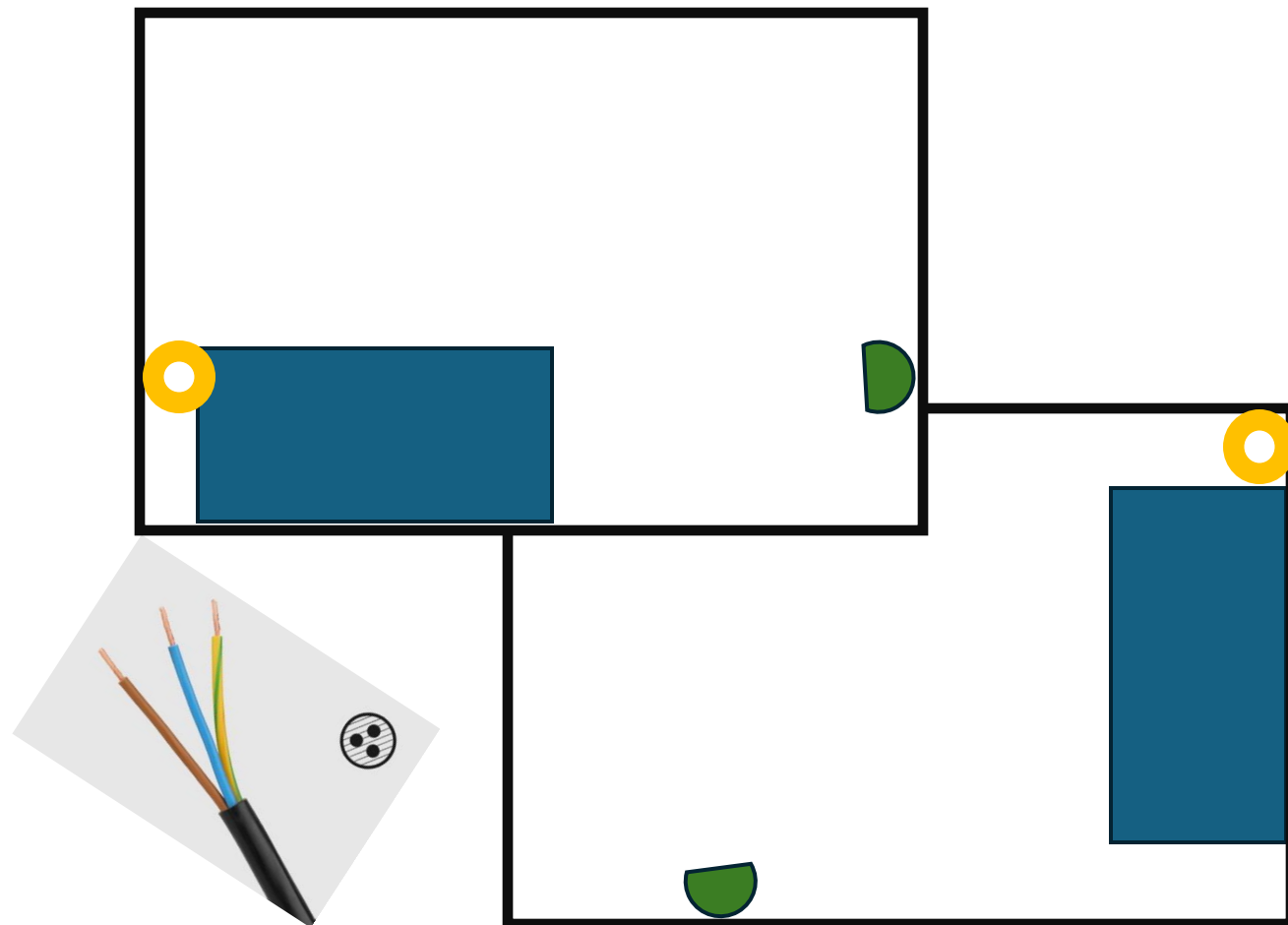


Gospod Watson, lahko pridete
sem, želel bi vas videti

Andrej Brodnik
UP FAMNIT, UL FRI

Peter preureja sobo

- Posteljo bo premaknil k drugi steni in mora zato prestaviti vtičnico in luč.
- Ko želi kupiti kabel, ga vprašajo, kakšen kabel potrebuje.
- Kakšnega naj kupi in zakaj?
- Od kje nam znanje?





Greva na čaj ob 5:12.

```
import socket
Borut = "127.0.0.1"
kje = 5005

vtic = socket.socket(socket.AF_INET,
                      socket.SOCK_DGRAM)
vtic.sendto(b"Greva na čaj ob 5:12?",
            (Borut, kje))
```

```
import socket

Borut = "127.0.0.1"
kje = 5005
nvs = 1024 # najv. velikost sporočila

vtic = socket.socket(AF_INET, SOCK_DGRAM)
vtic.bind((Borut, kje))

sporočilo, posiljatelj = vtic.recvfrom(nvs)
print(f"{posiljatelj}: {sporočilo}")
```



Storitev



in



Nudi **prenos sporočila** od  do  - **po najboljših močeh**

- Vmesnik do storitve  se imenuje vtič (*socket*)
- Kaj in kako se dogaja v  je stvar implementacije storitve, ki opredeljuje njeno delovanje, da lahko nudi storitev



Nekaj besed o vtičih

- Na isti napravi je lahko hkrati v uporabi več vtičev pri različnih aplikacijah => jih moramo nekako razlikovati
 - operacijski sistem nudi 2^{16} vrat in aplikacija izbere vrata, na katera bo zataknen vtič ali pusti operacijskemu, da ji dodeli vrata
- Naslov sogovornika sestoji iz **naslova naprave** in **številke vrat**

```
import socket
Borut = "127.0.0.1"
kje = 5005

vtic = socket.socket(socket.AF_INET,
                      socket.SOCK_DGRAM)
vtic.sendto(b"Greva na čaj ob 5:12?",
            (Borut, kde))
```

```
import socket
```

```
Borut = "127.0.0.1"
kde = 5005
```

```
nvs = 1024 # najv. velikost sporočila
```

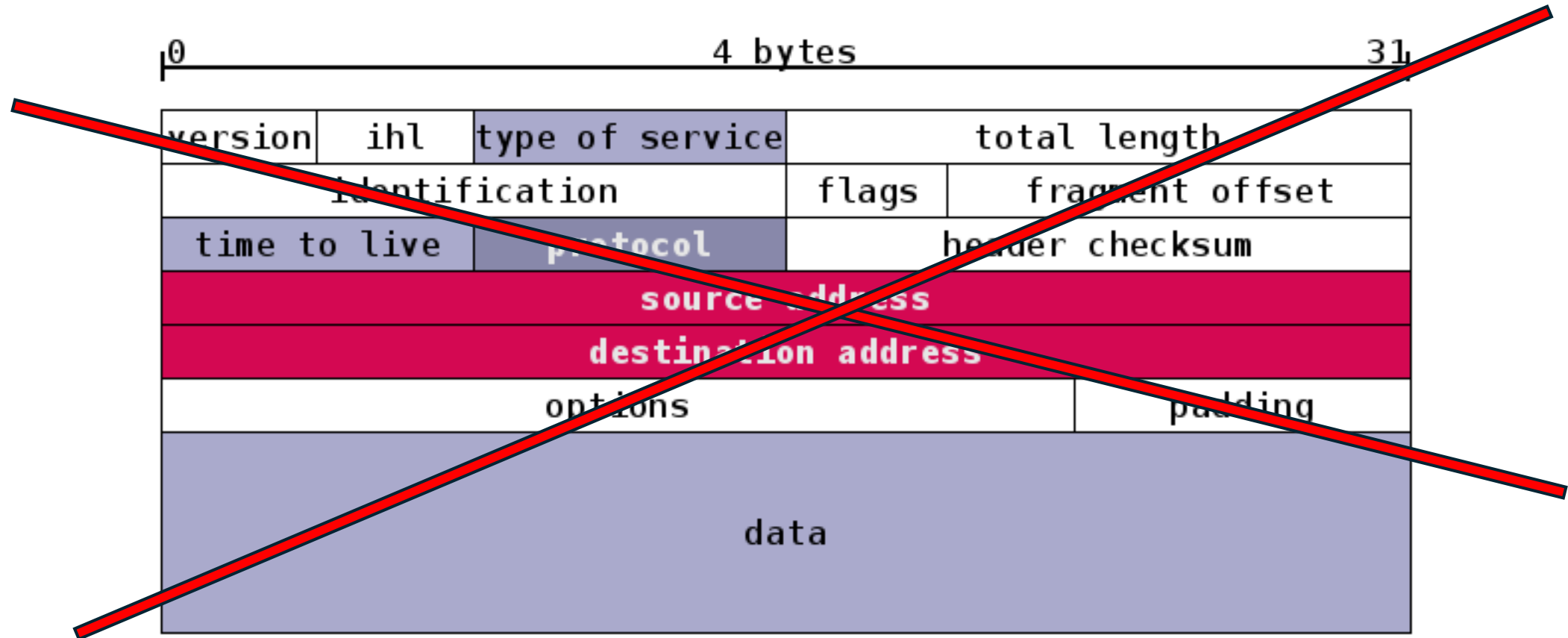
```
vtic = socket.socket(AF_INET, SOCK_DGRAM)
vtic.bind((Borut, kde))
```

```
sporočilo, posiljatelj = vtic.recvfrom(nvs)
print(f"{posiljatelj}: {sporočilo}")
```

Še zadnji komentar o vtičih

- Vtiči omogočajo aplikaciji, da **prenaša**
 1. posamezna sporočila – sporočilo se samo pošlje
 2. tok sporočil – ustvari se kanal prenos toka sporočil
 3. posebne zahteve – npr. šifriranje in predstavitev s certifikatom
- Na prenosni plasti imamo UDP in TCP storitvi ter pri slednji šifriranje s SSL/TLS

Računalniške komunikacije



Storitev nudi **funkcionalnost** in to skozi pojem **vtiča**.

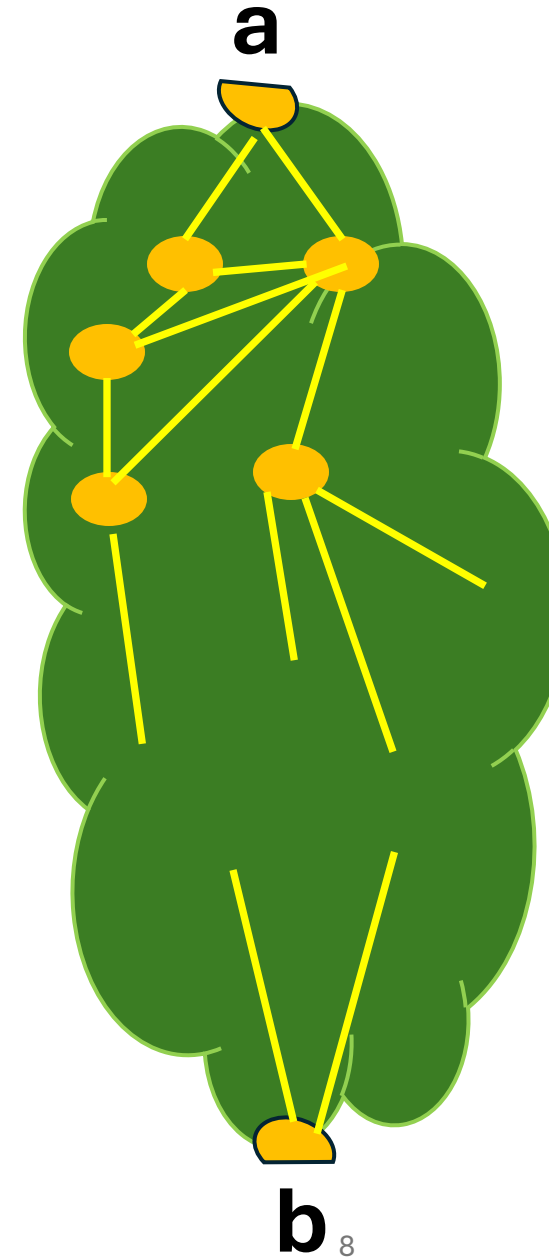
V nadaljevanju si bomo ogledali **koncepte delovanja storitve**, ki pa so tam samo in **izključno zato, da implemntirajo storitev z njeno funkcionalnostjo**.

Zato na nek način **vtič predstavlja vstopno točko v svet računalniških komunikacij**, saj definira funkcionalnost storitve.

Računalniških komunikacij bi ne bilo in bi jih ne potrebovali, če bi ne potrebovali storitve z njeno funkcionalnostjo.

Načrtovanje storitve in protokol

- v vsakem vozlišču je naprava. in na njej se naj izvaja aplikacija storitve
- usklajeno delovanje aplikacij omogoča izvedbo ponujene storitve
- Pomembno;
 - da lahko aplikacije delujejo usklajeno, govorijo skupni jezik – **protokol**; posledično lahko aplikacije izdelajo različni izvajalci
 - aplikacije **ne smejo uporabiti preveč virov** (čas in prostor)



Načrtovanje storitve

- nekako spraviti **sporočilo** (podatke, **data**) od **a** do **b**
- da lahko aplikacija v vozlišču to opravi, mora poznati poleg **sporočila naslov** prejemnika
- naslov prejemnika so **metapodatki** (**metadata**)
- podatki in metapodatki potujejo skupaj



Načrtovanje storitve

- storitev obljublja, da prejemniku sporoči kdo je pošiljatelj
 - Kako pomagati **b**, da Borutu sporoči, da je sporočilo prišlo od Alenke?
- podatek dodamo v metapodatke in potuje od **a** do **ba**



Upravljalni medklic

metapodatki

podatki

- metapodatki so pomembni in je potrebno vsaj zaznati, če ne preprečiti njihovo spreminjanje – **celovitost (meta)podatkov**
- primeri možne rešitve:
 - paritetni bit
 - paritetna vsota
 - izvleček in šifriran z javnim ključem
 - CRC
 - ...

metapodatki

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.

Tehnologija

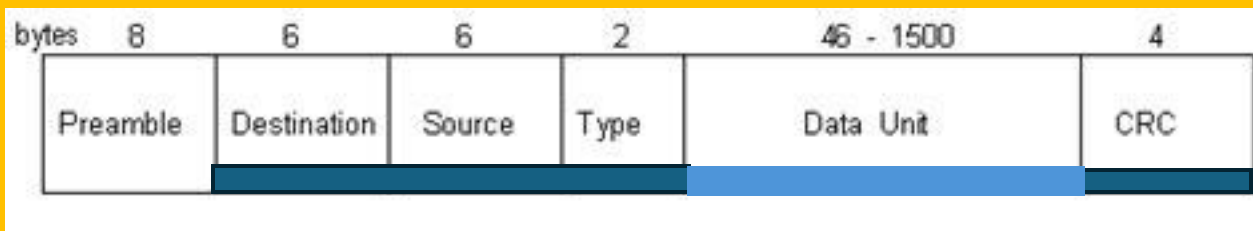
metapodatki

podatki

- Sklopu metapodatkov in podatkov skupaj rečemo kako?

okvir

- velikost okvirja in metapodatkov je fiksna



metapodatki
[okvir]

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.

Upravljalni medklic - nadaljevanje

metapodatki

podatki

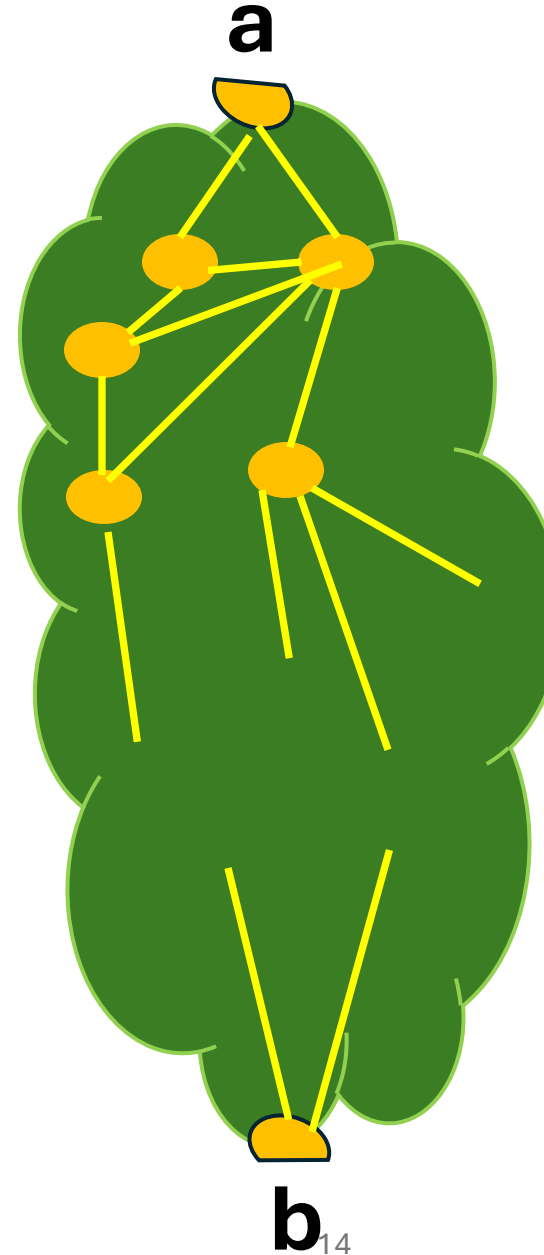
- Seveda tudi **paket** sestoji iz metapodatkov in podatkov
- ko je paket v pomnilniku pripravljen za obdelavo, vemo, kje se začne
 - in kje se konča?
 - potrebujemo vedenje o skupni velikosti paketa – **skupna dolžina**

metapodatki
[paket]

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.
skupna dolžina

Dostava

- nekako spraviti sporočilo (podatke, *data*) od **a** do **b**
 - v vsakem vozlišču je naprava. in na njej se izvaja aplikacija storitve
 - usklajeno delovanje aplikacij omogoča izvedbo ponujene storitve
- Aplikacija:
 - če je naslovnik na mojem vozlišču => predam Borutu
 - sicer => predam enemu od sosedov

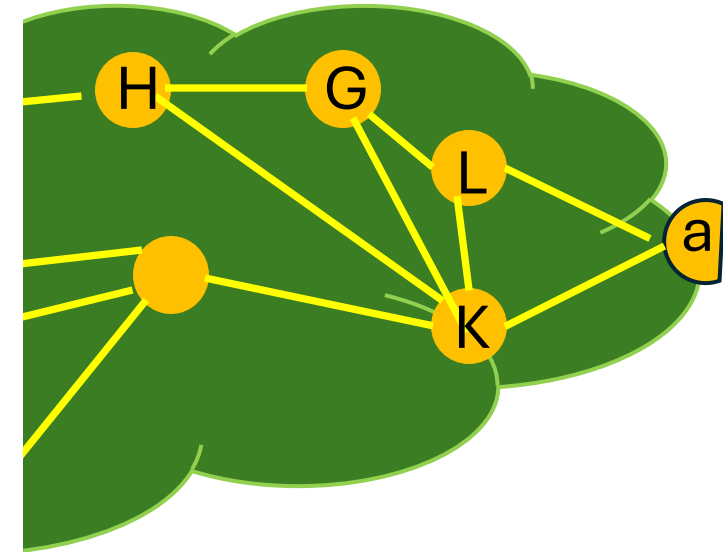


Predam enemu od sosedov – kateremu?

- Če vsakdo vidi vsakogar:
 - predam neposredno
 - vsakdo je povezan z vsakim (povezavna plast)
 - LAN, ethernet z mostički in stikali
- Sicer pa komu?

Kateremu sosеду predati?

- Nudi **prenos sporočila po najboljših močeh**
- Rešitev:
 1. naključnemu sosеду (vroč krompir)
 2. za vsak možen naslov hranimo informacijo, kateremu sosеду predati – **prepošiljevalna tabela**
- Vsaj dva problema:
 1. povezave se kvarijo in postavljajo => prep. tabele se spreminjajo na veliko vozliščih
 2. aplikacije **ne smejo uporabiti preveč virov** (čas in prostor) – prep. tabela
- Posledica: sporočila lahko blodijo **ZELO dolgo**



Kateremu sosedu predati – rešitve

- vsako sporočilo ima omejen dovoljen čas blodenja (obstaja) – zakaj v redu?
- naslove združujemo v veliko manjše število mrež: <naslov>= <mreža><naprava v m.>
- => prepošiljevalna tabela se bistveno zmanjša



Prepošiljevalna tabela

```
andy@svarun:~ % netstat -rn
Routing tables
```

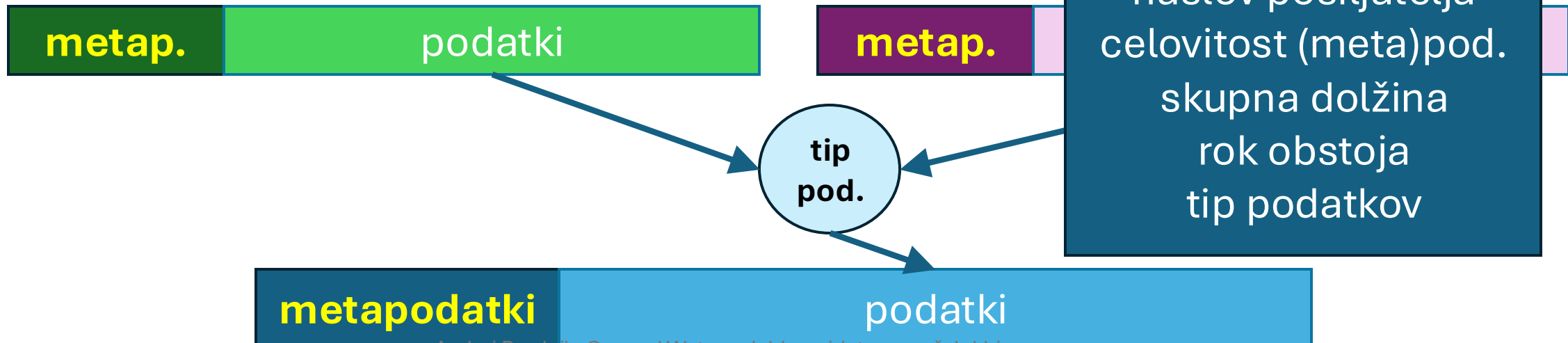
```
Internet:
Destination      Gateway          Flags           Netif  Expire
default          192.168.126.1   UGS             alc0
127.0.0.1        link#2          UH              lo0
192.168.2.0/24    link#1          U               alc0
192.168.2.3       link#2          UHS             lo0
192.168.3.0/24    link#1          U               alc0
192.168.3.3       link#2          UHS             lo0
192.168.4.0/24    link#1          U               alc0
192.168.4.3       link#2          UHS             lo0
192.168.5.0/24    link#1          U               alc0
192.168.5.3       link#2          UHS             lo0
192.168.126.0/24  link#1          U               alc0
192.168.126.2     link#2          UHS             lo0
```

Izpuščena podrobnost

- Borutov in Alenkin naslov sestojita iz naslova vozlišča in vrat

Prispelo sporočilo: (... , ('127.0.0.1', 55710))

- Vrata so del metapodatkov TCP oziroma UDP sto
- Metapodatki paketa pa označujejo ali gre TCP ali



Problemi na poti – zastoj

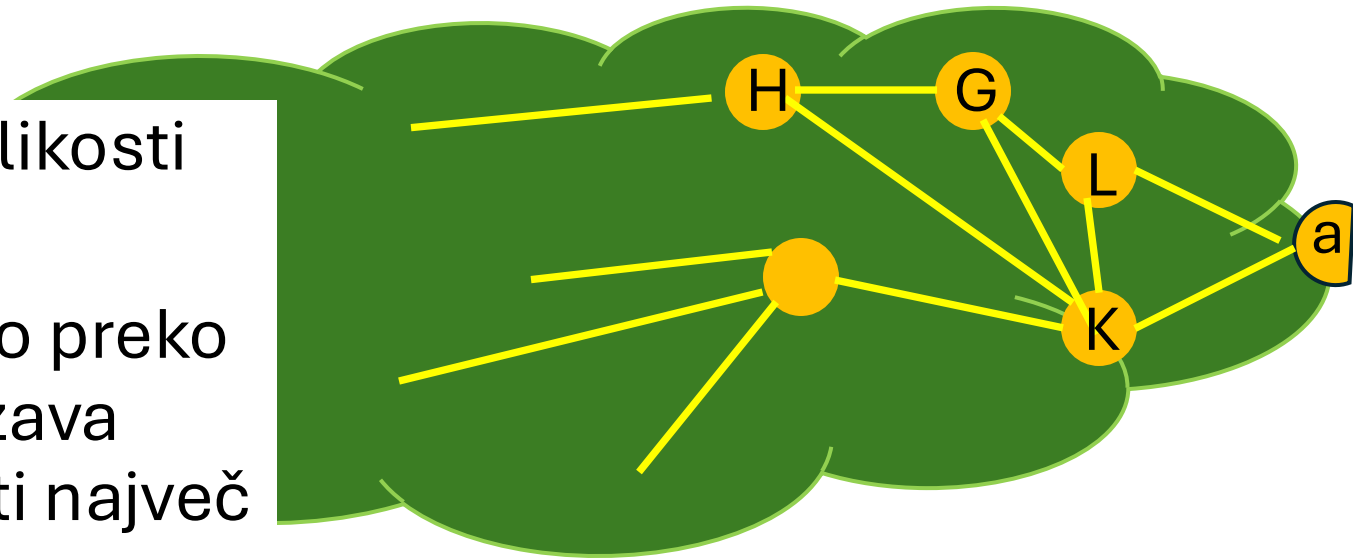
- se aplikacije v vozliščih obveščajo o tem – del metapodatkov
- za nekatera sporočila / pakete se bolj mudi, da pridejo na cilj – koliko se komu mudi

metapodatki **[paket]**

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.
skupna dolžina
rok obstoja
tip podatkov
prioriteta
zastoj na poti

Problemi na poti – neustrezna povezava

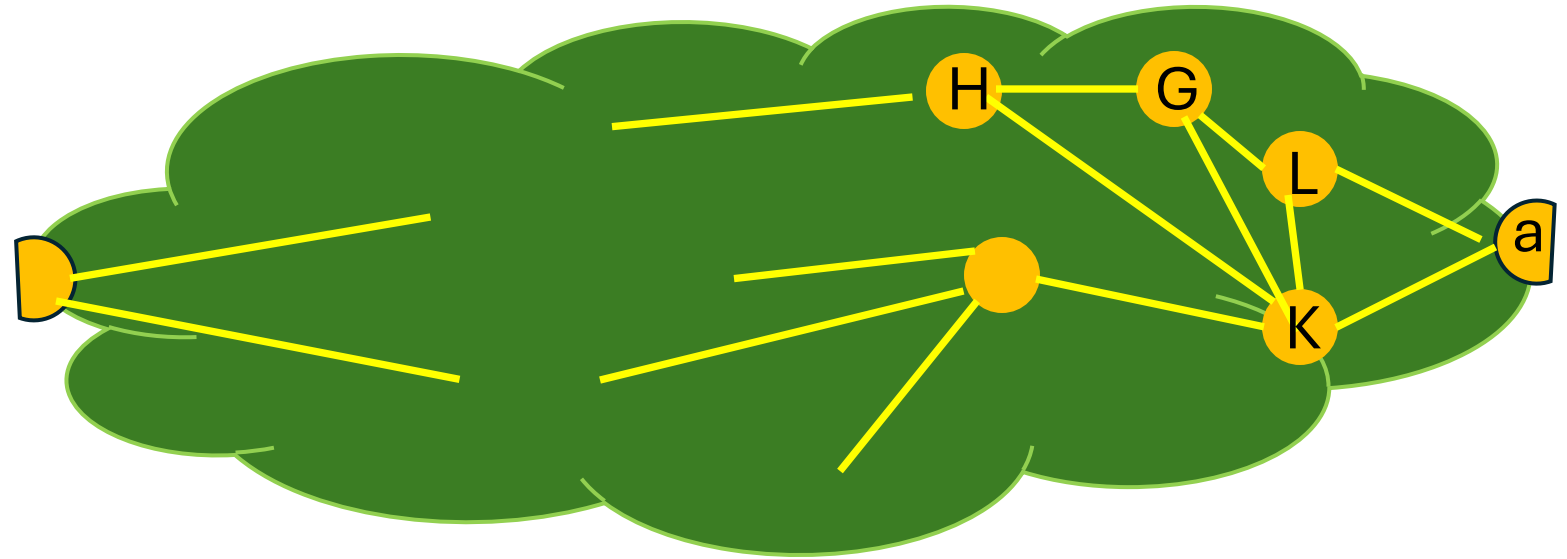
1. **G** prejme od **L** sporočilo velikosti 1.500B
2. **G** želi posredovati sporočilo preko **H** do prejemnika **b**, a povezava dovoljuje sporočila velikosti največ 600B
3. Rešitev: G razdeli sporočilo na koščke (fragmente), ki jih **b** sestavi skupaj



Problemi na poti – neustrezna povezava

metapodatki [paket]

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.
skupna dolžina
rok obstoja
tip podatkov
prioriteta
zastoj na poti
?fragmentacija?



metapodatki

podatki

metapodatki

ki

metapodatki

dat

metapodatki

po

Odprtost, razširljivost

- verzija protokola – uporabljeno tako pri okvirju, kot paketu
- novi metapodatki => velikost metapodatkov ni fiksna

metapodatki [paket]

naslov prejemnika
naslov pošiljatelja
celovitost (meta)pod.
skupna dolžina
rok obstoja
tip podatkov
prioriteta
zastoj na poti
?fragmentacija?
verzija
novi metapodatki
velikost metapod.

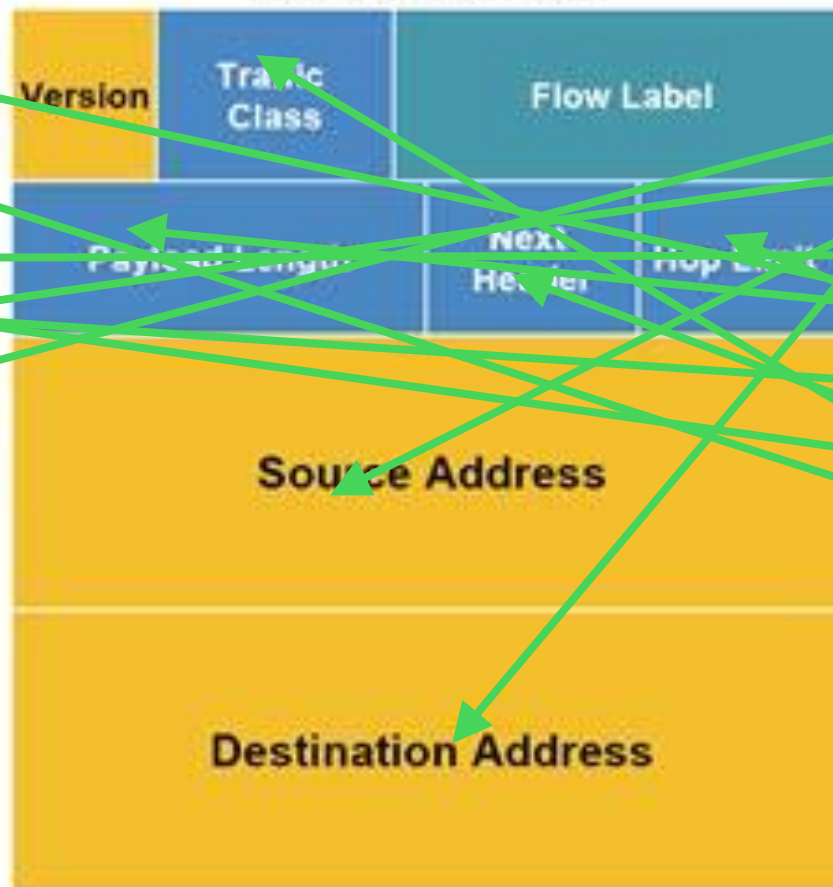
IPv4 Header



Legend

- Field's Name Kept from IPv4 to IPv6
- Fields Not Kept in IPv6
- Name and Position Changed in IPv6
- New Field in IPv6

IPv6 Header



metapodatki [paket]

- naslov prejemnika
- naslov pošiljatelja
- celovitost (meta)pod.
- skupna dolžina
- rok obstoja
- tip podatkov
- prioriteta
- zastoj na poti
- ?fragmentacija?
- verzija
- novi metapodatki
- velikost metapod.