

Distribuované systémy a výpočty

X36DSV

Jan Janeček
(dnes Peter Macejko)



P2P – Distribuce dat

Požadavky

- decentralizace
- škálovatelnost
- rovnoměrné rozložení dat a zátěže
- rychlé vyhledávání a výběr
- dynamická architektura
- obecnost – nezávislost na identifikaci

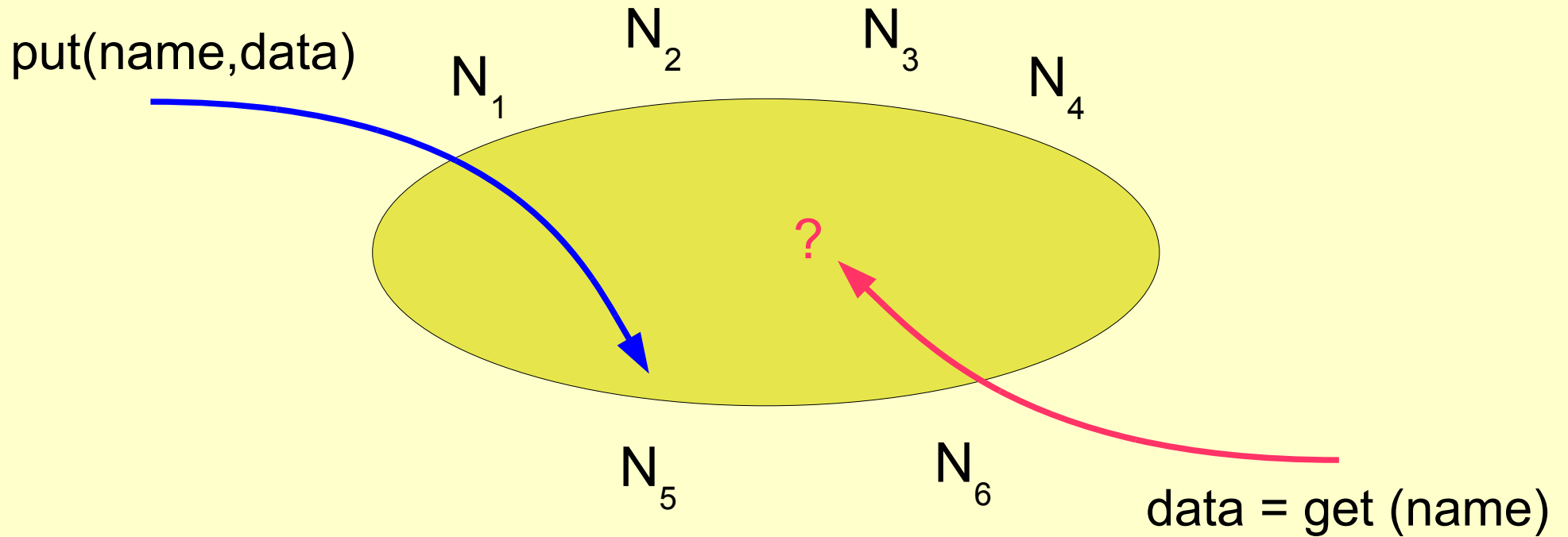
Řešení

- Chord – kruh
- Pastry/Tapestry – strom
- CAN - hyperkrychle



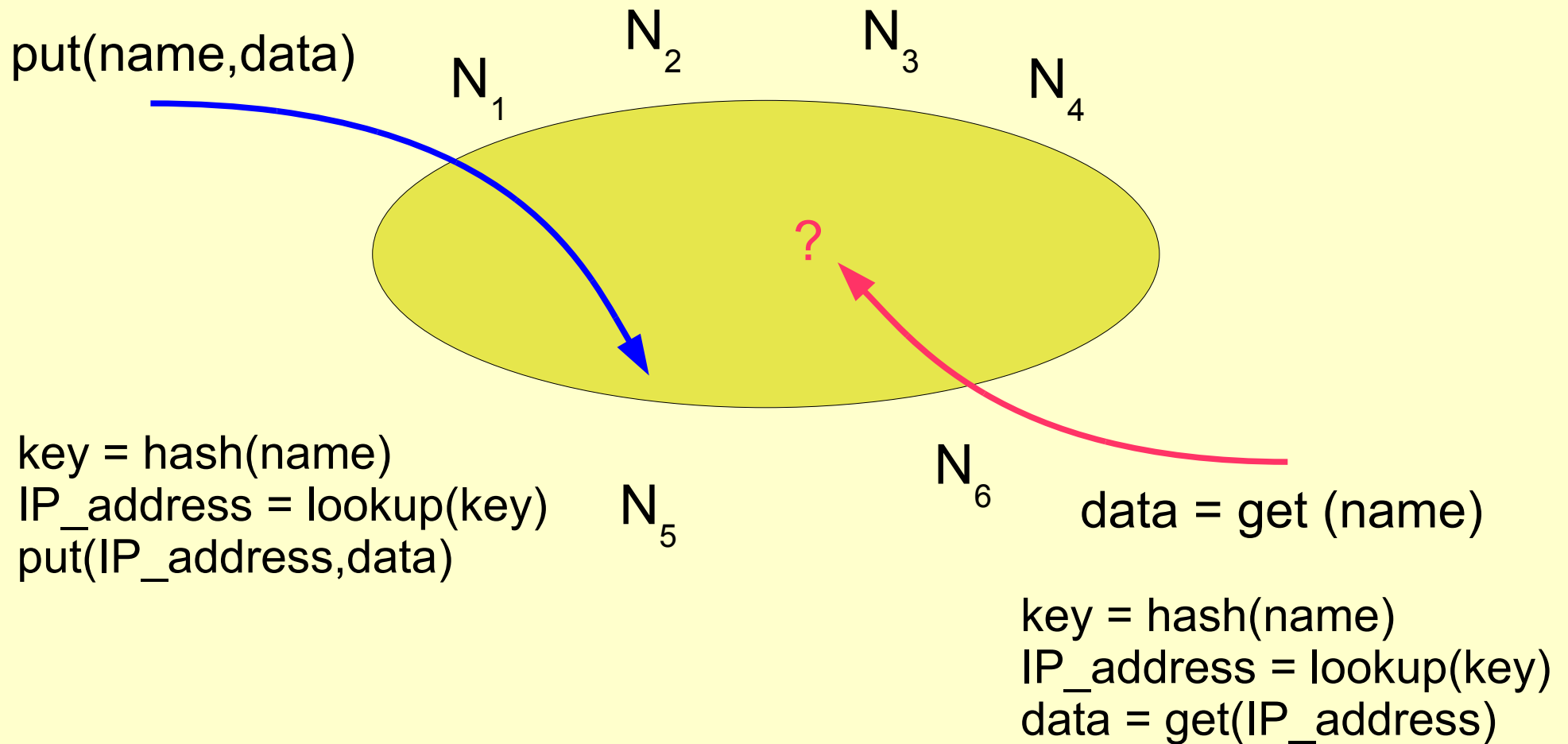
P2P – Vyhledávání

Problém



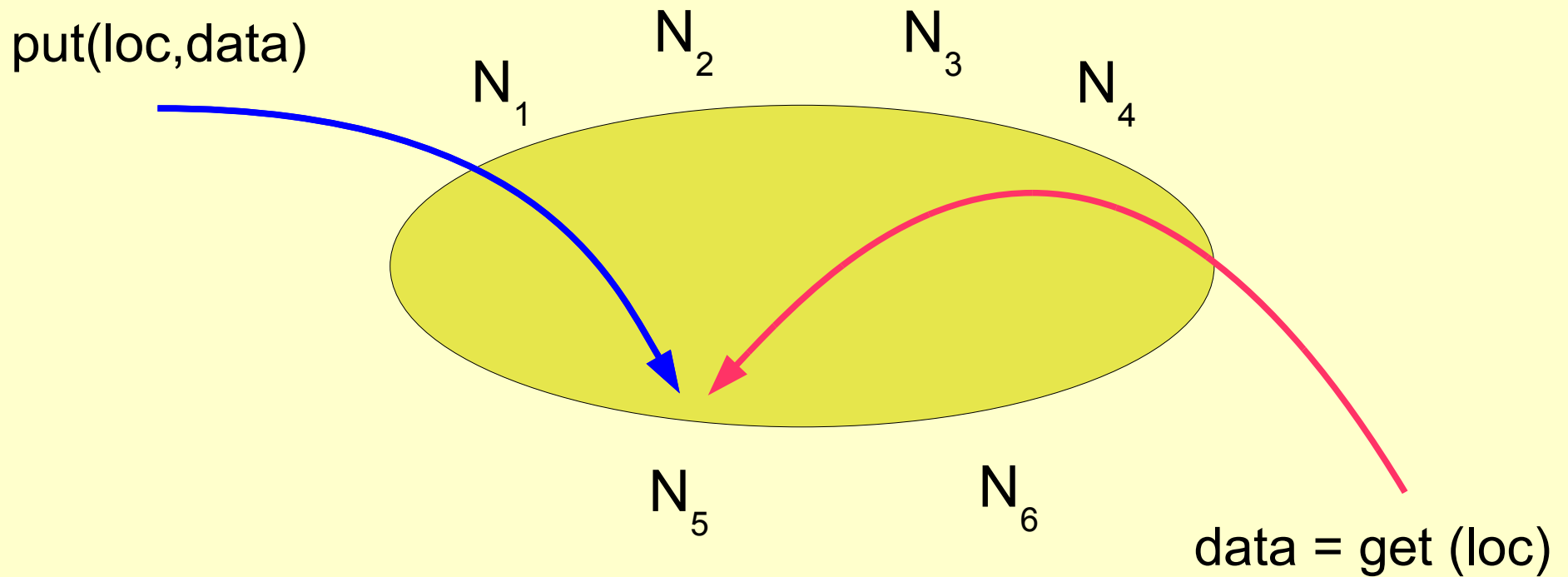
P2P – Vyhledávání

Problém



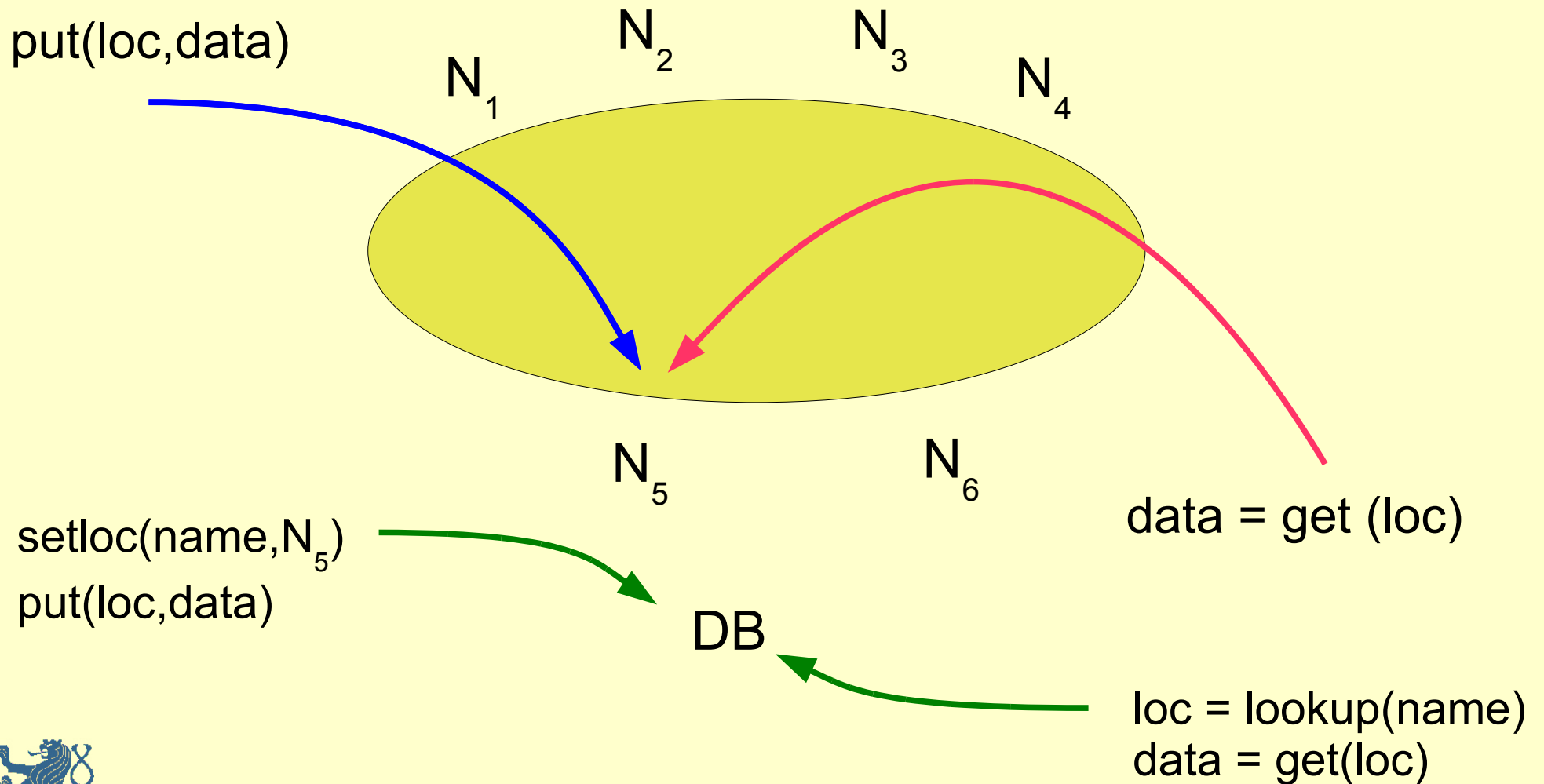
P2P – Vyhledávání

Centralizovaný adresář - Napster



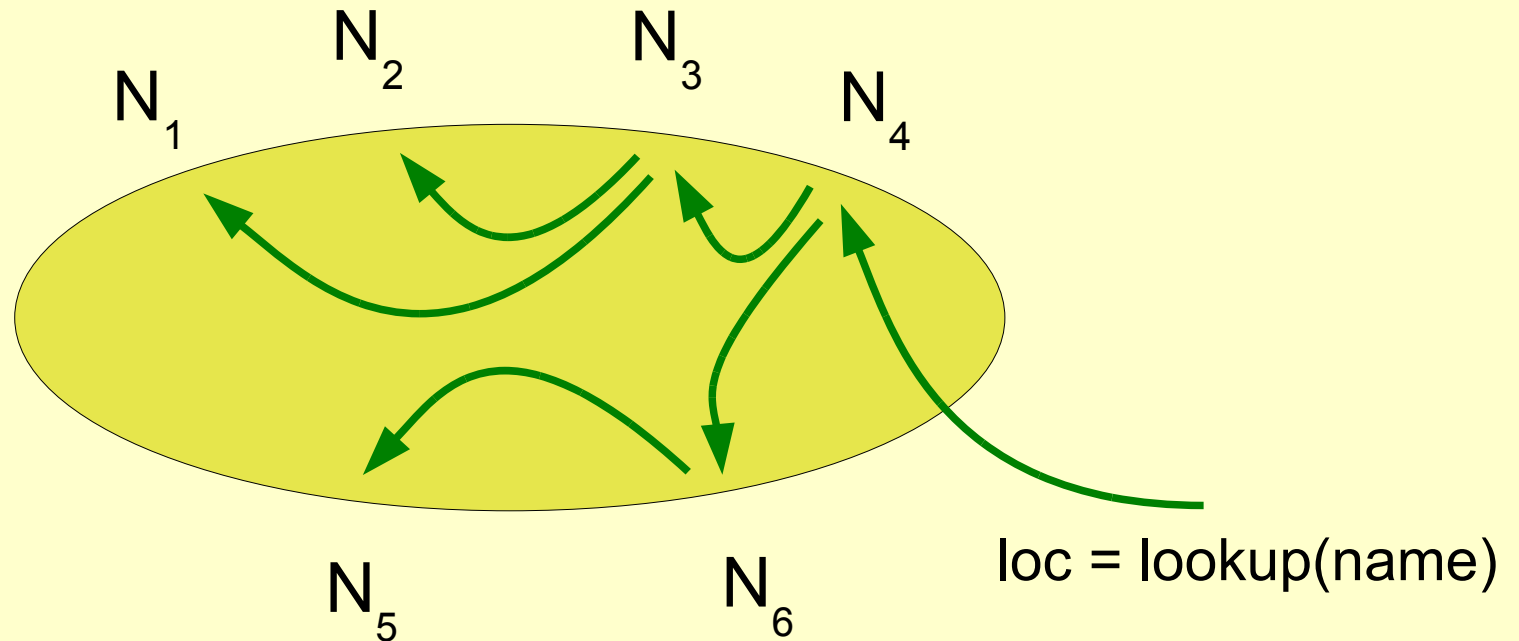
P2P – Vyhledávání

Centralizovaný adresář - Napster



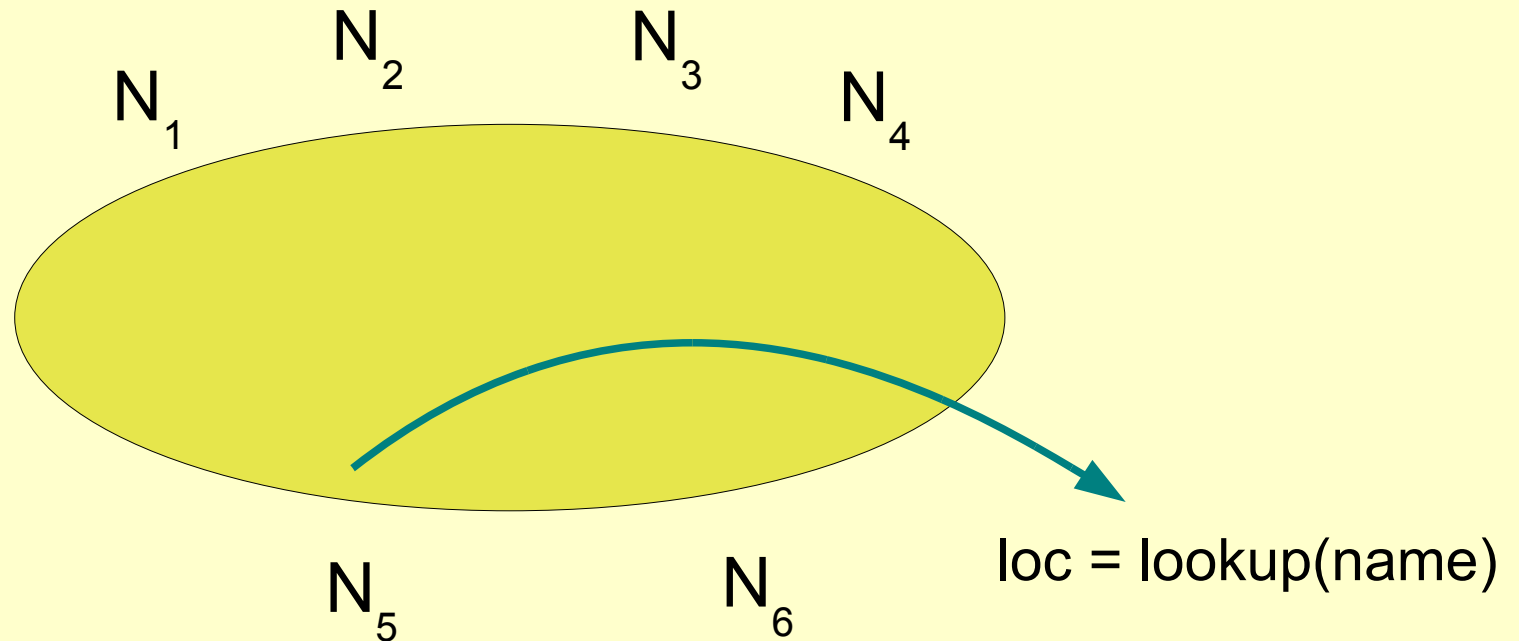
P2P – Vyhledávání

Broadcast - Gnutela



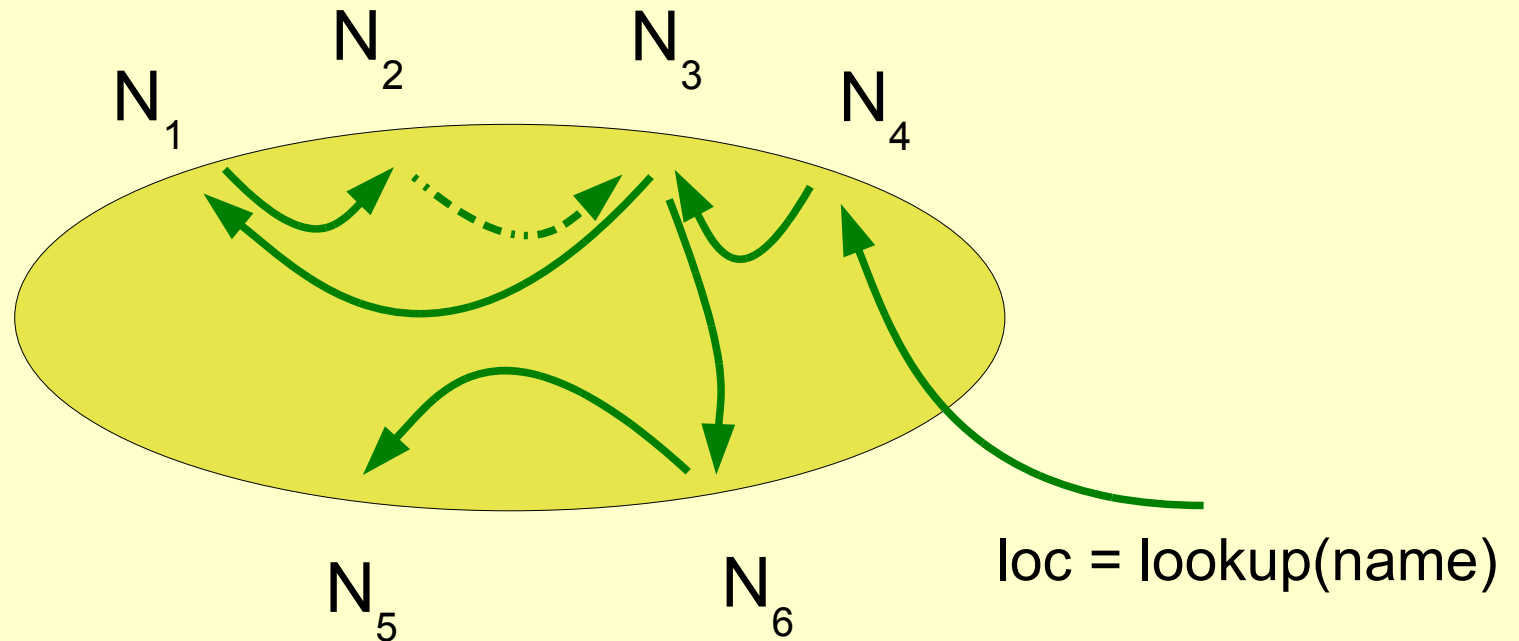
P2P – Vyhledávání

Broadcast - Gnutella



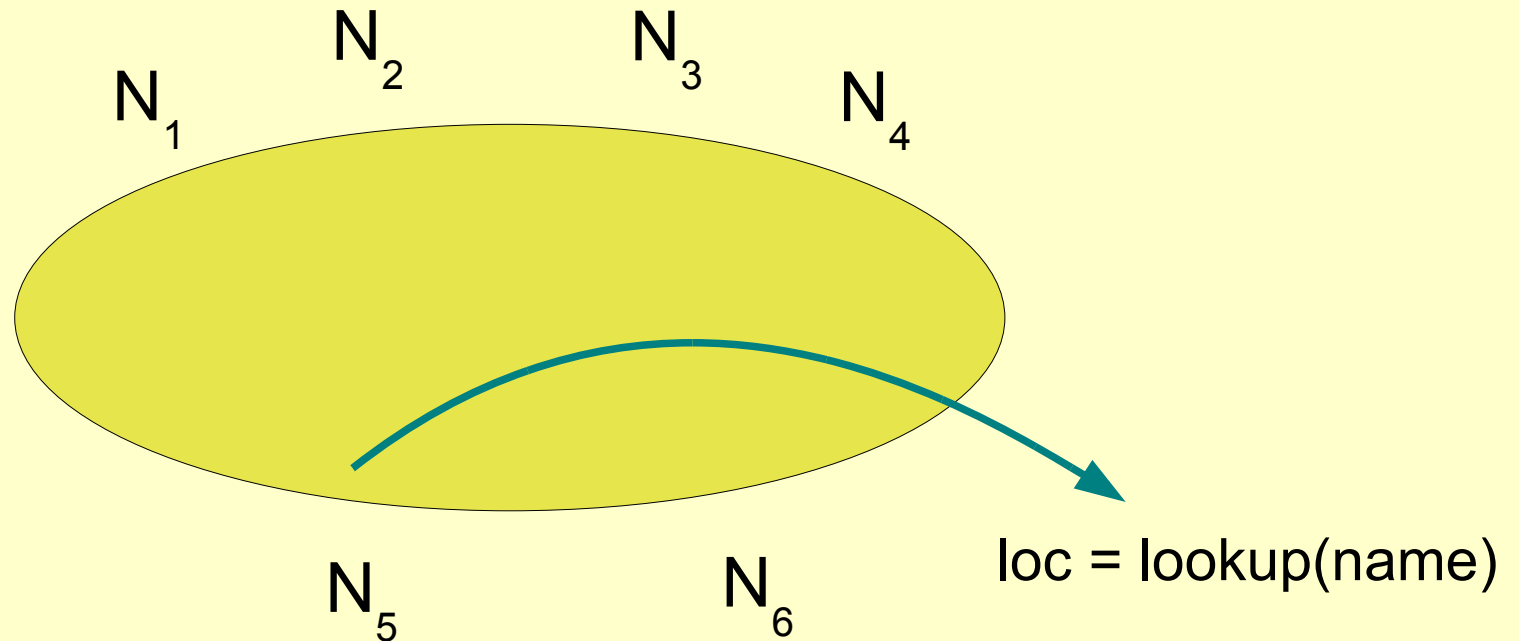
P2P – Vyhledávání

Prohledávání - FreeNet



P2P – Vyhledávání

Prohledávání - FreeNet



DHT – Distributed Hash Table

efektivita

počet zpráv na vyhledání
průměr grafu

škálovatelnost

počet stavových položek na uzel
stupeň uzlu

odolnost

proti výpadkům uzlu
proti výpadkům uzlů

replikace dat

počet redundantních kopií



DHT – Distributed Hash Table

Chord

jednoduchý interface

IP_address = lookup(key)

efektivní

$O(\log N)$ zpráv na vyhledání

škálovatelný

$O(\log N)$ stavových dat na uzel

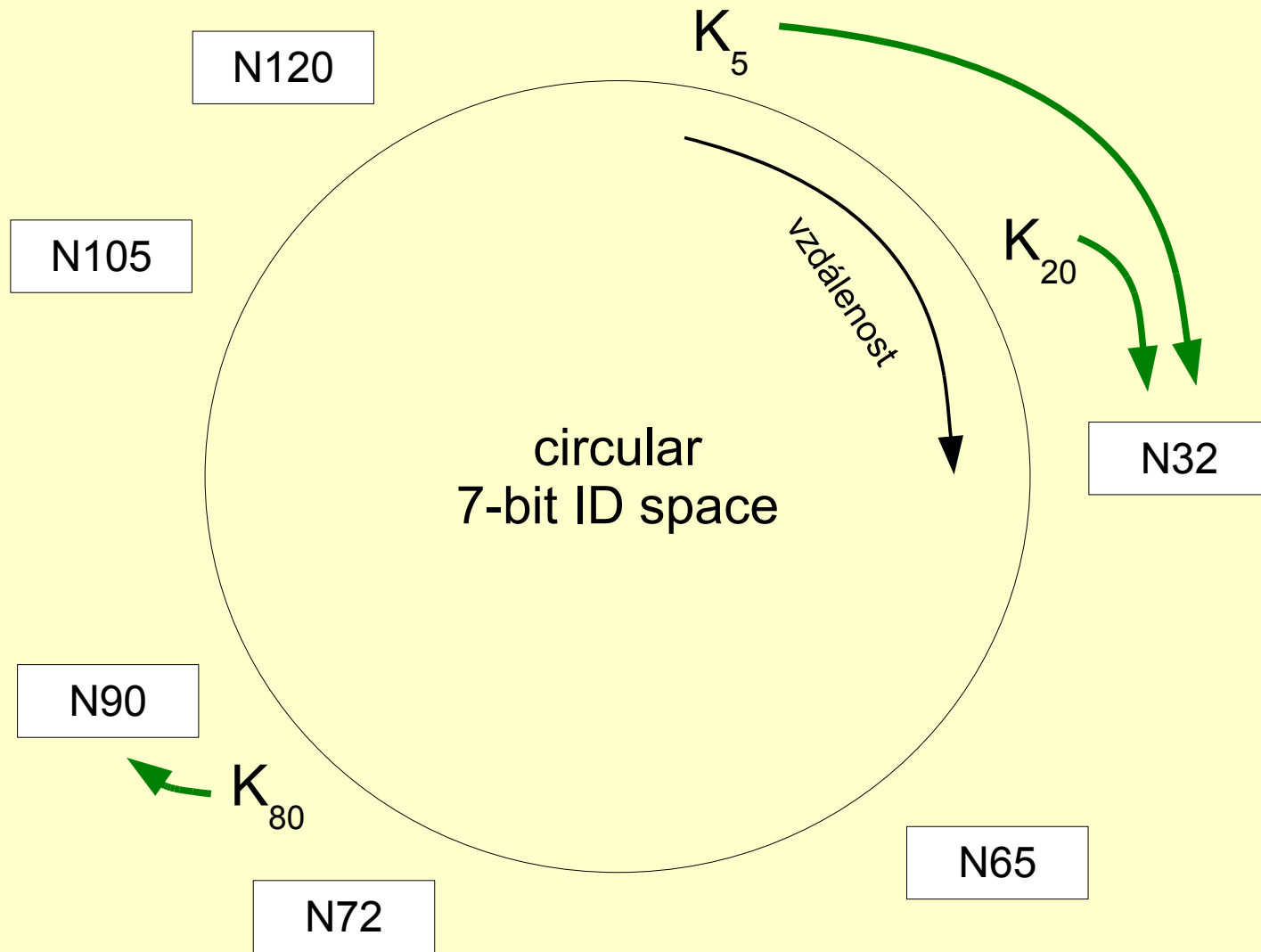
odolný

proti výpadkům uzlů



DHT – Chord

Hashing



DHT – Chord

Hashing

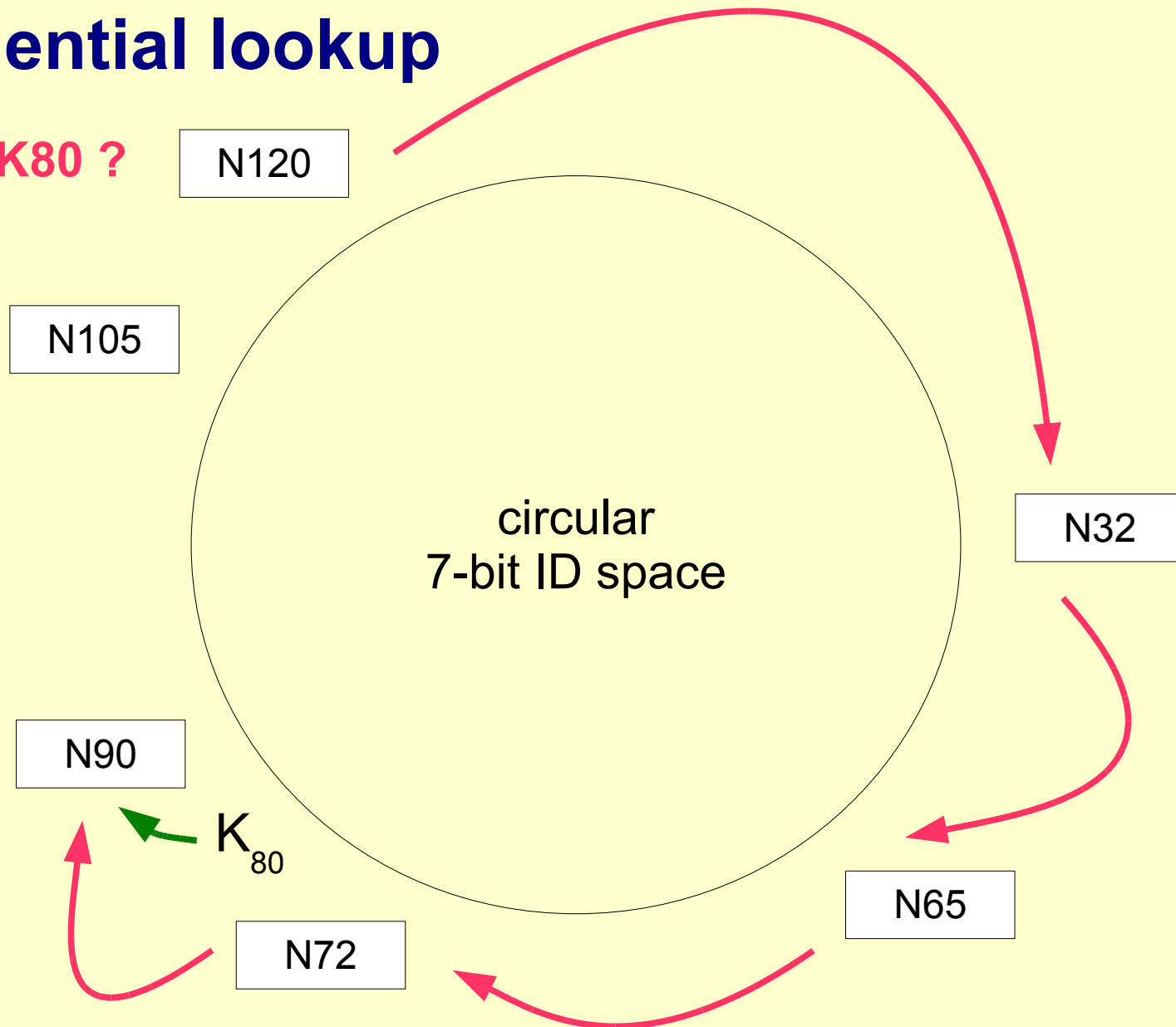
- konzistentní
- Chord se opírá o SHA-1
- skutečná délka klíče je 160 bitů
- vzdálenost – „pohyb“ po směru hodinových ručiček



DHT – Chord

Sequential lookup

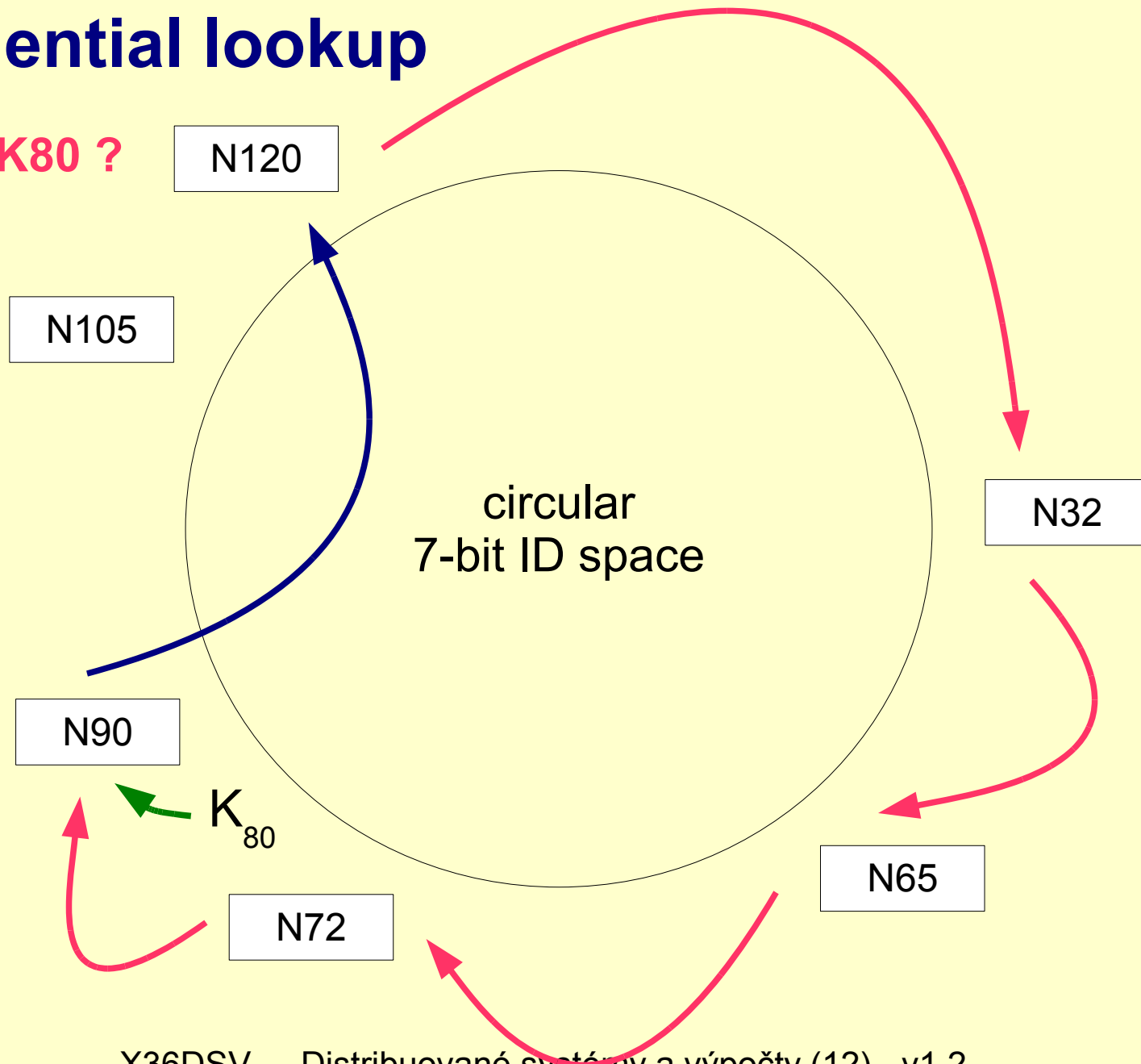
where is K80 ?



DHT – Chord

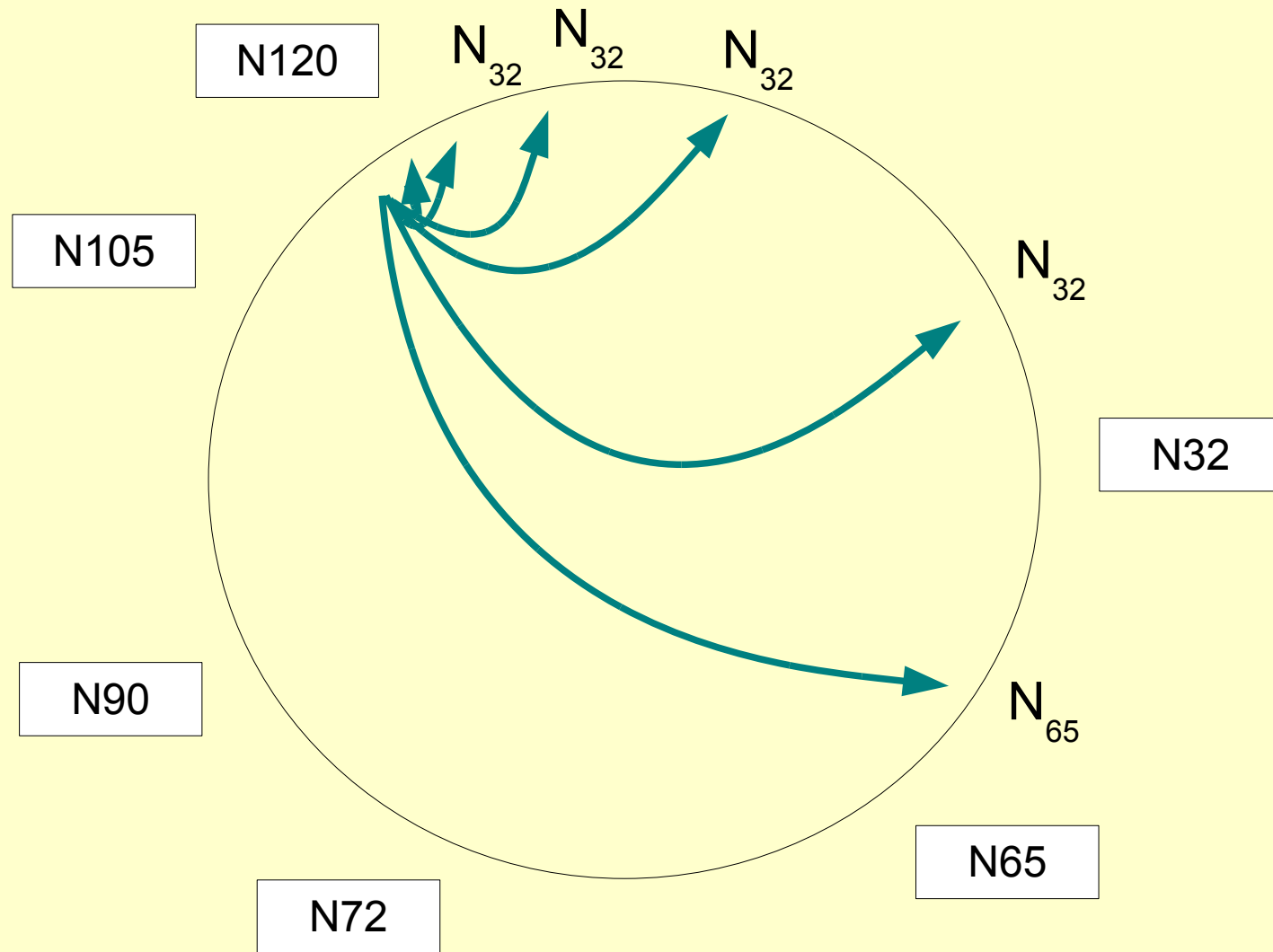
Sequential lookup

where is K₈₀ ?



DHT – Chord

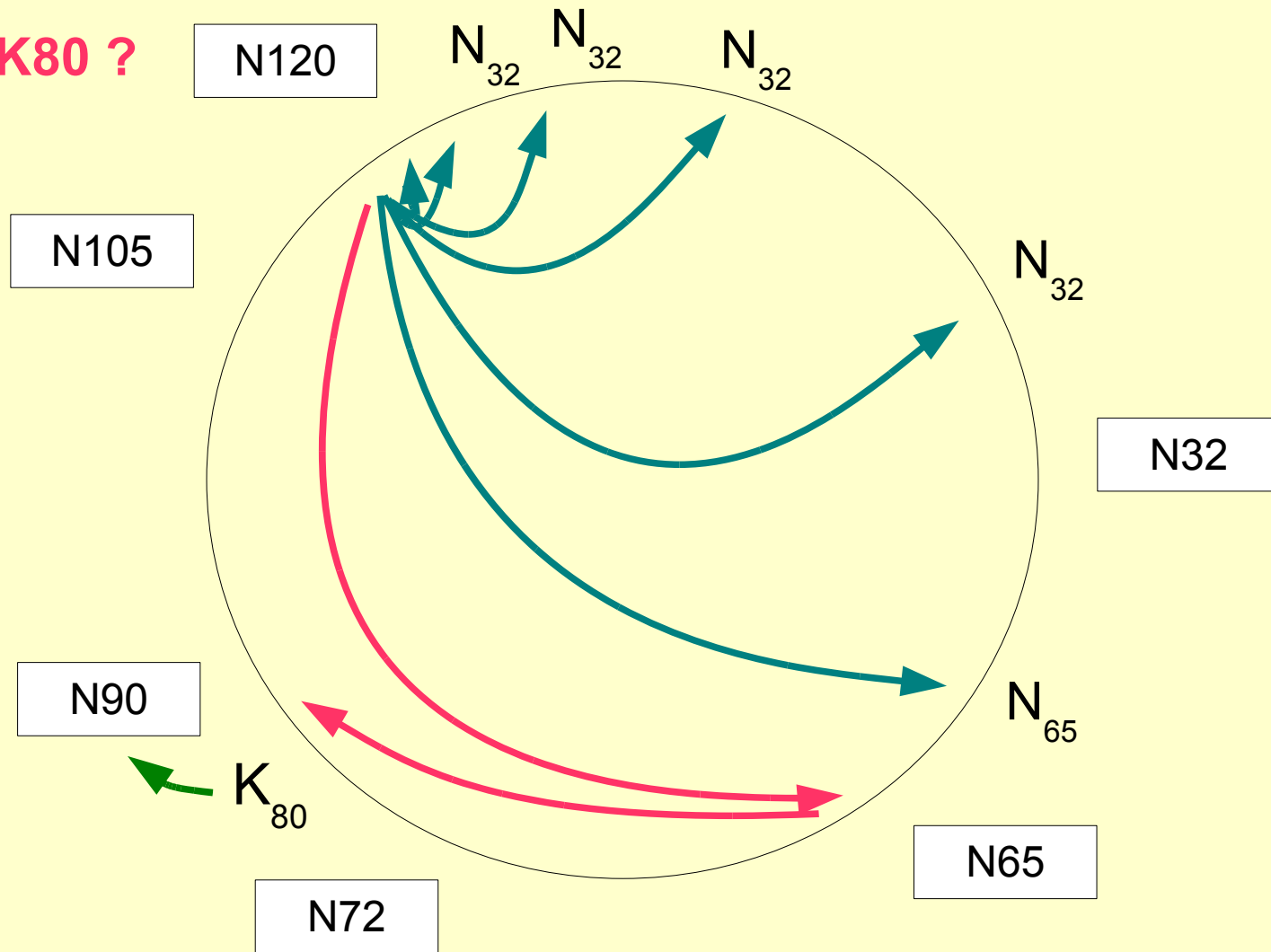
Finger table



DHT – Chord

Finger table

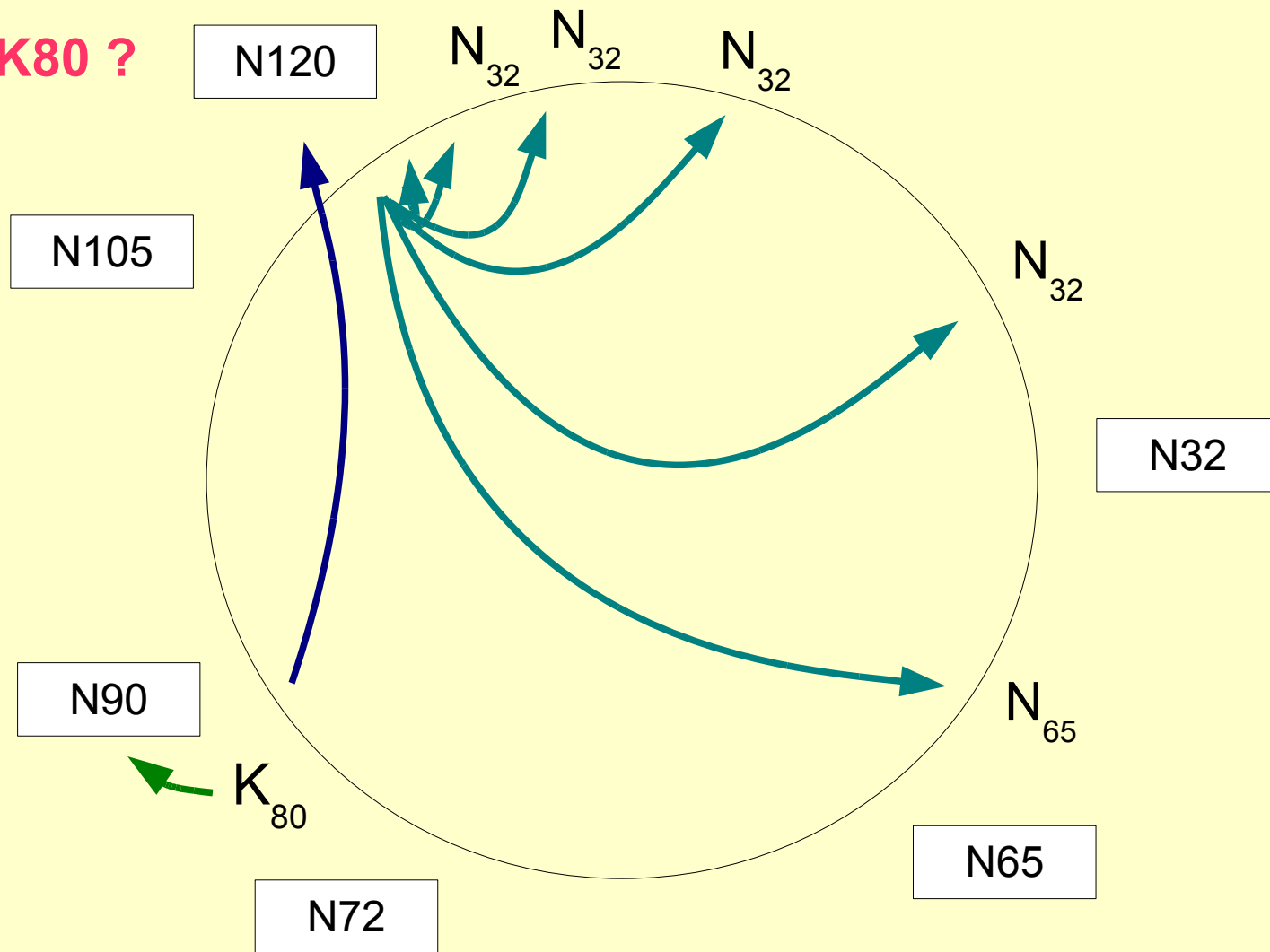
where is K80 ?



DHT – Chord

Finger table

where is K80 ?



DHT – Chord

Finger table

N120

i	$n+2^i$	succ
0	K121	N32
1	K122	N32
2	K124	N32
3	K0	N32
4	K8	N32
5	K24	N32
6	K56	N65



DHT – Chord

insert

- určení Id
- znalost jednoho uzlu Chord
- nalezení následníka a uložení odkazů
- vytvoření finger table

stabilize

- periodický dotaz na následníka
- ten sdělí svého (nového) předchůdce
- neshoda → update vazby
- uložení dat

fix_fingers

- periodický update jednotlivých odkazů



DHT – Chord

výpadek finger table

- fix_fingers()

násobný výpadek

- udržování seznamu r následníků

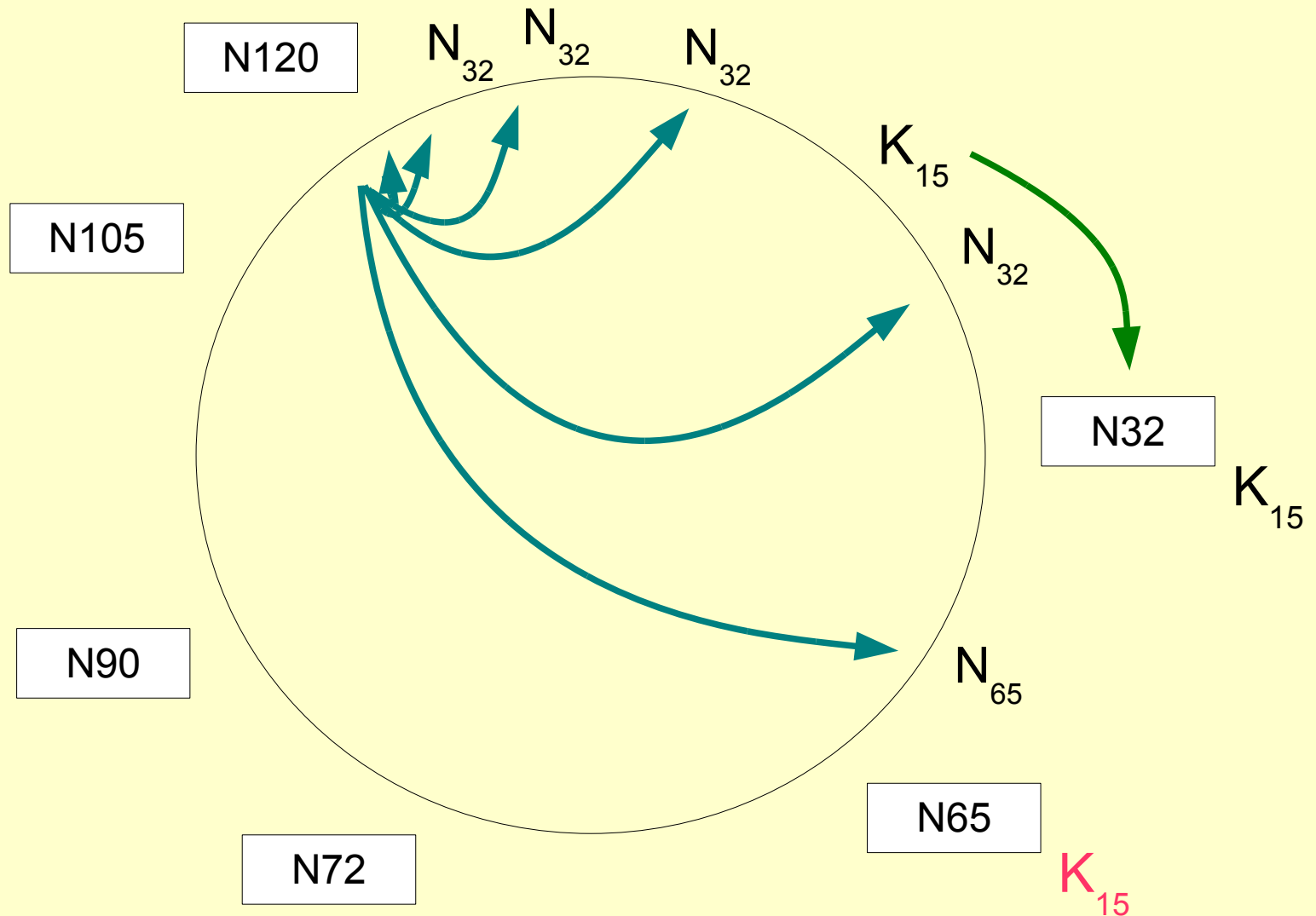
opuštění Chordu

- zkopírování dat k následníkovi
- update odkazů



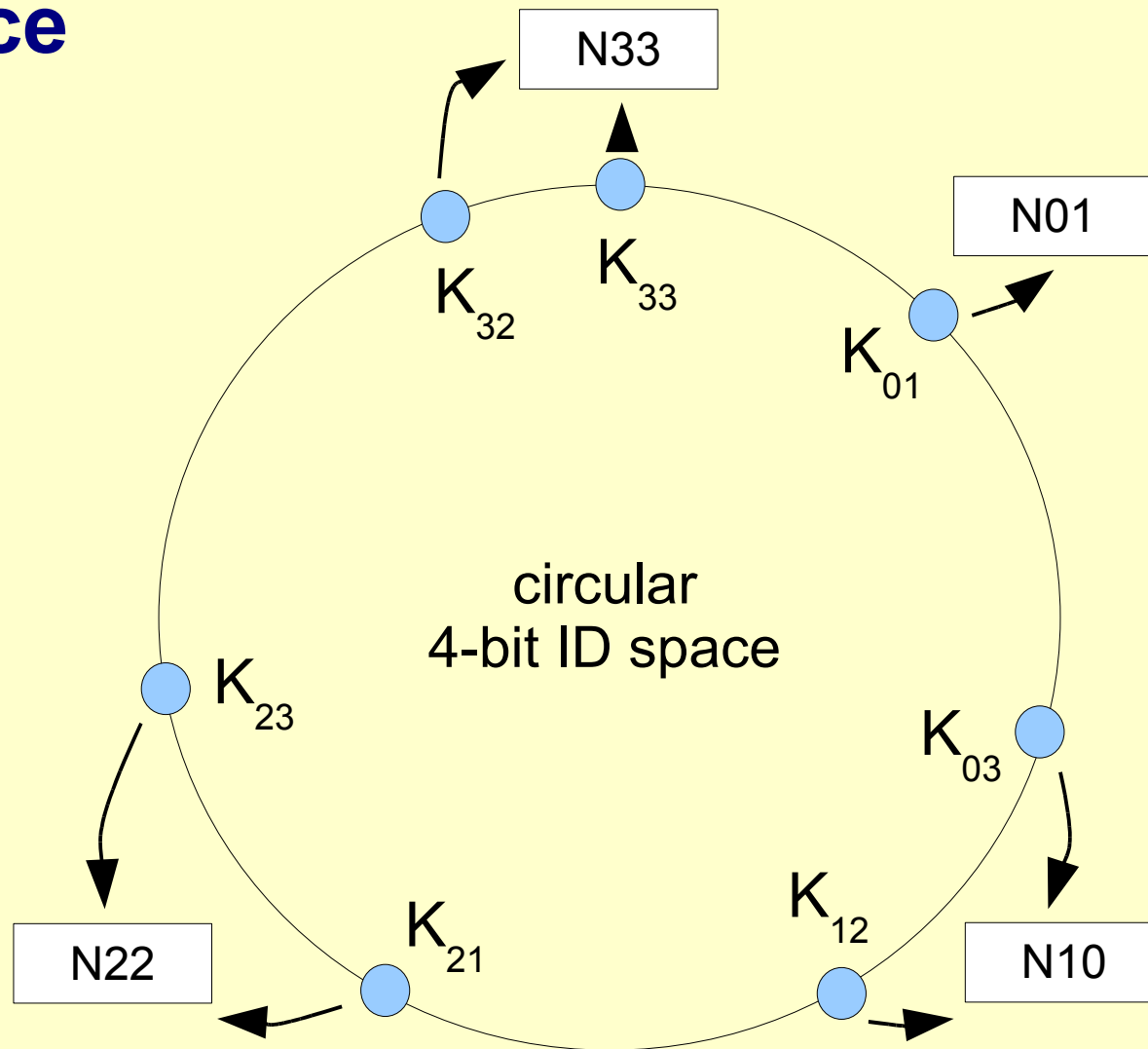
DHT – Chord

Replikace



DHT – Pastry

Id space



DHT – Pastry

Node state

n = 103220

12-bit ID space

routing table

0	031120	1	201303	312201
1	0	110003	120132	132012
2	100221	101203	102303	3
3		103112	2	103302
4		103210	2	
5	0		0	

leaf set

103123 103210 103302 103330

neighborhood set

031120 312201 120132 101203



DHT – Pastry

lookup

- blízký uzel v leaf table (pokud je v rozsahu)
- uzel s delší shodou prefixu v routing table
- nejbližší z uzlů se shodným prefixem

insert

- vyhledání síťově (geograficky, topologicky) blízkého uzlu **k**
- převzetí neighbourhood table
- “join“ zpráva směrem k vybrané ID pozici
- převzetí leaf table od uzlu **c** nejbližšího k ID pozici
- převzetí prvního řádku routing table od uzlu **k**
- převzetí dalších řádků routing table od uzlů cesty k **c**
- odeslání stavové informace směrem k **c**
- optimistické řešení kolize časovými známkami



DHT – Pastry

výpadky

- periodické testování platnosti tabulek
- udržování i ne nejvýhodnějších sousedů jako zálohu
- při opravě routing table pomohou sousedi ze stejného řádku
- při opravě leaf set pomohou sousedi z leaf set
- při opravě neighborhood set pomohou sousedi z neighborhood set

opuštění Pastry



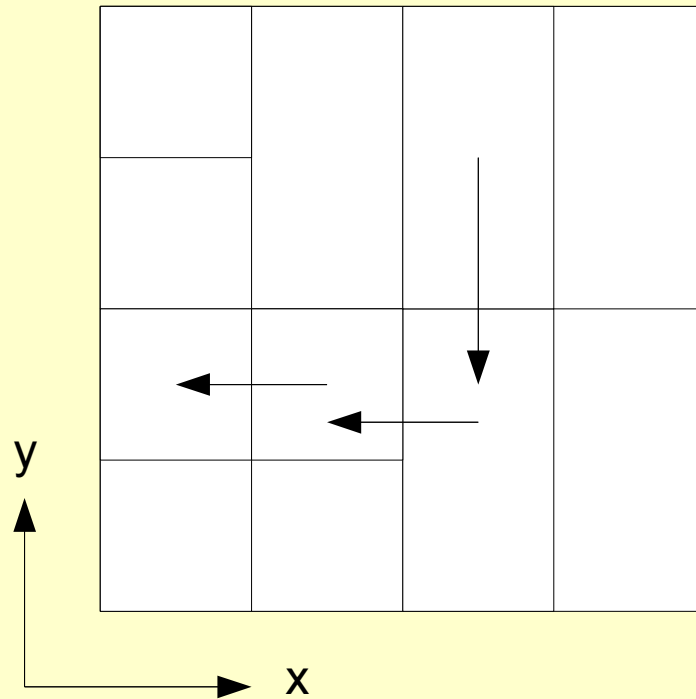
DHT – CAN

Id space

d-dimensional Id space

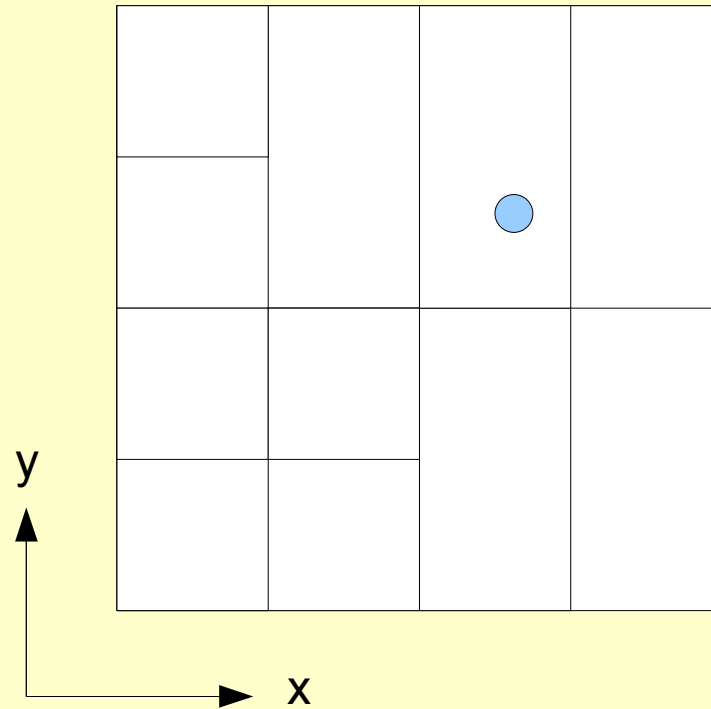
(x,y,...)

routing



DHT – CAN

insert



DHT – CAN

insert

