LAND

Stretch 1+ε Locality-Aware Networks

Ittai Abraham, Dahlia Malkhi, Oren Dobzinski (2004)

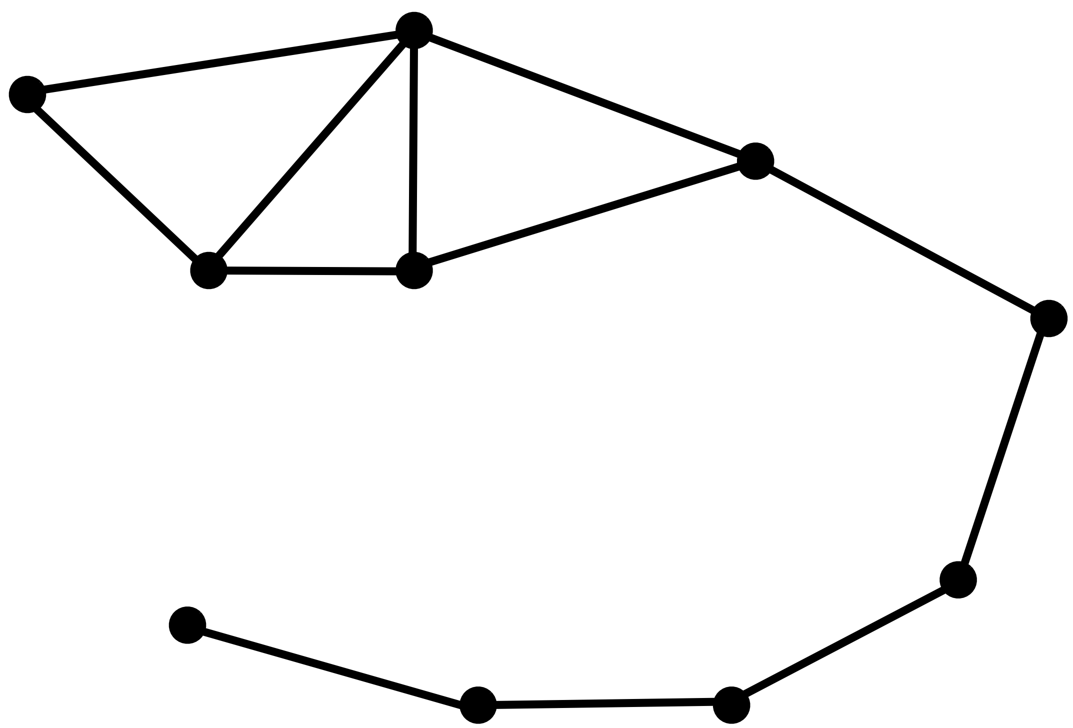
LAND

- **P2P-Netzwerk mit konstantem Lookup-Overhead**
  - Theoretisch garantierter Stretch von (1+ε)
- **Wachstumsorintierte Metrik**
  - Annahme: konstanter Ausdehnungsfaktor
- **Nutzt etablierte Techniken in P2P-Netzwerken**
  - DHT
  - Plaxton Routing
  - Starke Ähnlichkeit mit Tapestry (Zhao u.A. 2001)

Stretch-Faktor

- **Netzwerkgraph**
  - Knotenmenge V (|V| = n)
  - Verbindungen (Kanten)

- **Pfadkosten**
  - Kostenfunktion (z.B. RTT)
    $c : V^2 \mapsto \mathbb{R}_+$

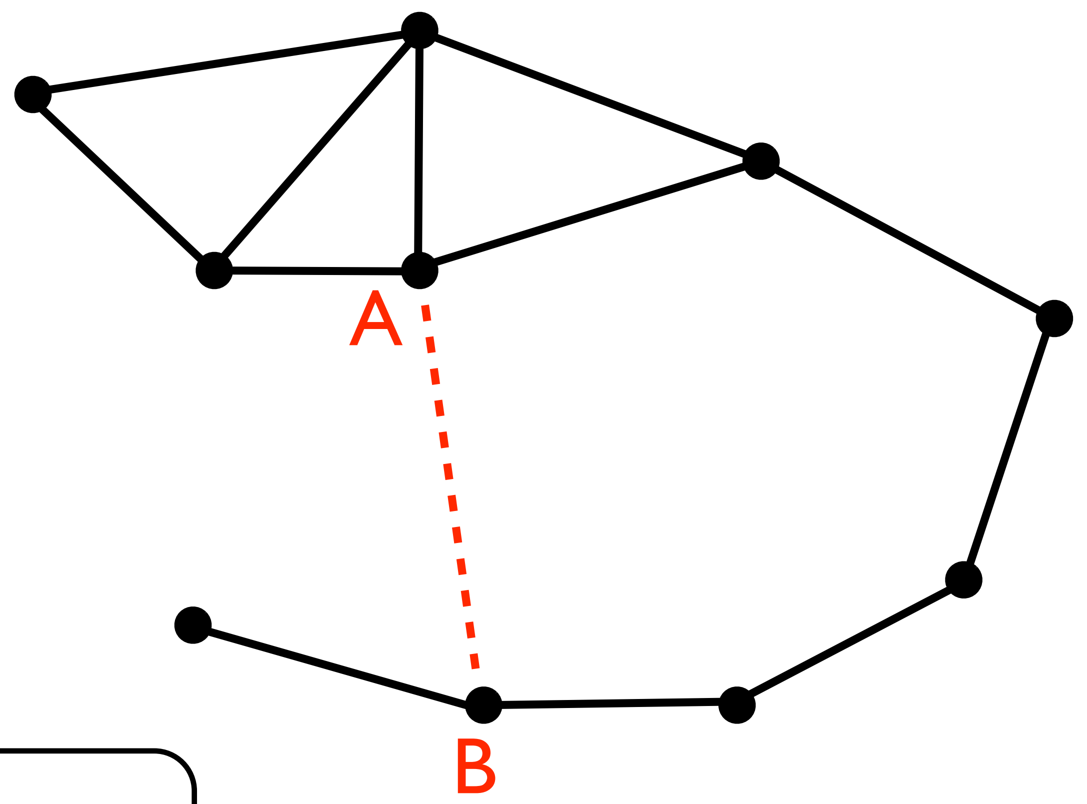


$$stretch = \frac{c(A,x_1)+\ldots+c(x_n,B)}{c(A,B)} \geq 1$$

Stretch-Faktor

- **Netzwerkgraph**
  - Knotenmenge V (|V| = n)
  - Verbindungen (Kanten)

- **Pfadkosten**
  - Kostenfunktion (z.B. RTT) $c : V^2 \mapsto \mathbb{R}_+$

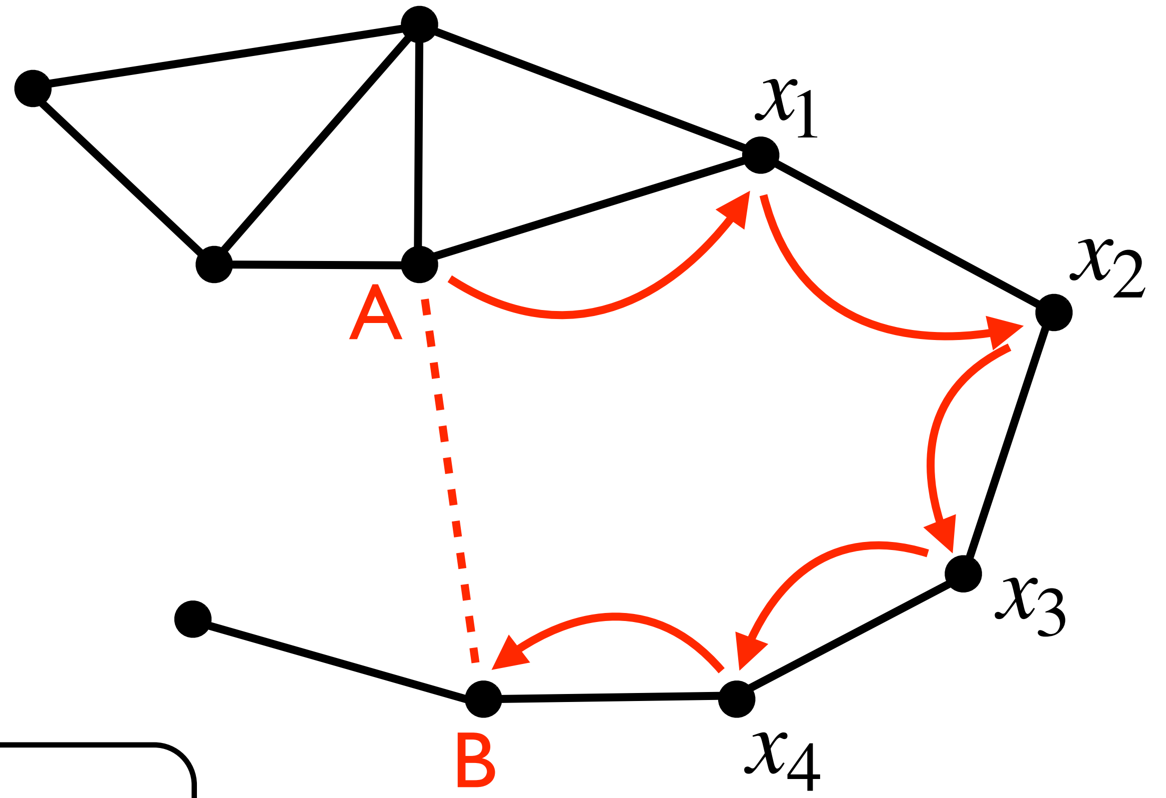


$$stretch = \frac{c(A,x_1)+\ldots+c(x_n,B)}{c(A,B)} \geq 1$$

Stretch-Faktor

- **Netzwerkgraph**
  - Knotenmenge V (|V| = n)
  - Verbindungen (Kanten)
- **Pfadkosten**
  - Kostenfunktion (z.B. RTT)

    $c : V^2 \mapsto \mathbb{R}_+$

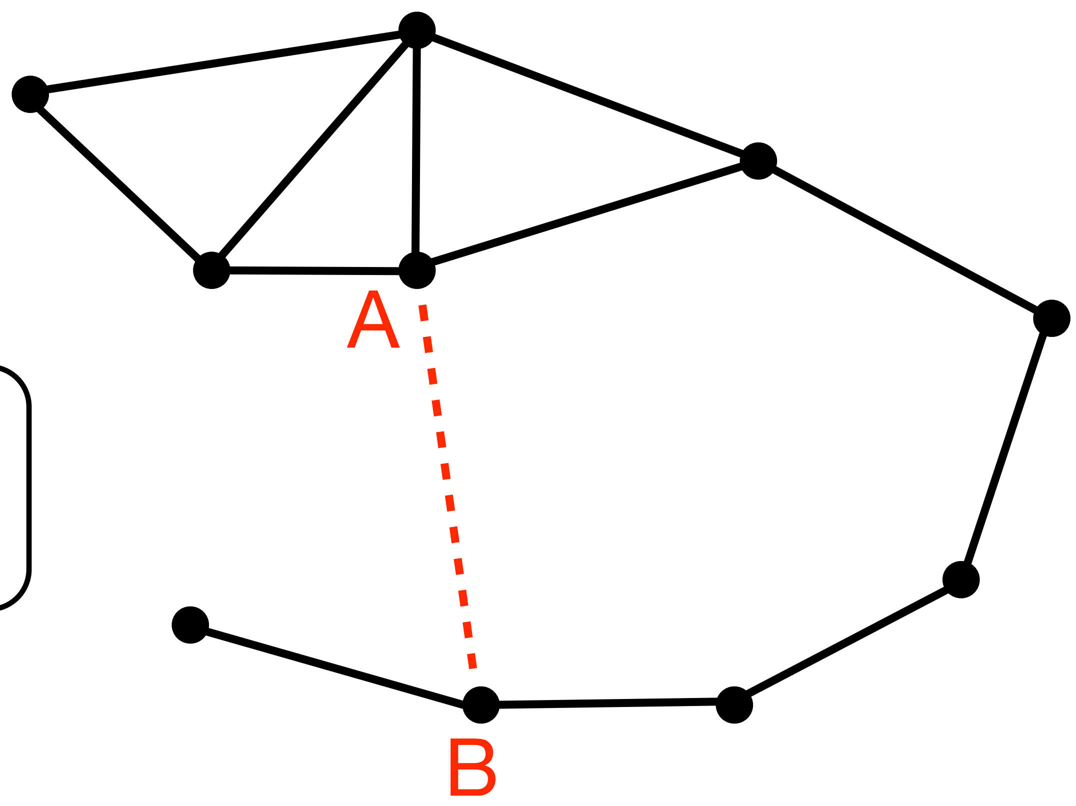


$$\boxed{stretch = \frac{c(A,x_1)+\ldots+c(x_n,B)}{c(A,B)} \geq 1}$$

Beitrag

- **Für beliebiges ε konstanter Stretch**
- **Strategie**
  - Aufbau eines adäquaten Netzwerks
  - Geeignete Wahl von Links
  - Ausgangsgrad O(log(n))

$$stretch = \frac{c(A,x_1)+\ldots+c(x_n,B)}{c(A,B)} \geq 1$$

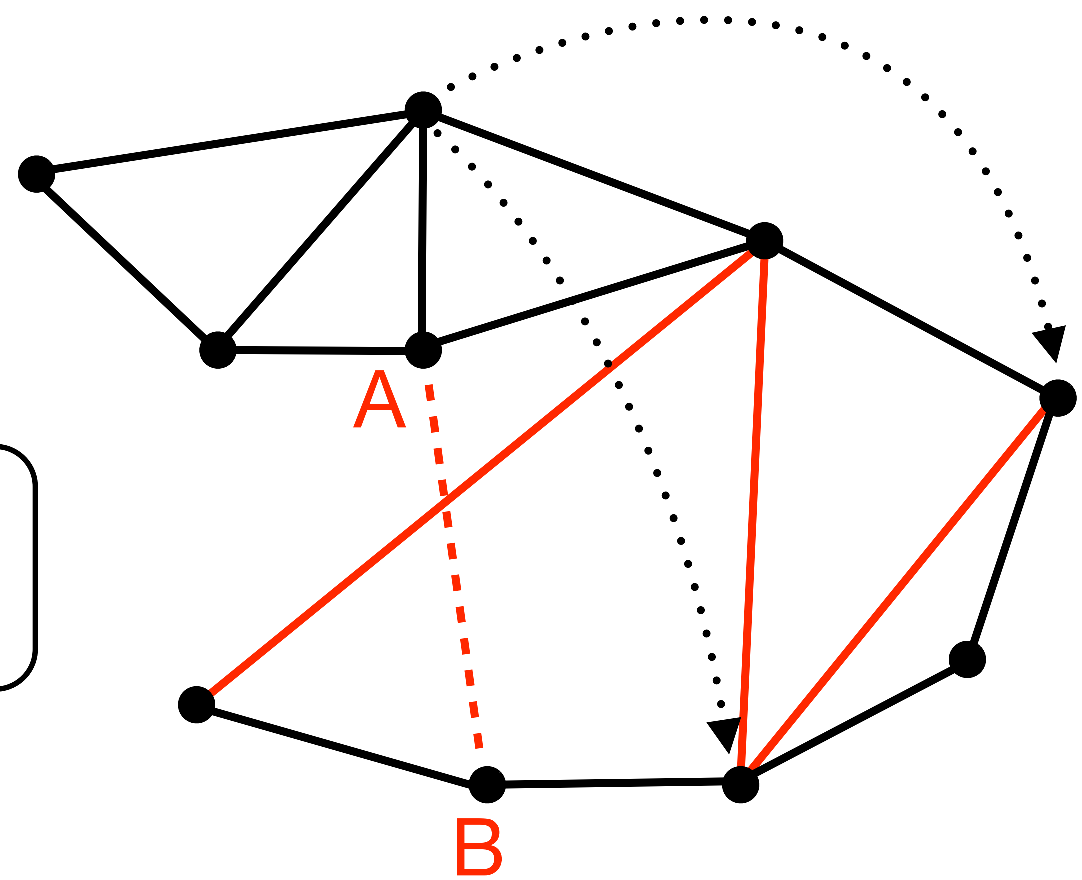


Beitrag

- **Für beliebiges ε konstanter Stretch**

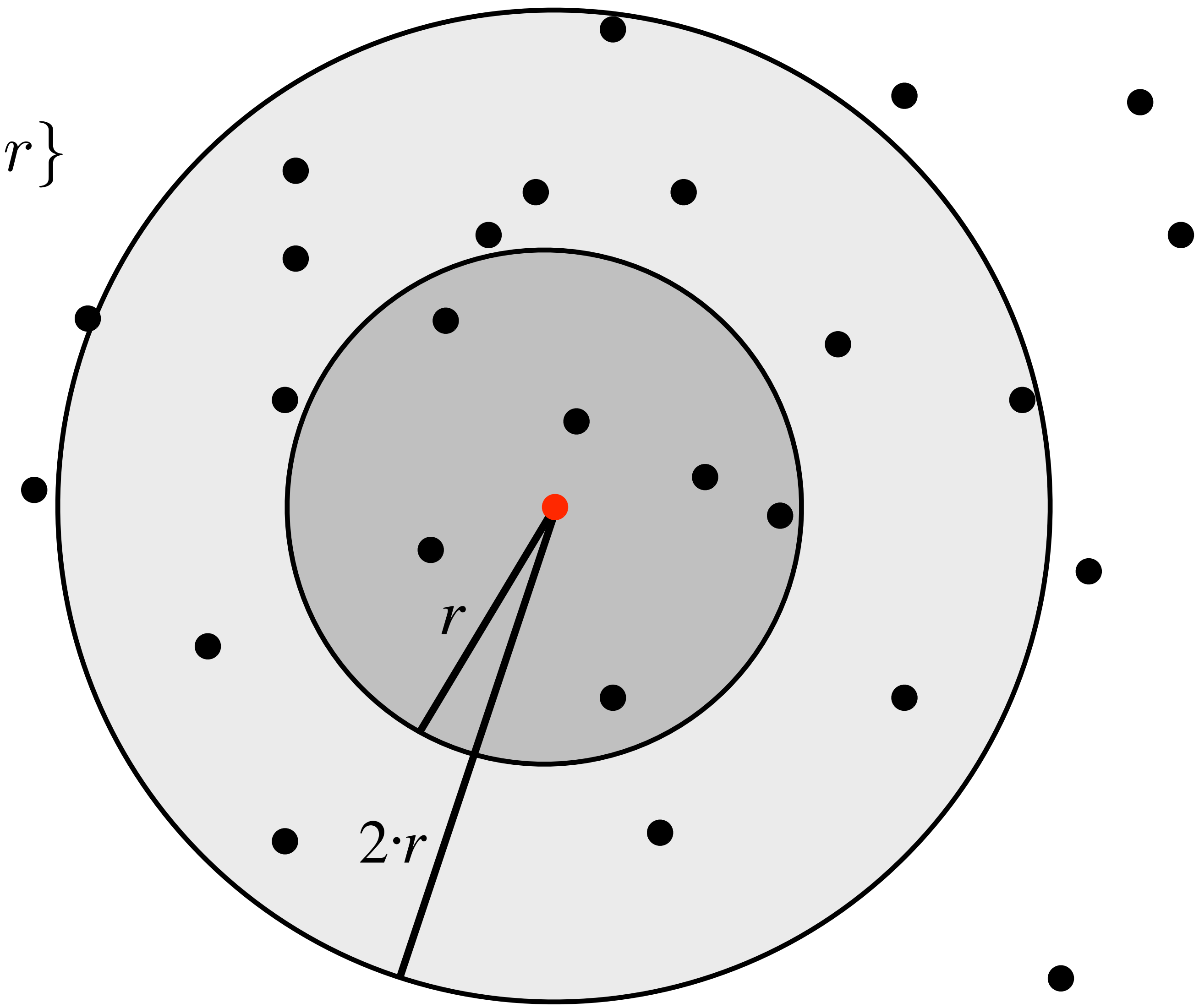
- **Strategie**
  - Aufbau eines adäquaten Netzwerks
  - Geeignete Wahl von Links
  - Ausgangsgrad O(log(n))

$$stretch = \frac{c(A,x_1)+\ldots+c(x_n,B)}{c(A,B)} \geq 1$$



Wachtumsrate

- **Knotenmenge N**
  - Knoten im Bereich
    $N(v, r) = \{w \in V | c(v, w) \leq r\}$
- **Annahme**
  - konstanter Ausdehnungsfaktor Δ
    $|N(v, 2 \cdot r)| \leq \triangle |N(v, r)|$
  - gegeben bei zufälliger Verteilung



Netzwerkstruktur

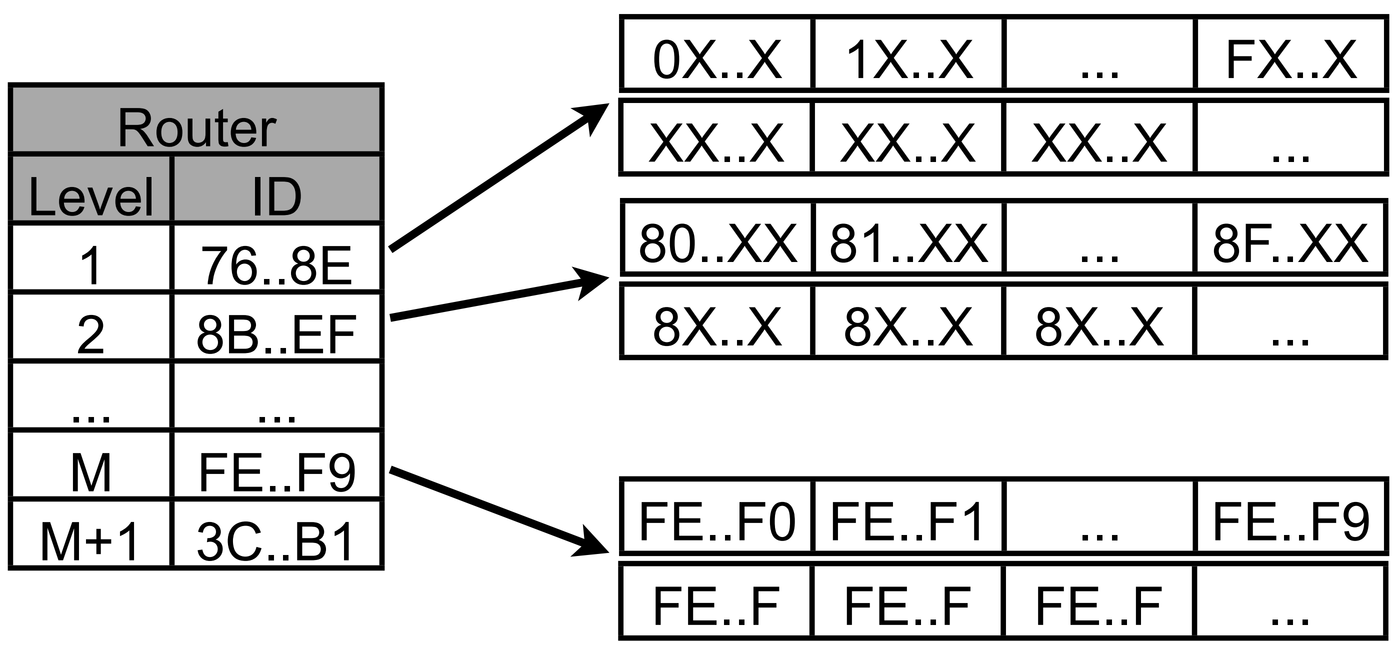
- **IDs**
  - Basis: $B = 2^b, B \geq \triangle^2$
  - Länge: $M = log_B(n)$

- **Knoten**
  - M+1 Router R mit eigener ID und Level $\ell \in \{1, ..., M+1\}$

Router	
Level	ID
1	76..8E
2	8B..EF
...	...
M	FE..F9
M+1	3C..B1

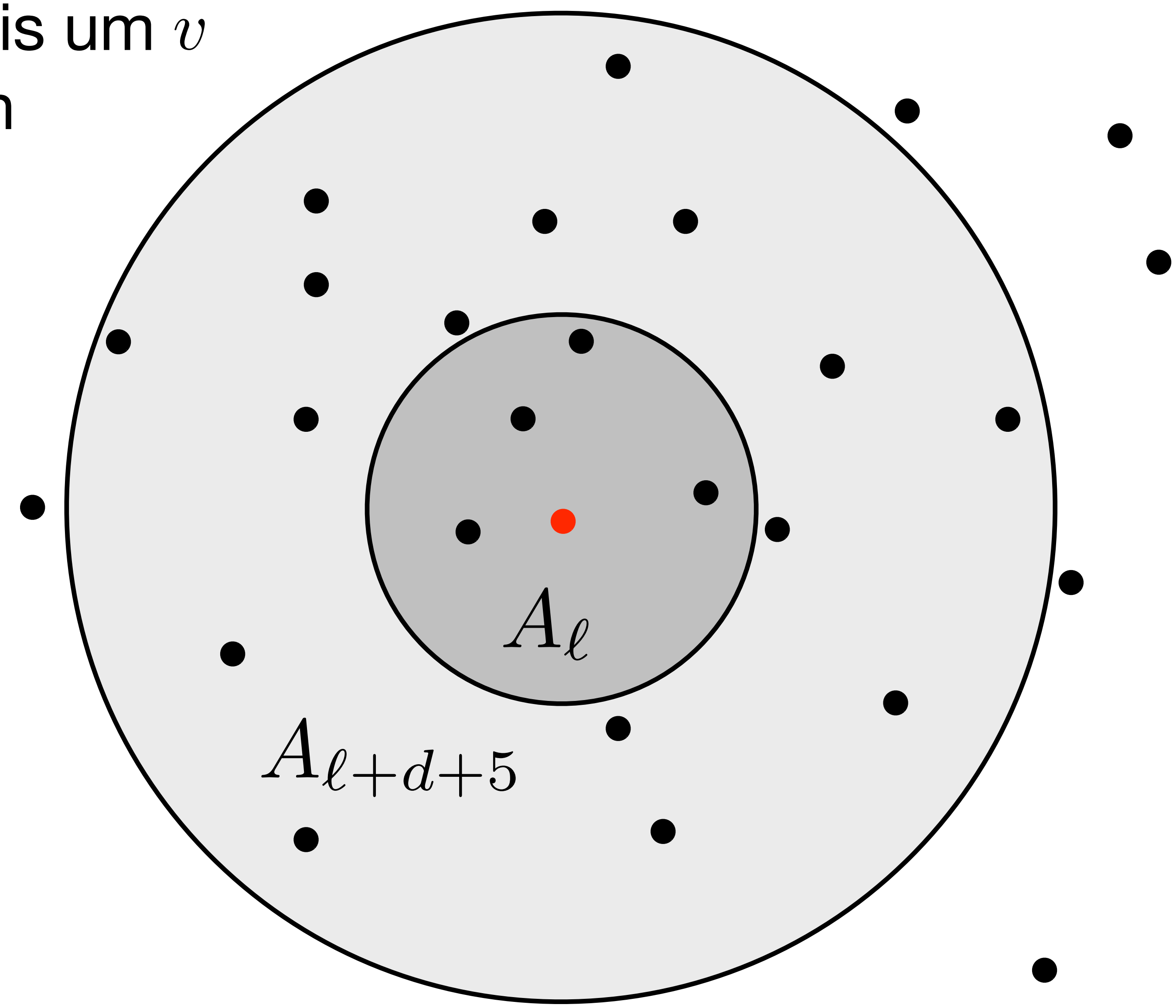
Netzwerkstruktur

- **Router**
  - B *neighbor* Links $L_0, ..., L_{B-1}$
    L_i adjustiert ℓ-te Stelle auf i
  - *publish* Links $P_1, P_2, ...$
  - L_i und P_i teilen das $(\ell - 1)$- Präfix von R.ID



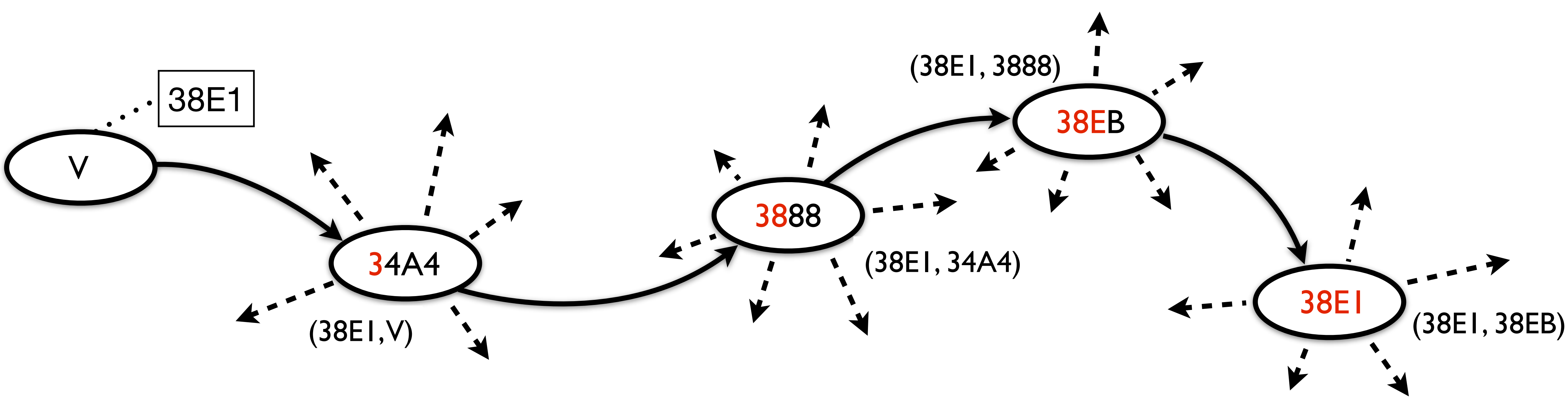
Netzwerkstruktur

- **Bereiche**
  - $A_i(v)$ ist der kleinste Kreis um v mit $min\{\alpha \cdot B^i, n\}$ Knoten
    $\alpha > ln(B)$
- **Links**
  - $R_\ell.L_i \in A_\ell$
  - $R_\ell.P_i \in A_{\ell+d+5}$
    $d = O(log(\frac{1}{\epsilon}))$
- ***shadow* Router**
  - Emulation eines Routers inkl. Links



Publish und Lookup

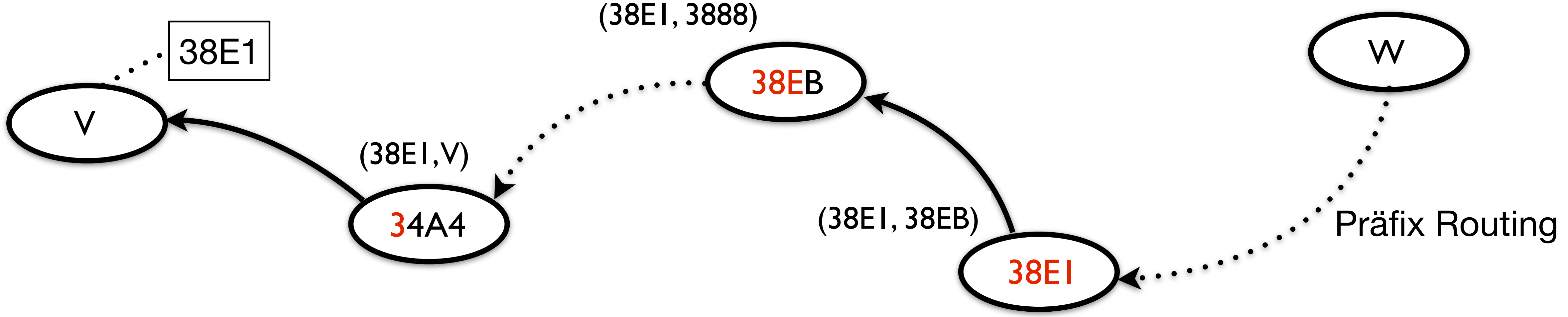
- **Publish**
  - Hashwert $\mathcal{H}(obj)$ des Objektes berechnen
  - Router R_{M+1} muss Speicherort kennen
    $R_{M+1}.ID = \mathcal{H}(obj)$
  - Publish-Route mittels Präfix-Routing zu R_{M+1}
  - Knoten auf Route Speichern Referenz



Knoten V publiziert Objekt mit $\mathcal{H}(obj) = 38E1$

Publish und Lookup

- **Lookup**
  - Hashwert $\mathcal{H}(obj)$ des Objektes berechnen
  - Präfix Routing zu Router R_{M+1}
    $R_{M+1}.ID = \mathcal{H}(obj)$
  - Referenzen folgen bis zum Speicherort



Lookup nach Objekt mit $\mathcal{H}(obj) = 38E1$

Änderungsdynamik

- **Veränderung der Netzwerk-Struktur**
  - Peers treten dem Netzwerk bei
  - Peers verlassen das Netzwerk spontan

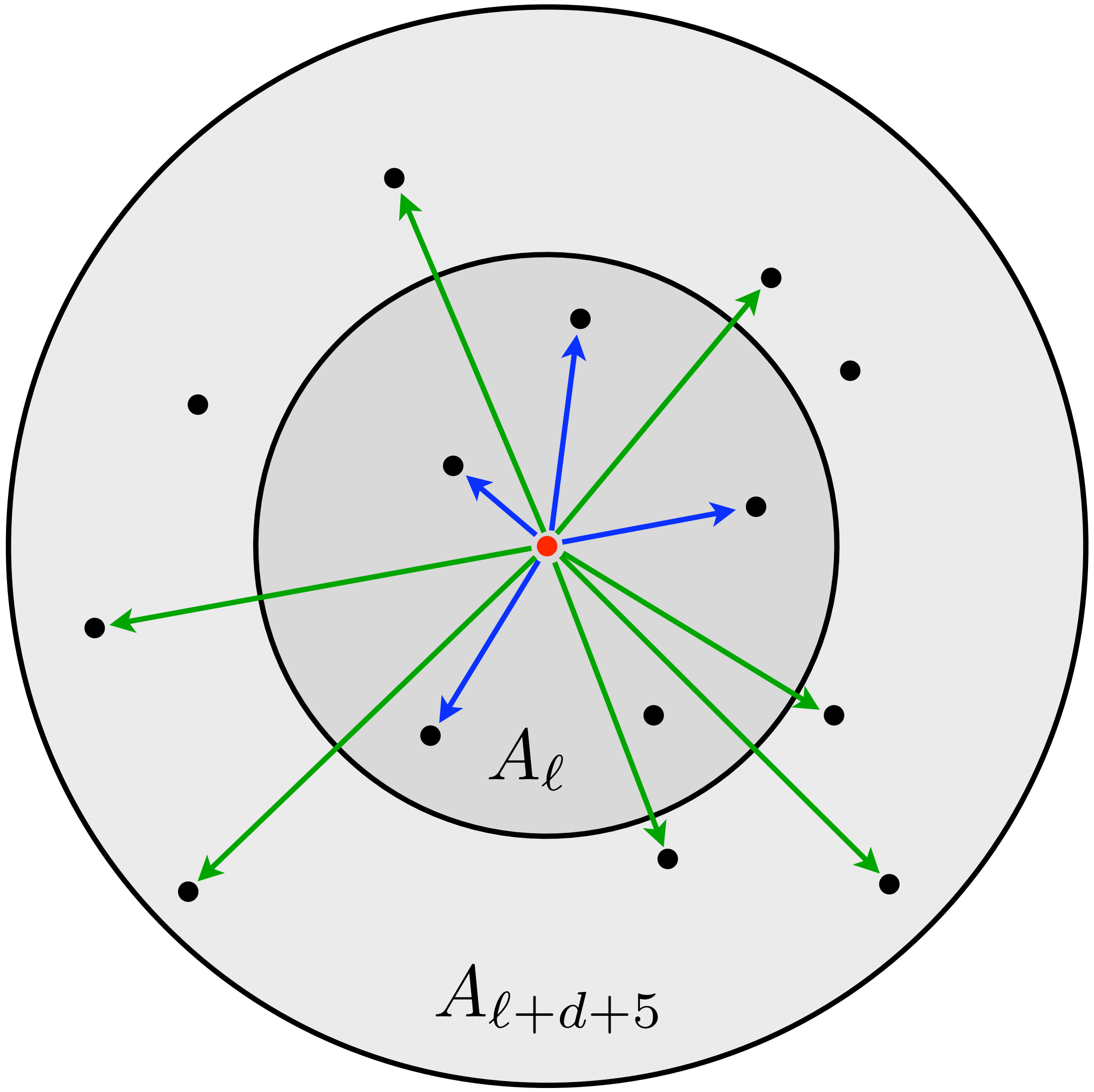
- **Anforderungen**
  - Konnektivität erhalten
  - Umstrukturierung durch Link-Updates

Änderungsdynamik

- **Knoten hinzufügen**
  - IDs für initiale Router erstellen
  - Anlegen der Links

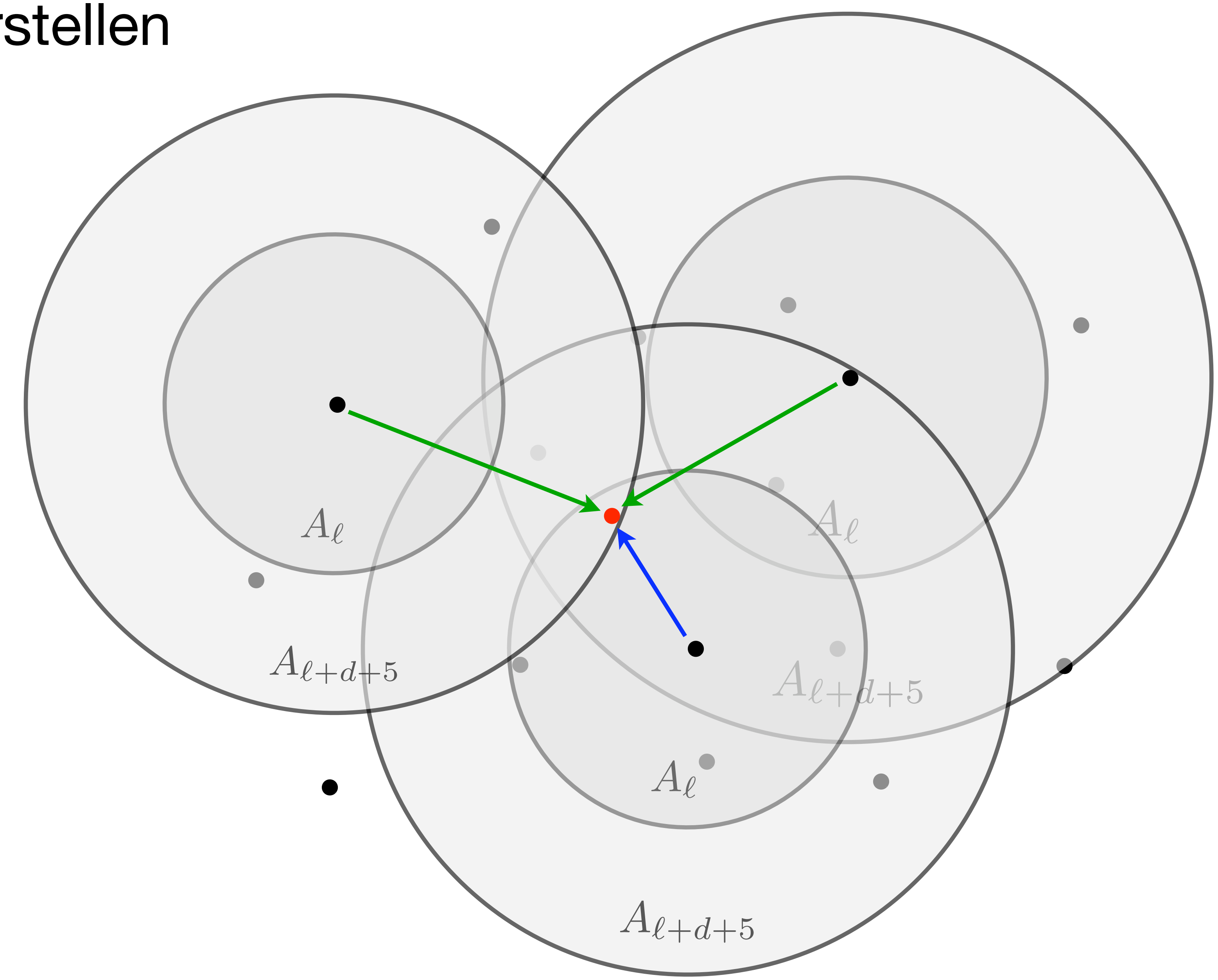
    $R_\ell.L_i \in A_\ell$

    $R_\ell.P_i \in A_{\ell+d+5}$
  - Eingehende Links?



Änderungsdynamik

- **Knoten hinzufügen**
  - IDs für initiale Router erstellen
  - Anlegen der Links
    $R_\ell.L_i \in A_\ell$
    $R_\ell.P_i \in A_{\ell+d+5}$
  - Eingehende Links?

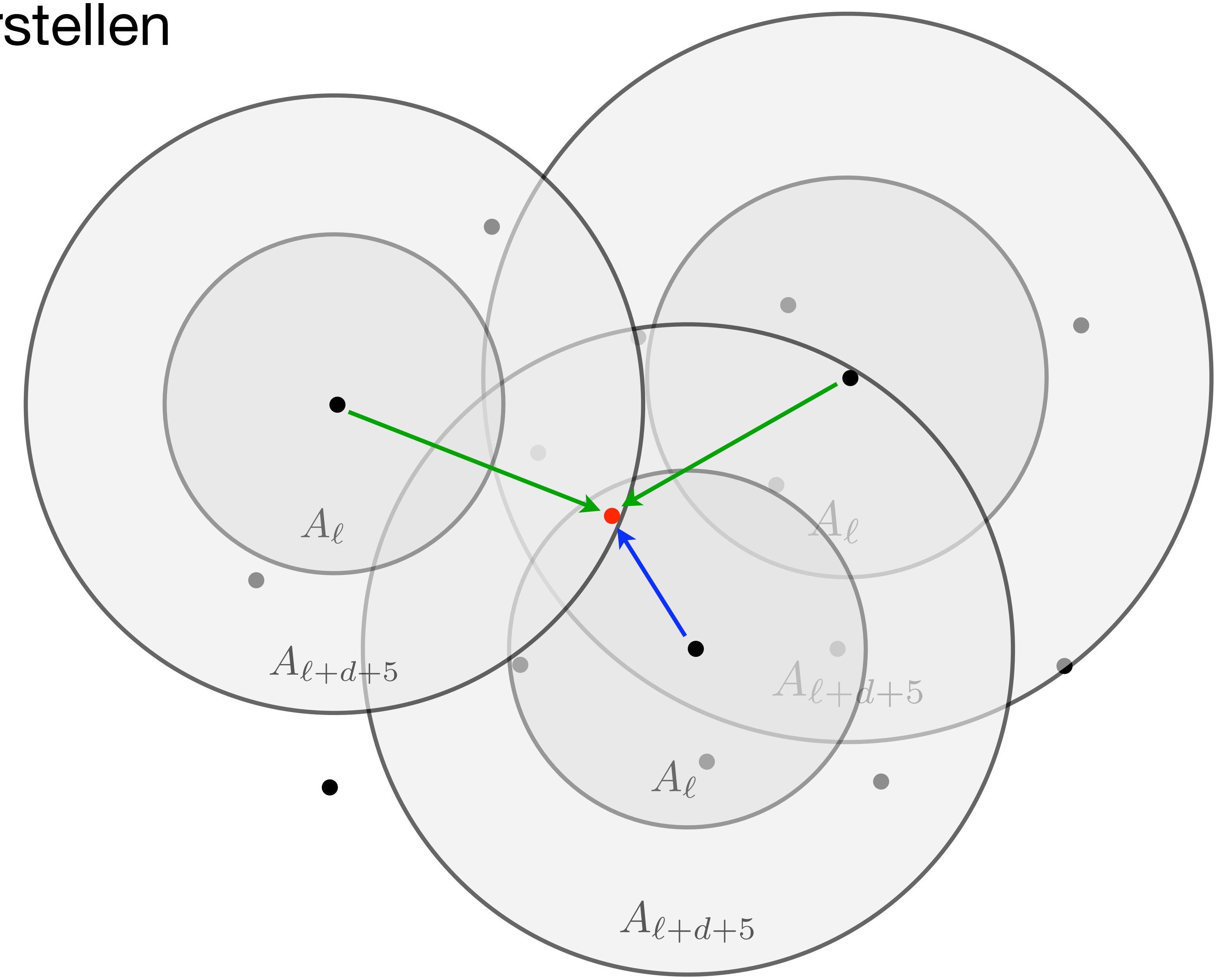


Änderungsdynamik

- **Knoten hinzufügen**
  - IDs für initiale Router erstellen
  - Anlegen der Links

    $R_\ell.L_i \in A_\ell$

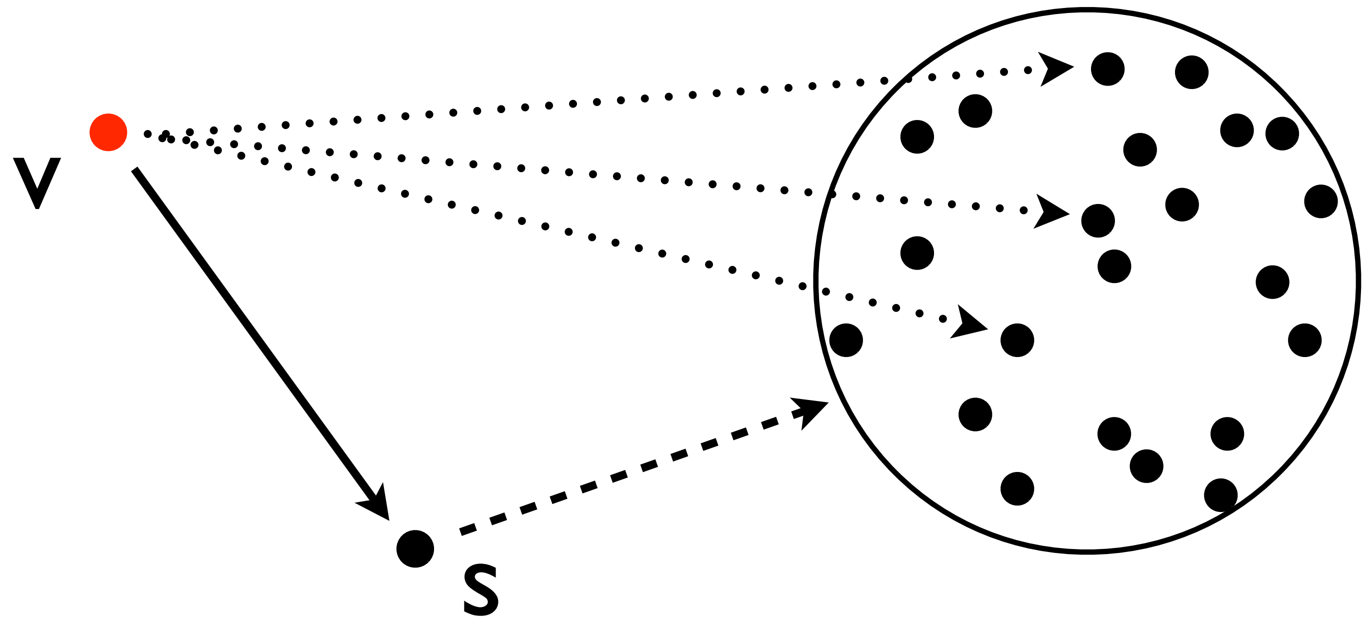
    $R_\ell.P_i \in A_{\ell+d+5}$
  - Eingehende Links?



Problem:
Geeignete Knoten müssen im *näheren Umfeld* gefunden werden

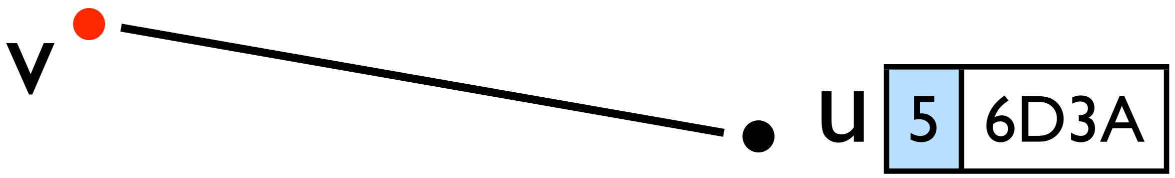
Änderungsdynamik

- **Potentiele Linkziele aus naher Umgebung**
  - Ansatz: Nutze den naheliegensten Knoten s, um potentielle Linkziele zu ermitteln



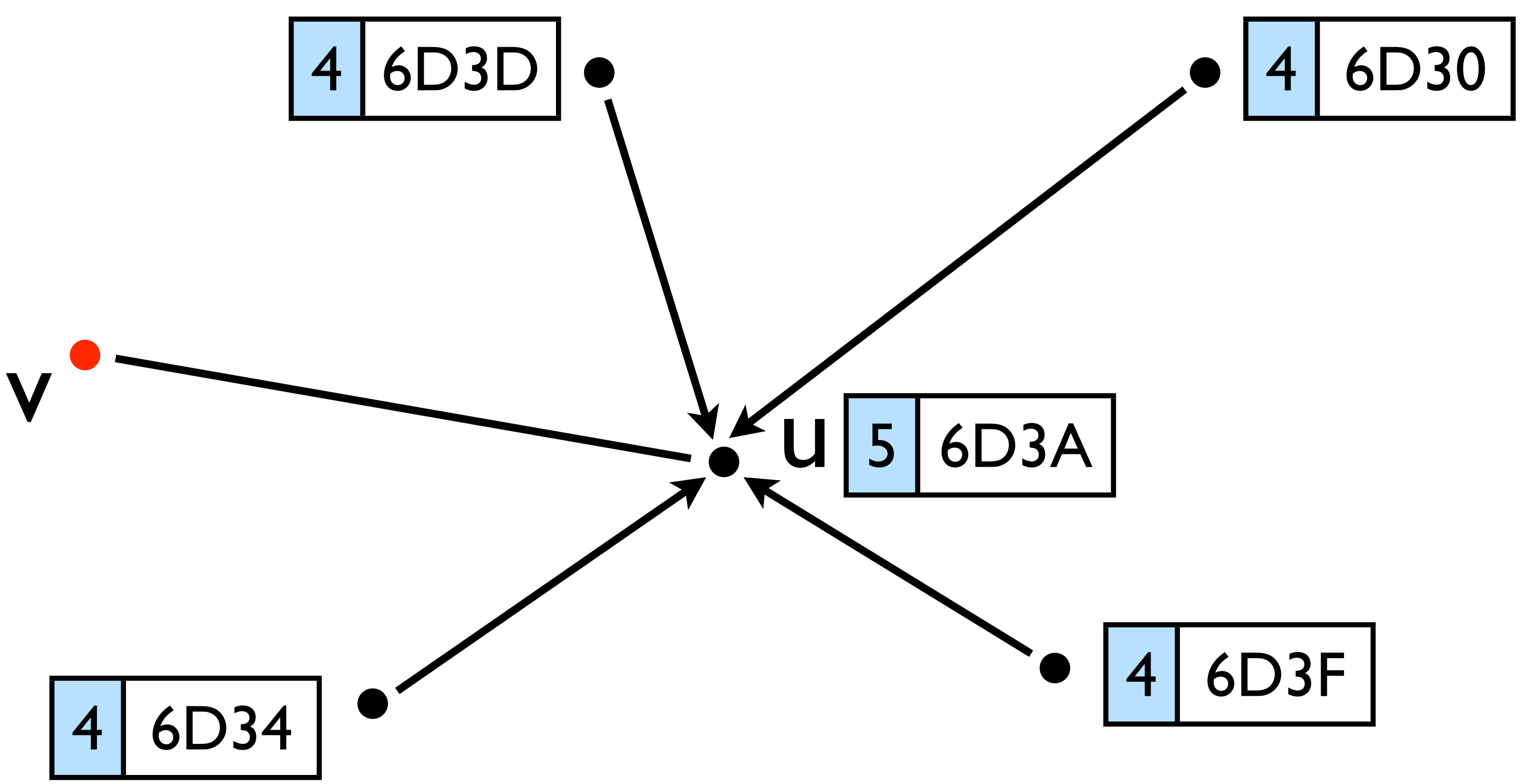
Änderungsdynamik

- **Naheliegendsten Knoten finden**
  - Beliebiger bekannter Knoten u
  - Beginne bei Router $u.R_{M+1}$ $(\ell = M+1)$
  - Wähle in jedem Schritt $\ell \geq 1$ den naheliegendsten Router $R_{\ell-1}$ der einen Link auf R_ℓ hat $argmin_{v.R_{\ell-1}}\{c(u,v)|u.R_\ell \in (v.R_{\ell-1}.L \cup v.R_{\ell-1}.P)\}$



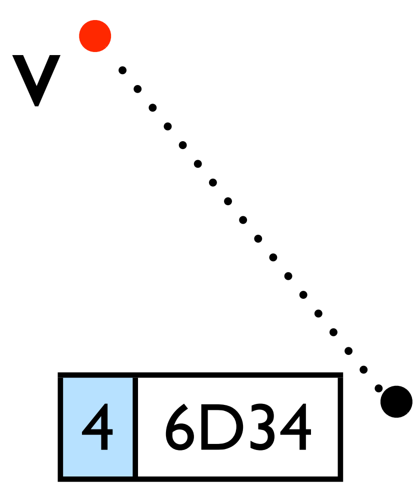
Änderungsdynamik

- **Naheliegendsten Knoten finden**
  - Beliebiger bekannter Knoten u
  - Beginne bei Router $u.R_{M+1}$ $(\ell = M+1)$
  - Wähle in jedem Schritt $\ell \geq 1$ den naheliegendsten Router $R_{\ell-1}$ der einen Link auf R_ℓ hat $argmin_{v.R_{\ell-1}}\{c(u,v)|u.R_\ell \in (v.R_{\ell-1}.L \cup v.R_{\ell-1}.P)\}$



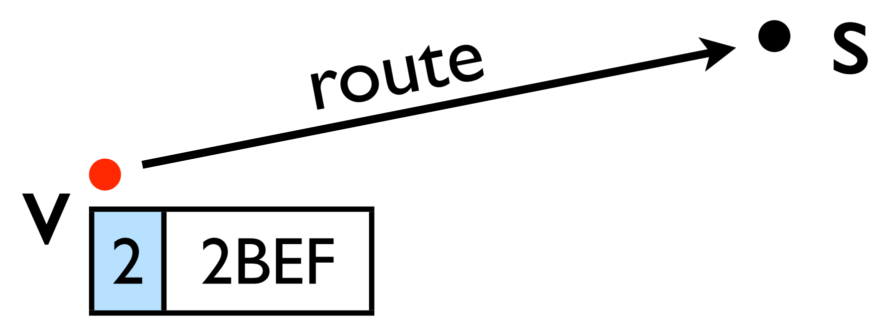
Änderungsdynamik

- **Naheliegendsten Knoten finden**
  - Beliebiger bekannter Knoten u
  - Beginne bei Router $u.R_{M+1}$ $(\ell = M+1)$
  - Wähle in jedem Schritt $\ell \geq 1$ den naheliegendsten Router $R_{\ell-1}$ der einen Link auf R_ℓ hat
    $argmin_{v.R_{\ell-1}}\{c(u,v)|u.R_\ell \in (v.R_{\ell-1}.L \cup v.R_{\ell-1}.P)\}$



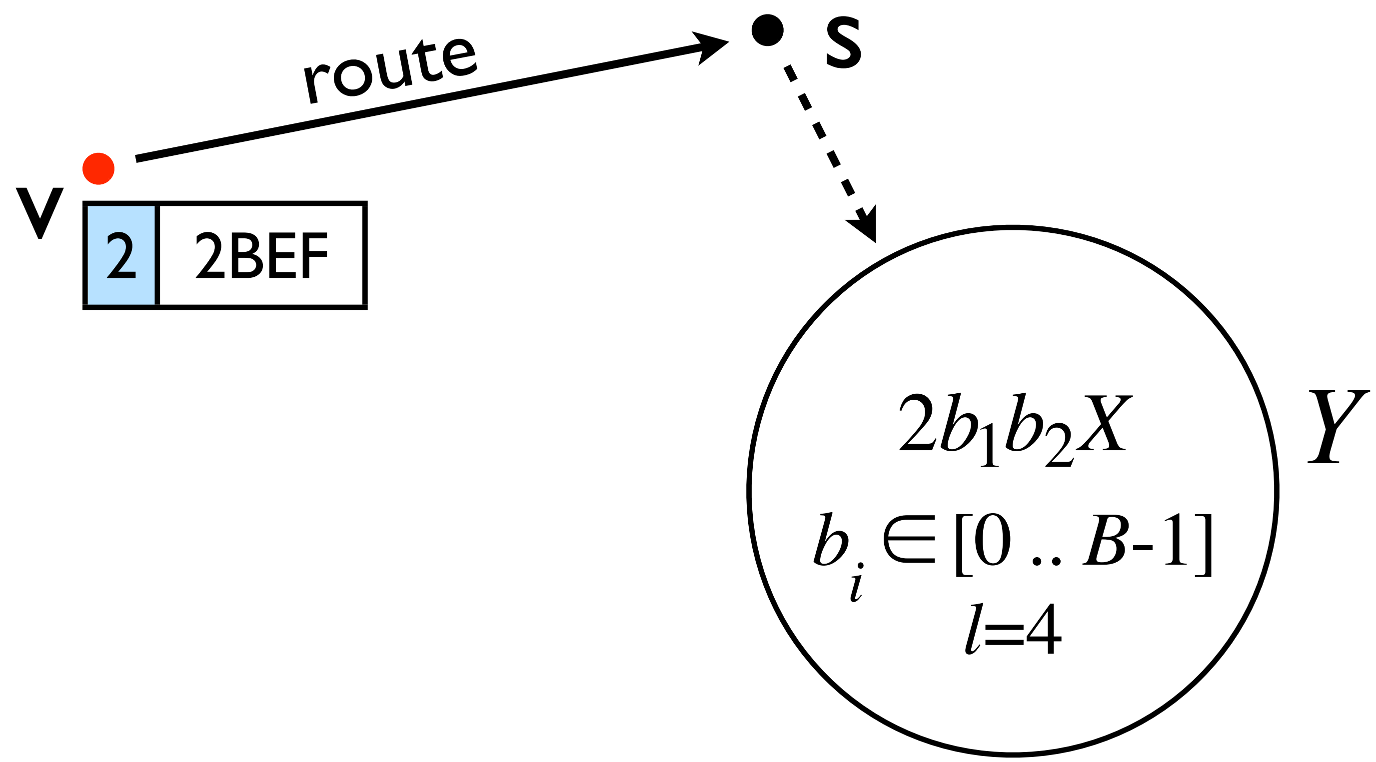
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



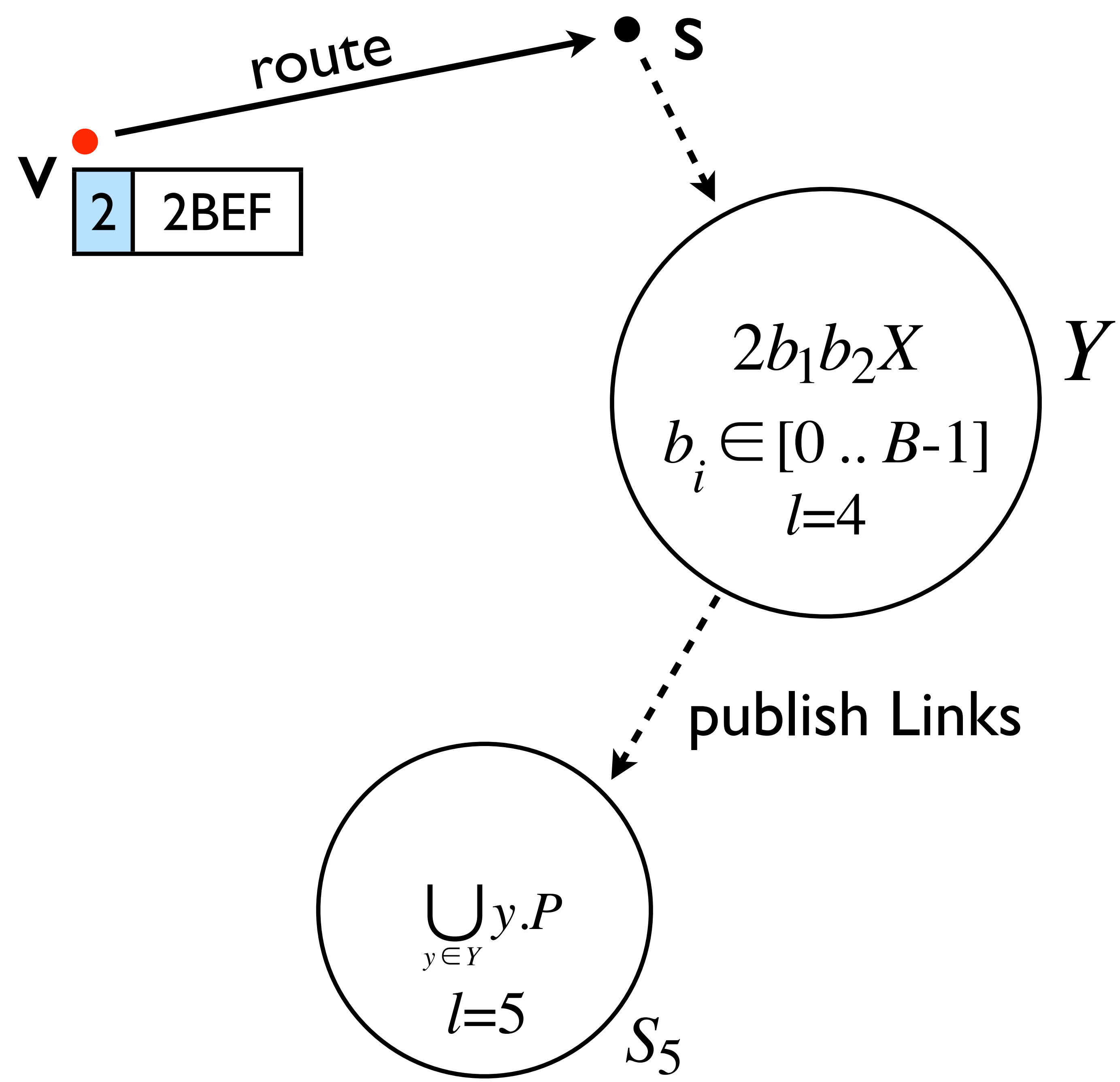
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



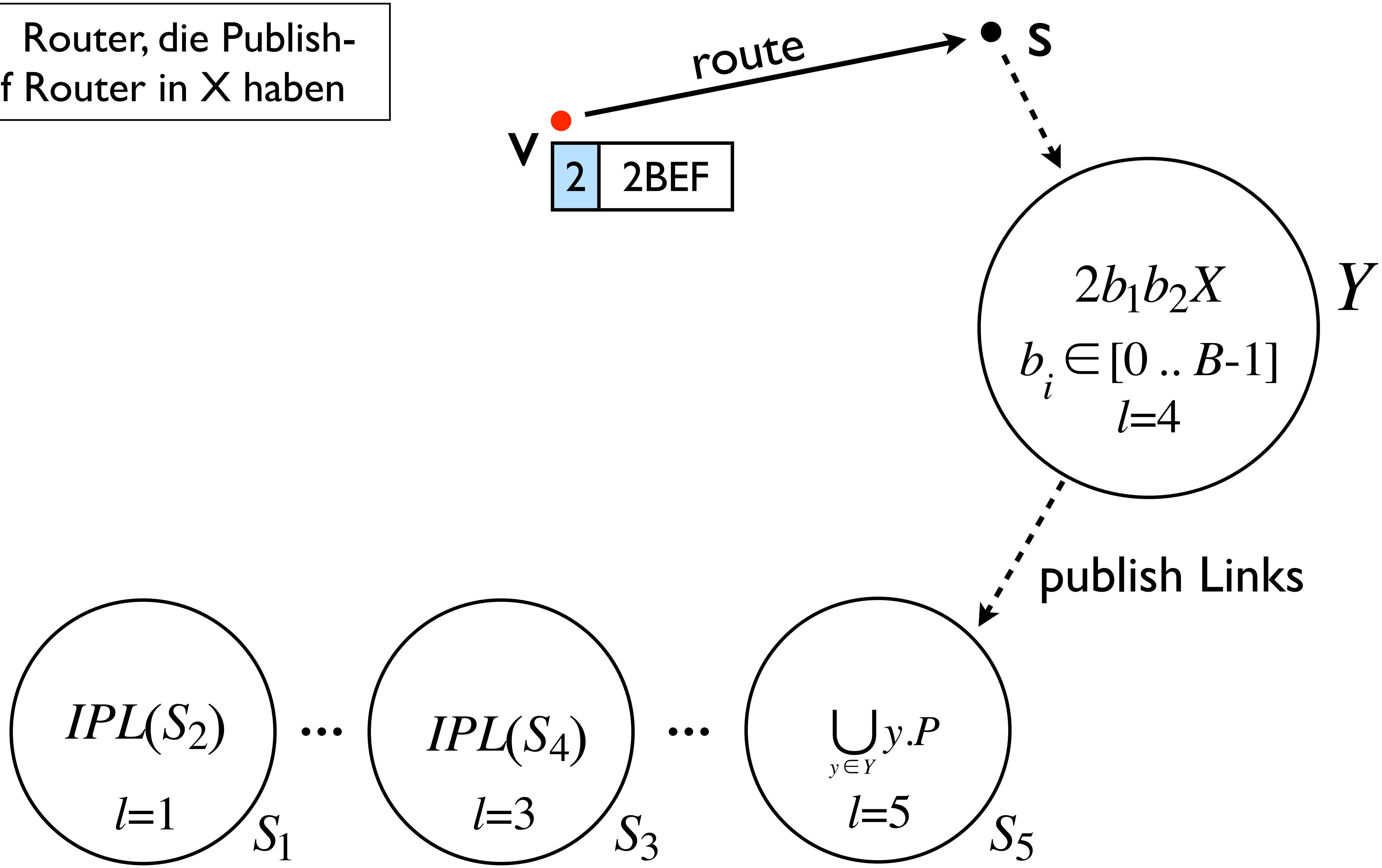
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



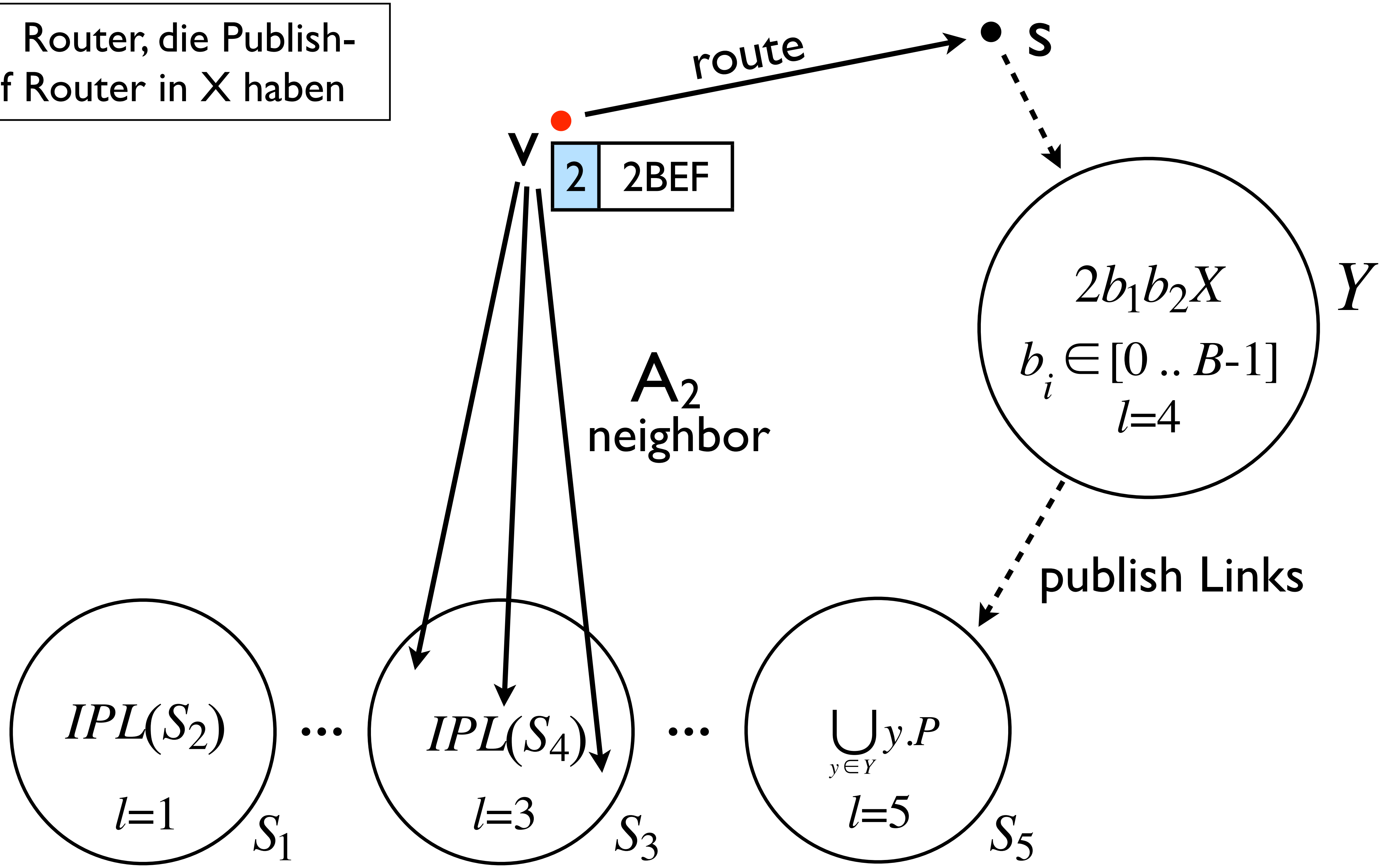
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



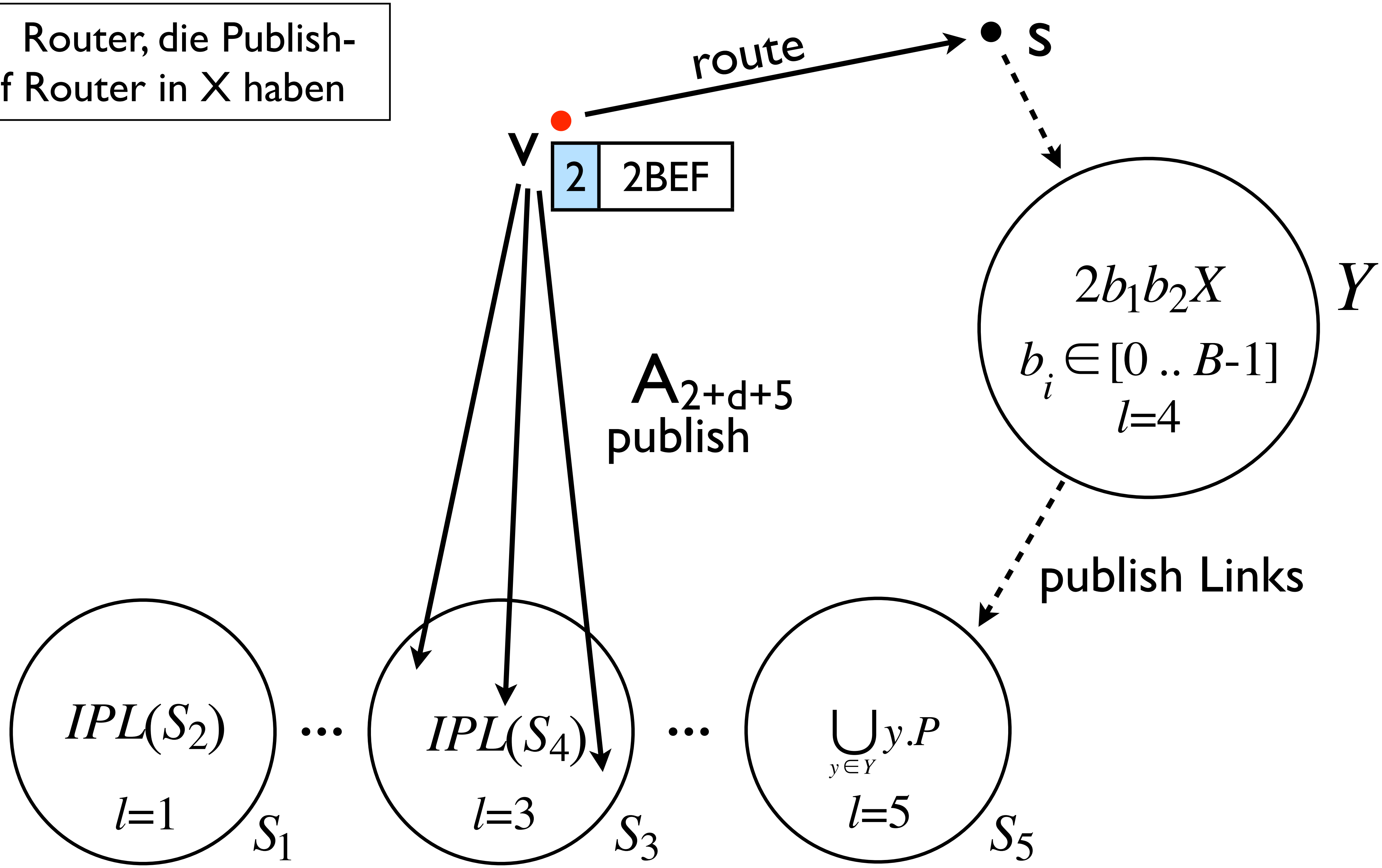
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



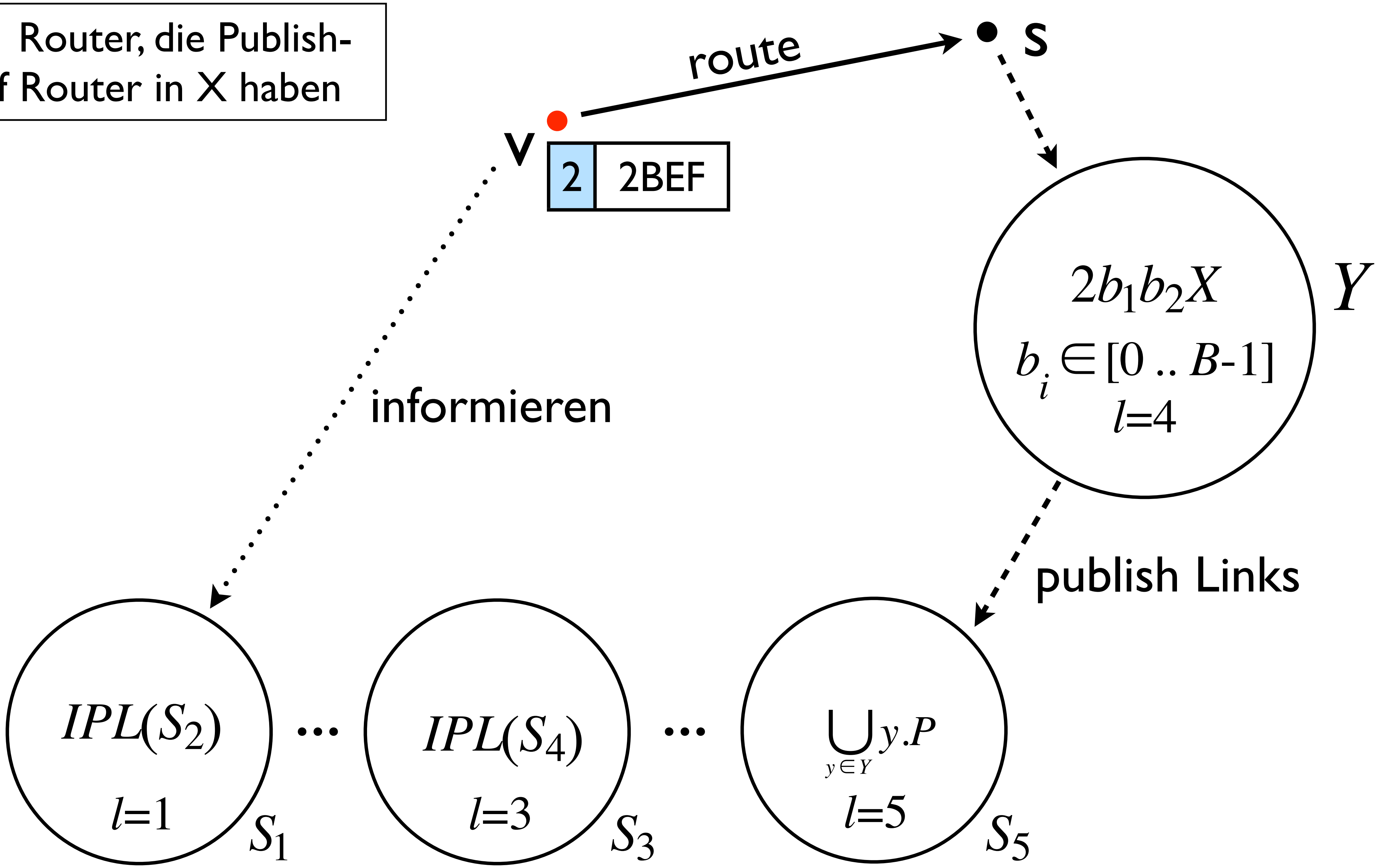
Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



Änderungsdynamik

IPL(X) = Router, die Publish-Links auf Router in X haben



Änderungsdynamik

- **Knoten entfernen**
  - Erneuerung der Links verlinkender Router
  - Alternativrouter aus *publish*-Link Pool
  - Erkennungsmechanismus?

Analyse

- **Beweisskizze für (1+ε) Stretch**
  - Betrachte Pfad von Knoten s nach Knoten t

$$s = \underbrace{x_1, x_2, x_3, ..., x_k}_{A}\underbrace{,\ }_{B}\underbrace{w_{k-1}, w_{k-2}, ..., w_1}_{C} = t$$

Präfix Routing nach $\mathcal{H}(obj)$

Folgen der Referenzen

Analyse

- **Kosten der Route A**

  $$\sum_{i=1}^{k-1} c(x_i, x_{i+1}) \leq \frac{\gamma}{\gamma-1} \cdot a_{i+1}(s) \leq \frac{2\cdot\gamma^{1-d}}{\gamma-1} \cdot c(s,t)$$

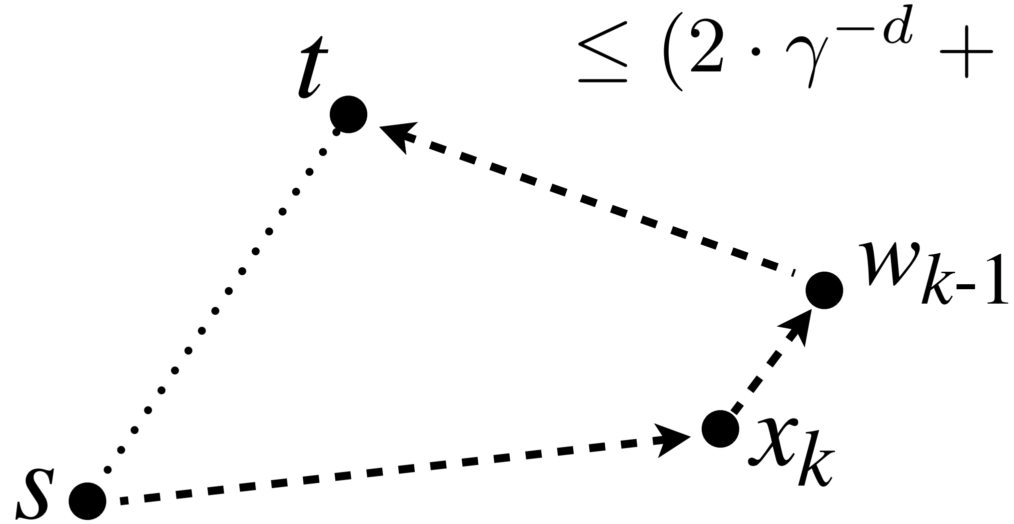
- **Kosten der Route C**

  $$\sum_{k-1}^{1} c(w_i, w_{i+1}) \leq \frac{\gamma^{-d}}{\gamma-1} \cdot c(s,t)$$

  $$\boxed{\gamma = B^{log_{\triangle} 2}}$$

- **Kosten der Route B**
  - Abgeschätzt mit Dreiecksungleichung

  $$\begin{aligned} c(x_k, w_{k-1}) &\leq a_{k+1}(s) + c(s,t) + a_k(t) \\ &\leq (2 \cdot \gamma^{-d} + 1 + \gamma^{-d-1}) \cdot c(s,t) \end{aligned}$$



$$\boxed{x_i \in A_i(x_{i-1}) \subseteq A_{i+1}(s)}$$

Analyse

- **Kumulierte Kosten**
  $(1 + (\frac{3}{\gamma - 1} + \frac{1}{\gamma} + 4) \cdot \gamma^{-d}) \cdot c(s, t)$

- **Wähle geeignetes d**
  $$\epsilon \geq \frac{1}{\gamma^d} \cdot (\frac{2 \cdot \gamma}{\gamma - 1} + 2 + \frac{1}{\gamma} + \frac{1}{\gamma - 1})$$
  $$= (\frac{3}{\gamma - 1} + \frac{1}{\gamma} + 4) \cdot \gamma^{-d}$$

Analyse

- **Kumulierte Kosten**
  $(1 + \boxed{(\frac{3}{\gamma-1} + \frac{1}{\gamma} + 4) \cdot \gamma^{-d}}) \cdot c(s,t)$

- **Wähle geeignetes d**
  $$\epsilon \geq \frac{1}{\gamma^d} \cdot (\frac{2\cdot\gamma}{\gamma-1} + 2 + \frac{1}{\gamma} + \frac{1}{\gamma-1})$$
  $$= \boxed{(\frac{3}{\gamma-1} + \frac{1}{\gamma} + 4) \cdot \gamma^{-d}}$$

Analyse

- **Anzahl der Links**

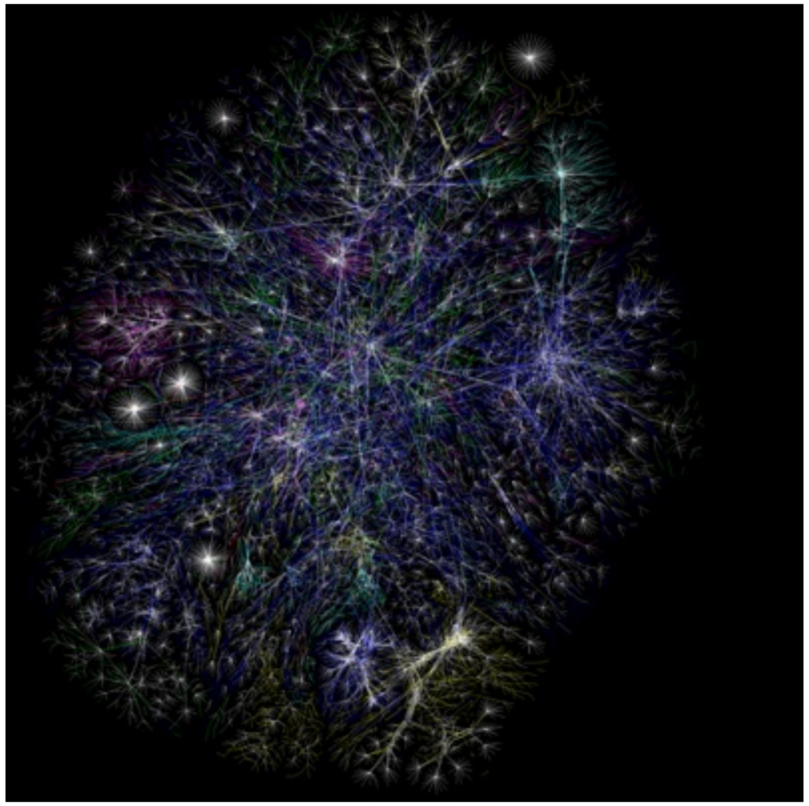
$$|L| = (M+1) \cdot B = B + B \cdot log_B(n)$$

$$|P| = \frac{\alpha \cdot B^{d+6}}{1 - B \cdot e^{-\alpha}} \cdot log_B(n)$$

- **Der Ausgangsgrad eines Knotens kann sehr hoch sein!**

Fazit

- **Idealisiertes Netzwerkmodell**
  - Stretch kann auf theoretischer Basis bewiesen werden
  - Konstanter Ausdehnungsfaktor unrealistisch
- **Grundlegende Annahmen**
  - Dreiecksungleichung gilt <u>nicht</u>
- **Ungeklärte Probleme**
  - *shadow router* Konstrukt
  - Erkennen von fehlenden Knoten
  - Sinnvolle Netzwerkgröße



Vielen Dank für Ihre Aufmerksamkeit!