Netzwerke und Kommunikation
B-LS-MI 004
NDK 02-050
Dynamisches Routing und Routing Protokolle

rolf.schmutz@fhnw.ch

FHNW

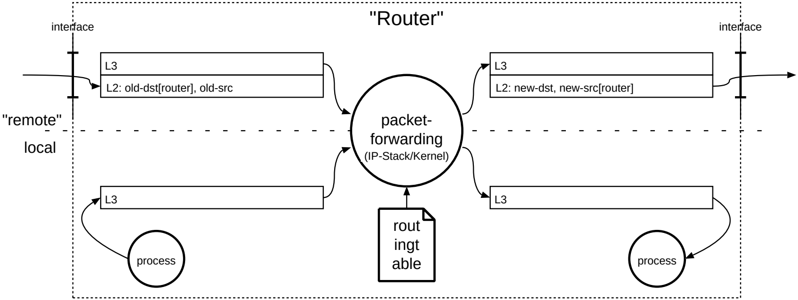
21. Oktober 2020

Ziele

- Sie kennen die Aufgabe der Routing Protokolle
- Sie kennen die Funktionsweise und den Einsatzzweck von OSPF, RIP und BGP
- *Sie kennen den Unterschied zwischen routing (forwarding) und Routing Protokollen*

Routing: Packet-Forwarding

- jeder Router leitet Pakete aufgrund der Information in der *routing-table* weiter
- das umgangssprachliche “routing” ist eigentlich ein *packet-forwarding*

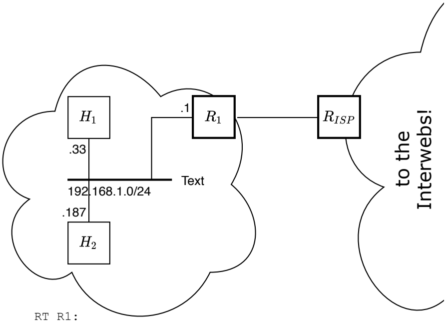


Routing: statisches Routing

- für kleine Netzwerke genügt statisches Routing:
  - stub-net: nur ein Netzwerk und ein Router mit Verbindung zum "Rest" der Welt vorhanden[1]
  - statische Netze mit wenigen Verbindungen dazwischen – z.B. Firma mit Aussenstandorten über Mietleitungen
- dabei werden die Routing-Tabellen auf den beteiligten Geräten manuell nachgeführt
- es können "alternative" Wege eingetragen werden, die Routingtabelle bleibt aber statisch

[1] das ist genau die Konfiguration bei Ihnen zuhause

Routing: Stub-Network (statisch)




```
RT R1:
Dest            Next-Hop/via
----            ------------
192.168.1.0/24  local
0.0.0.0         R_{ISP}

-----> only one exit
```

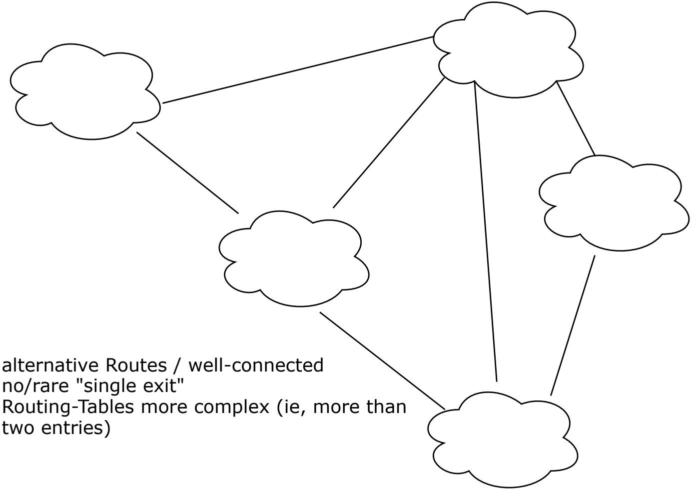
Routing: Dynamisches Routing

- automatische/dynamische Adaption bietet sich an bei:
  - grossen Netzwerken mit vielen Routern bei denen manuelle Anpassung der RT fehleranfällig/mühsam wäre
  - sich dynamisch verändernden Netzwerken
- dies wird durch *anpassen der Routing-Tabelle* aufgrund von *Topologie-Informationen* erreicht
- ein *Routing-Protokolle* (RP) hat die Aufgaben:
  - mit "peers" (anderen Routern) zu kommunizieren und Topologie-Informationen auszutauschen
  - die Routing-Tabelle entsprechend anpassen
  - **das Routing-Protokoll macht selber kein forwarding!**

Dynamisches Routing

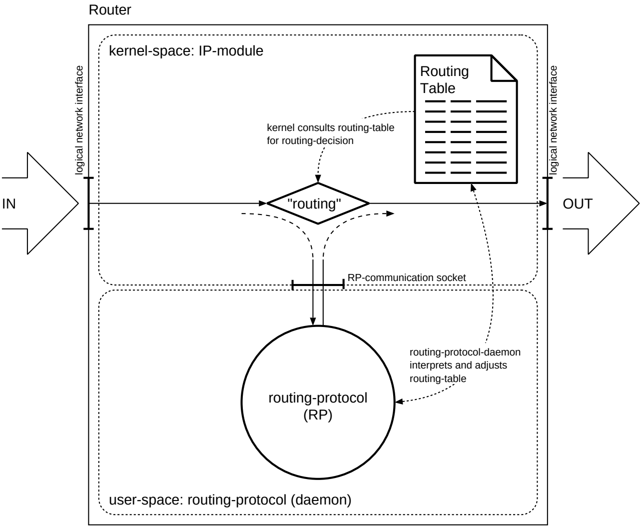
Anpassen der Routing-Tabelle aufgrund von Topologie-Informationen. Dynamisches Routing wird durch "normale" Prozesse/Programme mit Socket-Kommunikation erreicht

Routing: Well-Connected-Network(s) (dynamisch ist angesagt!)



- alternative Routes / well-connected
- no/rare "single exit"
- Routing-Tables more complex (ie, more than two entries)

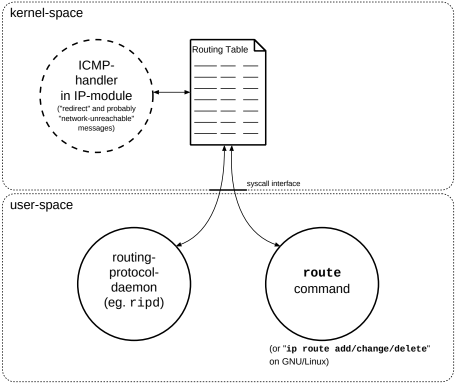
Routing: RP↔RT Interaktion



Routing: RT Updates

Die Routing-Tabelle kann auf folgende Weise manipuliert werden:

- manuell durch editieren der RT ("statisches Routing")
- durch ein Routing-Protokoll ("dynamisches Routing")
- durch `ICMP-REDIRECT`-Meldungen ("redirects")



Routing: Routing/Forwarding und Routing-Protokoll

ein Host kann in Bezug auf "Routing" folgende Rollen einnehmen:

- kein forwarding, kein routing-protocol: normales Endgerät[2]
- forwarding, kein routing-protocol: statischer Router
- kein forwarding, routing-protocol aktiv: "Route Reflector" oder "Route Server"
- forwarding und routing-protocol: dynamischer Router

	Forwarding	*No Forwarding*
Routing Protocol	dynamic Router	Route Server
No Routing Protocol	static Router	standard Host/Endpoint

[2] Client oder Server

Routing: Interior und Exterior Routing

die Anforderungen an ein Routing-Protokoll unterscheiden sich je nach Anwendungszweck:

- **Interior Routing Protocol**: innerhalb einer Organisation[3]/**AS**[4] soll das Routing-Protokoll:
  - sich schnell an neue Situation anpassen ("konvergiert schnell")
  - effizient arbeiten (nicht zuviele Daten senden/empfangen)
  - den *besten/schnellsten* Weg finden
  - möglichst konfigurationslos arbeiten ("automatisch")
- **Exterior Routing Protocol**: *zwischen* Organisation, resp "im Internet" soll das Routing-Protokoll:
  - nicht-technische, d.h. "politische" Entscheidungen verwalten ("wer darf mit wem", transit, etc)
  - Langzeitstabil sein (d.h. keine schnelle Oszillation "route-flapping")

Unabhängig davon sollte ein Routing-Protokoll:

- "loop-free" arbeiten, d.h. keine Routing-Schlaufen generieren

[3] Hochschule, Firma, Service-Provider, etc

[4] "autonomous system"

Routing: Protokoll Familien

Es wird im Allgemeinen zwischen drei Ansätzen unterschieden:

Distance Vector z.B. RIP

information element[5]: Listen {`network, metric`} – d.i. eine abgekürzte Routing-Tabelle
communication peers[6]: mit allen direkten Nachbar-Router
topology inference[7]: aufgrund vorverarbeiteten Information[8] von anderen Routern (summary, "routing by rumors")

Link State z.B. OSPF

information element: Listen {`link/interface, state, metric`} – d.i. eine Liste der *Netzwerk-Interfaces* und ihr Zustand (up, down)
communication peers: mit *allen Routern im Netzwerk*
topology inference: aufgrund gesicherten *lokalen Informationen* der Router

Path Vector z.B. BGP

information element: komplexe Listen {`net`, $\texttt{path-element}_1$, $\texttt{path-element}_2$, `...`, `other-attributes*`} – d.i *Pfad* zum Ziel
communication peers: mit ausgewählten *Peer-Routern* (peering)
topology inference: magisch

[5] "was wird ausgetauscht"

[6] "mit wem kommuniziert das Routing-Protokoll"

[7] "wie wird die Netztopologie bestimmt"

[8] Sicht dieses Routers vom Netzwerk

Routing: Distance Vector am Beispiel von RIP

Das **R**outing **I**nformation **P**rotocol[9] ist ein lebendes Fossil aus der IP-Steinzeit:

Information Element : distance vector — a list of {`network, distance`} tuples. Distance is measured in count of “hops” to reach a network; one hop being a router

Communication Peers : broadcast on all connected networks, UDP 520

Topology Inference : (none), distributed Bellman-Ford algorithm[10]

Operation : RIPv1 sends DV-elements in fixed intervals to the broadcast addresses of all connected networks (interfaces):

- send own routing-table (only network w/o mask and hop-count metric) broadcast
- received DVs are compared element-wise with the existing routing-table; entries with minimal metric are kept, all other information is dropped
- every (dynamically learned) routing-table entry will eventually time-out if it is not updated by new received DV

Pros : ubiquitous (everyone talks and understands RIP)

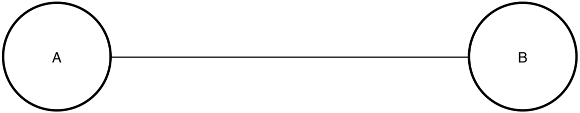
Cons : (too many, see next slide)

[9] nein, nicht “Rest In Peace”, wobei das in diesem Fall angebracht wäre

[10] RIP is based on the fact that only the next-hop to a certain destination must be known for correct routing-operation. A single RIP-instance is not able to determine the real network topology

Routing: Notation Beispiele

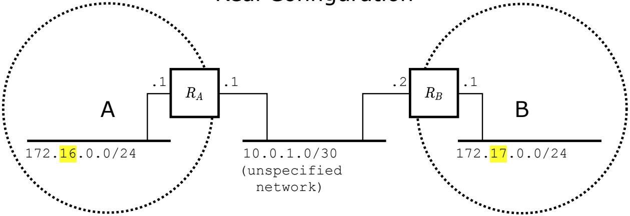
Simplyfied View




```
RT A:
Dest Next-Hop
---- --------
A    local
B    B
```

```
RT B:
Dest Next-Hop
---- --------
B    local
A    A
```

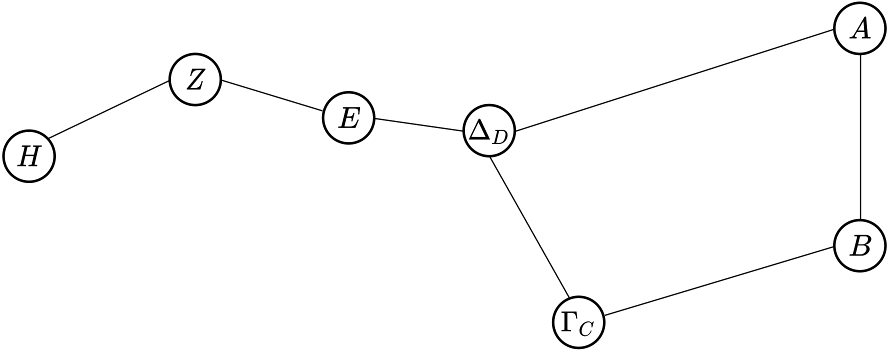
Real Configuration



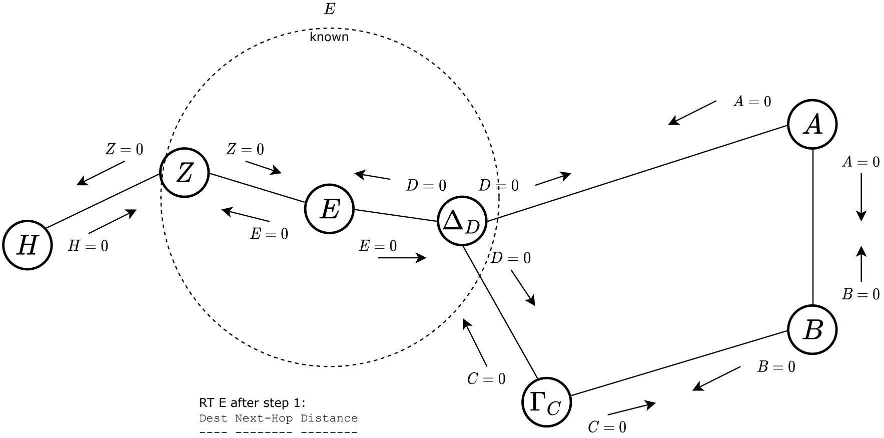

```
RT A:
Dest           Next-Hop
----           --------
172.16.0.0/24  local
10.0.1.0/30    local
172.17.0.0/24  10.0.1.2
```

```
RT B:
Dest           Next-Hop
----           --------
172.17.0.0/24  local
10.0.1.0/30    local
172.16.0.0/24  10.0.1.1
```

Routing: RIP Beispiel



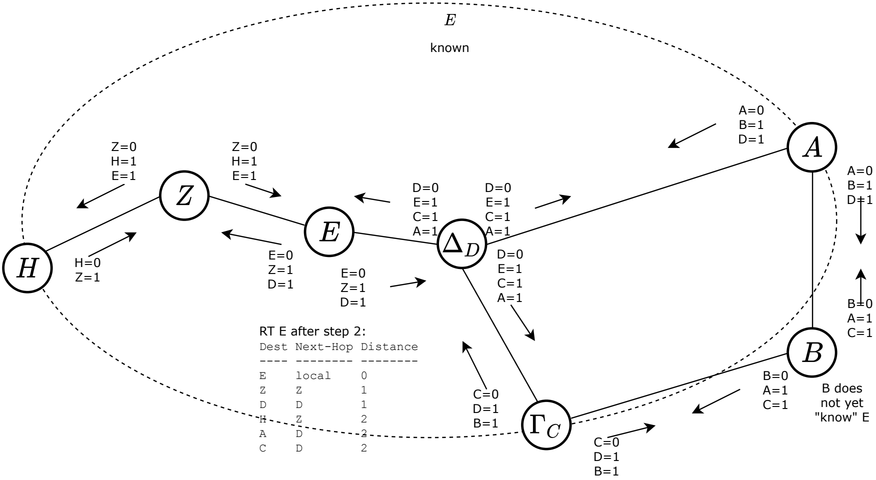
Routing: RIP Beispiel, Step 1



RT E after step 1:

```
Dest Next-Hop Distance
---- -------- --------
E    local    0
Z    Z        1
D    D        1
```

Routing: RIP Beispiel, Step 2



RT E after step 2:

```
Dest Next-Hop Distance
---- -------- --------
E    local    0
Z    Z        1
D    D        1
H    Z        2
A    D        2
C    D        2
```

Routing: RIP Tabellen

important: adjancency of nodes is not reflected in column-adjaceny

	H			*Z*			*E*			*D*			*C*			*A*			*B*		
	Dest	*Next*	*Dist*	*Dest*	*Next*	*Dist*	*Dest*	*Next*	*Dist*	*Dest*	*Next*	*Dist*	*Dest*	*Next*	*Dist*	*Dest*	*Next*	*Dist*	*Dest*	*Next*	*Dist*
Step 0:	H	local	0	Z	local	0	E	local	0	D	local	0	C	local	0	A	local	0	B	local	0
Step 1:	H Z	local Z	0 1	Z H E	local H E	0 1 1	E Z D	local Z D	0 1 1	D A C E	local A C E	0 1 1 1	C B D	local B D	0 1 1	A B D	local B D	0 1 1	B A C	local A C	0 1 1
Step 2: duplicates eliminated	H Z E	local Z Z	0 1 2	Z H E D	local H E E	0 1 1 2	E Z D H A C	local Z D Z D D	0 1 1 2 2 2	D A C E Z B	local A C E E A	0 1 1 1 2 2	C B D A Z	local B D D E	0 1 1 2 2	A B D C E	local B D B D	0 1 1 2 2	B A C D	local A C A	0 1 1 2

● ● ●

"Next" only contains direct neighbors (adjacent nodes)

Routing: RIP Nachteile

- "routing by rumors"
- too simple metric
- limited diameter (max 15 hops)
- classful behavior (implicit class-netmask)
- slow convergence; fixed intervals and flawed method
- loops/gaps; "counting to infinity", etc
- no authentication
- traffic grows uncanny for bigger networks
- broadcast communication

Routing: RIPv2 Fixes

metric : tweak "interface cost" such that slower links will count more than one host[11]

classful behavior : RIPv2 solves this problem: DV contains `(network,mask,distance)` tuples

slow convergence : "triggered updates", event-driven DV-broadcast. Implemented as an option to RIPv1 and mandatory in RIPv2

loops/gaps : "split horizon" and "poisoned reverse". Implemented as an option to RIPv2, mandatory in RIPv2

authentication : implemented in RIPv2

broadcast communication : RIPv2 supports multicast instead of broadcast

RIPv2 still suffers from "routing by rumors", "simple metric" (although there is support for additional metrics), "limited diameter" and "traffic growth". Besides this, RIPv2/RIPv1 coexistence is not as seamless as it may be: why not just a real routing-protocol™?

[11] `eg man ifconfig`

Routing: Link State am Beispiel von OSPF

*O*pen *S*hortest *P*ath *F*irst is capable of handling even the largest corporate networks[12]

Information Element: LSA, the *L*ink *S*tate *A*dvertisment.
A List of tuples `(link, state, metric...)`[13]

Communication Peers: Communication in OSPF if twofold:

- HELLO-protocol to discover and check reachability of neighbours
- LSA distribution through a *flooding* mechanism to all OSPF-routers in the network

Topology Inference: Djikstra's spanning tree algorithm — full topology available from LSDB[14]

Operation:

- neighbor/link integrity: through periodic HELLO-packet to neighbor routers very short packets, interval adjustable (common values 5, 10, 30 seconds)
- flooding LSA: on start-up or changes to interface/link-state, a LSA-packet is *flooded* to the entire OSPF-network (area)
- RT-calculation: by Djikstra's algorithm

Pros: (too many, see next slide)

Cons: CPU- and memory-load may be a concern if used on *old crappy* hardware

---

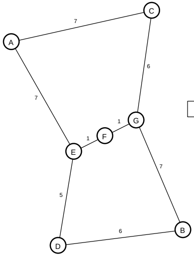
[12] we are talking about *interior routing* here... The Internet would be happy with OSPF too – if not for the politics... (see slide 28)

[13] the combined length of all LSAs in a network grows sub-polynomial, the length of a single LSA only changes if links where added or removed — compare this situation to RIP

Routing:Link-State Djikstra (1/3)

Network Topology "Orion"

numbers on links indicates "cost"



LSAs

distributed by flooding

A: (C=7, E=7)

⋮

G: (B=7, C=6, F=1)

LSDB

collection of all LSAs

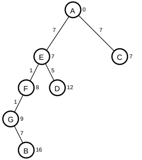
	A	B	C	D	E	F	G
A	0	-	7	-	7	-	-
B	-	0	-	6	-	-	7
C	7	-	0	-	-	-	6
D	-	6	-	0	5	-	-
E	7	-	-	5	0	1	-
F	-	-	-	-	1	0	1
G	-	7	6	-	-	1	0

symmetrical if metric is the
same for both directions (eg. not ADSL)

Djikstra's
Shortes Path
Algorithm

Minimal Spanning Tree

for node "A"



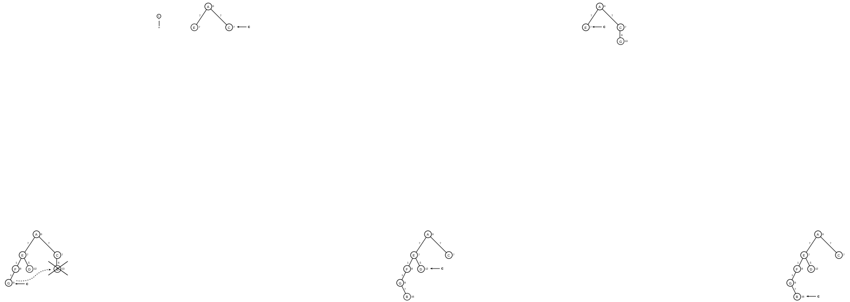
Routing: Link-State Djikstra (2/3)

"a not-so-formal description of Djikstras Spanning-Tree-Algorithm"

Initialization	Loop	Result
c $\leftarrow$ selected-root-node	while **F** $\neq$ {} do	**F** $=$ {}
A $\leftarrow$ { c }	for each neighbor n of c still in **F** do	**A** $=$ { topologically sorted list }
F $\leftarrow$ { all-nodes } \ c	if n $\exists$ **T**	
T $\leftarrow$ {}	if cost(n) $<$ cost(n $\in$ **T**)	
	T $\leftarrow$ **T** - (n $\in$ **T**) $+$ n	
	else	
	T $\leftarrow$ **T** $+$ n	
	end-if	
	else	
	T $\leftarrow$ **T** $+$ n	
	end-if	
	end-for	
	set c to minimum cost node from **T**	
	A $\leftarrow$ **A** $+$ c[15]	
	T $\leftarrow$ **T** \ c	
	F $\leftarrow$ **F** \ c	
	end-while	

[15] this involves also adding it permanently to the tree... ie: add new c as a child of former c

Routing:Link-State Djikstra (3/3)



Routing: OSPF Vorteile

Djikstra	Schleifen-freie Topologie-Ableitung ("spanning-tree")
Areas	Unterteilung in Teilgebiete, z.B. "Backbone" und "Site" mit Delegation
Virtual Links	nicht-physikalische Verbindungen können abgebildet werden
Multicast	es werden nur OSPF-Router angesprochen (vergl. RIPv1 broadcast)
Aggregating	ein Teilgebiet kann als "kleines AS" (blackbox) zusammengefasst werden
Security	authenticated messages, viel schwieriger anzugreifen als RIP

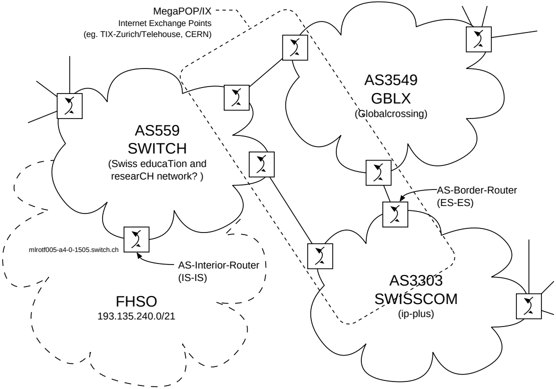
Routing: Path Vector

and now for something completely different. . .

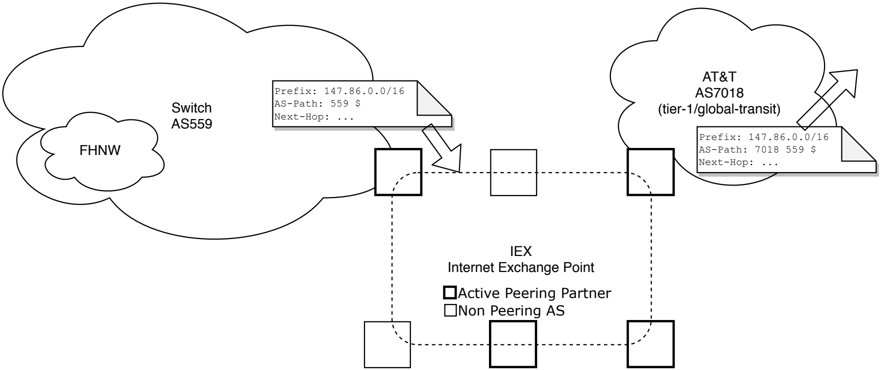
Routing: Internet “AS-jungle”

Routing in the Internet is based on “autonomous systems” (AS), regarded as black boxes, ie the *AS-internal* routing procedure is hidden at the AS-border

- ASes are usually ISPs or large corporations with redundant connections to the Internet (*multi-homed*)
- AS-numbers are allocated globally[21], ie this are unique identifiers
- ISPs usually gather together at “Internet Exchange” points[22], a physical location to form a star-network topology
- BGP is used between ASes to exchange routing-information *based on AS structures* (ie, large-scale routing)
- BGP allows to perform *policy based routing* (ie, not only based on destination)



Routing: BGP-Announcements



Routing: BGP

Dynamic routing in the Internet is done by the **B**order **G**ateway **P**rotocol

Information Element : path-vector, a prefix[16] and a list of ASes (in-reverse) to reach it

an actual example (edited) from AT&T=AS7018 (query is part of Solnet=AS9044):

```
BGP routing table entry for 212.101.0.0/19:
  7018 3549 9044 9044, (received & used)
```

read: to reach 212.101.0.0/19 from AT&T, the packet must pass through AT&T (AS7018), Global Crossing (GBLX, AS3549) and finally Solnet (AS9044)

Communication Peers : strictly point-to-point using TCP port 179. The connections must be configured manually (UDP 179 only for I-BGP)

Topology Inference : this is remains a miracle

Operation :

- OPEN connection to peer
- send NOTIFICATION in case of errors or to close the session
- send KEEPALIVE periodically (60 seconds beeing reasonable, hold-up is usually set to three times keepalive)
- send UPDATE as: "`prefix:` *`MY-AS#`* `[`*`existing-AS-path`*`], NEXT-HOP=`*`MY-IP`*", ie *prepend* the own AS# to the path
- select best (shortest, policy-based, etc) path from the received UPDATES, generate RT

Pros : there is no alternative

Cons : there is no alternative

Routing: BGP Factlets

Loop-Prevention: loops are resolved by filtering the path for repetitious elements

Best-Path: in absence of other criteria, the best path is the shortest[17] one. Although BGP (the process) allows almost arbitrary filtering and mangling of attributes, thus very sophisticated selection of paths are possible

I-BGP, E-BGP: large AS may run BGP AS-internal. This allows load-sharing and flexible adaption to AS-AS connection changes

[17] AS-wise, ie there is no possibility to infer the actual hop-count *inside* one AS

Routing: BGP Informationen 1/2

BGP "politics" Informationen kann mit `whois` oder über eine geeignete Webseite gefunden werden[18] Dabei wird der *AS-Path* zu einem Ziel angezeigt, d.h. nicht einzelne Router wie bei `traceroute` sondern AS (z.B. Internet Service Providers) als "black box"

radb: die routing arbiter database http://www.ra.net/, Beispiel (edited):

```
rschmutz@callisto routing $ host www.post.ch
www.post.ch has address 194.41.161.1

# query RADB http://www.ra.net/ with 194.41.161.1

route:       194.41.128.0/18
descr:       CH-POST-040816
origin:       AS12511

# query RADB http://www.ra.net/ with AS12511

aut-num:      AS12511
descr:        Die Schweizerische Post
import:       from AS6730
              action pref=100;
              accept ANY
import:       from AS3303
              action pref=100;
              accept ANY
export:       to AS6730
              announce AS12511
export:       to AS3303
              announce AS12511
```

[18] wenn sie nicht selbst ein AS verwalten, resp. ein BGP-peer sind. http://www.ripe.net/ für Europa

Routing: BGP Informationen 2/2

oder über einen *Route-Server*[19]:

gblx
```
telnet route-server.gblx.net
...
route-server.phx1>show ip bgp 194.41.161.1
BGP routing table entry for 194.41.128.0/18, version 39211644
Bestpath Modifiers: always-compare-med, deterministic-med
Paths: (1 available, best #1)
  Not advertised to any peer
  6730 12511 12511 12511 12511, (received & used)
    67.17.64.89 from 67.17.82.146 (67.17.82.146)
      Origin IGP, localpref 300, valid, internal, best
      Community: 3549:4723 3549:31756
      Originator: 67.17.80.142, Cluster list: 0.0.0.141
```

[19] auch route-reflector. Liste bei http://www.traceroute.org/#Route%20Servers

Routing: BGP Coole Tools

- "Looking-Glass/LG" Server sind Web-GUIs auf bekannte tools wie `traceroute`, `ping`, `show ip bgp`: http://traceroute.org/#Looking%20Glass, z.B. https://lg.he.net/
- RIPE[20] hat ein sehr cooles "live" BGP-Tool: https://ris-live.ripe.net/
- farbenfrohe Meta-Seite zu BGP: https://www.bgp4.as/tools
- RIPE hat ebenfalls eine Zusammenfassung von "whois" und Routing-Informationen: https://stat.ripe.net/ – dies klappt auch mit Netzwerken ausserhalb der RIPE-Registrierung
- mit einem lokalen `telnet` oder `nc/netcat/ncat`-Client können auch interaktive Sessions auf "echte" Route-Server im Internet gemacht werden:

```
rschmutz@callisto [,öô 1]NDK6 $ telnet route-server.he.net
Trying 64.62.142.154...
Connected to route-server.he.net.
Escape character is '^]'.
**********************************************************************
*****                   route-server.he.net                      *****
*****              Hurricane Electric IP Route Monitor           *****
*****                         AS 6939                            *****
**********************************************************************

This router maintains peering sessions with some of the core routers in
Hurricane Electric's network. Hurricane Electric operates an international
```