálu médiem)

Rozdíl v době šíření signálu – Time Difference of Arrival (TDoA) – signál z hlavní (master) a vedlejších (slave) stanic dorazí v různých časech.

1. **Jak funguje lokalizace v mobilních sítích – co je to Cell ID, RX level, TA, AoA, E-OTD a jak je lze využít?**

**3. Co je to GNSS, z jakých částí (segmentů) se skládá a jaké GNSS znáte? Co je to augmentační systém?**

* Kosmický segment
* Řídící a kontrolní segment
* Uživatelský segment
* Navstar GPS
* Glonass
* Compass (BeiDou-2)
* Galileo

augmentační systémy, které mají jejich přesnost zvyšovat. Protože jsou převážně určeny pro pokrytí určité oblasti, využívají augmentační systémy ke své činnosti geostacionárních statelitů

**4. Jaké chyby mohou ovlivnit určení polohy?**

chyby při měření času a okamžiku příchodu signálu, vlivy atmosféry (zejména ionsoféry u GNSS), vícecestným šířením signálu, chybami v určení polohy vysílačů (špatné efemeridy u GNSS, chyby hodin na vysílači), geometrickou chybou či myslně vnášenou chybou (např. Selective availability)

**5. Jaký je rozdíl mezi vektorovou a rastrovou mapou?**

Dlaždice rastrové mapy jsou de facto obrázky s definovanou pozicí, které nemusíme upravovat a vykreslíme je tak velmi rychle. Oproti tomu vektorové mapy obsahují geometrické tvary, které jsou do mapy vykreslovány.

**6. Jaké projekce a souřadné systémy znáte?**

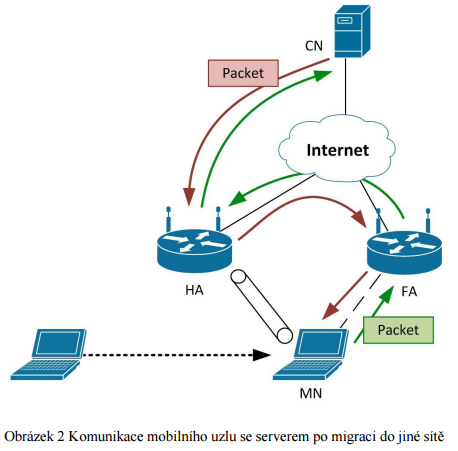
Mercatorovo zobrazení nebude vhodné pro severní oblasti Norska

3 souřadnými systémy

* WGS-84 (World Geodetic System)
* UTM (Universal transverse Mercator)
* S-JTSK (systém jednotné trigonometrické sítě katastrální), též Křovákovo zobrazení

# T4

Hlavním cílem technologie Mobile IP (zkráceně MIP) je zajistit kontinuitu spojení. Tok packetů nesmí být přerušen bez ohledu na to, kde se síťové zařízení právě nachází či jakým způsobem se momentálně pohybuje.



1. **Jakými technikami lze realizovat mobility v IP sítích a na jakých vrstvách ISO-OSI RM?**

* na 2. vrstvě ISO-OSI – distribuci VLAN na konkrétní přístupový bod a provoz doručujeme podle asociace klienta k přístupovému bodu na základě jeho MAC adresy
* paketová data v **buňkových** sítích, kde je realizována správa sezení, transport datagramů a správa mobilit. **protokol GTP** data jsou přenášena na servisní uzel GPRS a z těchto uzlů agregována uzlem v roli brány do světa IP
* na 3. vrstvě ISO-OSI – mobilní IP a protokol LISP
* na 4. vrstvě je možné využít Multipath TCP
* mobilní VPN, která bude provoz tunelovat do cílové lokality při zachování původní IP adresy
* manipulaci směrovacích informací Anycast
* manipulaci DNS záznamů (např. dynamickou DNS)

1. **Jaké jsou základní rozdíly mezi IPv4 a IPv6 mobilitou?**

* Mobilita v prostředí IPv4 nebyla původně předpokládána ani vyžadována. Při návrhu se nepočítalo s mobilními zařízeními, které by byly schopny rychle přecházet mezi různými sítěmi.
* IPv6 mobilita jedním ze stavebních kamenů IPv6 a koncové uzly by ji měly alespoň v minimální podobě podporovat.

U IPv4 se zavádi domácí a cizí agent HA/FA, kteří monitorují novou adresu cestujícího uzlu a předávají si tunelem provoz do nové sítě na původní adresu. Při změně přípojného bodu následující činnosti: Vyhledání agenta, Registrace dočasné adresy u cizího a domácího agenta, potvrzení do spojovacích tabulek, tunel mezi lokalitami. Nevýhodou je asymetrické směrování při zpětném provozu nebo nákladné přeposílání.

Základní odlišností IPv6 je, že nevyužíváme cizího agenta a CN může za určitých podmínek komunikovat s MN přímo. Každý mobilní uzel má globálně unikátní HoA, musí být schopen zachovat spojení během přechodumezi sítěmi a komunikovat i s uzly, které MIPv6 vůbec nepodporují. Pro samotnou IPv6 mobilitu je použita speciální hlavička – mobility header.

1. Jaké pojmy pro IP mobilitu definujeme?

* home network – HN
* foreign network – FN
* home agent – HA
* home address – HoA
* care-of address – CoA
* mobile node – MN
* correspondent node – CN (komunikující s mobilním)

1. **Je nutné komunikaci MN s HA autentizovat, pokud ano, proč?**

ano, protože by jinak bylo snadné ukradnout identitu (ověřit)

1. **Co znamená zkratka NEMO?**

mobilita celých sítí (Network Mobility – NEMO) v IPv6

1. **Může CN komunikovat s MN přímo bez využítí HA u IPv4 mobility?**

ne, nemá jak, neví kam

1. **Může CN komunikovat s MN přímo bez využítí HA u IPv6 mobility? Pokud ano, co musí být splněno?**

ano, musí být vyplněna příslušná mobility hlavička v záhlaví

1. **Lze nějak zjednodušit IPv6 mobilitu z pohledu klientů?**

není potřeba implementovat mobilitu, pokud je na síti přítomna mobilitní proxy, která komunikuje s HA místo samotného MN.

1. **Jaké bezdrátové technologie znáte a na co se které z nich hodí?**

buňkové sítě

* komunikaci v rozlehlejší geografické oblasti zajišťuje množství základnových stanic, které svým dosahem vytvářejí soustavu vzájemně se překrývajících poměrně malých buněk

Bezdrátové PAN

* založené na zařízeních, která nemusí být připojena k Internetu a tvoří síť složenou z hlavních a vedlejších zařízení, které spolu komunikují, většinou na relativně malou vzdálenost. Mezi typické zástupce patří technologie Bluetooth a ZigBee
* Bezdrátové LAN sítě, WiFi a mesh sítě
* přístup k bezdrátovému broadbandu, WiMAX

1. **Charakterizujte oblast použití mesh sítí a jejich vztah k sítím 802.11**

[Základ](https://www.marigold.cz/item/mesh-site-p2p-architektura-v-bezdratovych-sitich)

Mesh sítě jsou tedy založeny na takzvaném „ad hoc peer to peer routingu“ – na směřování provozu mezi rovnocennými adaptéry podle potřeby. Tímto systémem se podařilo prodloužit dosah takové distribuce signálu, protože uzly si data mezi sebou „předávaly“.

Problémem proti 802.11 je šířka používaného pásma, protože se se vzrůstajícím počtem hopů snižuje.

1. **Jaké směrovací protokoly a jejich typy se u bezdrátových mesh sítí používají?**

Proaktivní (udržují spojení), reaktivní (vytváří spojení v případě požadavku), hybridní, hierarchické (přístup záleží na úrovni uzlu)

* OLSR - Optimized Link State Routing Protocol
* B.A.T.M.A.N.
* HWMP

# T5

1. **Popište, pro jaké aplikační oblasti se bezdrátové senzorové sítě hodí?**

jsou určena pro nasazení v prostoru rozmístěných senzorů, které kooperativně monitorují fyzikální veličiny či dané prostředí

1. **Co lze prostřednictvím WSN monitorovat?**

Mezi monitorované veličiny můžeme zařadit teplotu, vlhkost (relativní, pokožky), tlak (barometrický, krevní), vibrace, zrychlení, rychlost pohybu a další telemetrická data u vozidel, intenzitu osvětlení, vzdálenost, okolní hluk, frekvenci (např. tepovou), apod.

průmyslových procesů a průmyslových strojů (včetně špatně přístupných či zcela nepřístupných částí), oblast zdravotnictví (podávání léků, monitorování fyziologických dat na dálku, sledování pohybu lékařů a pacientů), monitorování a řízení energetické spotřeby kancelářských budov a domácností, flotil vozidel, skladů a řešení pro chytré domácnosti, infotainment a monitorování sportovního výkonu a souvisejících biologických dat.

1. **Jaká jsou specifika WSN, čím se liší od mesh sítí?**

Topologie WSN se často mění, místo mesh sítě se směrováním se data distribuují broadcastem, výpočetní výkon, množství operační paměti a možnosti napájení jsou

výrazně omezené a senzory nemají globálně unikátní identitu (to by při velkém množství senzorů vedlo k zbytečně velkému množství bitů pro adresy).

1. **Obsahují WSN specifický vrstvený model, pokud ano, čím se liší od ISO-OSI?**

používá se zde zjednodušený model o 5 vrstvách, kde chybí 5. a 6. vrstva ISO OSI RM. navíc jsou zavedeny ještě 3 roviny, které pak definují požadavky na jednotlivé protokoly.

* rovina správy napájení (vypínání rádia po přijetí zprávy, zapínání a vypínání směrování zpráv podle aktuálního
* stavu zdroje energie)
* rovina správy mobility (detekce a zaznamenání přesunu uzlu, zajištění doručení dat, určení sousedních uzlů, apod.)
* rovina správy úloh (rovnoměrná distribuce a plánování měření v daném regionu, ne všechny uzly musí být v daném okamžiku aktivní a provádět měření)

1. **Jaké jsou hlavní rysy sítí 802.15.4 a technologie ZigBee?**

IEEE 802.15.4 is a technical standard which defines the operation of low-rate wireless personal area networks in general. Zigbee je poddruh. Díky použití multiskokového [ad-hoc](https://cs.wikipedia.org/wiki/Ad_hoc) směrování umožňuje komunikaci i na větší vzdálenosti bez přímé [radiové](https://cs.wikipedia.org/wiki/Radio) viditelnosti jednotlivých zařízení.

Rysy. Nody mají různé úlohy - koordinátor, směrovač, endpoint. sítí u nichž není požadován přenos velkého objemu dat.

Technologie ZigBee postavená na fyzické linkové vrstvě [IEEE 802](https://cs.wikipedia.org/wiki/IEEE_802).15.4 definuje tři různé síťové topologie. Základní topologií je [topologie hvězdicová](https://cs.wikipedia.org/wiki/Hv%C4%9Bzdicov%C3%A1_topologie) s centrálním řídícím uzlem (koordinátorem sítě). Druhým typem je [stromová struktura](https://cs.wikipedia.org/wiki/Stromov%C3%A1_struktura) jež umožňuje zvětšit vzdálenost mezi koordinátorem a koncovým zařízením. Protokol též umožňuje vytvoření redundancích spojení a vzniká tak topologie typu sítě – [*mesh*](https://cs.wikipedia.org/wiki/Mesh).

1. Jaké typy nositelných zařízení znáte?

* fitness náramky, hrudní pásy, měřiče kyslíku v krvi, vlkosti pokožky apod.
* chytré hodinky (kalkulačka / trackery / mobil / programovatelná zařízení)
* rozšířená realita (brýle Google, náhlavní sady)
* podpůrné doplňky pro postižené
* platení systémy

1. Čím se odlišují chytré hodinky od běžných hodinek?

výdrží baterie