ROMA TRE
UNIVERSITÀ DEGLI STUDI

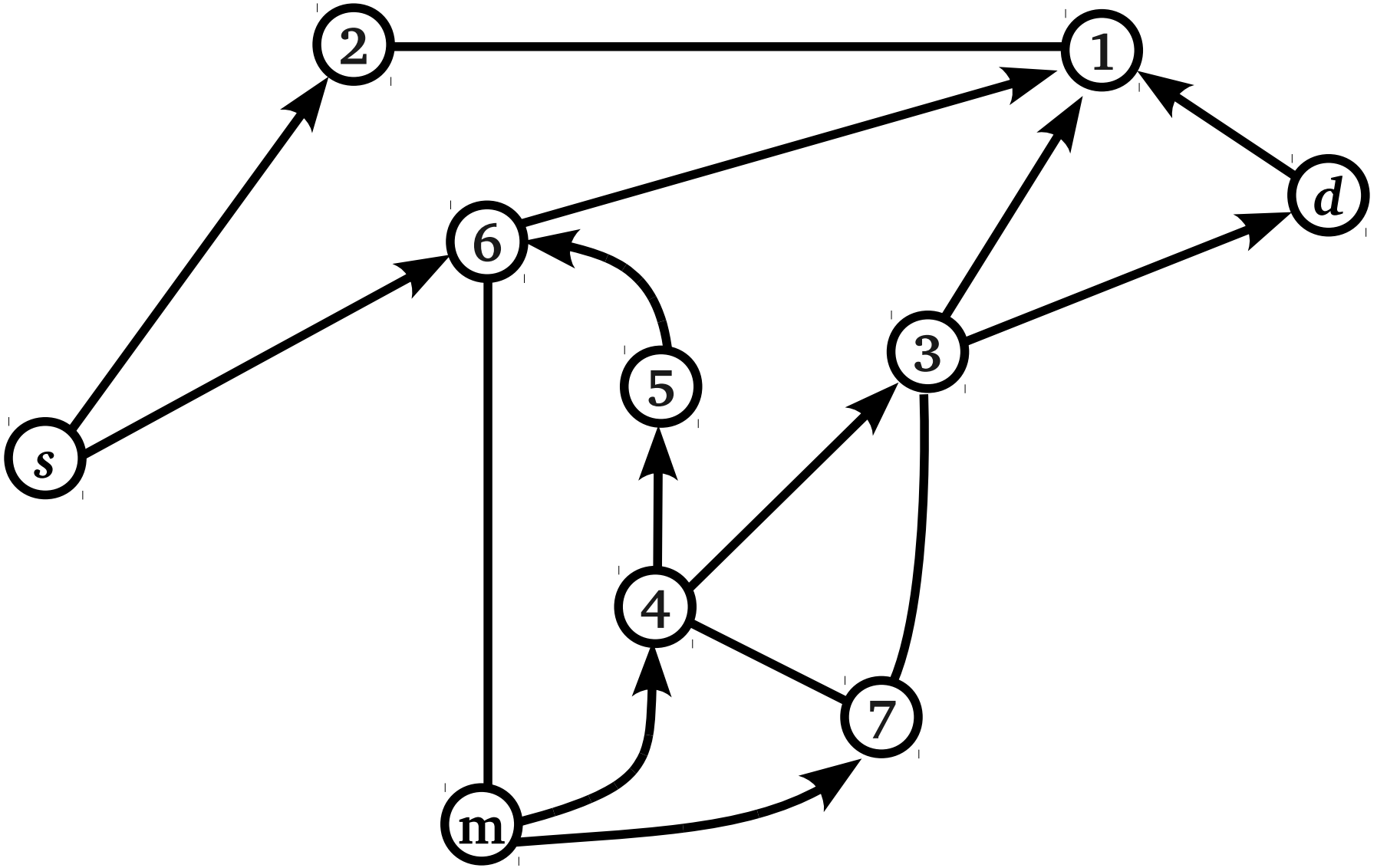
University of Leicester

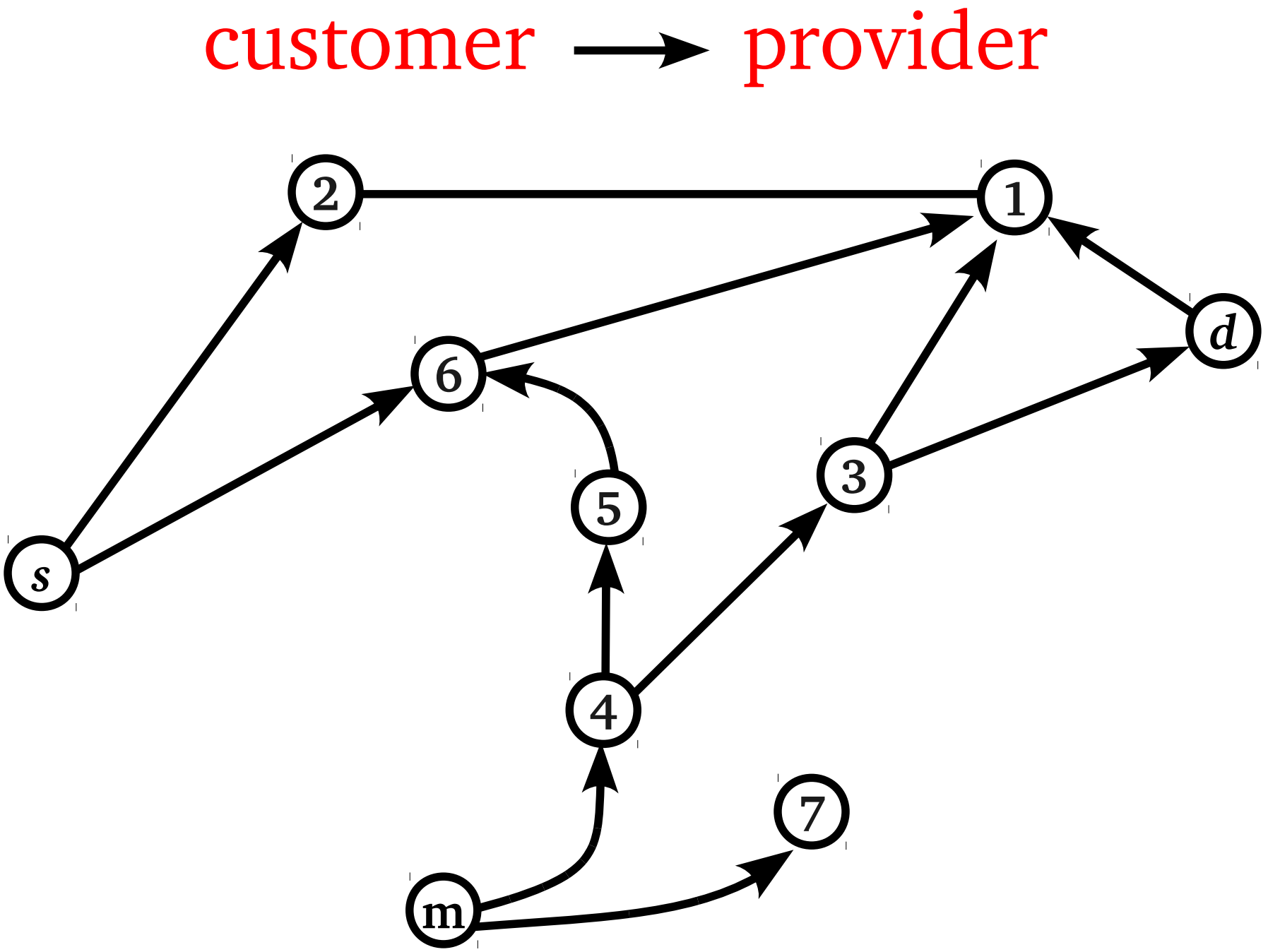
ICALP 2012 July 11th

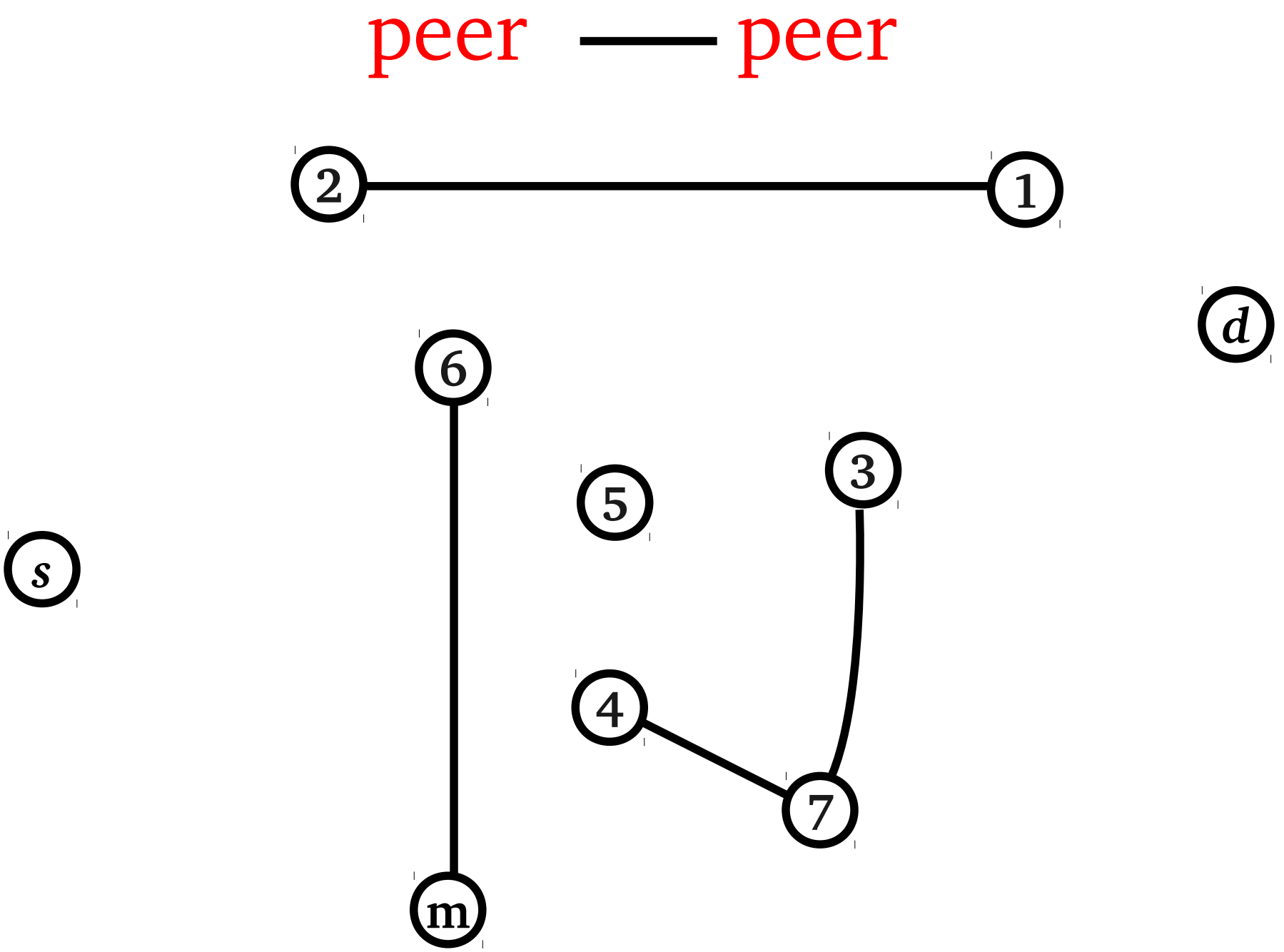
Computational Complexity of Traffic Hijacking under BGP and S-BGP

Marco Chiesa (Roma Tre University)
Giuseppe Di Battista (Roma Tre University)
Thomas Erlebach (University of Leicester)
Maurizio Patrignani (Roma Tre University)

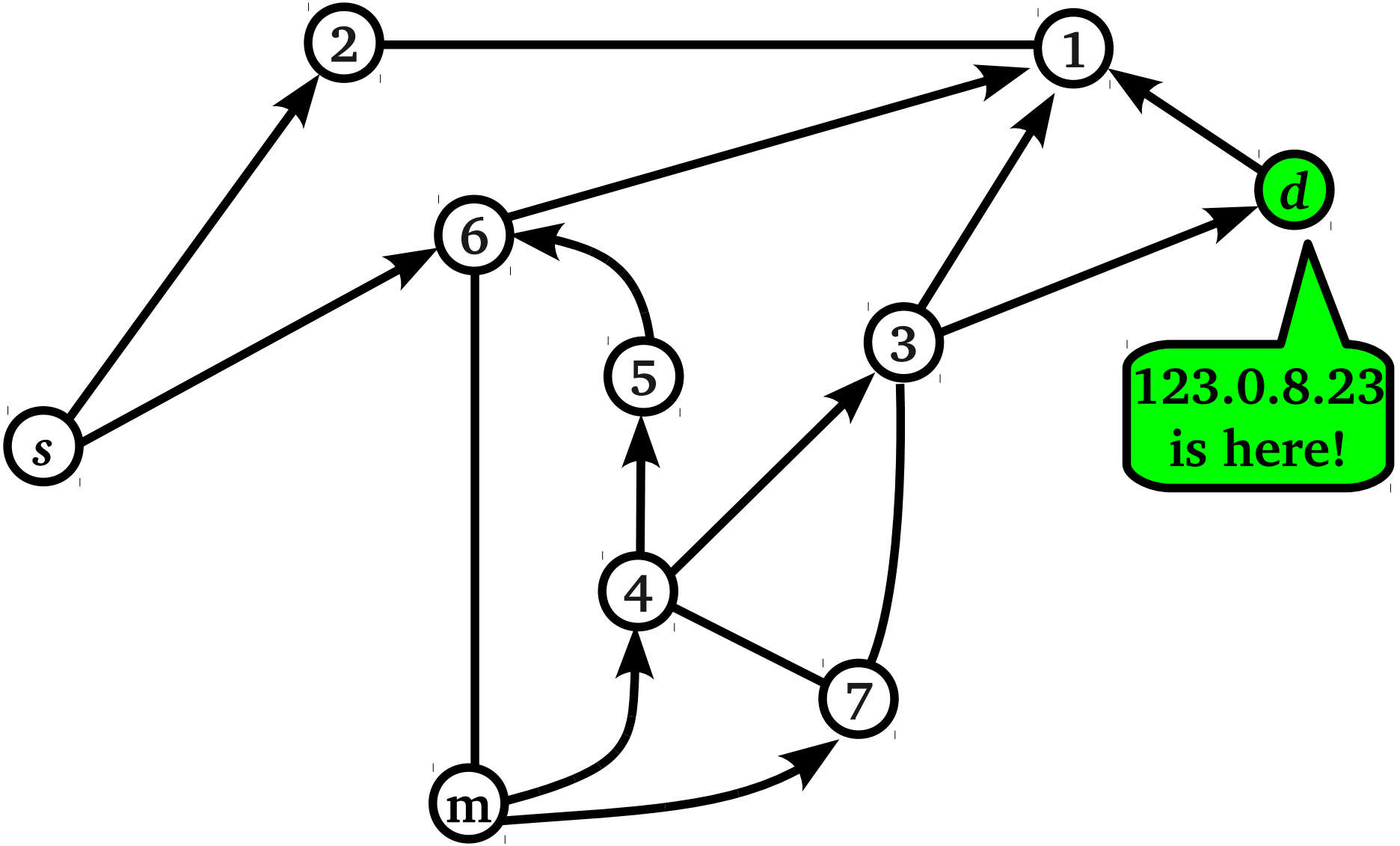
the BGP graph



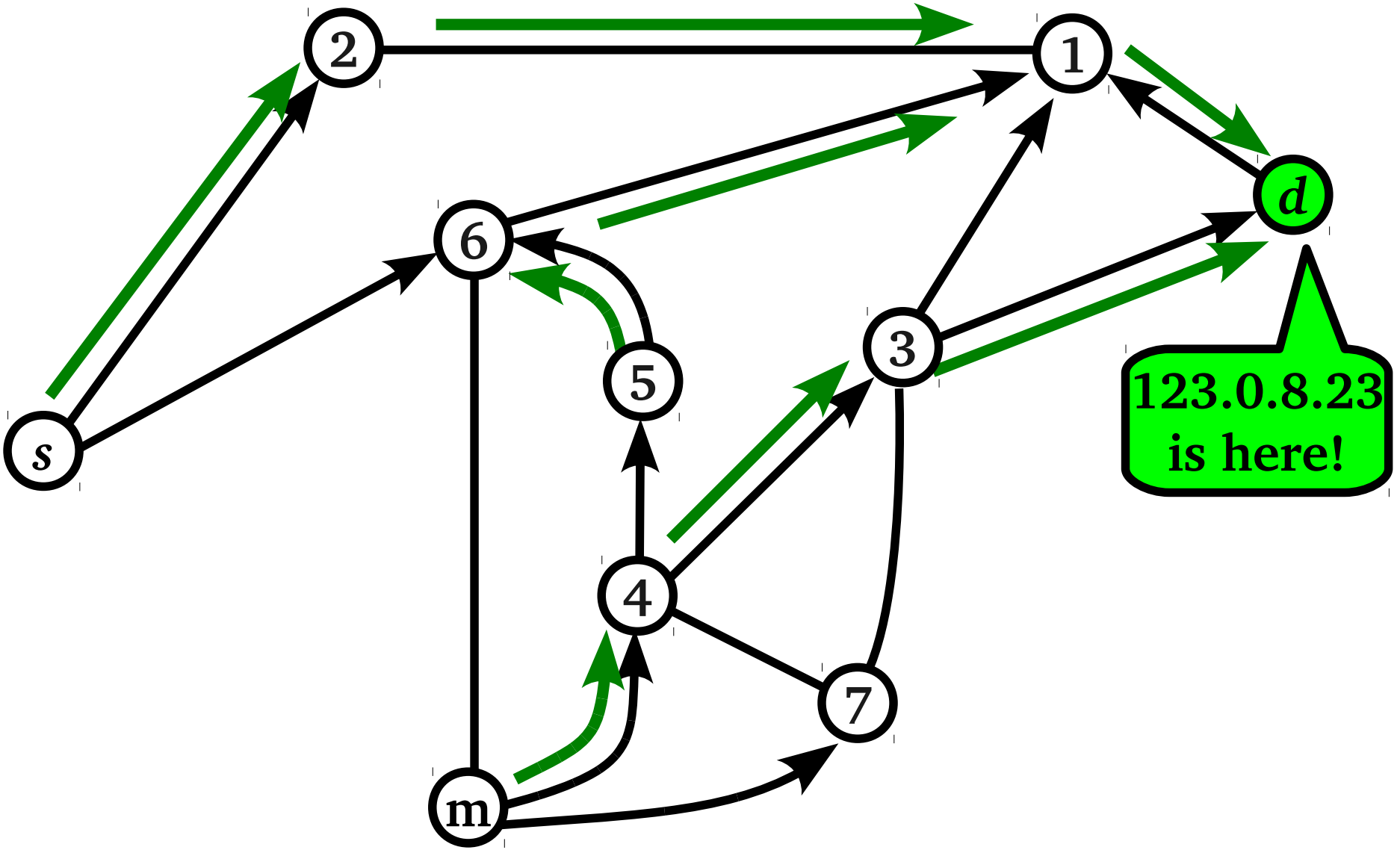

customer → provider
2
1
d
6
3
5
s
4
7
m


peer
peer
2
1
d
6
3
5
s
4
7
m

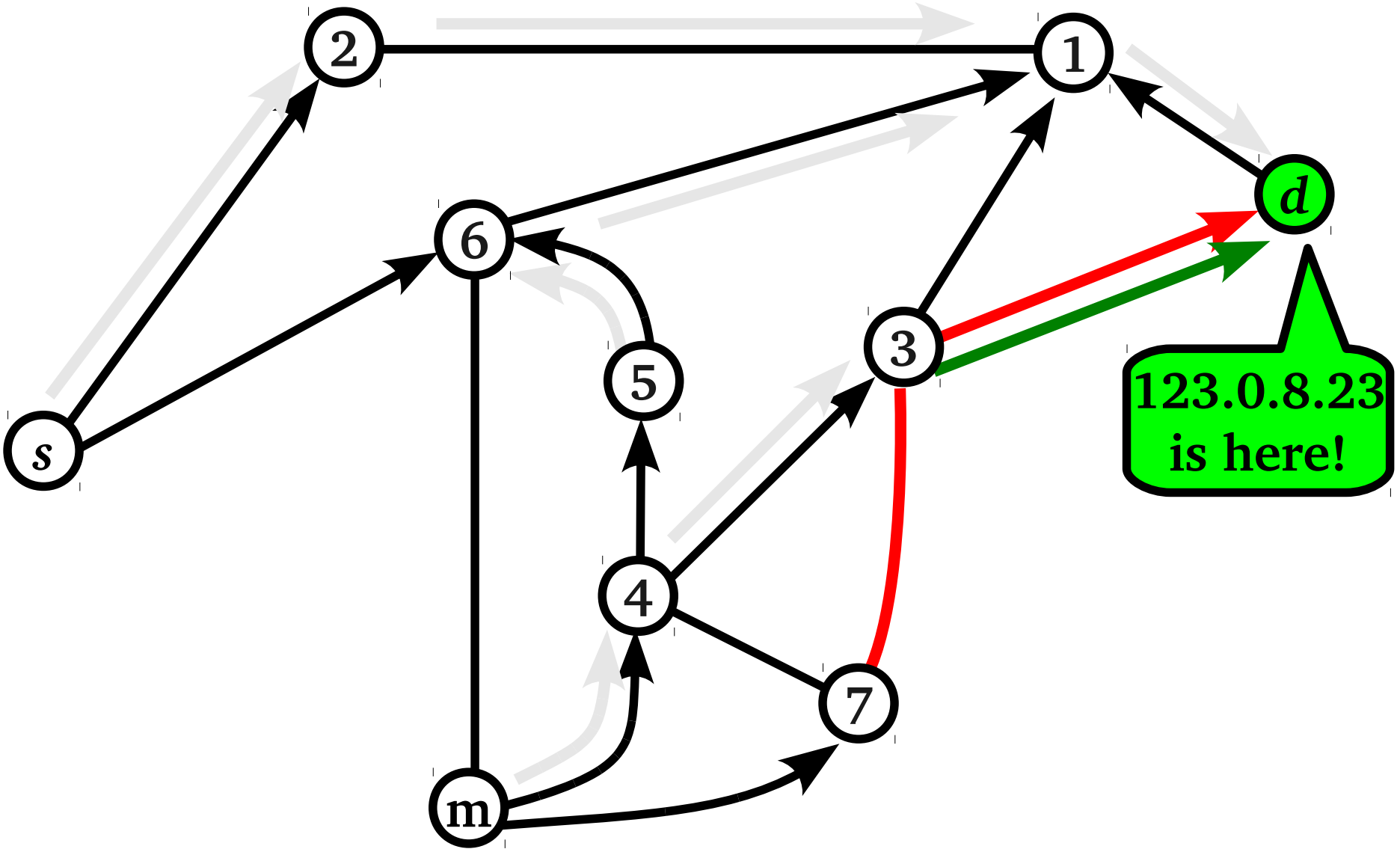
destination vertex



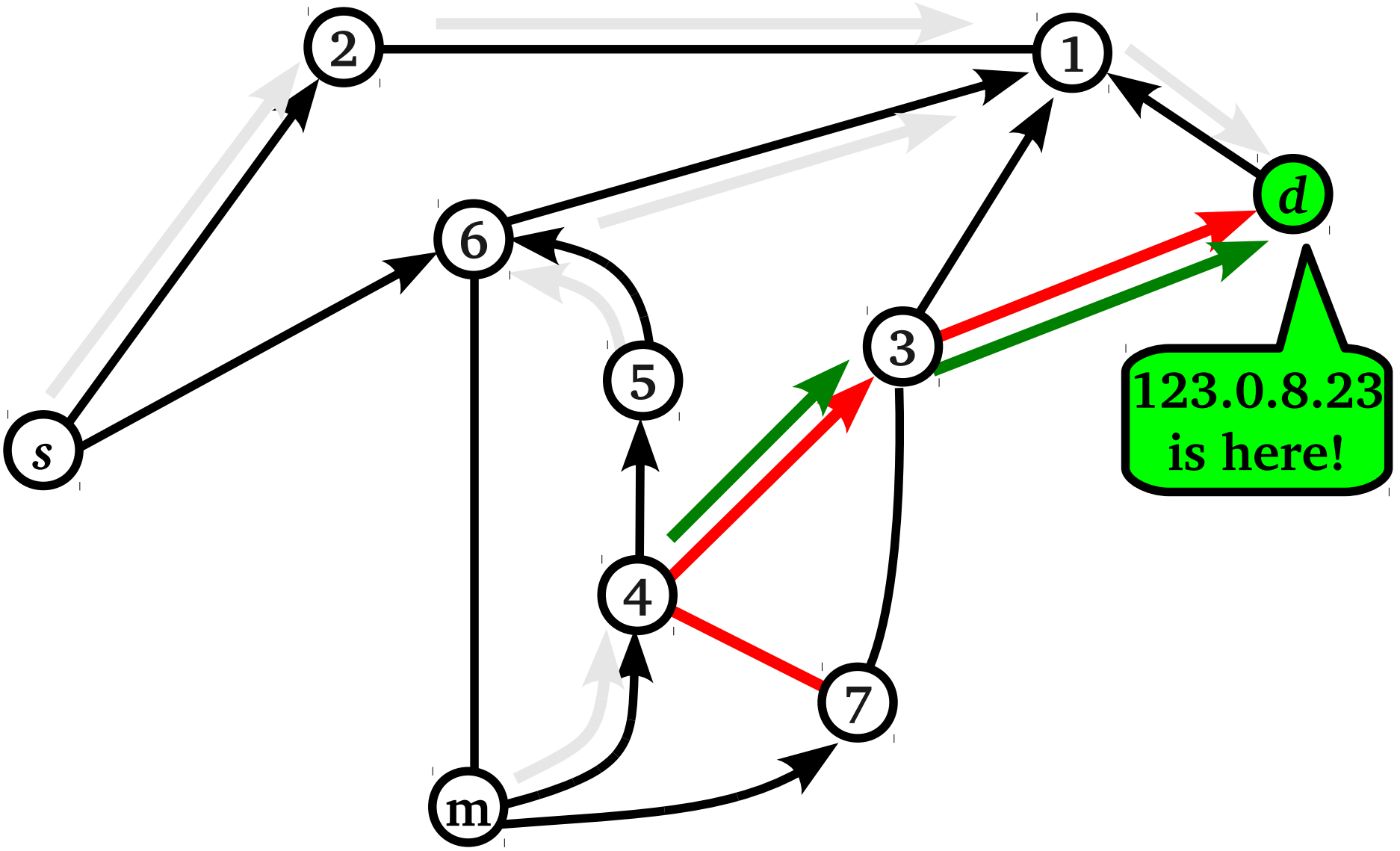
BGP routing tree



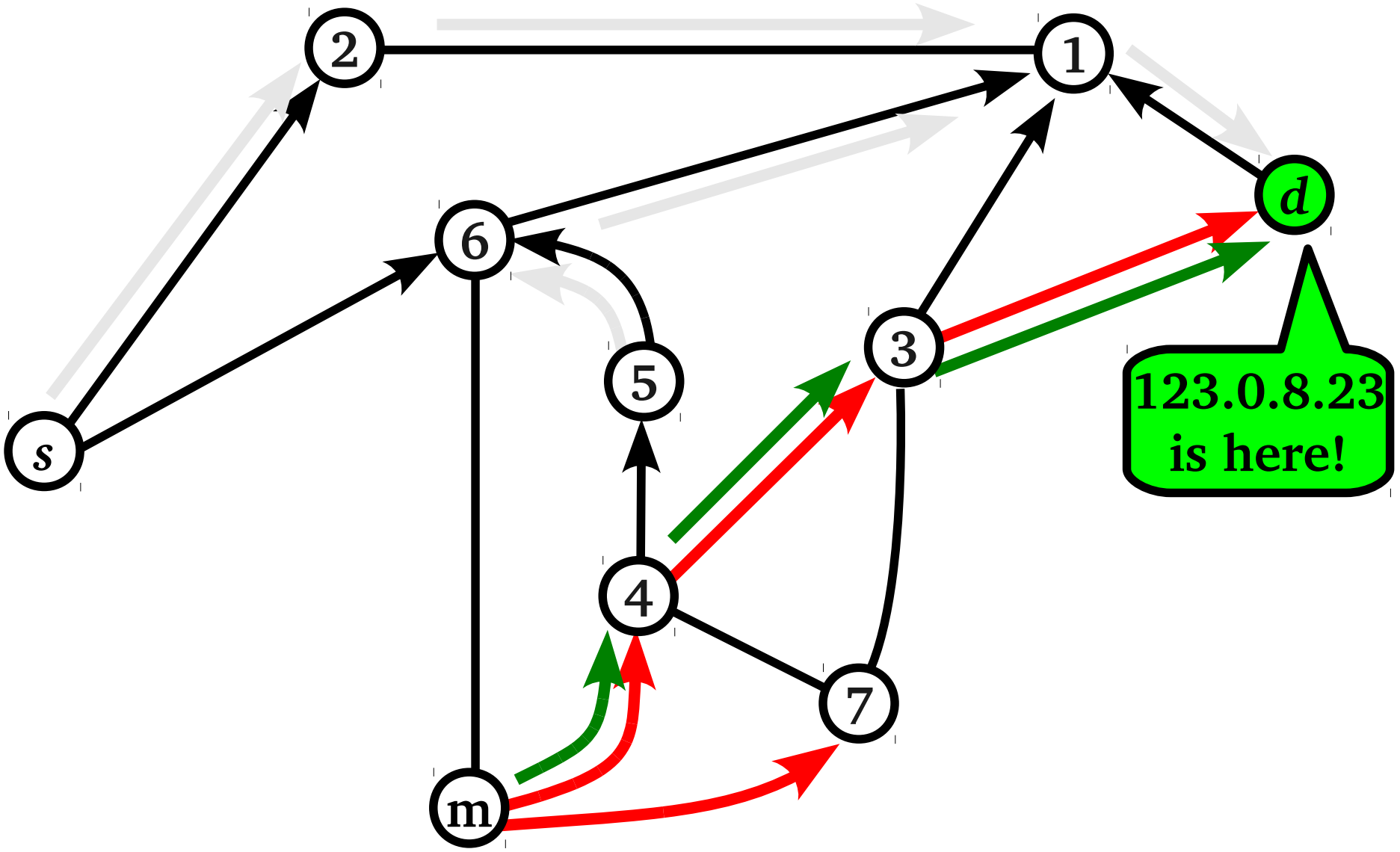
only valley-free paths are valid



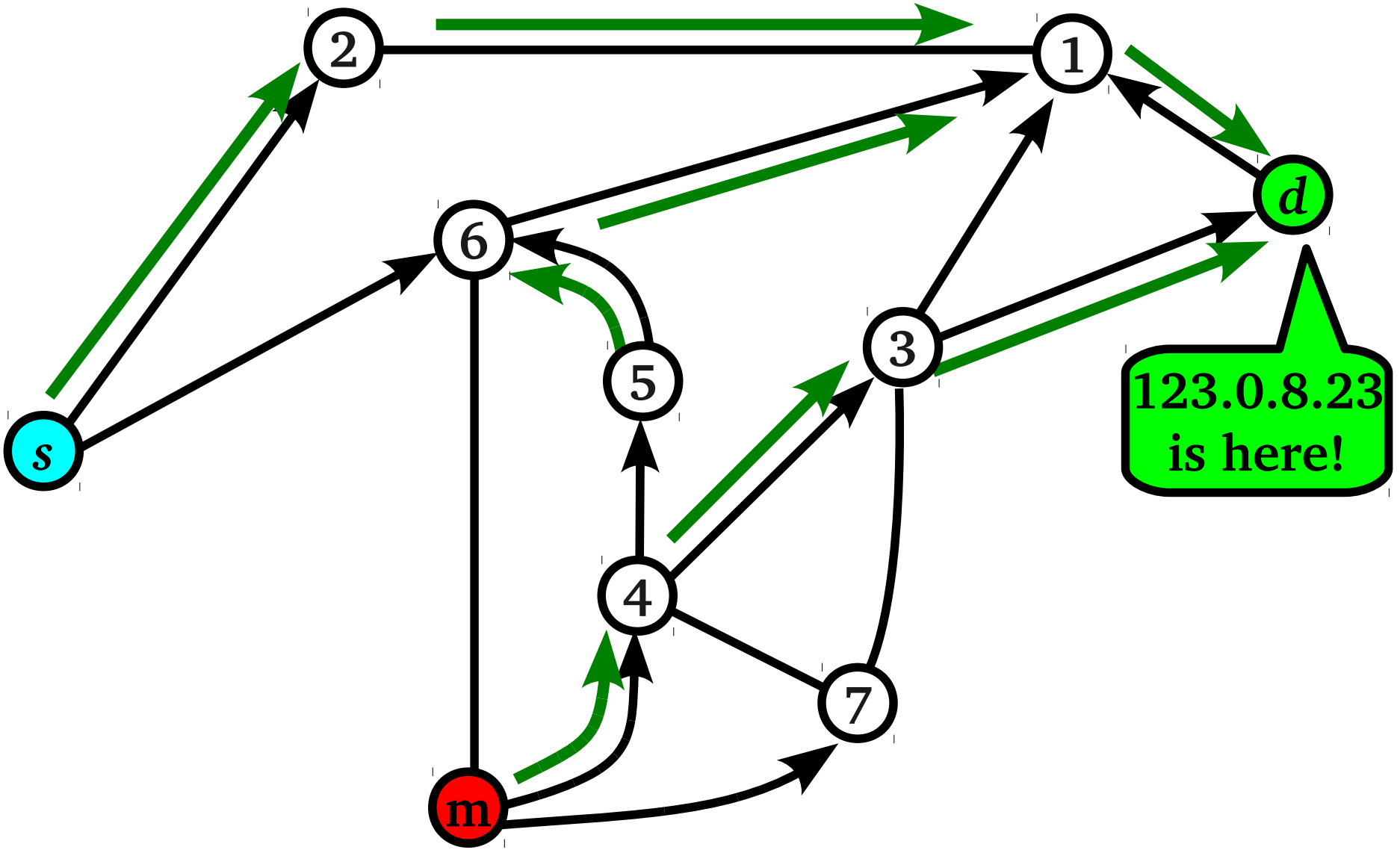
only valley-free paths are valid



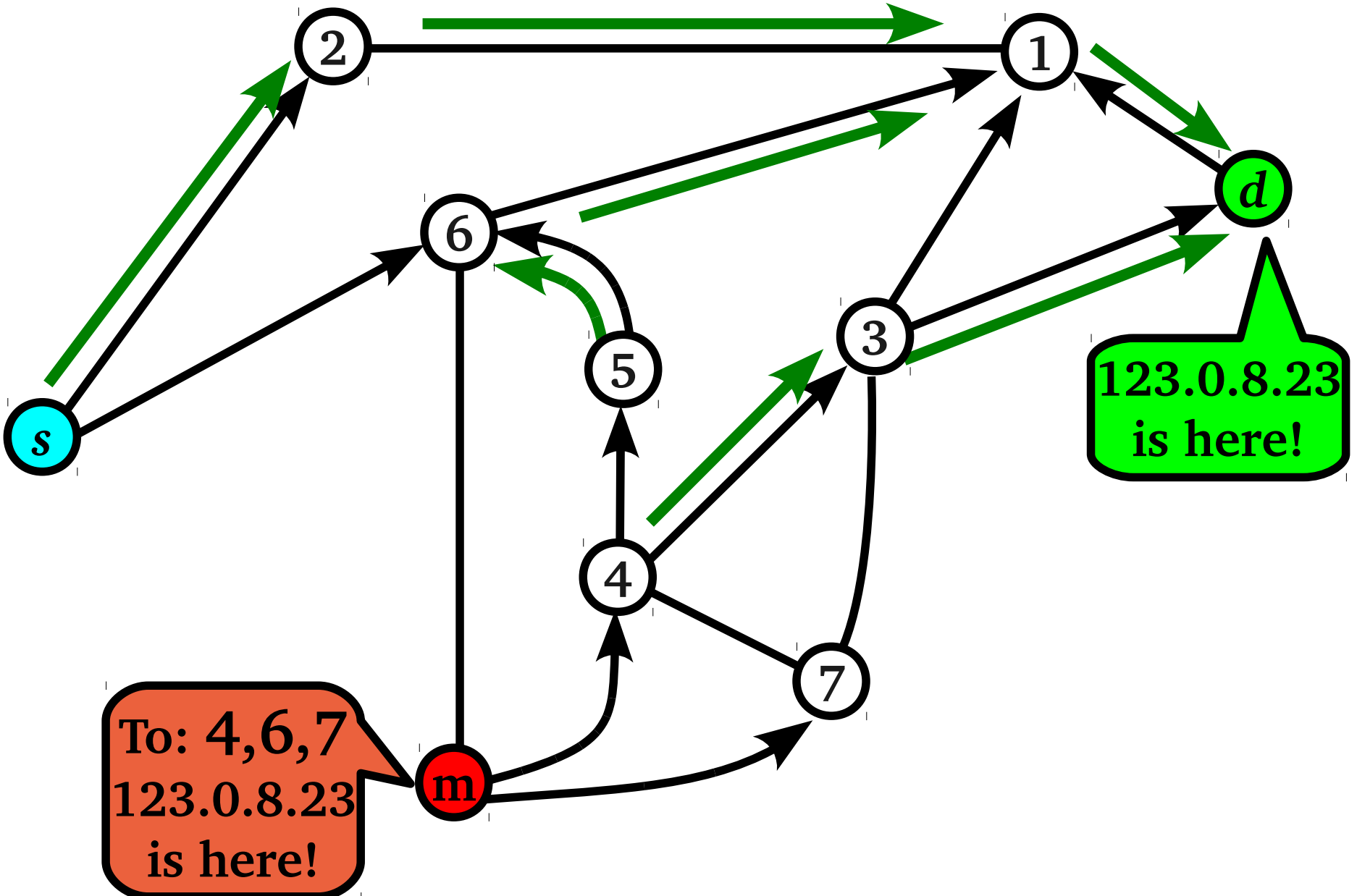
only valley-free paths are valid



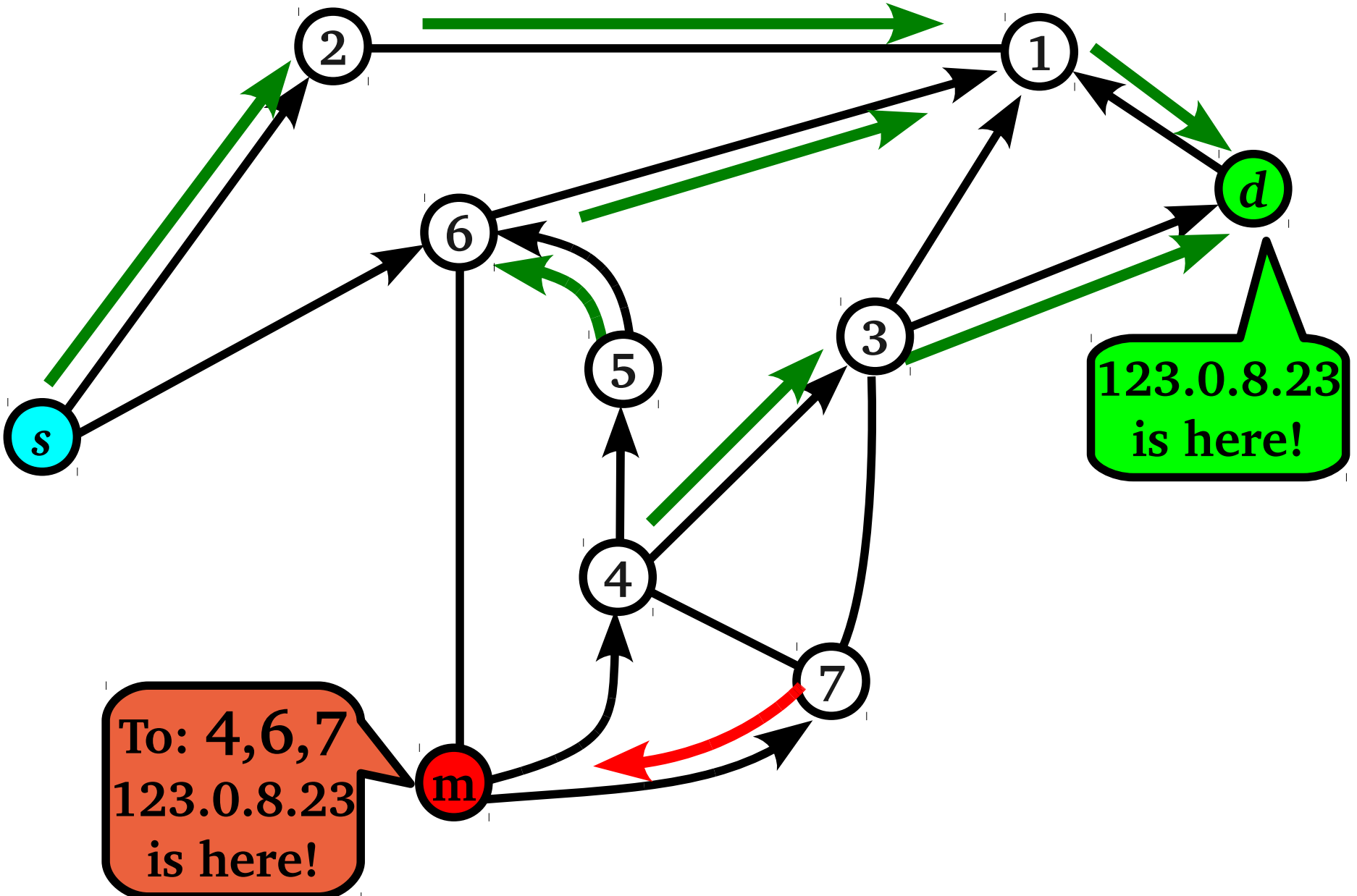
source and manipulator



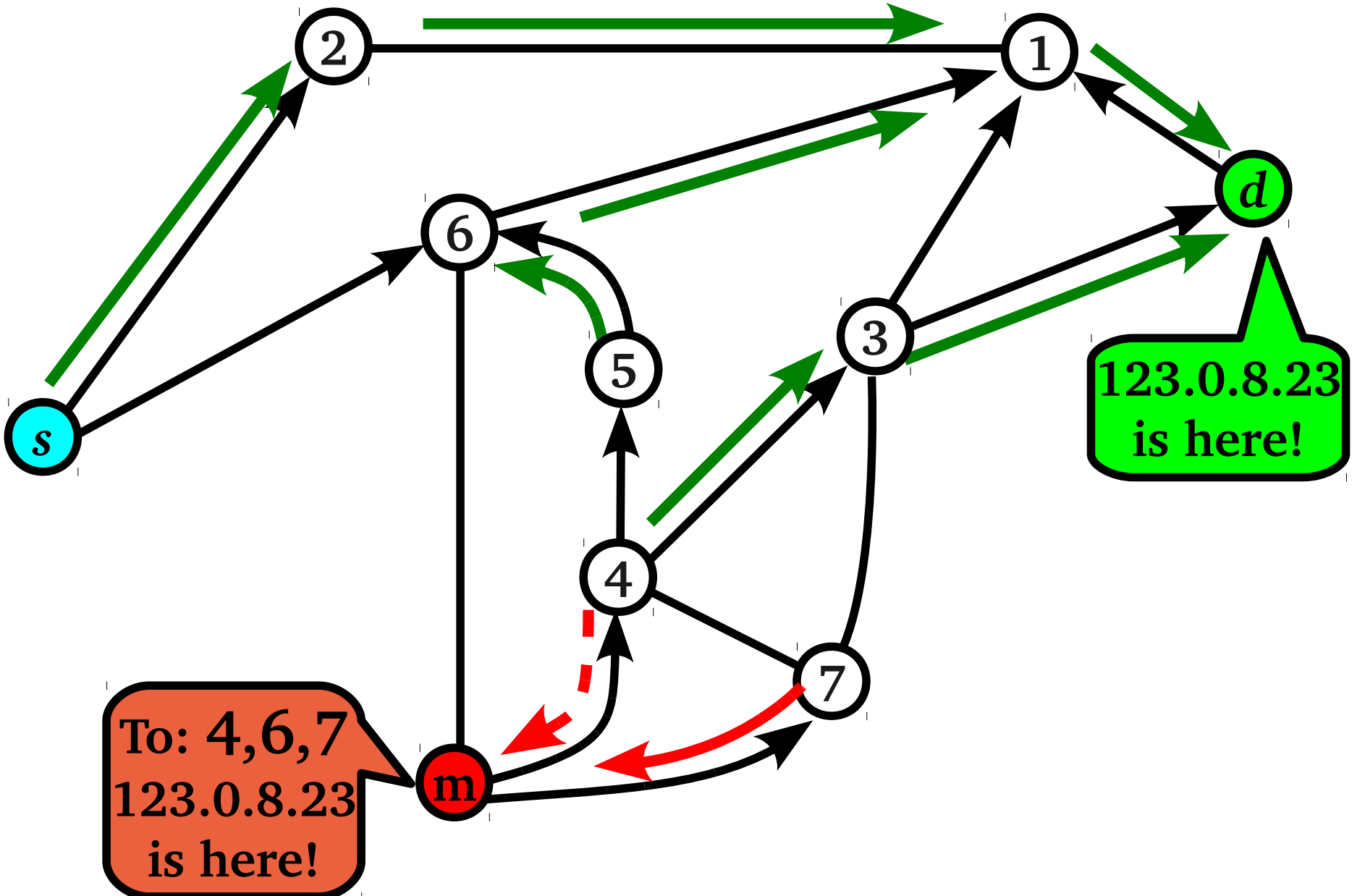
manipulator goal: hijack source



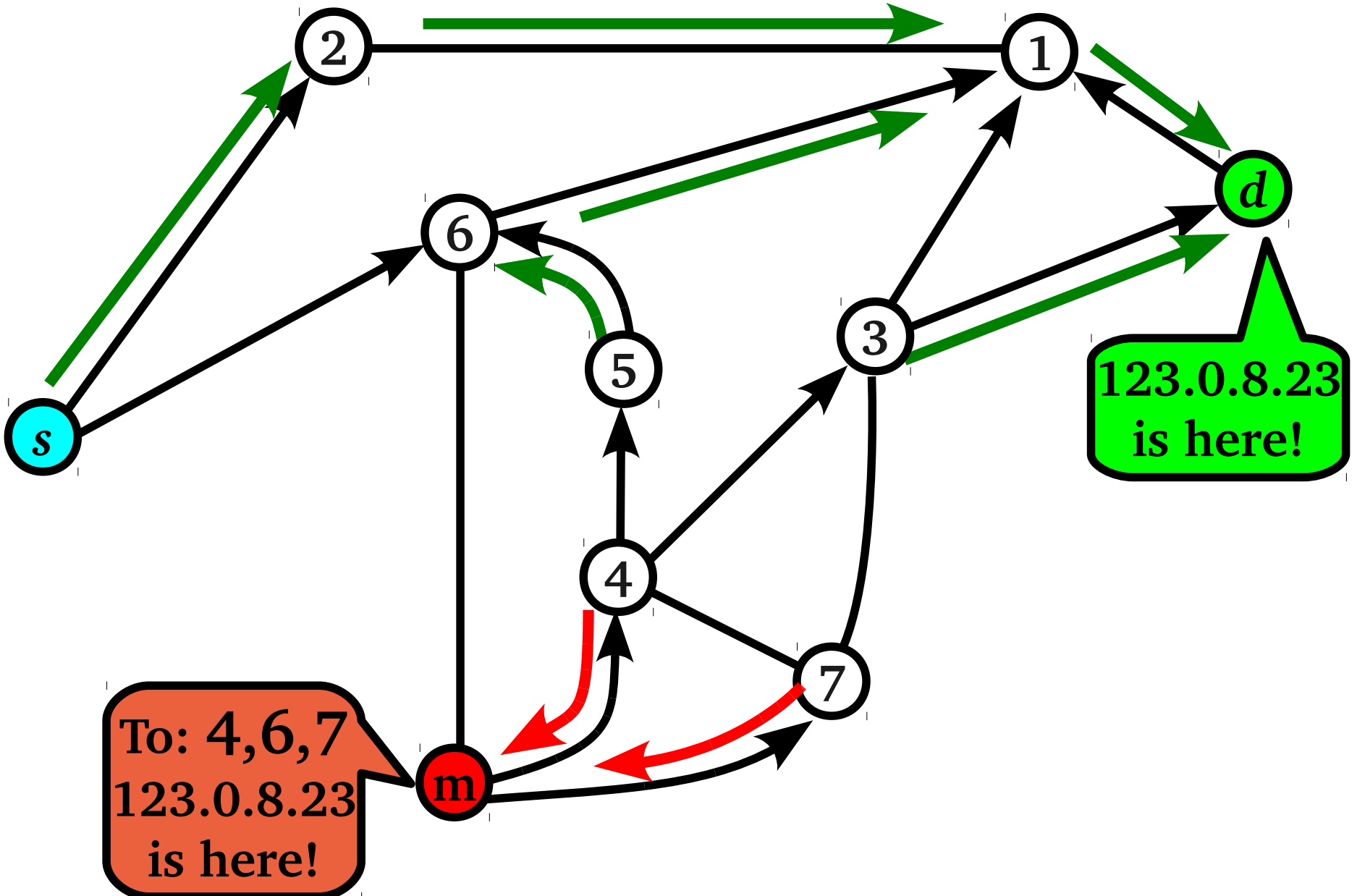
example of hijacking



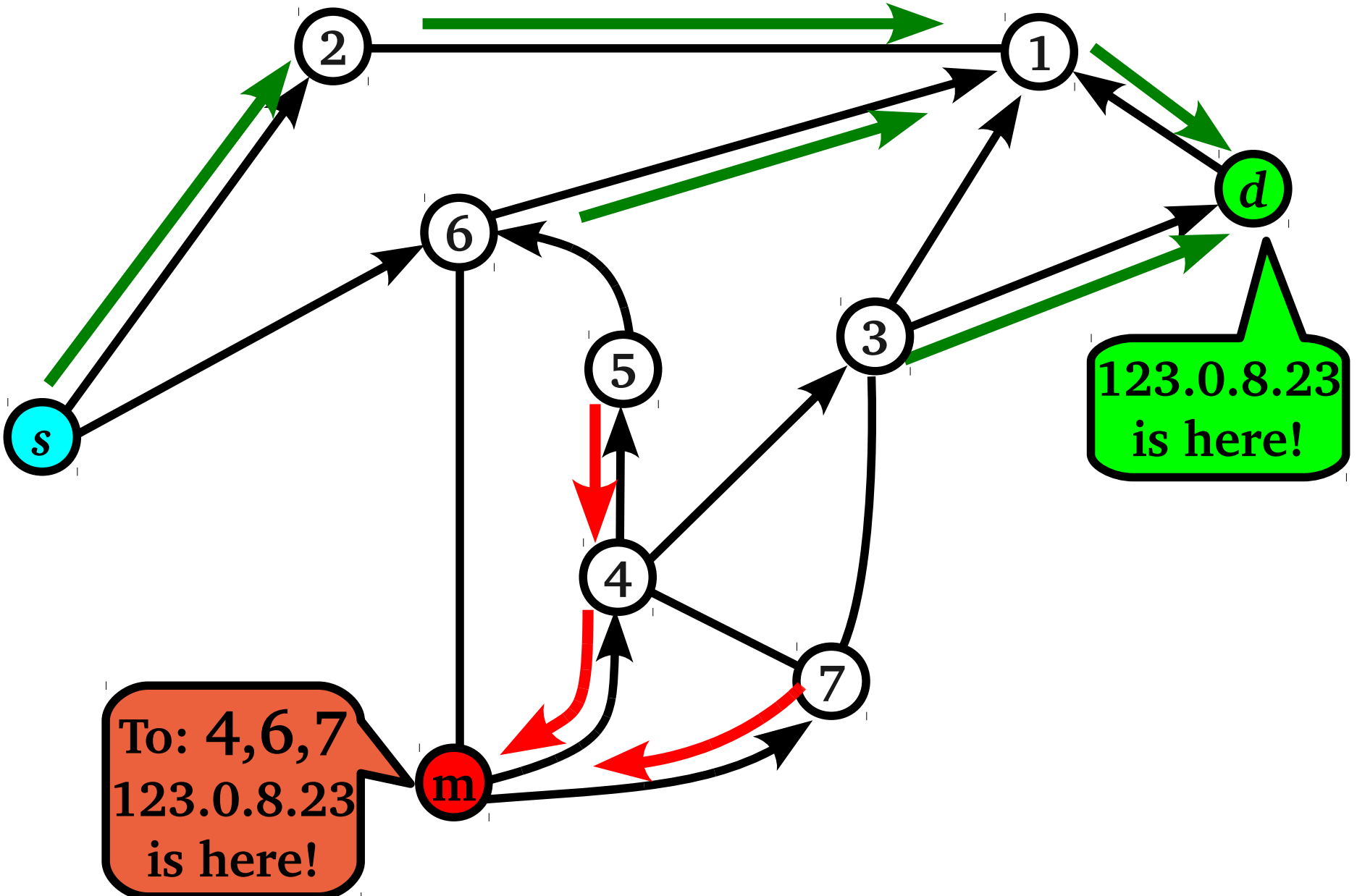
prefer customer rule



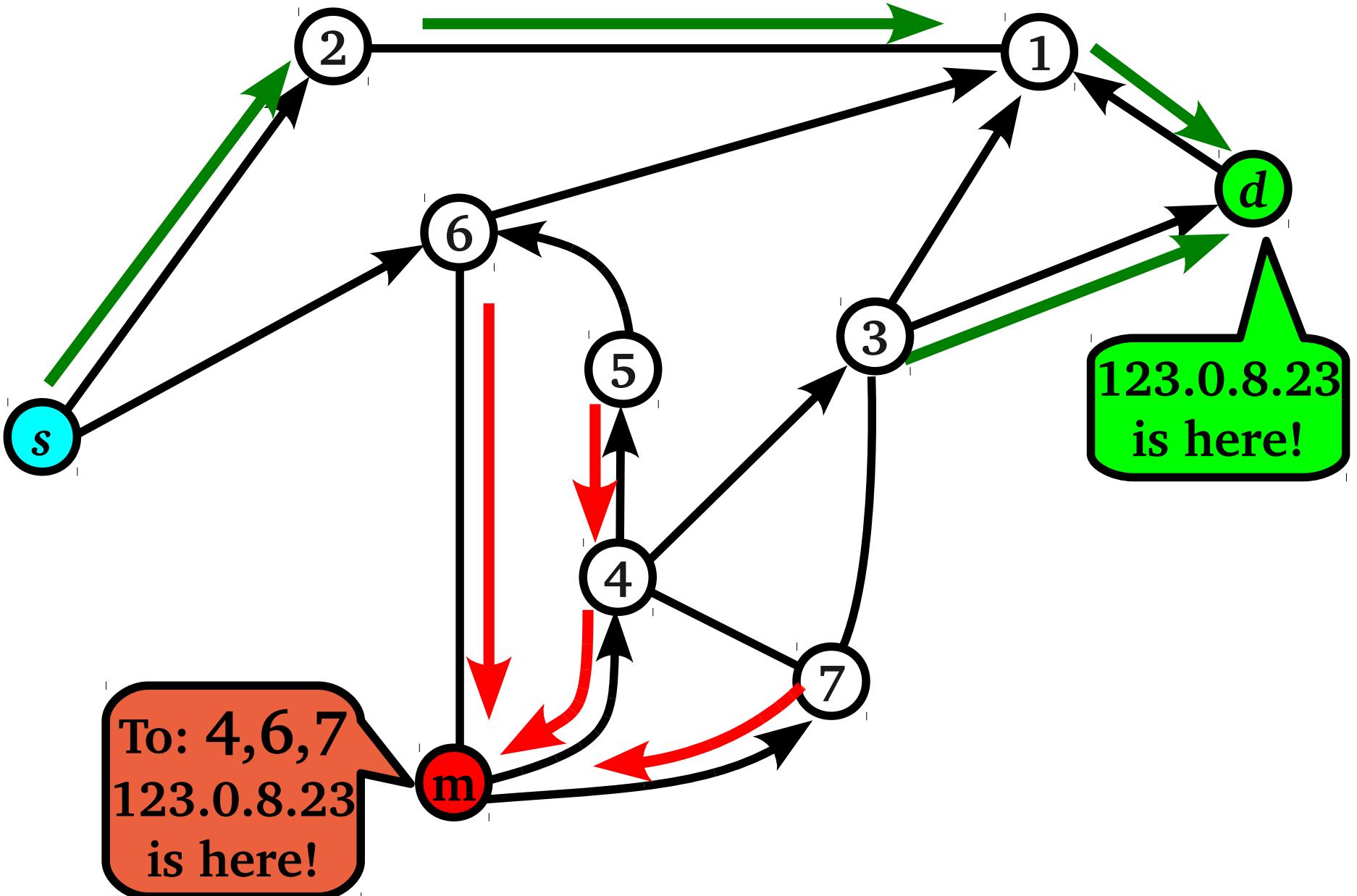
path (4 3 d) is disrupted by better class by path (4 m)



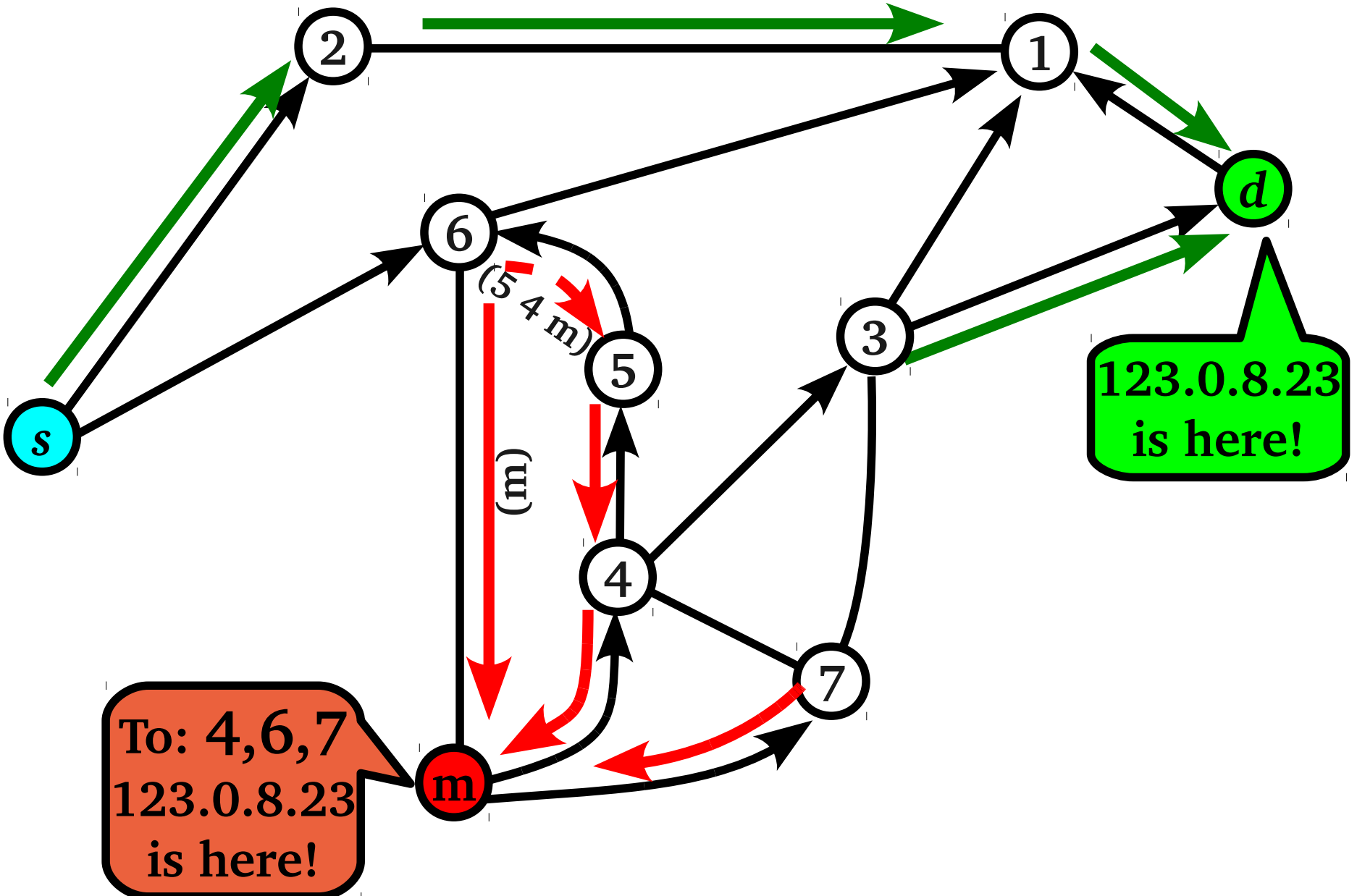
example of hijacking



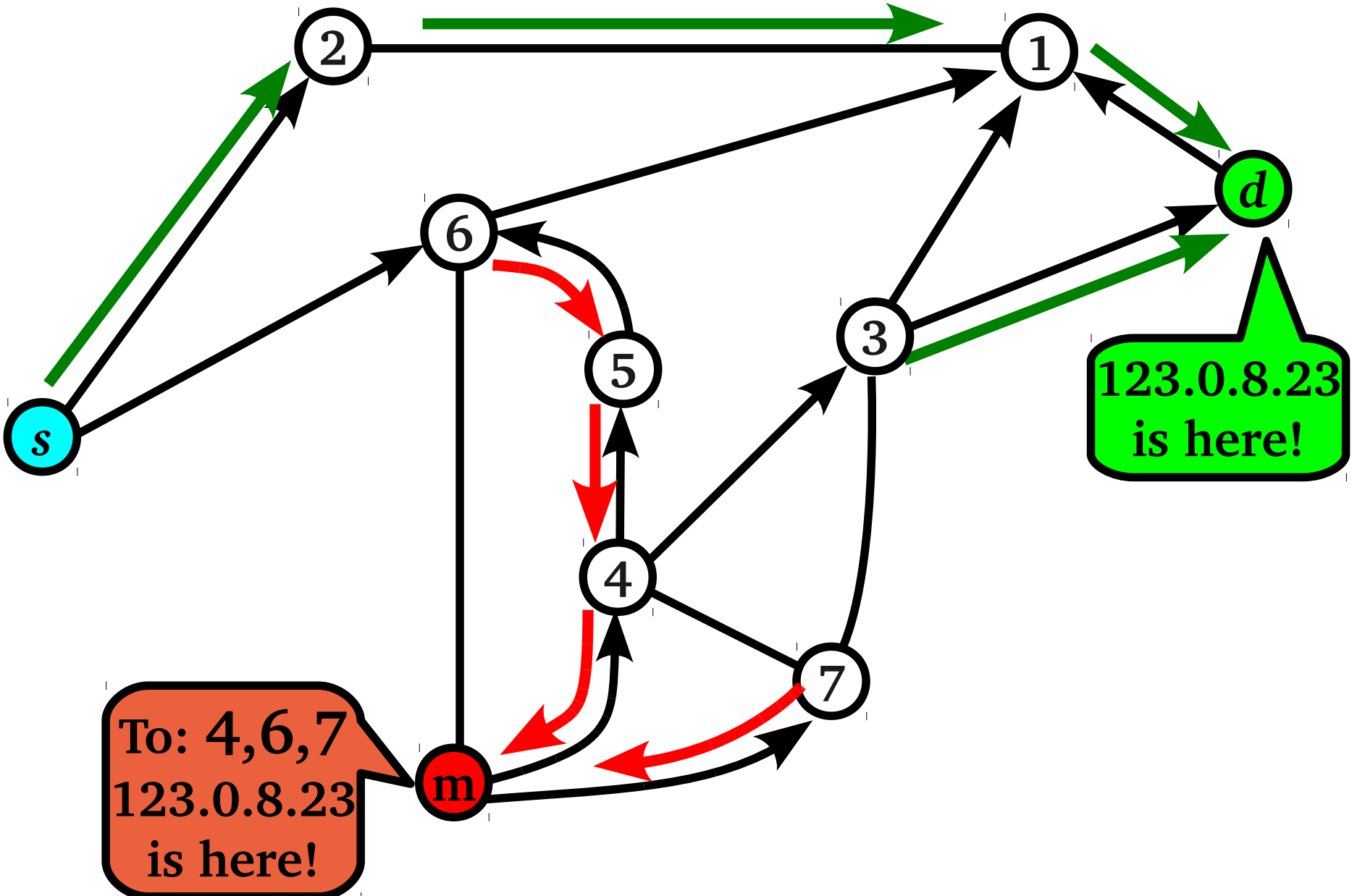
example of hijacking



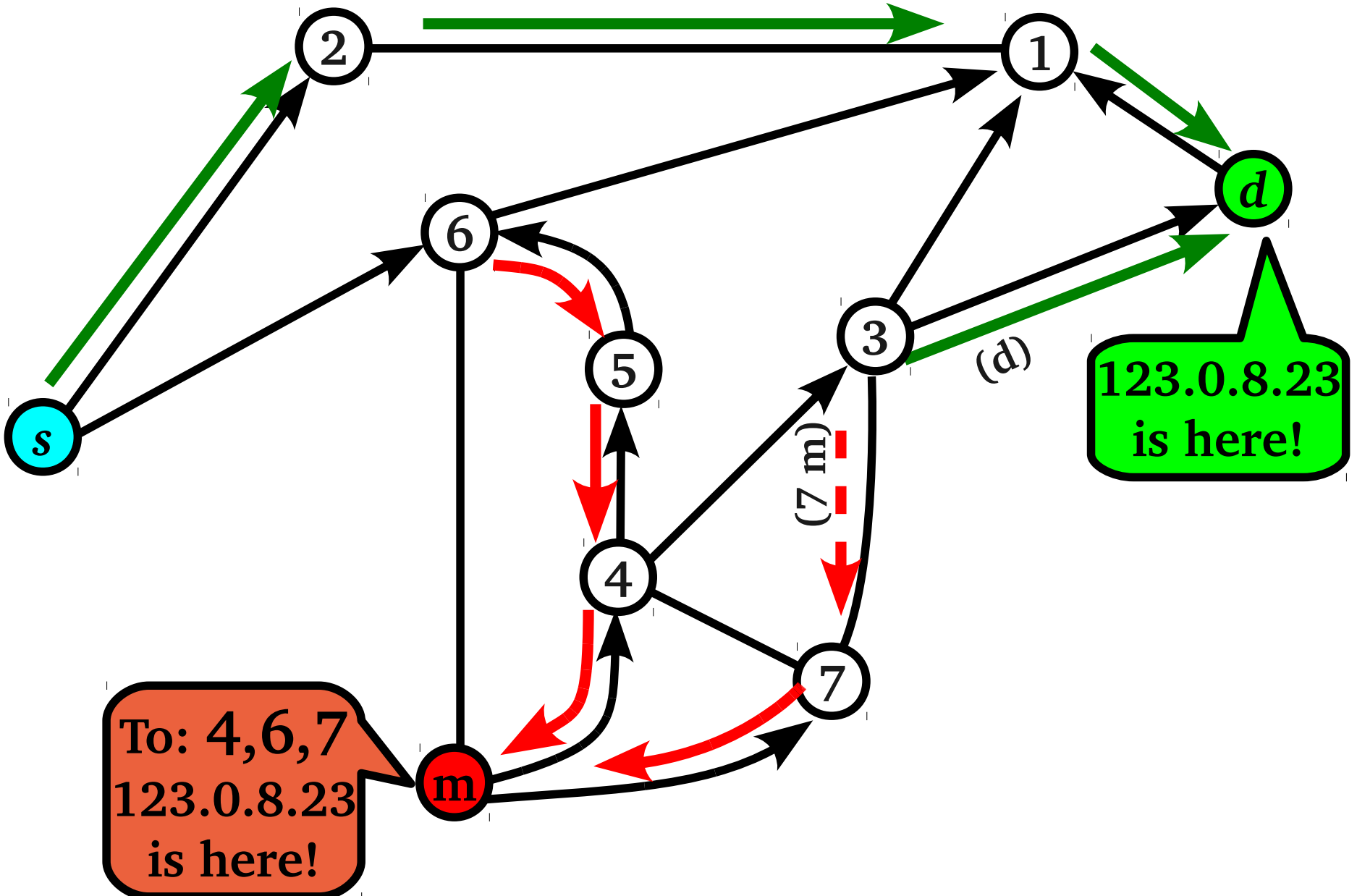
prefer customer rule



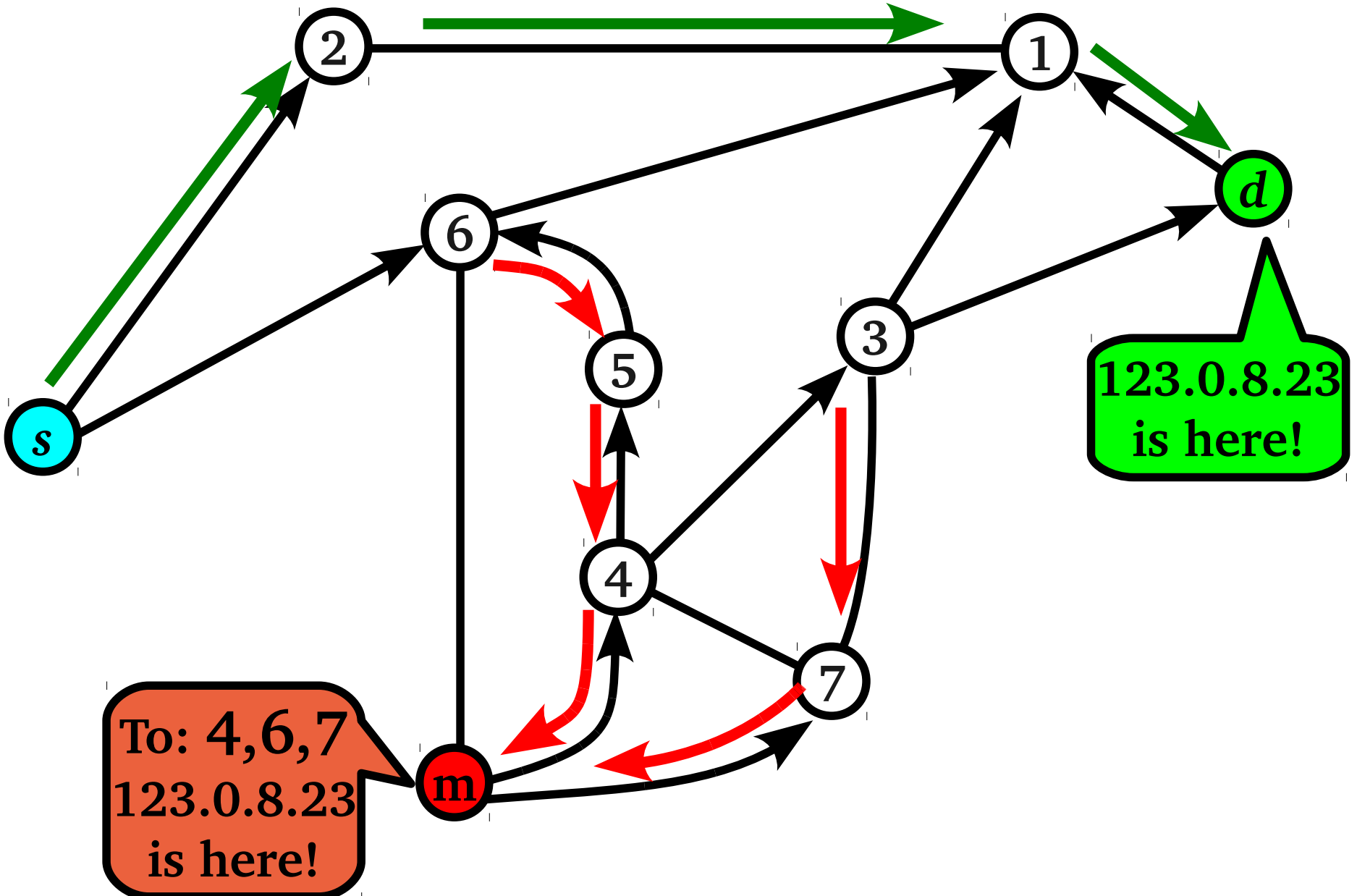
prefer customer rule



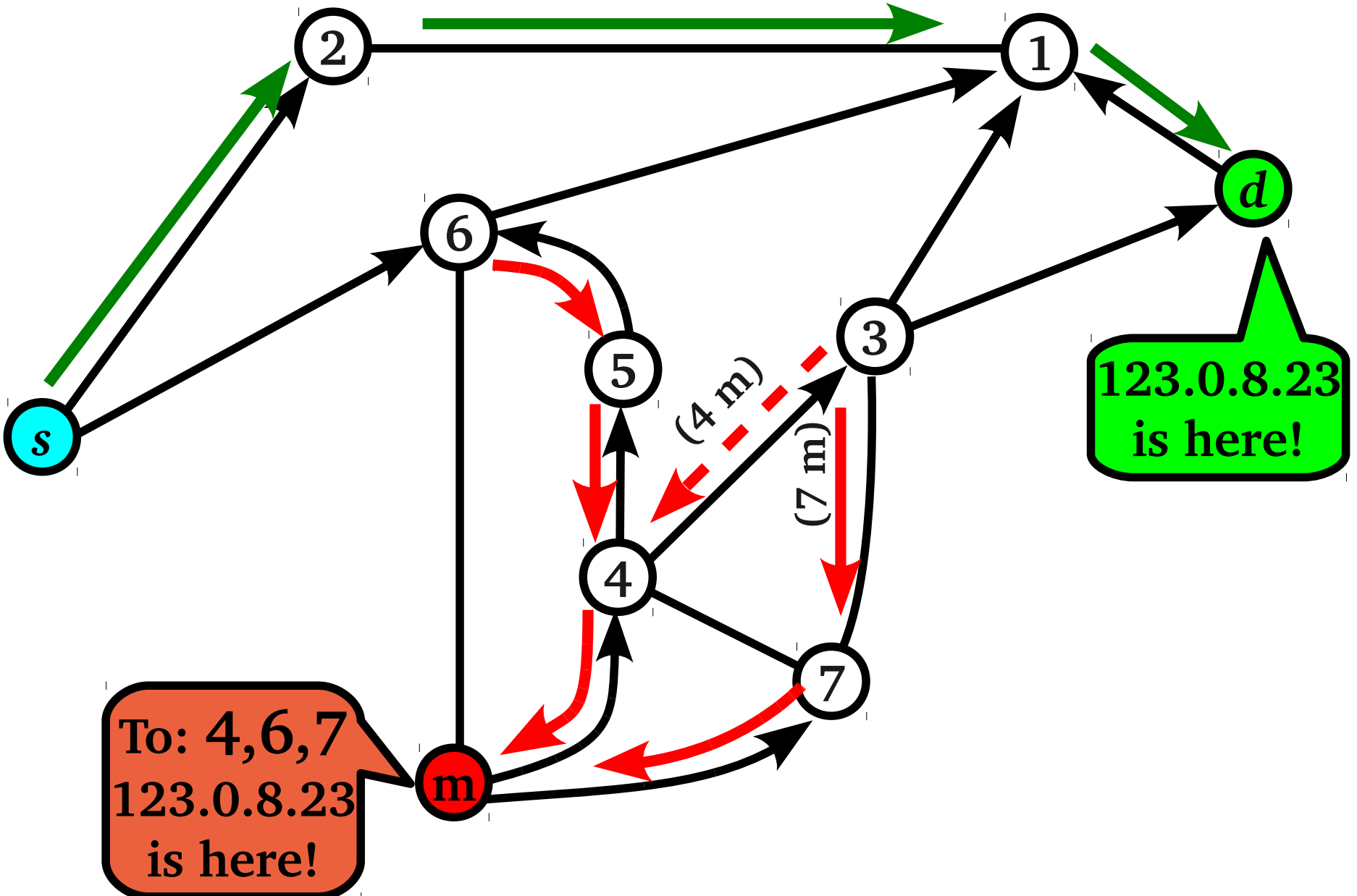
prefer peer rule



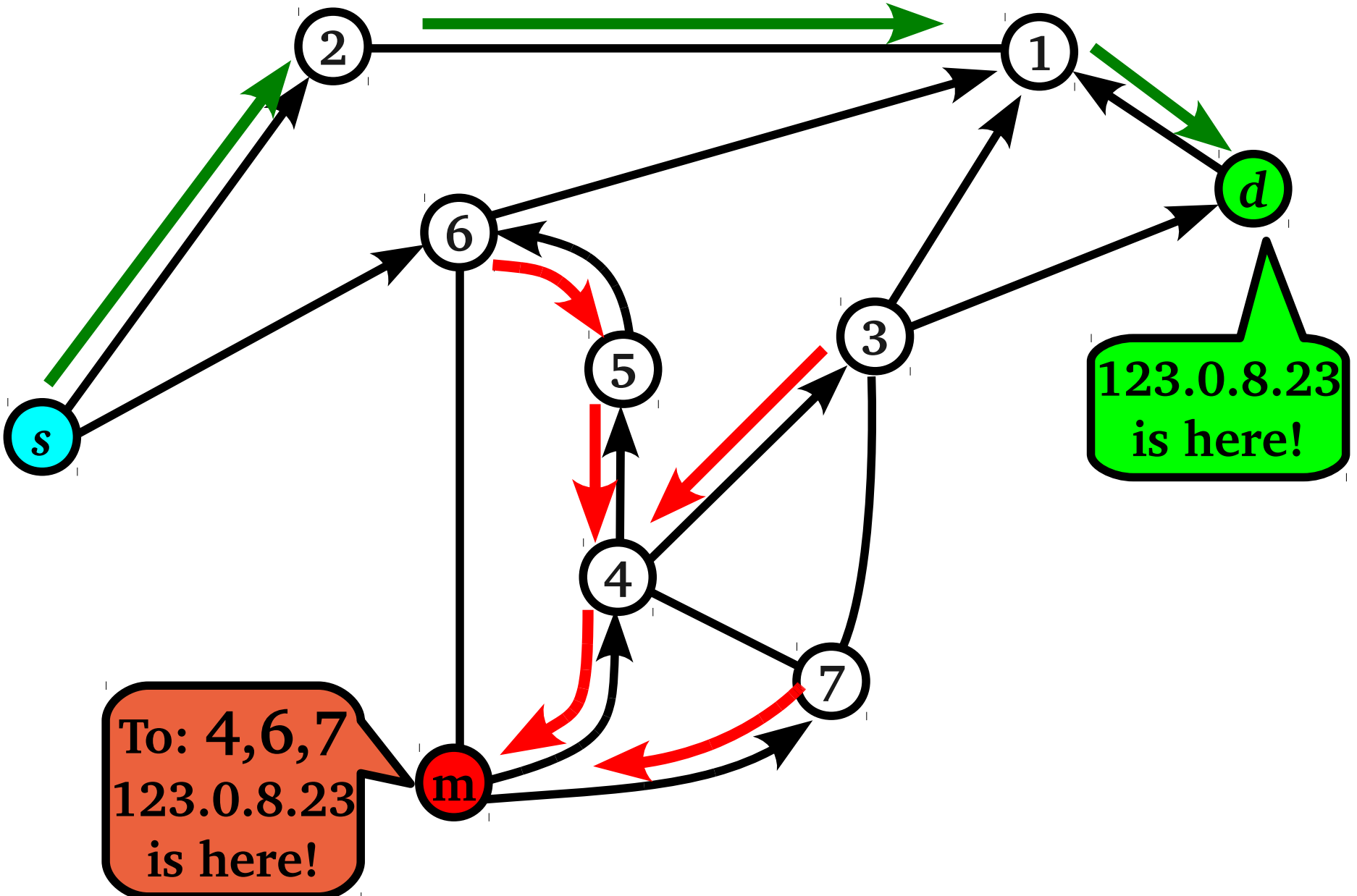
prefer peer rule



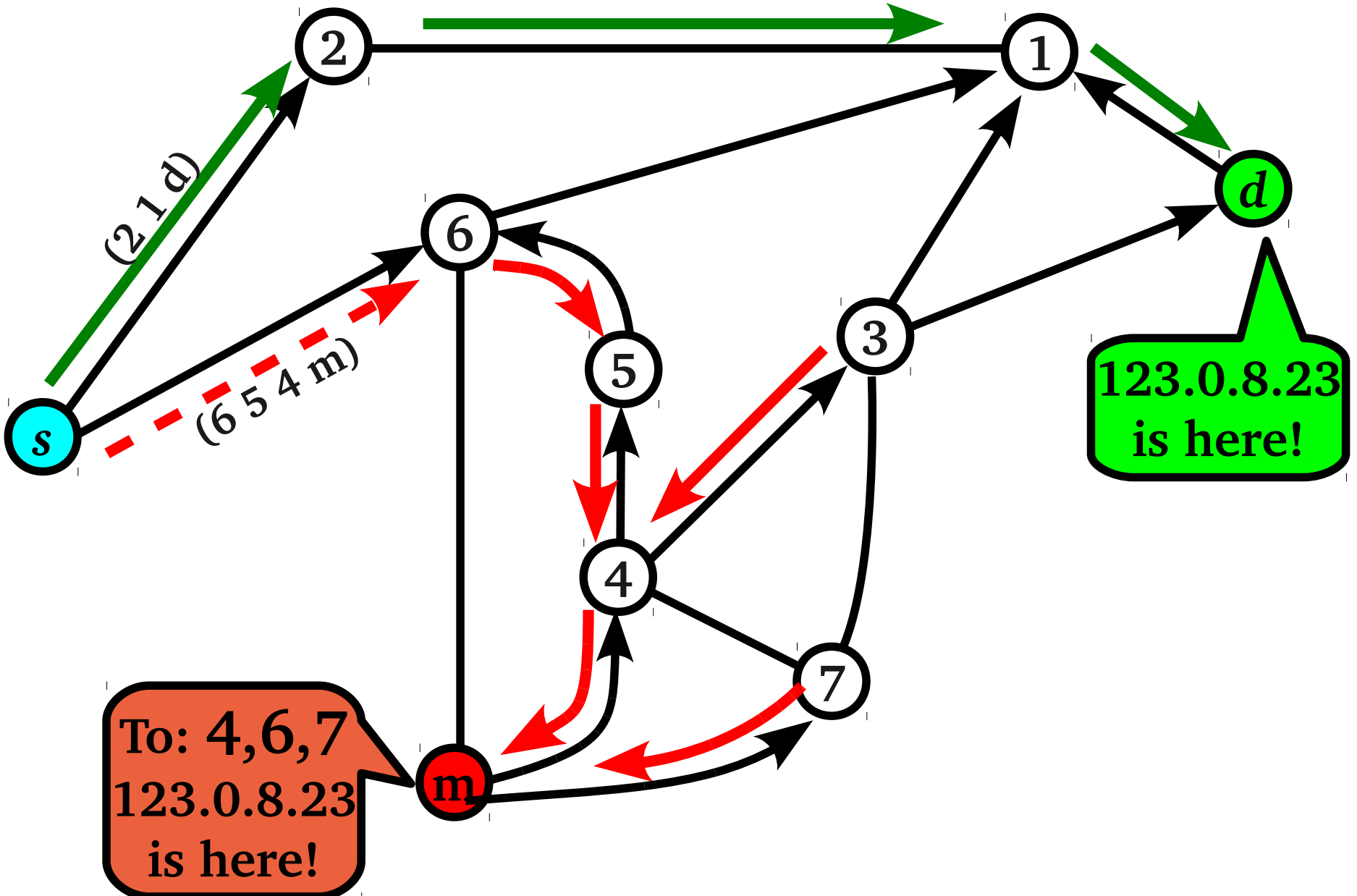
prefer customer rule



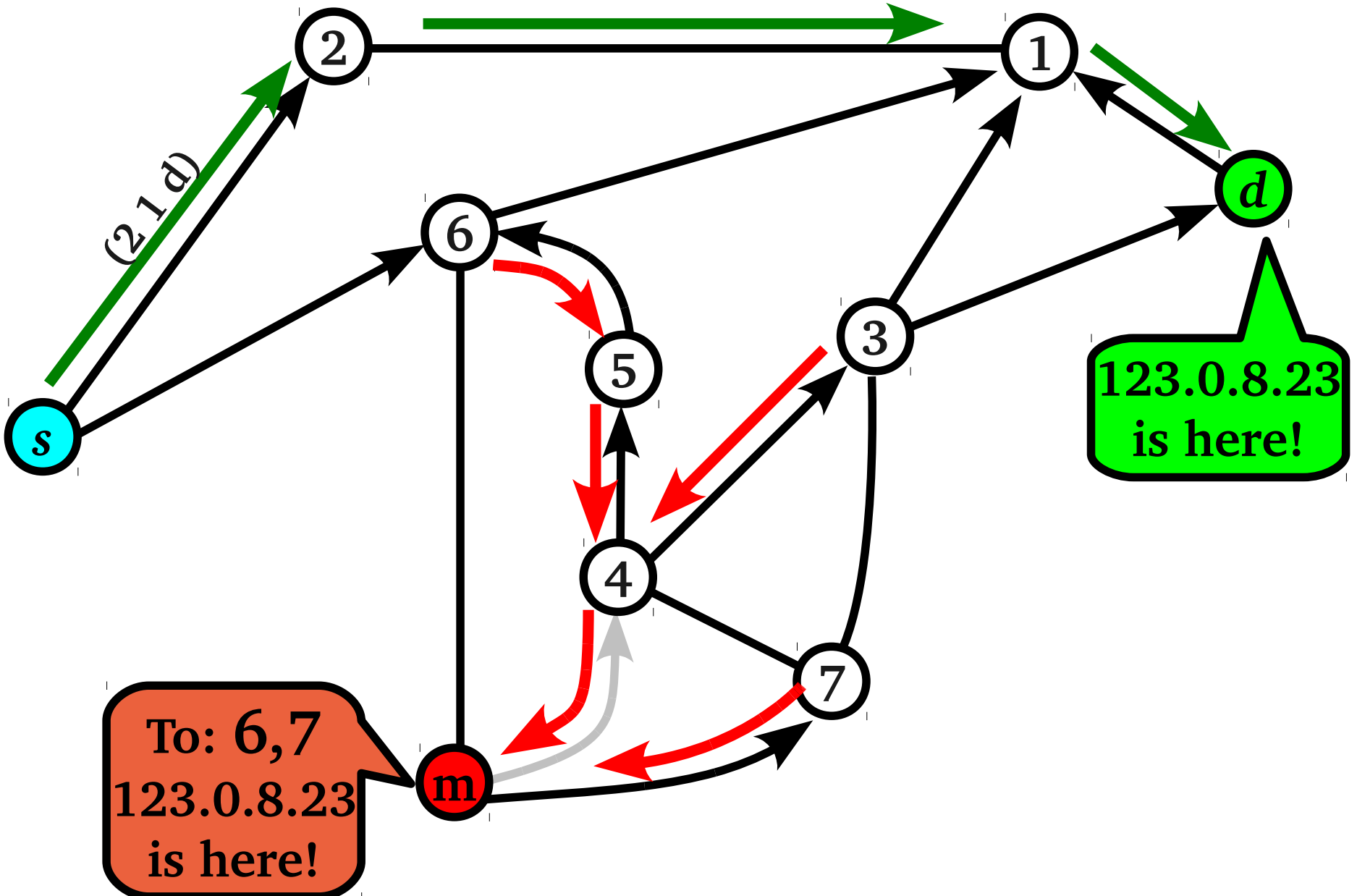
prefer customer rule



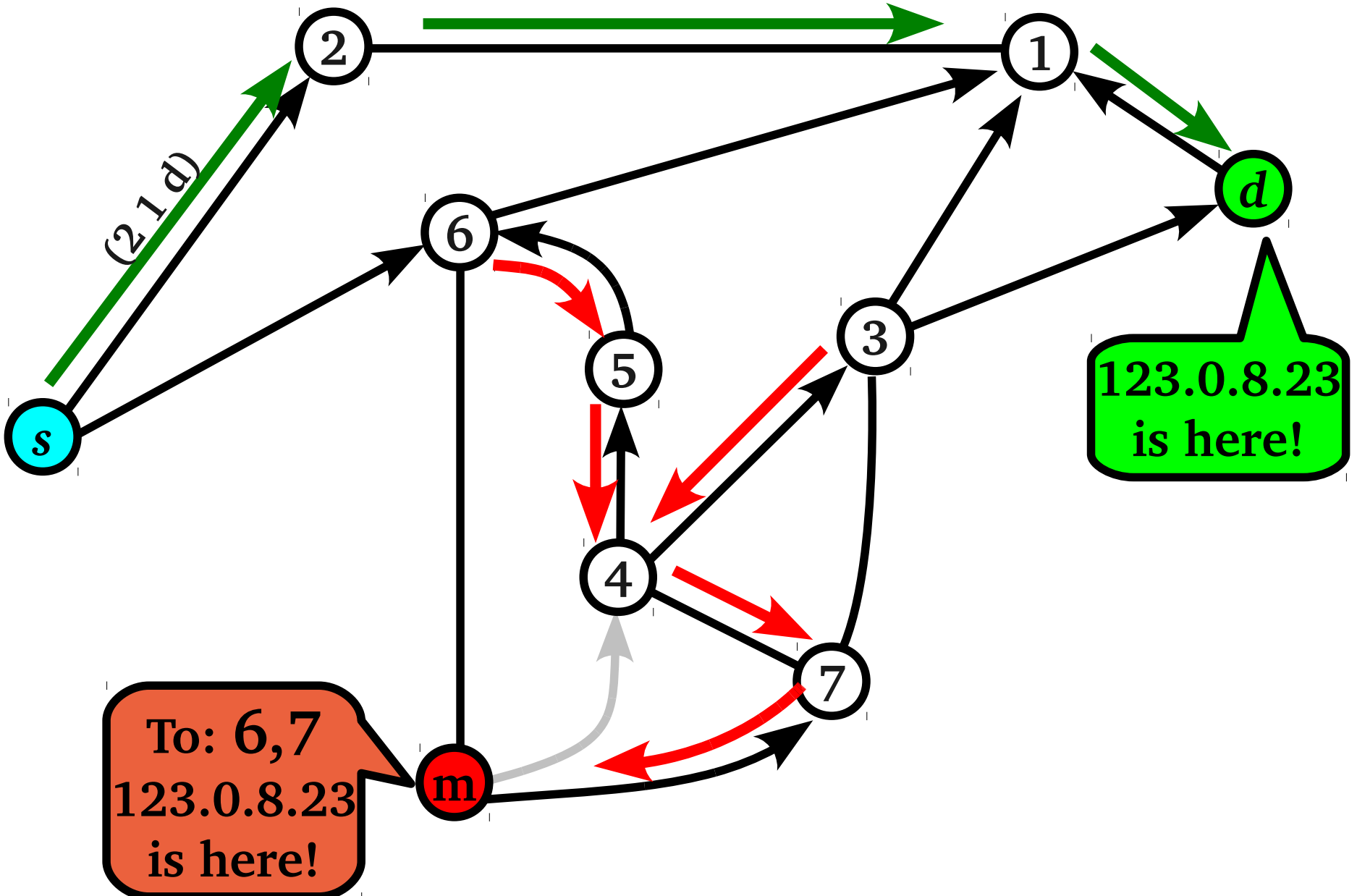
example of unsuccessful hijacking



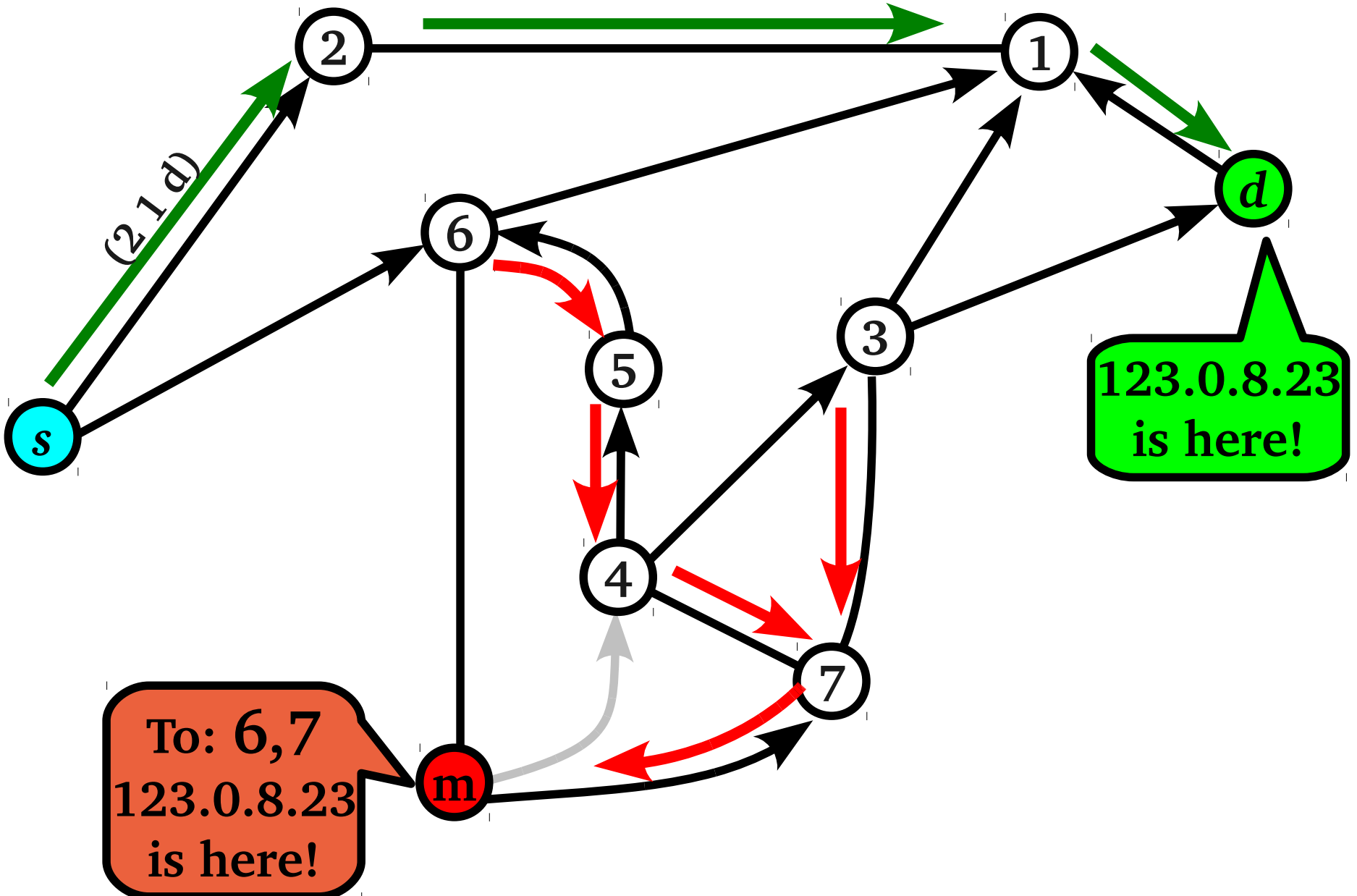
example of hijacking:
announce to a subset of neighbors



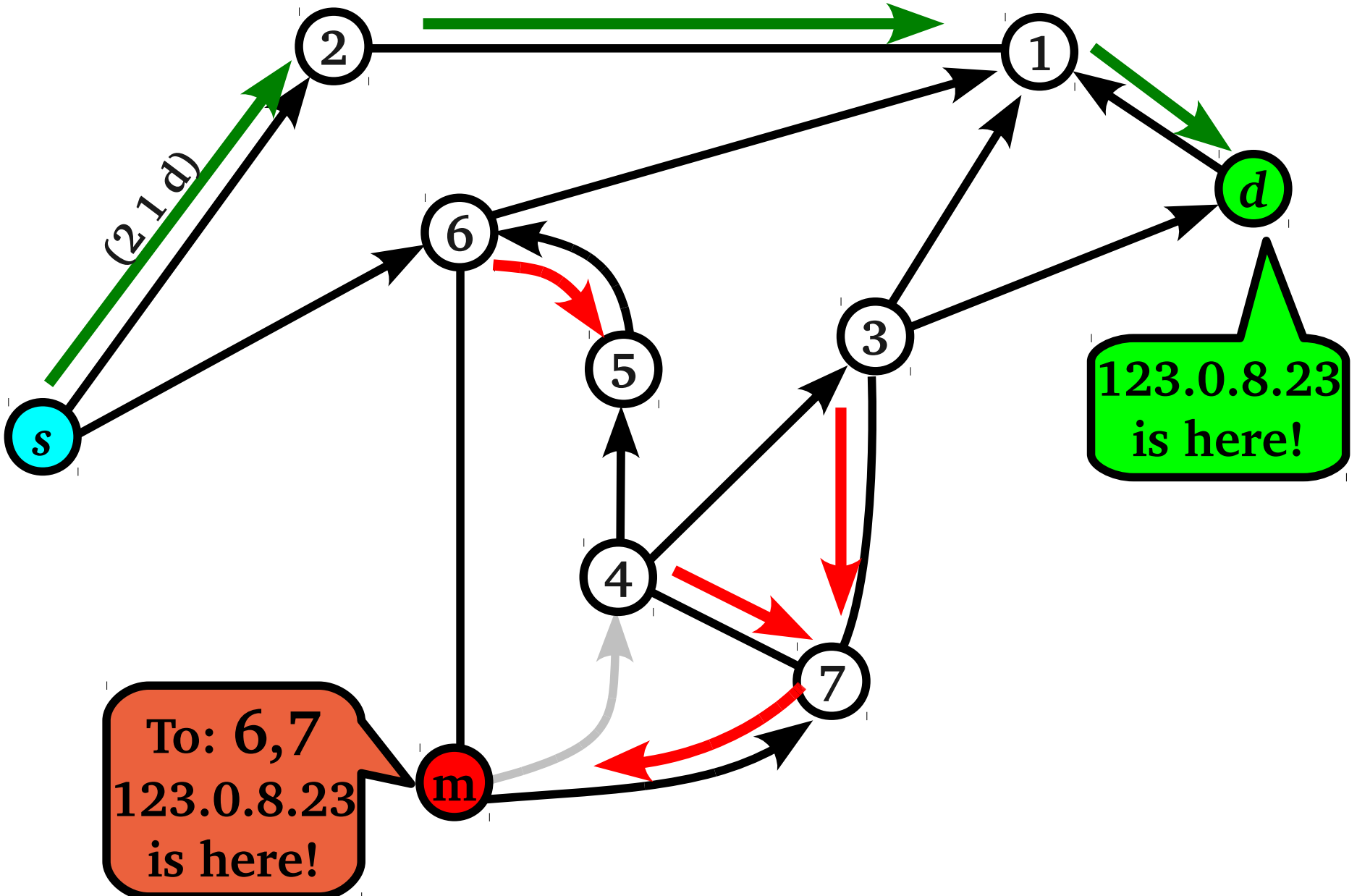
example of hijacking:
announce to a subset of neighbors



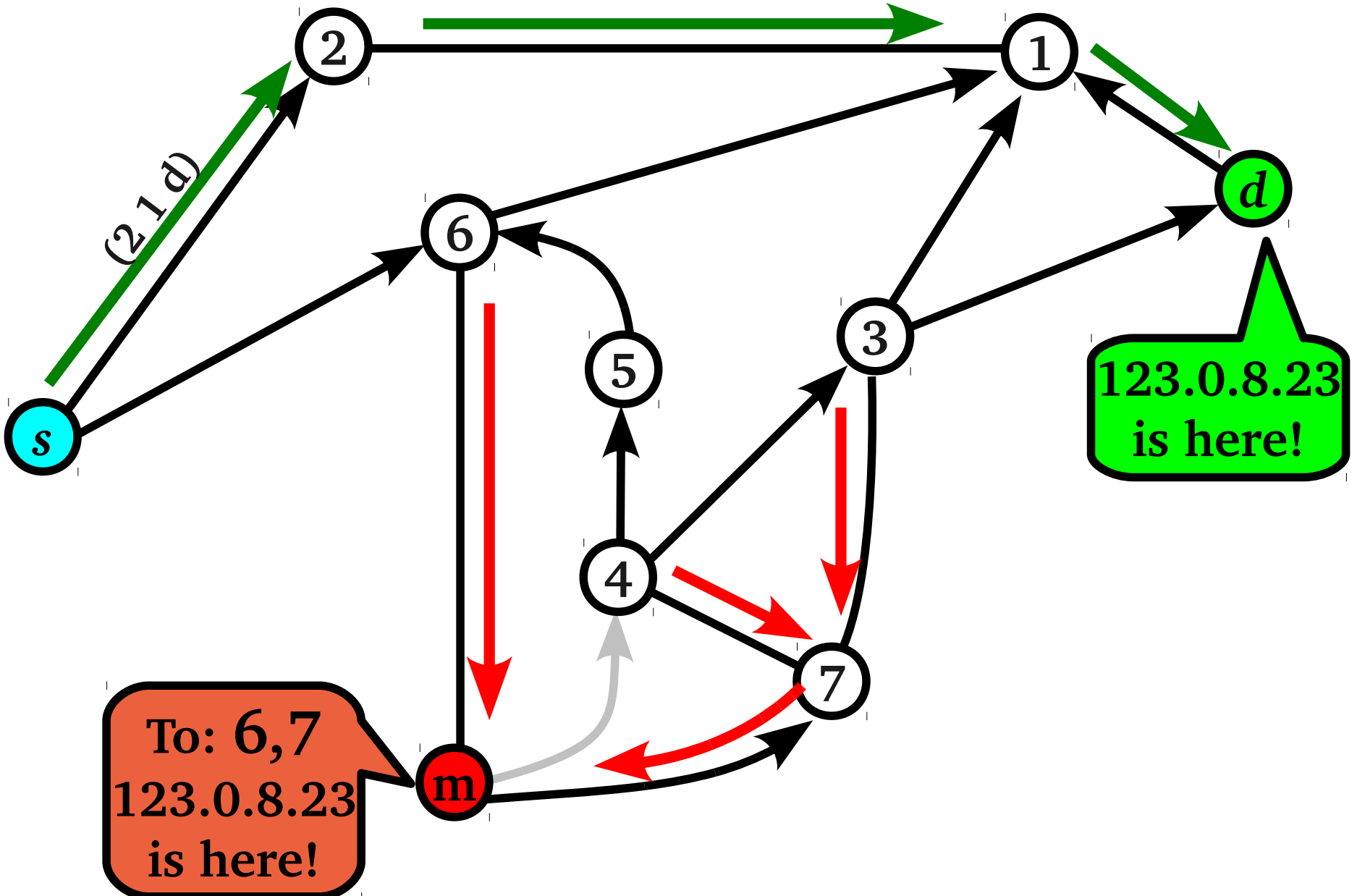
example of hijacking:
announce to a subset of neighbors



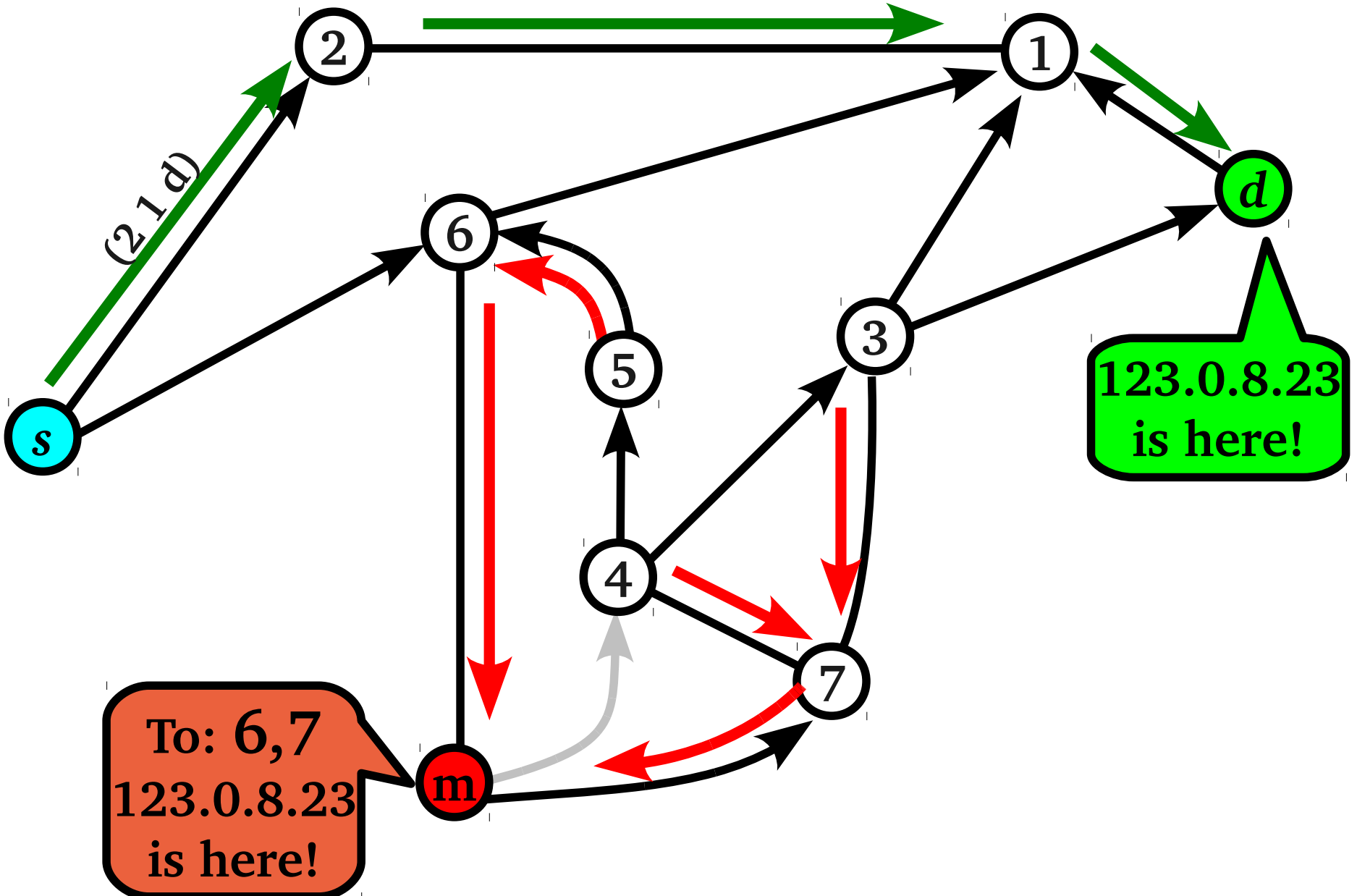
example of hijacking:
announce to a subset of neighbors



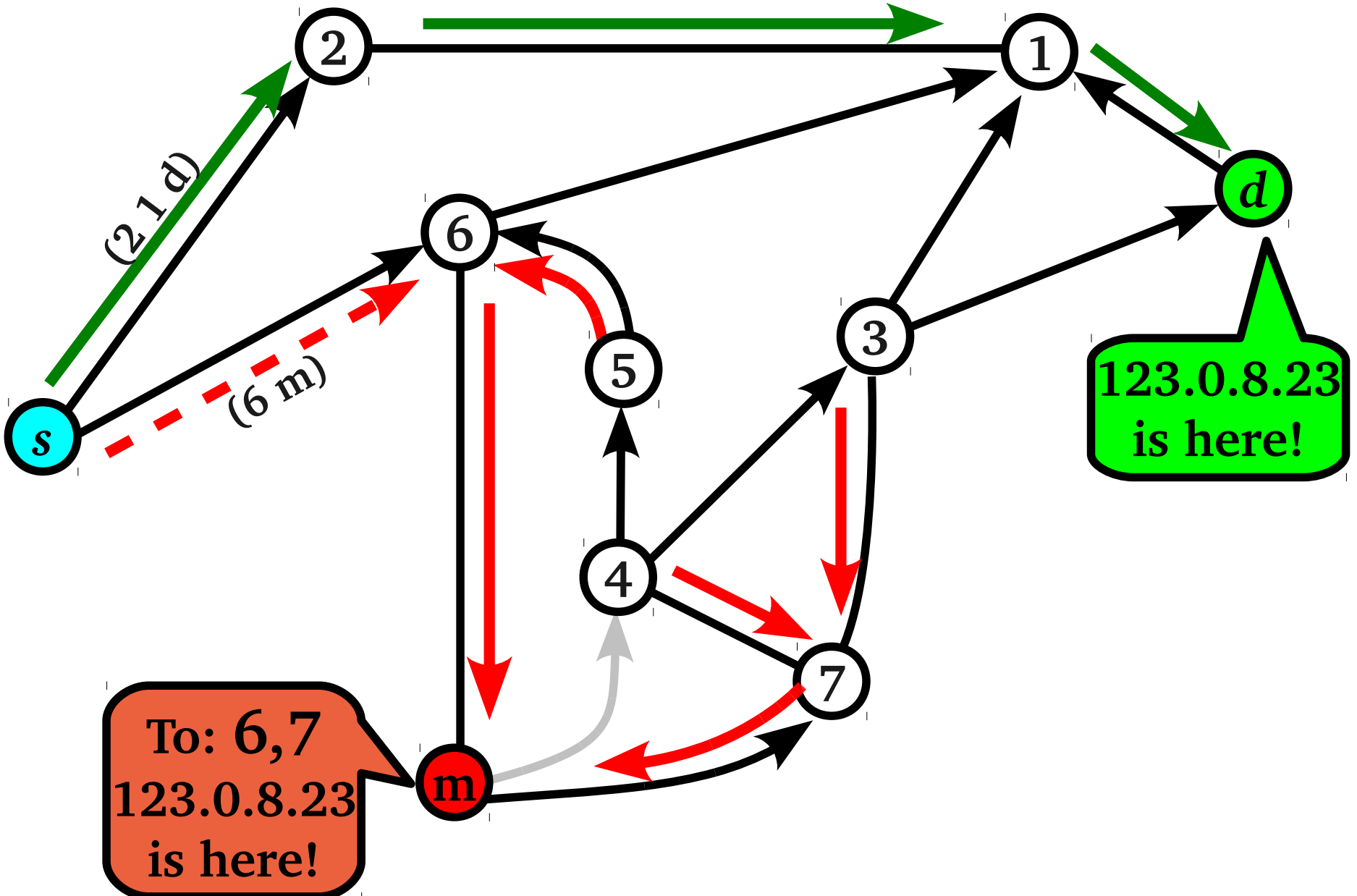
example of hijacking:
announce to a subset of neighbors



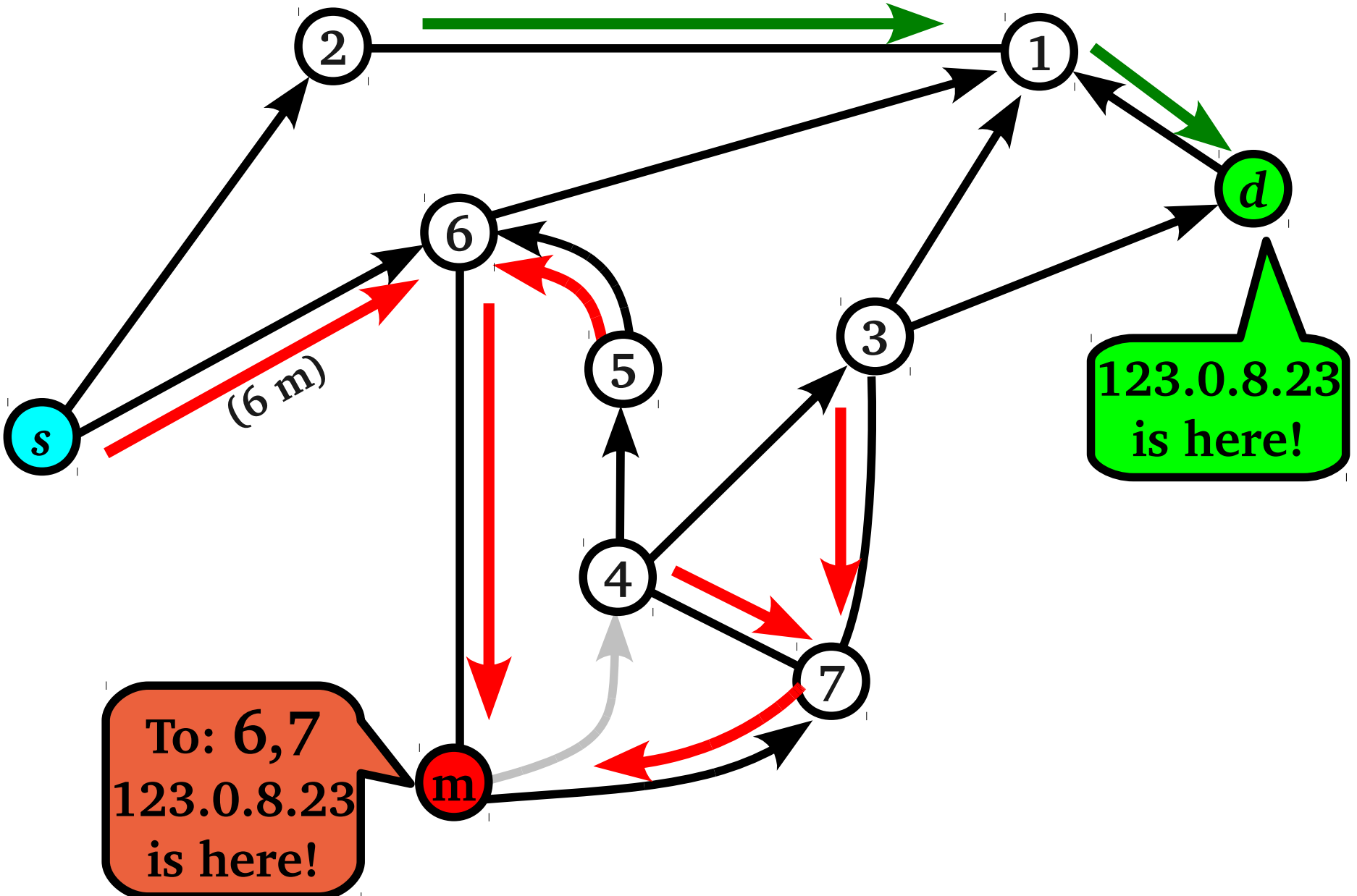
example of hijacking:
announce to a subset of neighbors



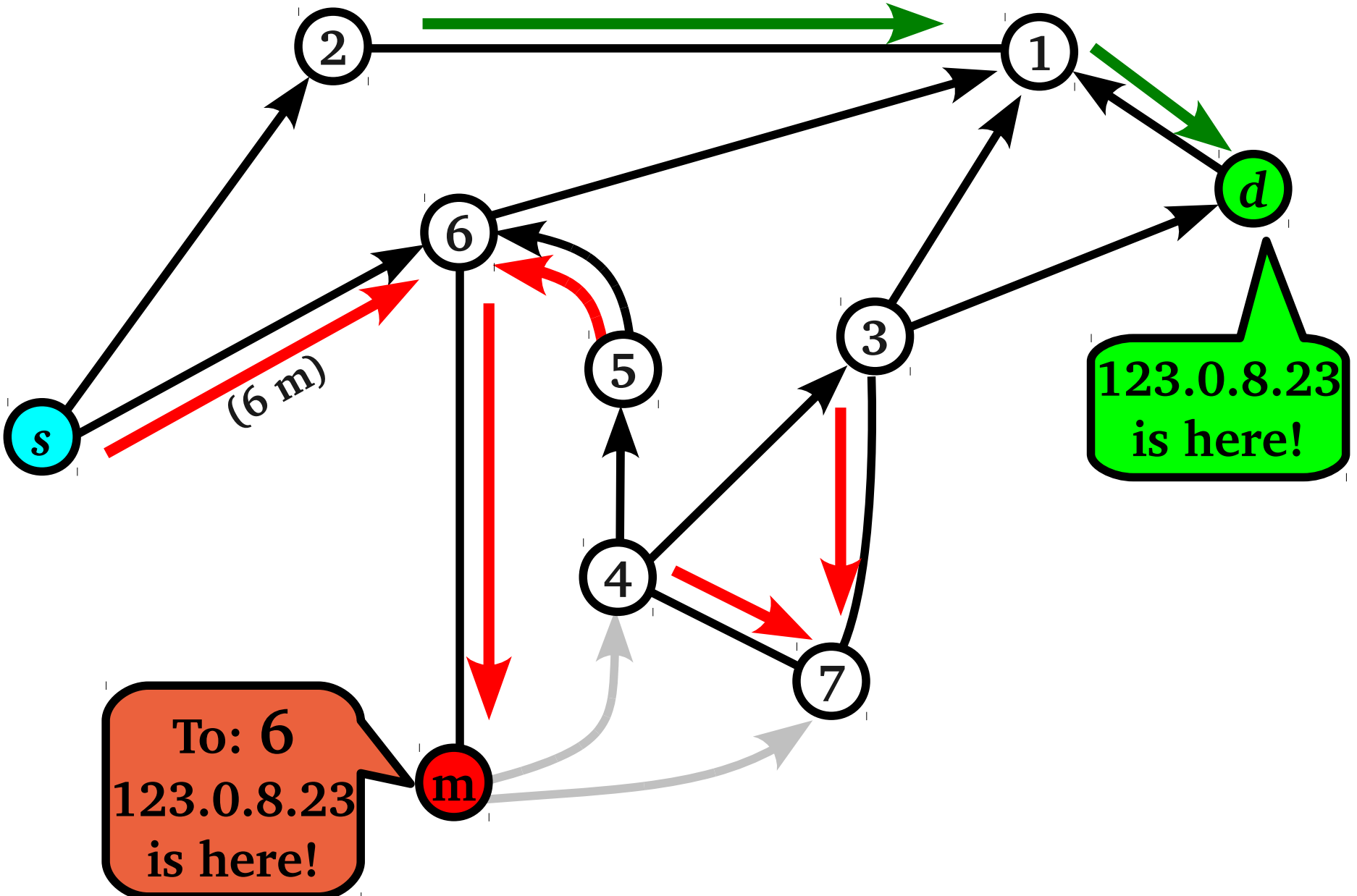
example of hijacking:
announce to a subset of neighbors



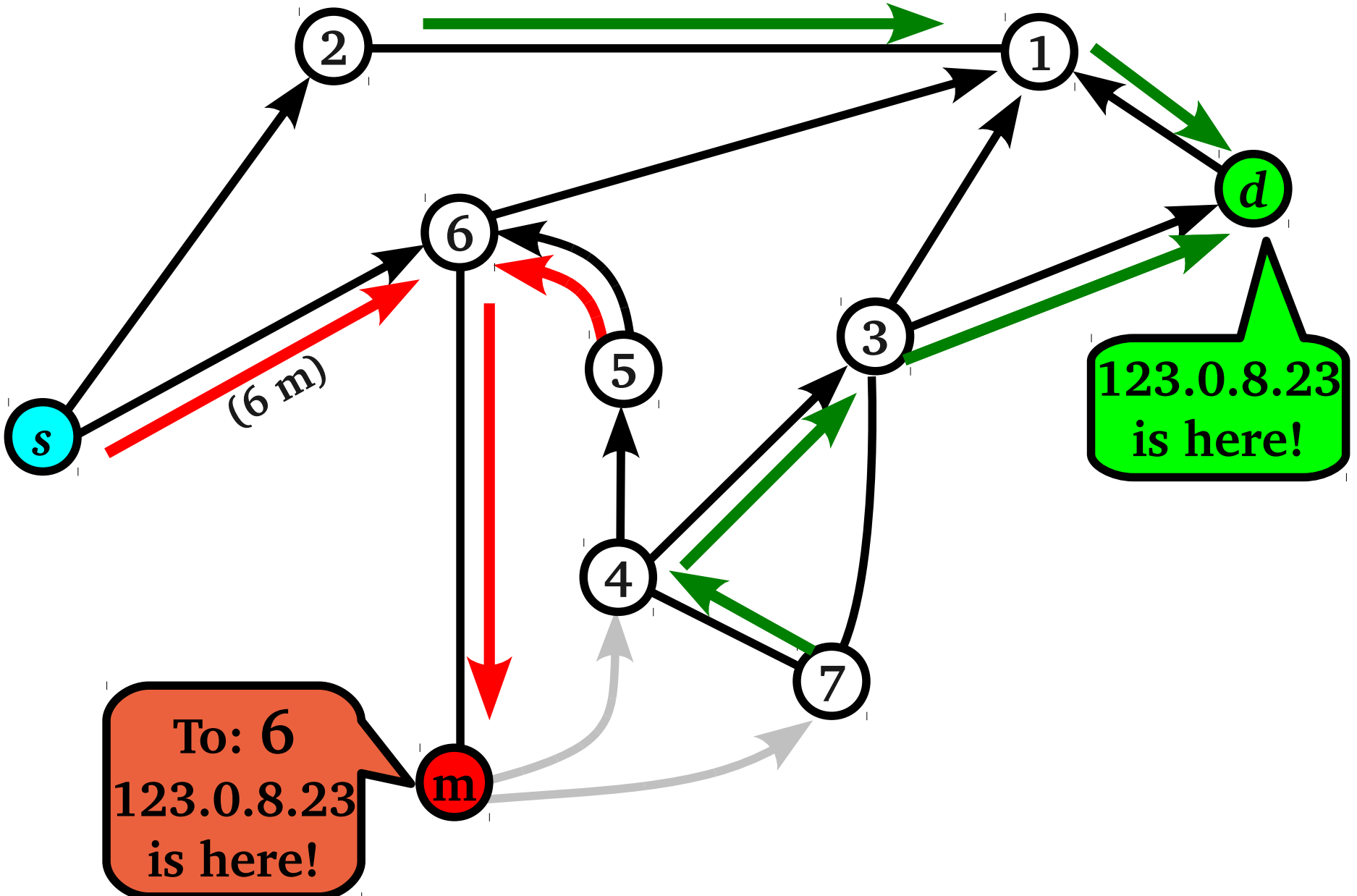
example of successful hijacking: announce to a subset of neighbors



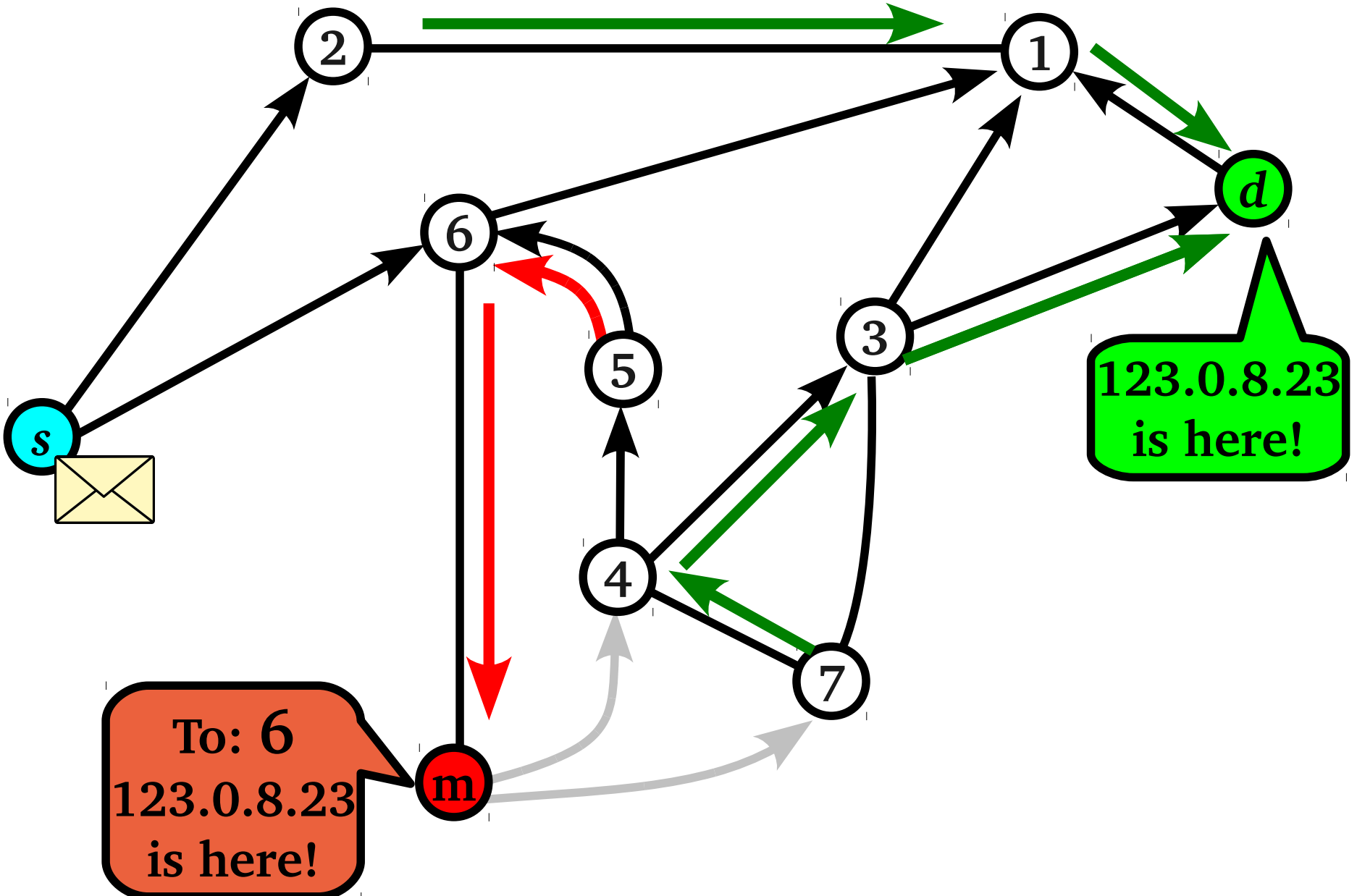
example of successful hijacking: announce to a subset of neighbors



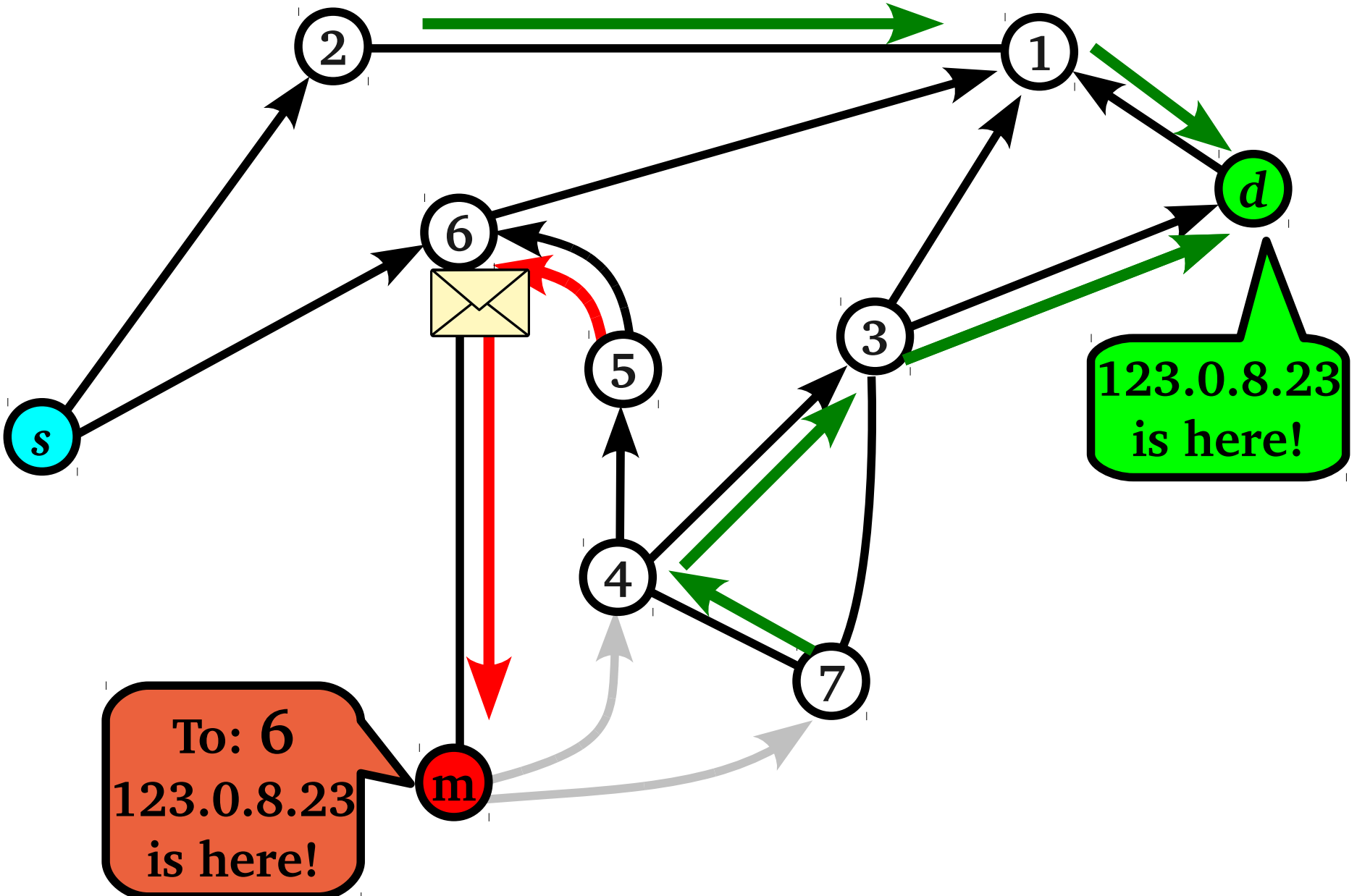
example of a successful interception



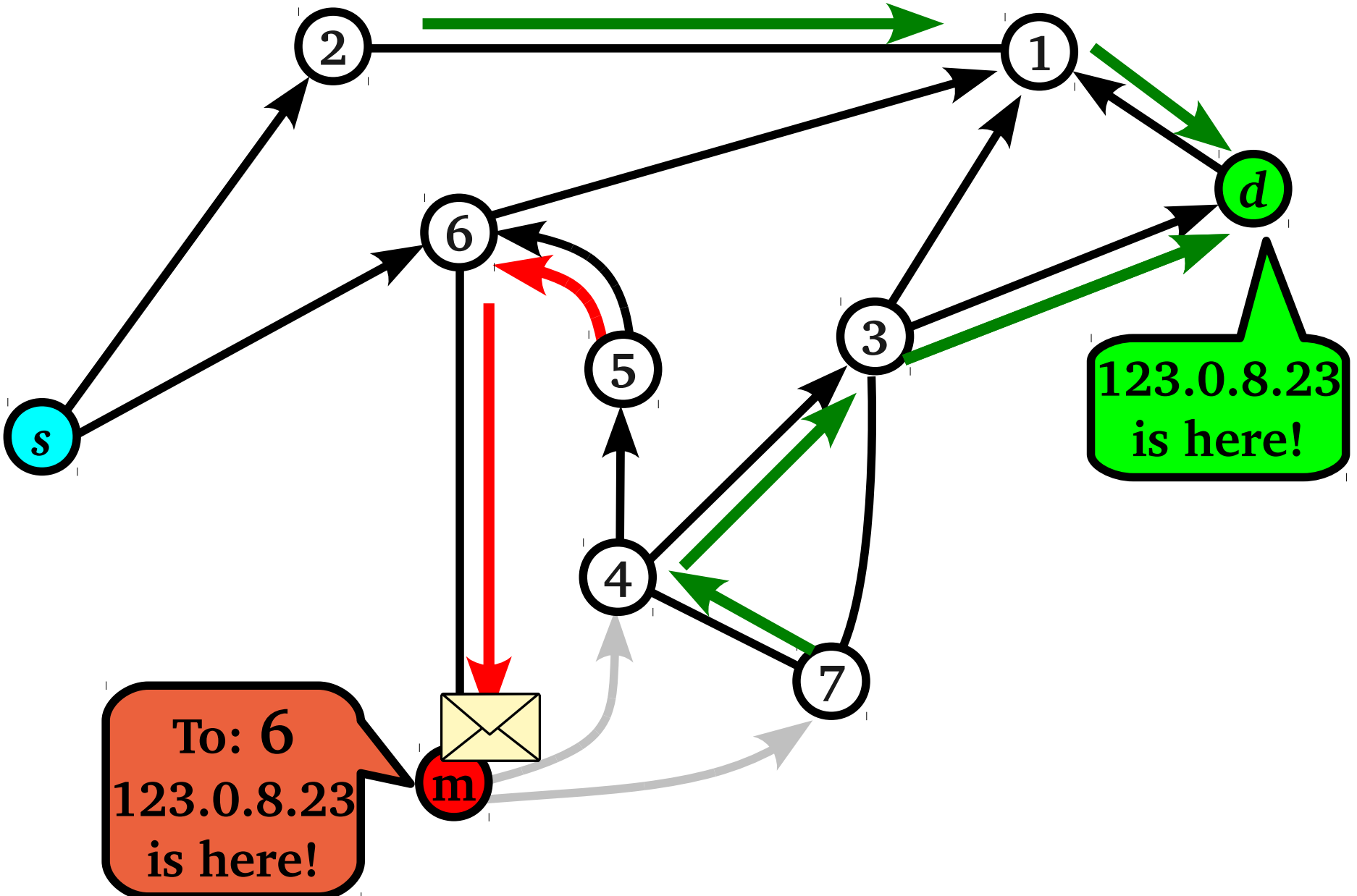
example of a successful interception



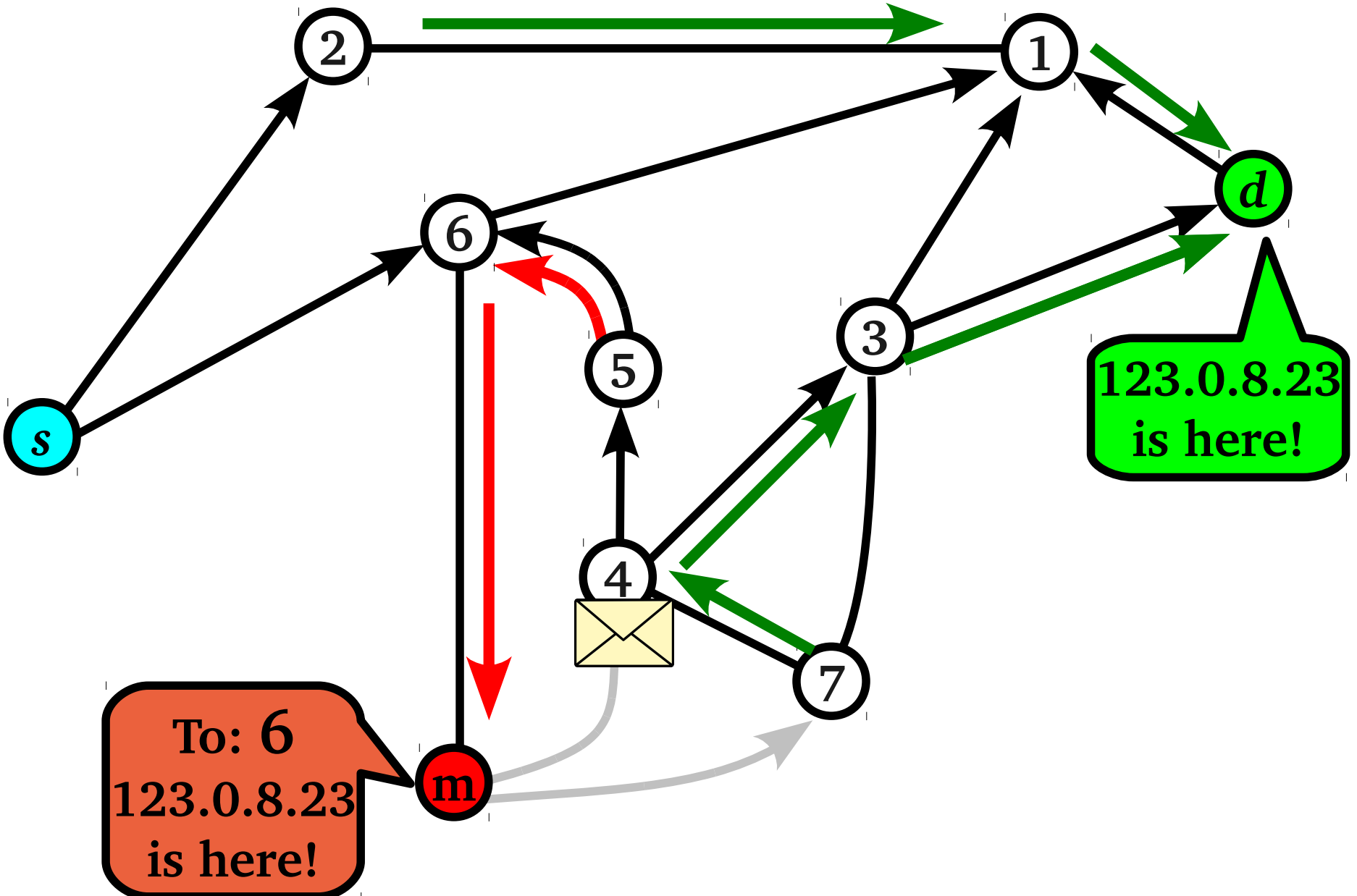
example of a successful interception



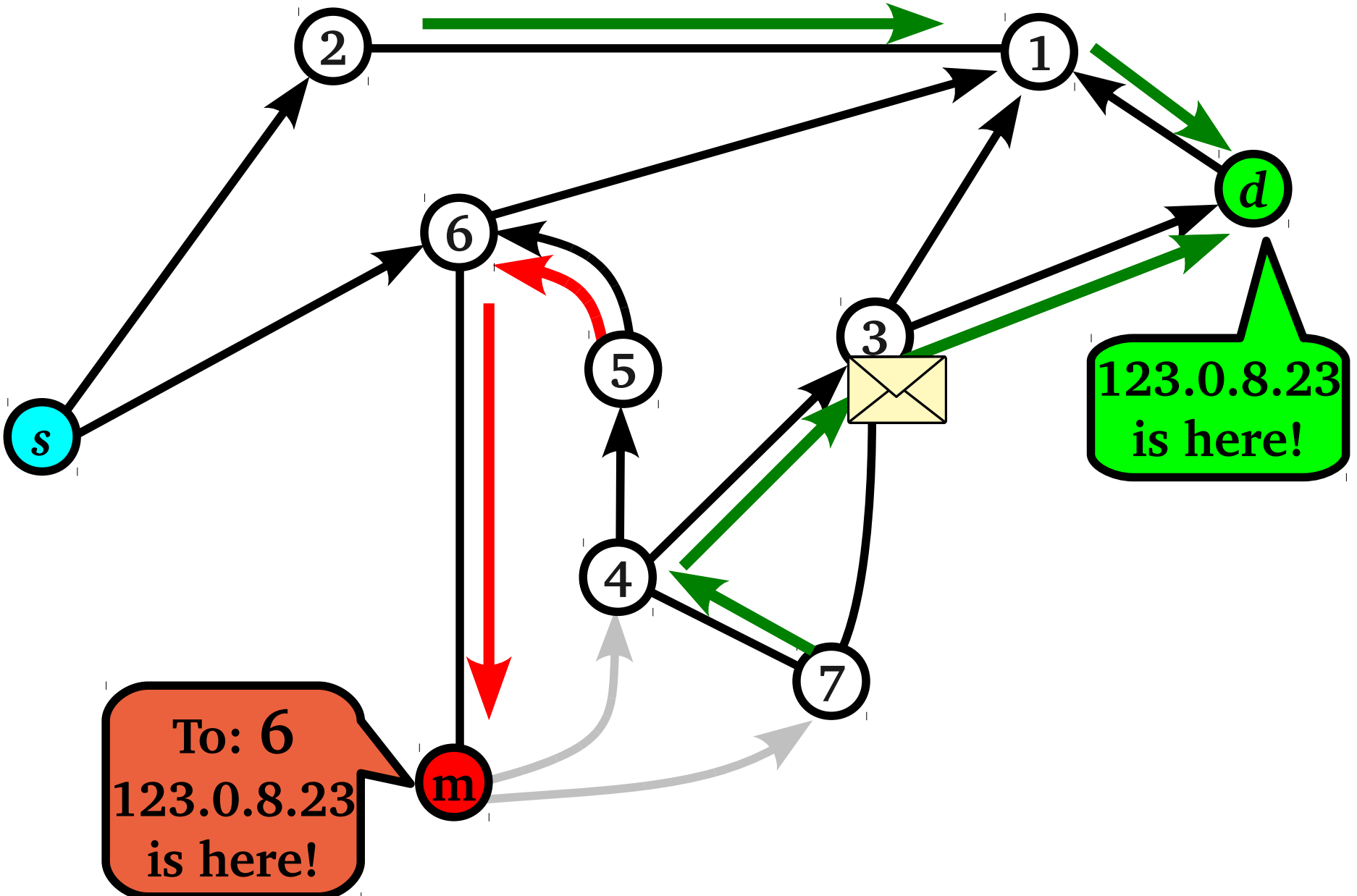
example of a successful interception



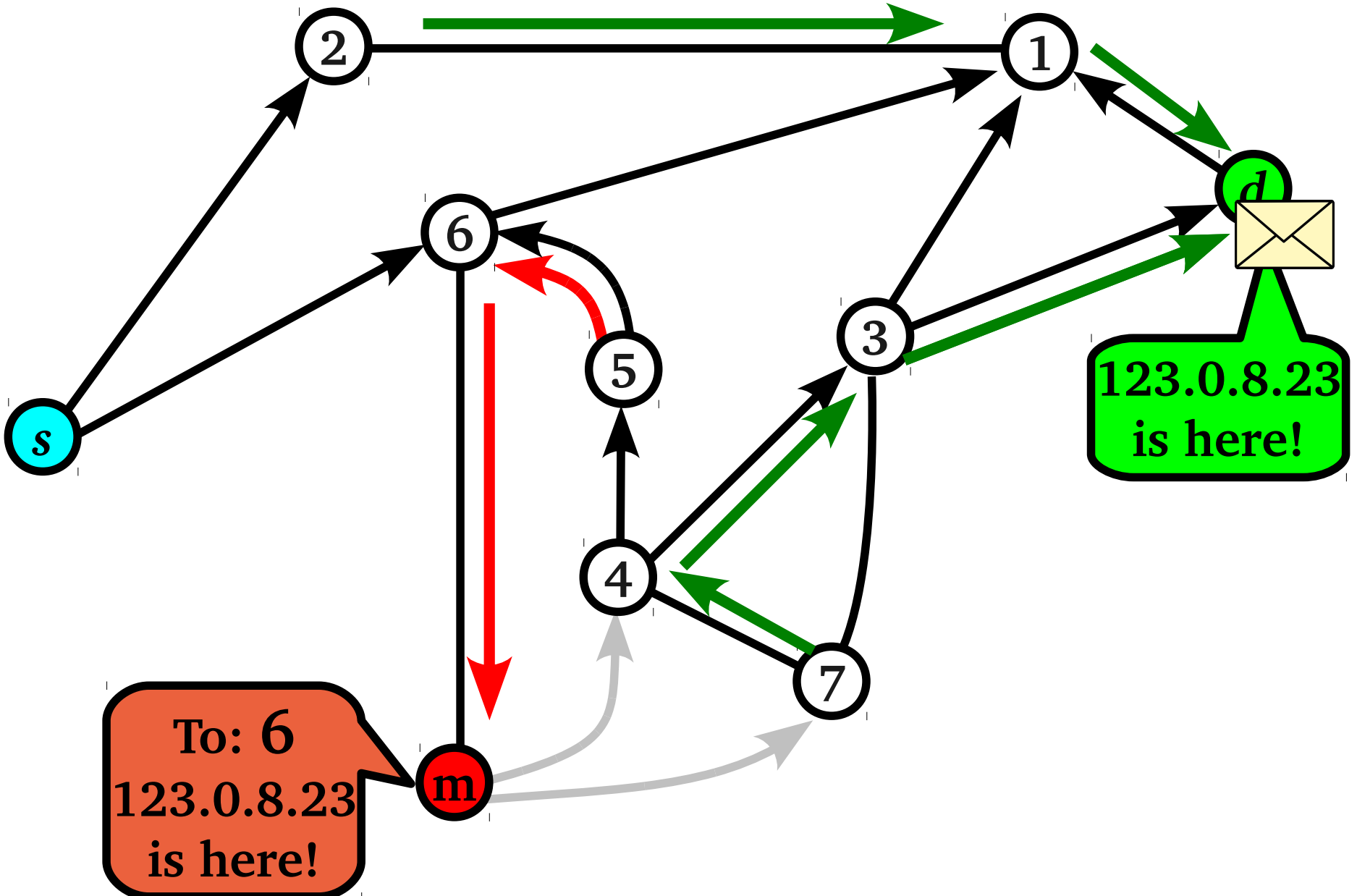
example of a successful interception



example of a successful interception



example of a successful interception



cheating capabilities of the manipulator

origin-spoofing cheating capabilities:

- a router can only pretend to be the origin of an arbitrary IP prefix.

cheating capabilities of the manipulator

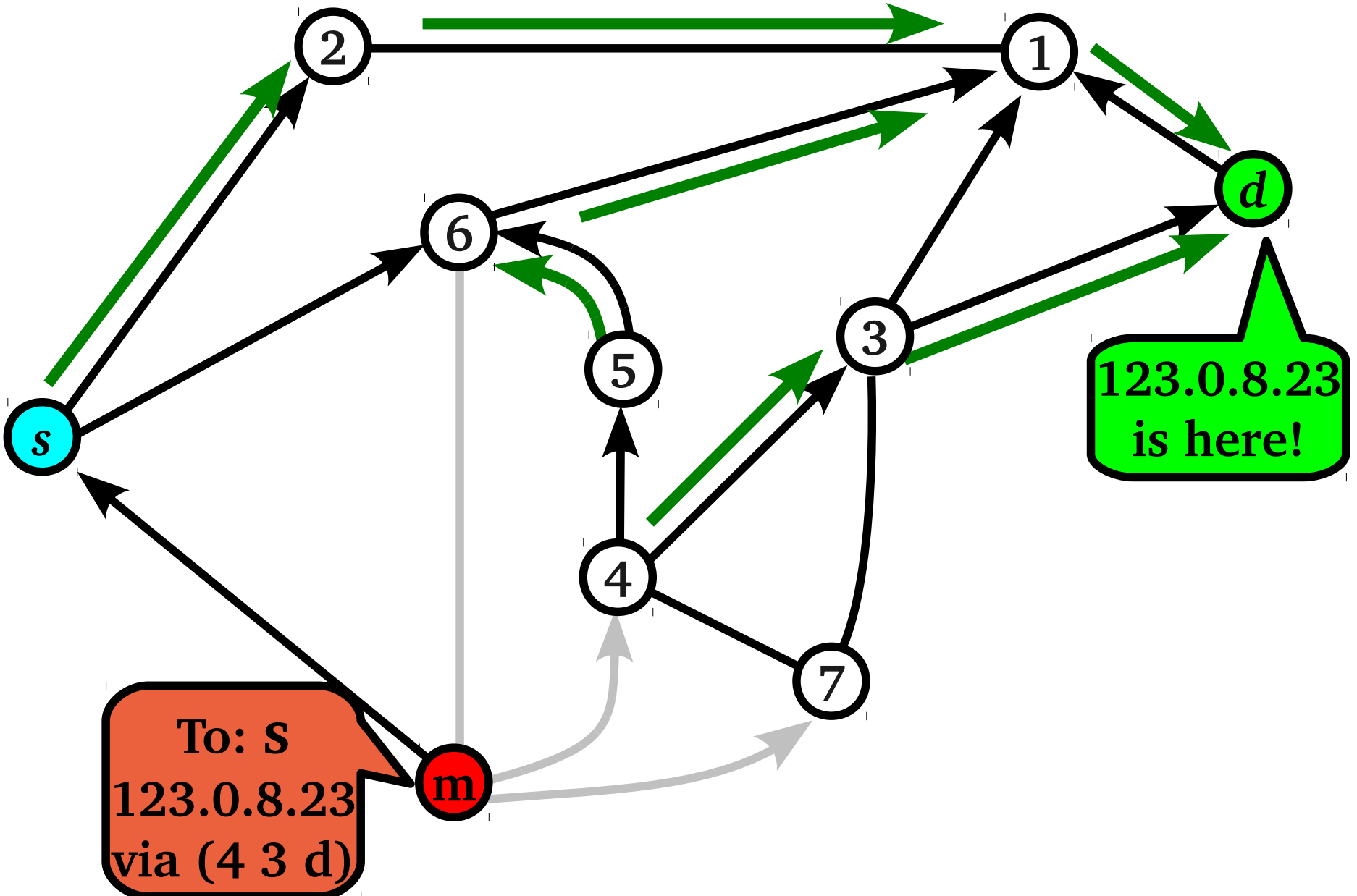
origin-spoofing cheating capabilities:

- a router can only pretend to be the origin of an arbitrary IP prefix.

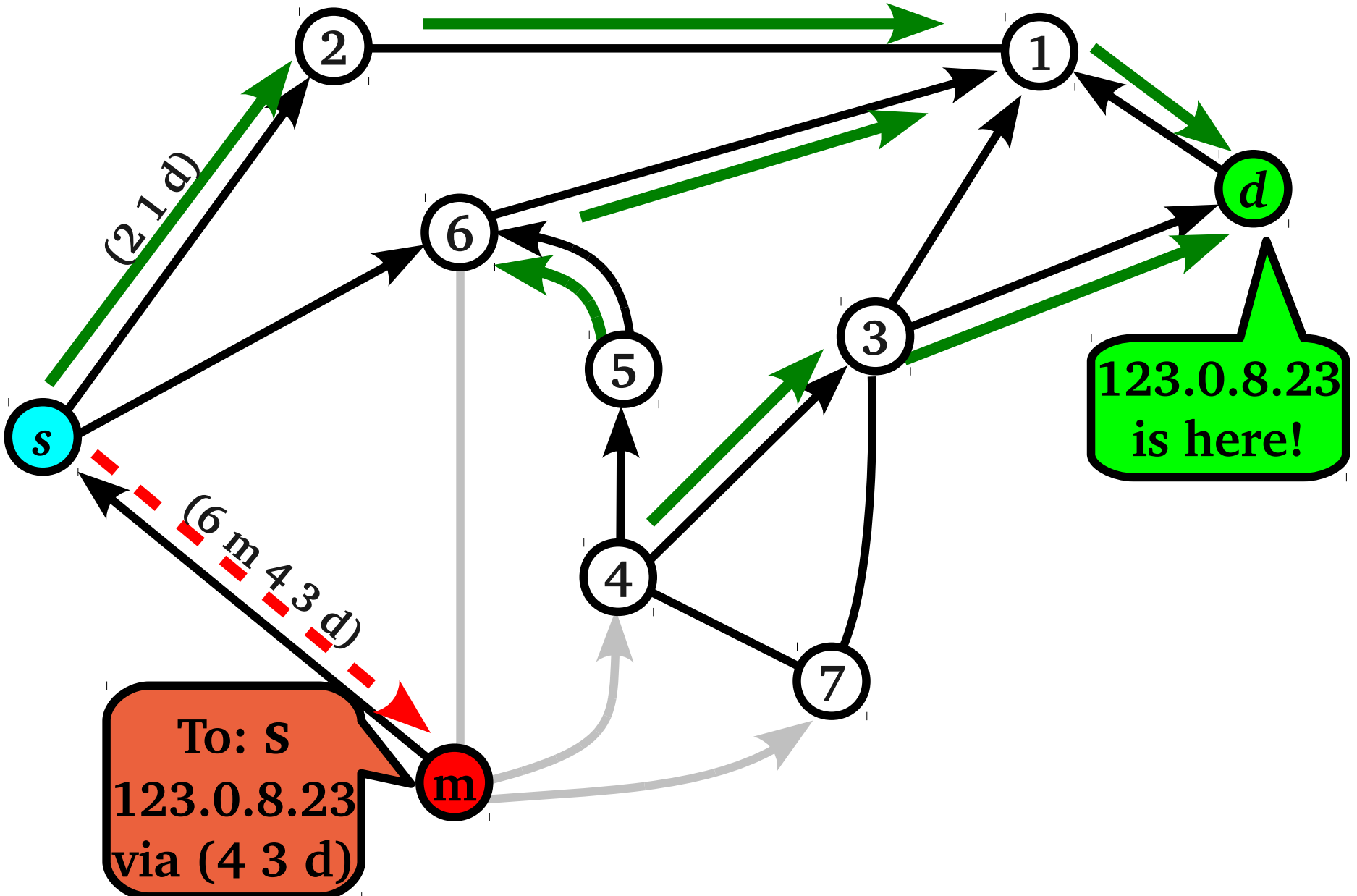
S-BGP cheating capabilities:

- a router cannot pretend to be the origin of an IP prefix that does not own.
- a router ***v*** can announce a path ***(v u)P*** only if ***u*** announced ***P*** to **v** in the past.

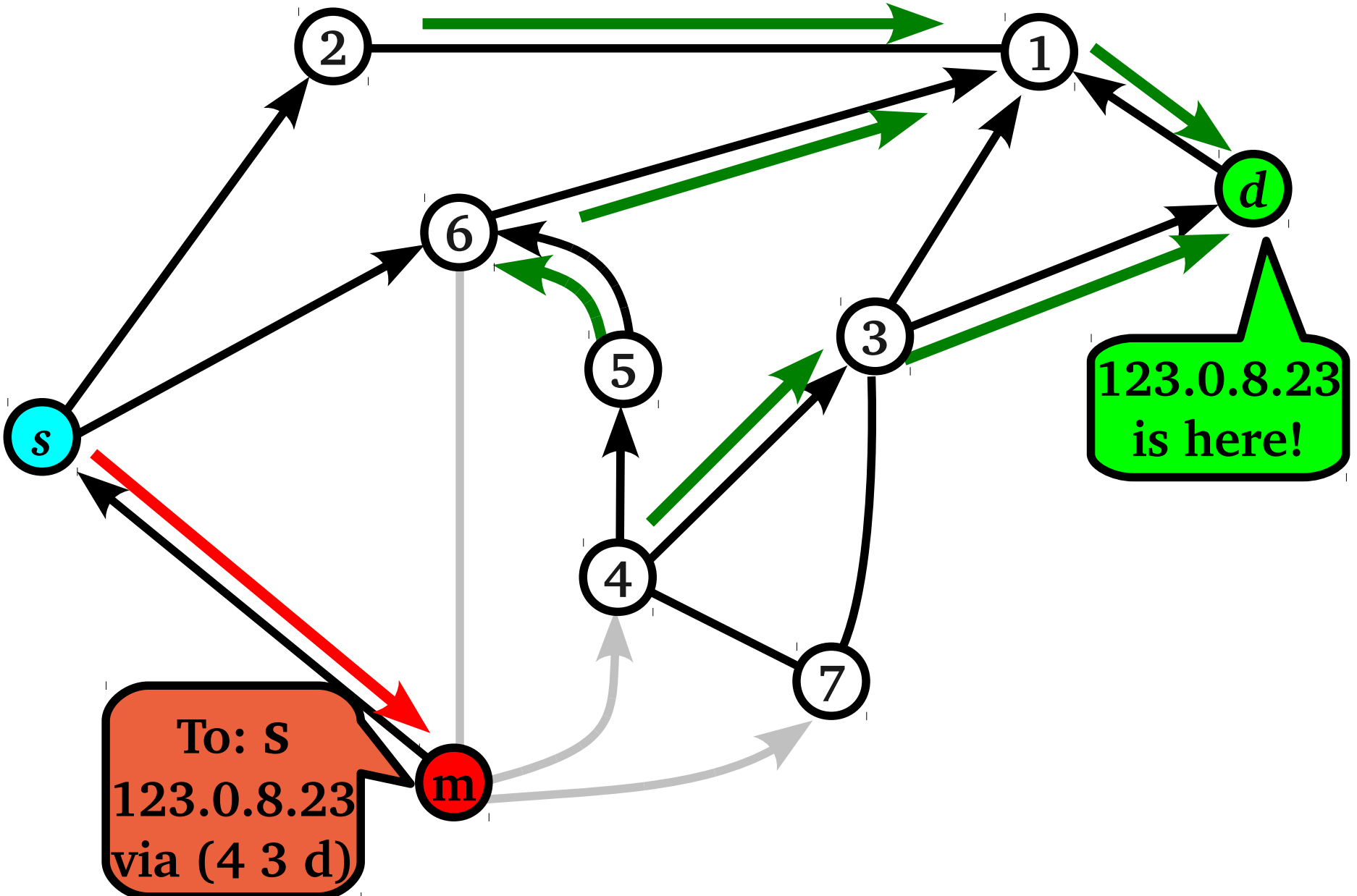
example of hijacking in S-BGP



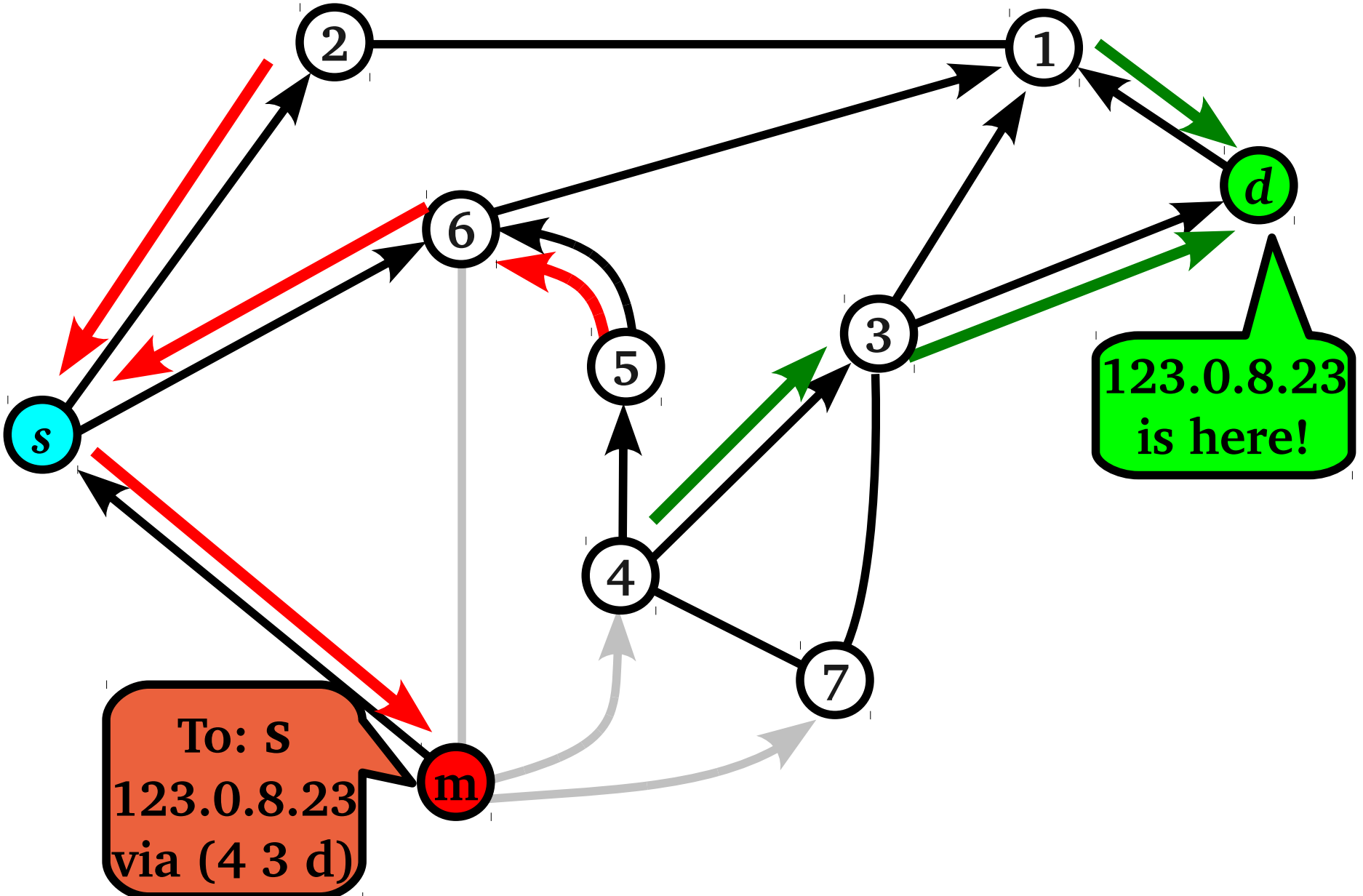
example of hijacking in S-BGP



example of hijacking in S-BGP



example of hijacking in S-BGP … and interception



related work

- Golberg Schapira Hummon Rexford, 2010
  - conjecture: is every hijack attack also an interception attack in S-BGP?
  - maximize traffic attraction is NP-hard
- Ballani Francis Zhang., 2007
  - methodology to intercept traffic + experimental study
- Levin Schapira Zohar, 2008
  - for several utility functions S-BGP is incentive compatible

our results (1)

- Golberg Schapira Hummon Rexford, 2010
  - is every hijack attack also an interception attack in S-BGP?

    Yes, it is! we'll discuss it later.

our results (2)

hijack problem. Given a BGP graph, a source, a destination, and a manipulator vertex, does there exist a strategy for **hijacking** the source vertex?

cheating capabilities	AS paths of any length	Bound AS-path length	Bound AS-path length and AS degree
origin-spoofing	**NP-hard**	**P**	P
S-BGP	NP-hard	**NP-Hard**	**P**

our results (2)

hijack problem. Given a BGP graph, a source, a destination, and a manipulator vertex, does there exist a strategy for **hijacking** the source vertex?

cheating capabilities	AS paths of any length	Bound AS-path length	Bound AS-path length and AS degree
origin-spoofing	NP-hard	**P**	P
S-BGP	NP-hard	**NP-Hard**	P

marks a sharp difference between BGP and S-BGP

hijack with origin-spoofing capabilities is in P

Theorem. If

- manipulator has origin-spoofing cheating capabilities and
- the length of the longest valley-free path is bounded by a constant,

then problem *hijack* is solvable in polynomial time.

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

input:

- BGP graph
- destination d, source s, manipulator m

output:

- a set A of neighbors of m

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

1: for each valley-free path p_{sm} from s to m

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

1: for each valley-free path p_{sm} from s to m

2: A = a set containing the neighbor of m in p_{sm}

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

1: for each valley-free path p_{sm} from s to m
2: A = a set containing the neighbor of m in p_{sm}
3: for each neighbor n of m
4: if there is no valley-free path through (m,n) that disrupts p_{sm} by better class
5: add n to A

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

1: for each valley-free path p_{sm} from s to m
2: A = a set containing the neighbor of m in p_{sm}
3: for each neighbor n of m
4: if there is no valley-free path through (m,n) that disrupts p_{sm} by better class
5: add n to A
6: if attack succeeds announcing to A
7: return A

hijack with origin-spoofing capabilities is in P (proof)

Hijack Algorithm.

1: for each valley-free path p_{sm} from s to m
2: A=a set containing the neighbor of m in p_{sm}
3: for each neighbor n of m
4: if there is no valley-free path through (m,n) that disrupts p_{sm} by better class
5: add n to A
6: if attack succeeds announcing to A
7: return A
8: return empty-set

hijack with origin-spoofing capabilities is in P (proof)

correctness:

6: if attack succeeds announcing to A

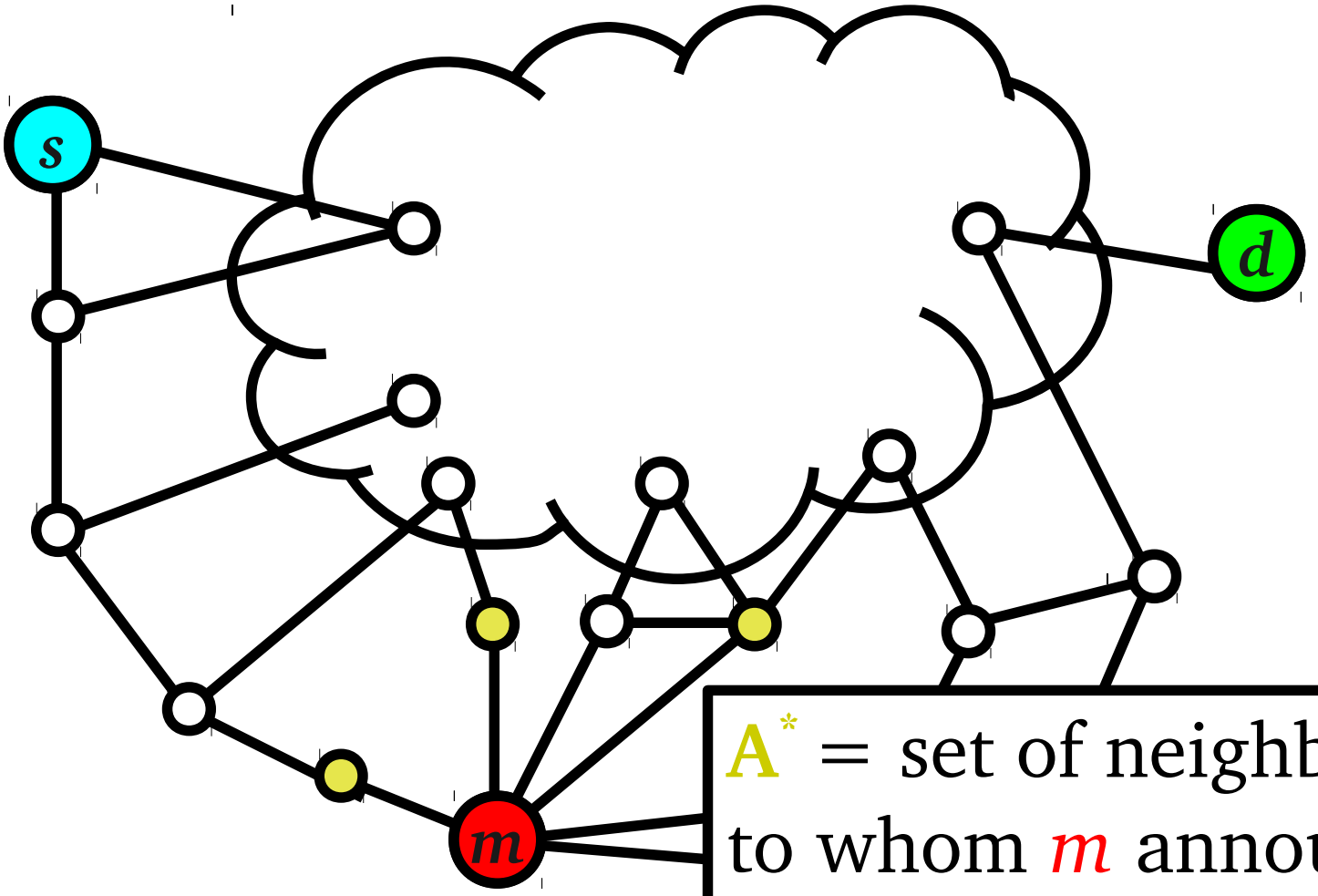
7: return A

completeness:

Suppose by contradiction there exists a successful attack strategy A^* from m but the Hijack Algorithm fails

algorithm completeness (proof)

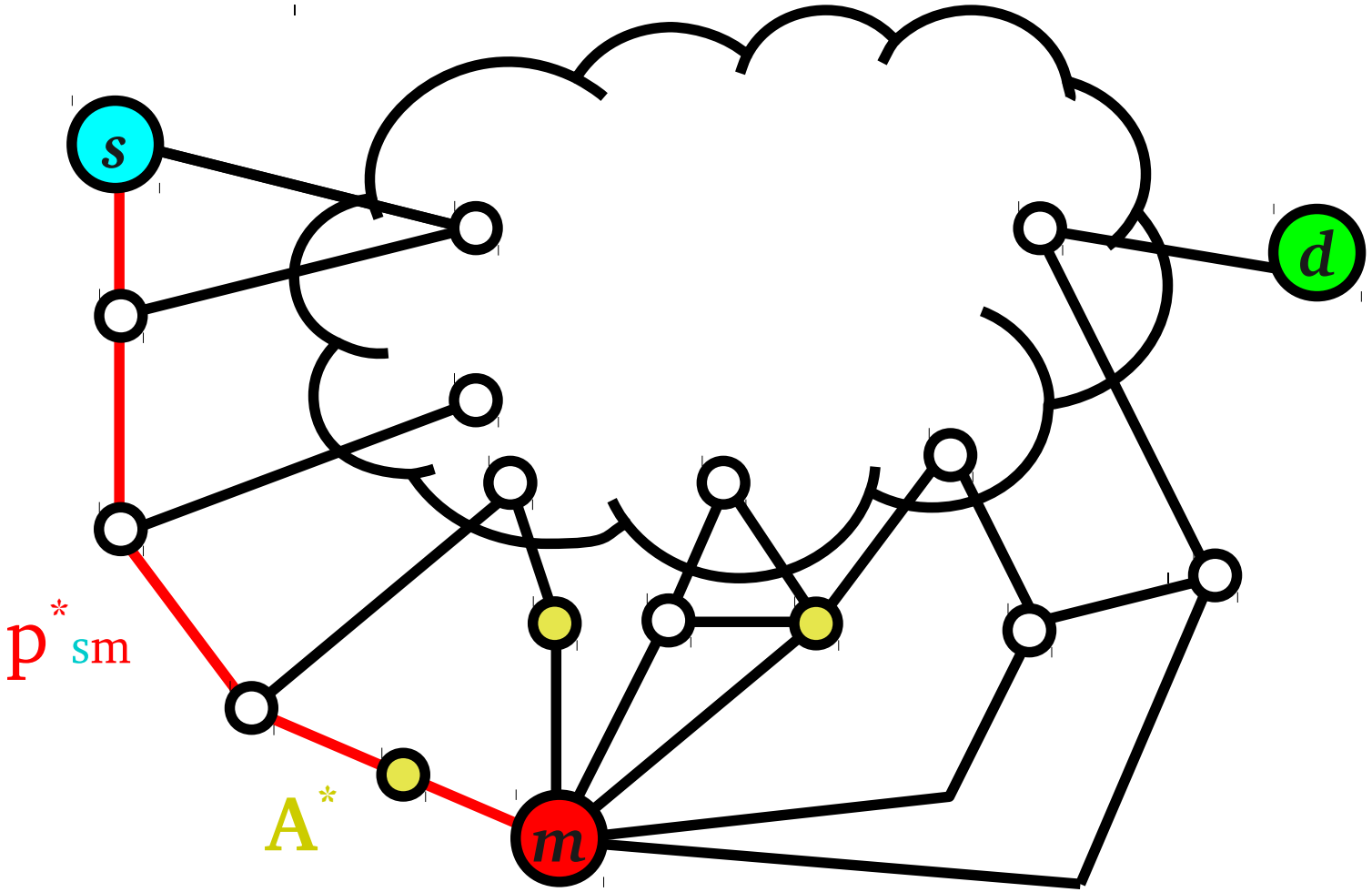
m attracts traffic announcing to A^*



A^* = set of neighbors to whom m announces the prefix

algorithm completeness (proof)

m attracts traffic through p^*_{sm}



Hijack Algorithm iteration on p^*_{sm}

Hijack Algorithm.

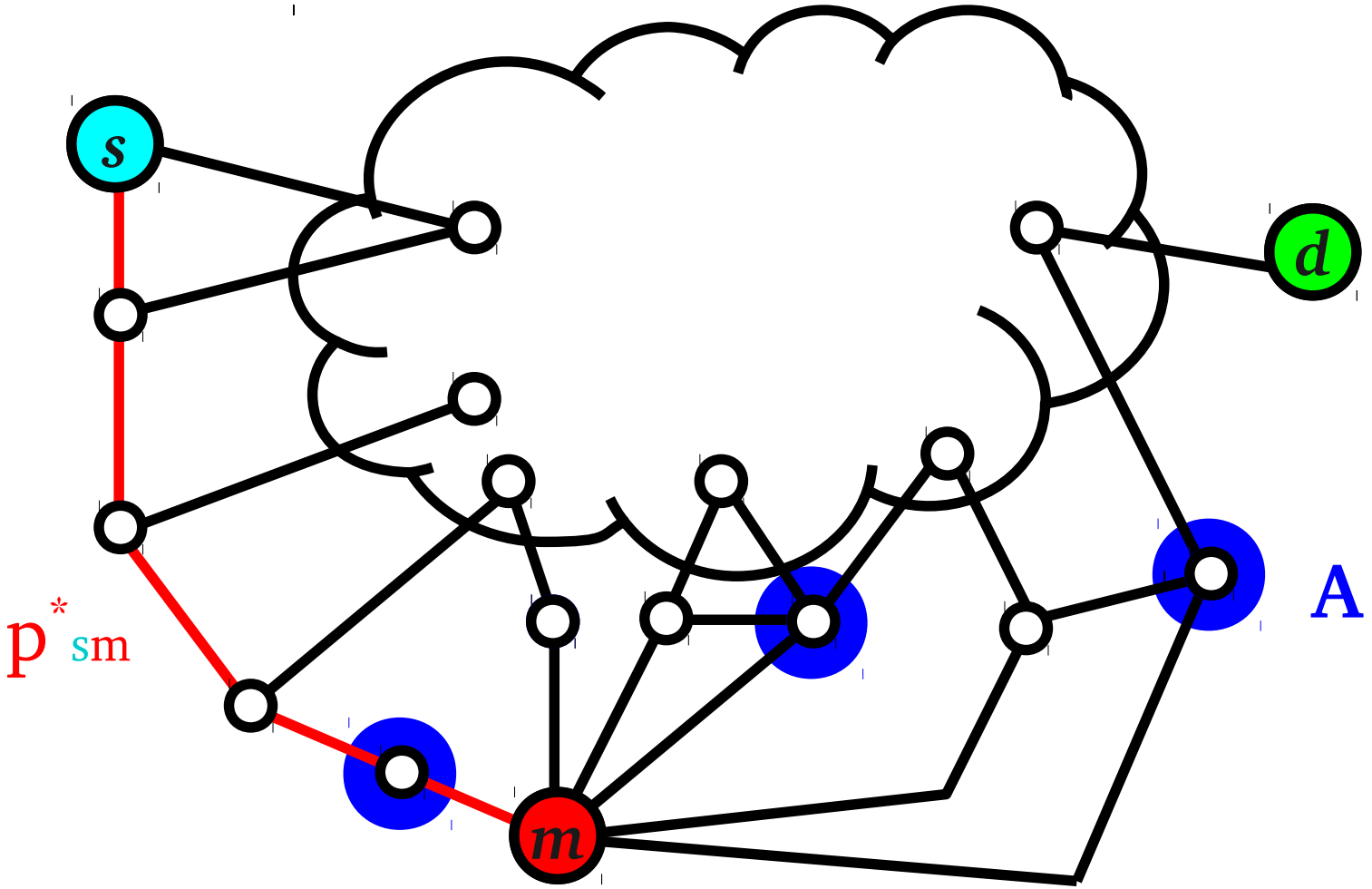
1: for each valley-free path p_{sm} from s to m

Hijack Algorithm iteration on p^*_{sm}: creation of A

Hijack Algorithm.

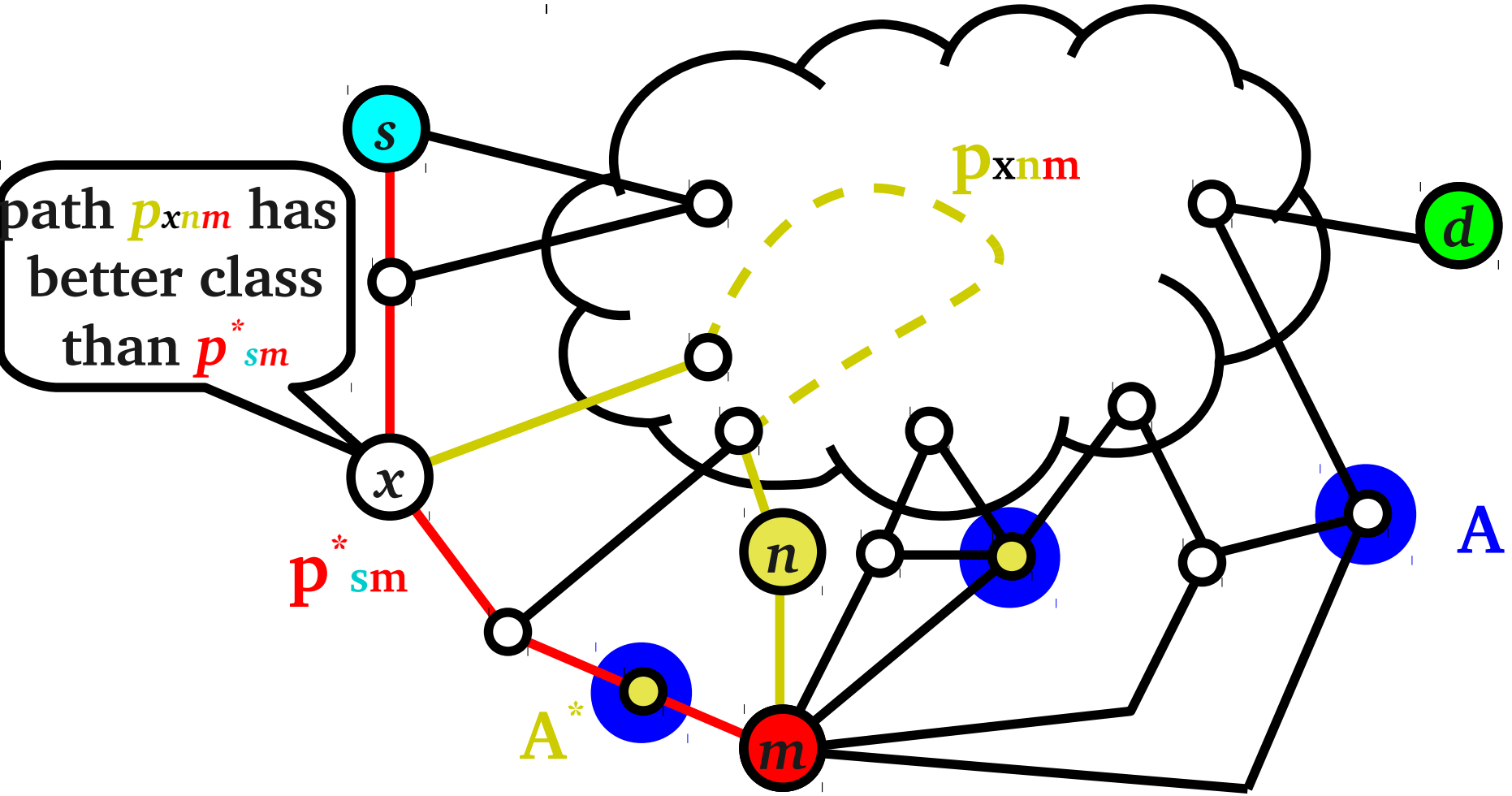
1: for each valley-free path p_{sm} from s to m
2: A = a set containing the neighbor of m in p_{sm}
3: for each neighbor n of m
4: if there is no valley-free path through (m,n) that disrupts p_{sm} by better class
5: add n to A
6: if attack succeeds announcing to A
7: return A

Hijack Algorithm iteration on p^*_{sm}

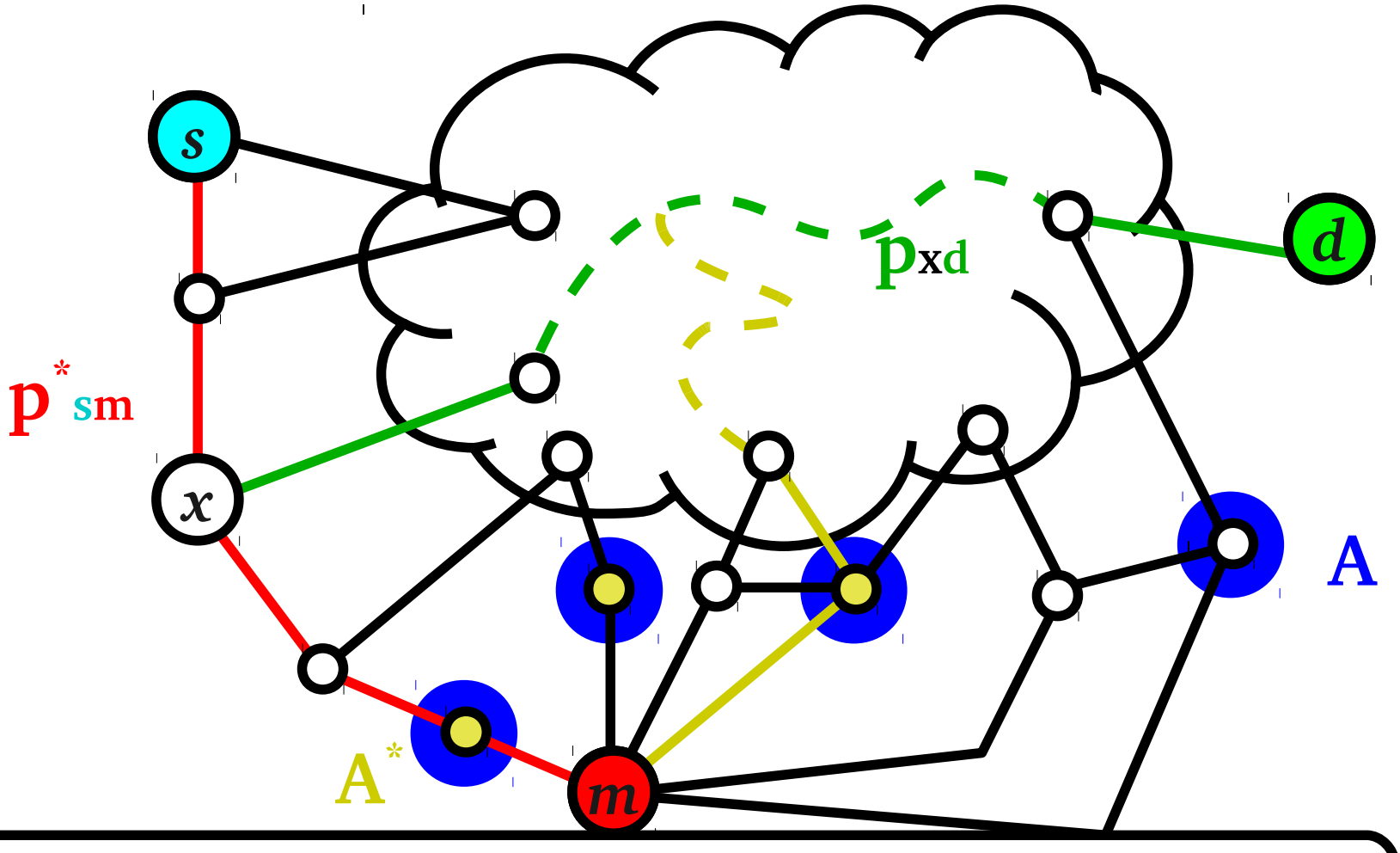


Hijack Algorithm iteration on p^*_{sm}:

A^* is contained in A (proof)



Hijack Algorithm iteration on p^*_{sm}: announcing to A → successful hijack

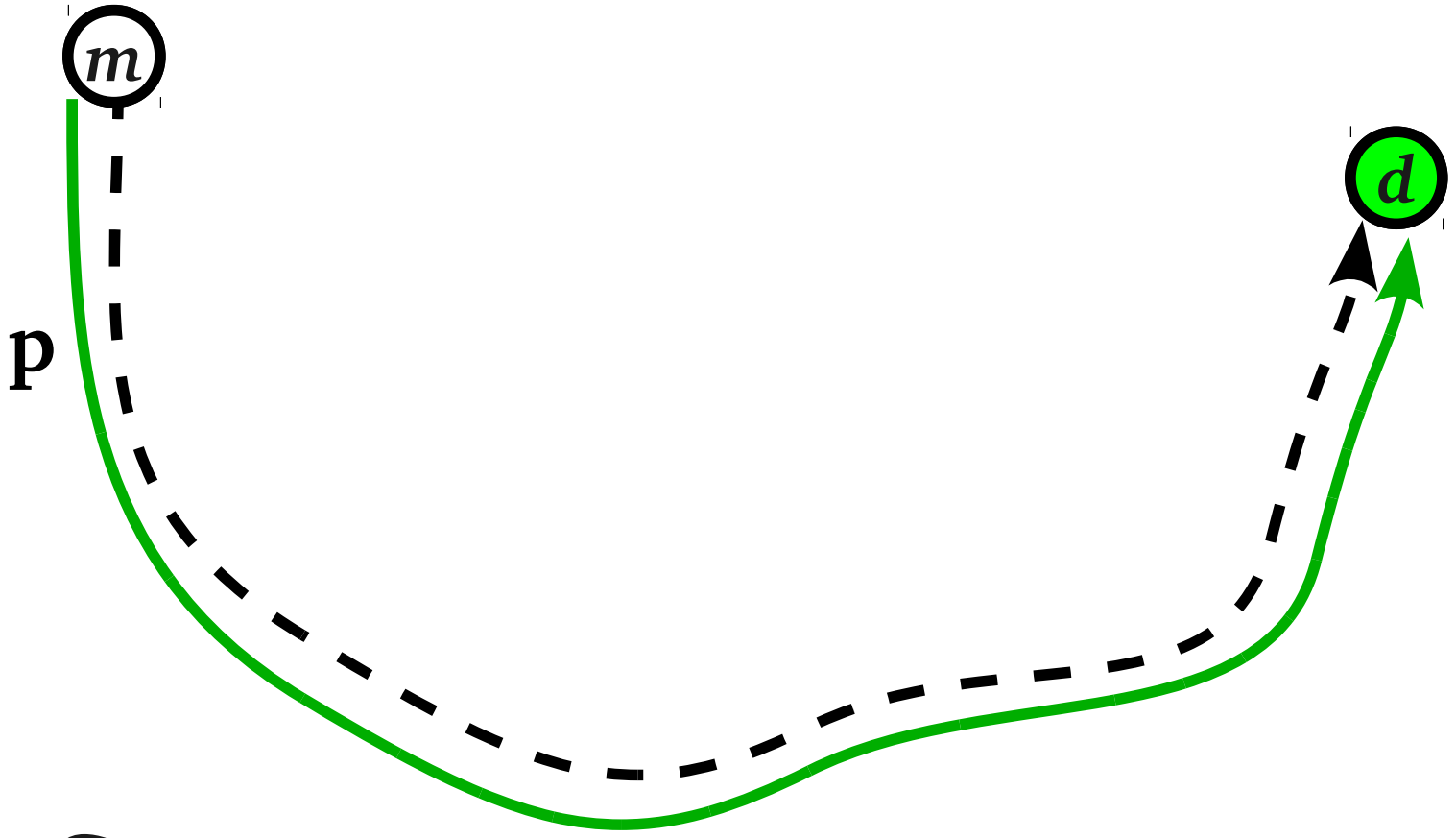


Lemma. p_{xd} is disrupted by better class

hijacking=interception in S-BGP

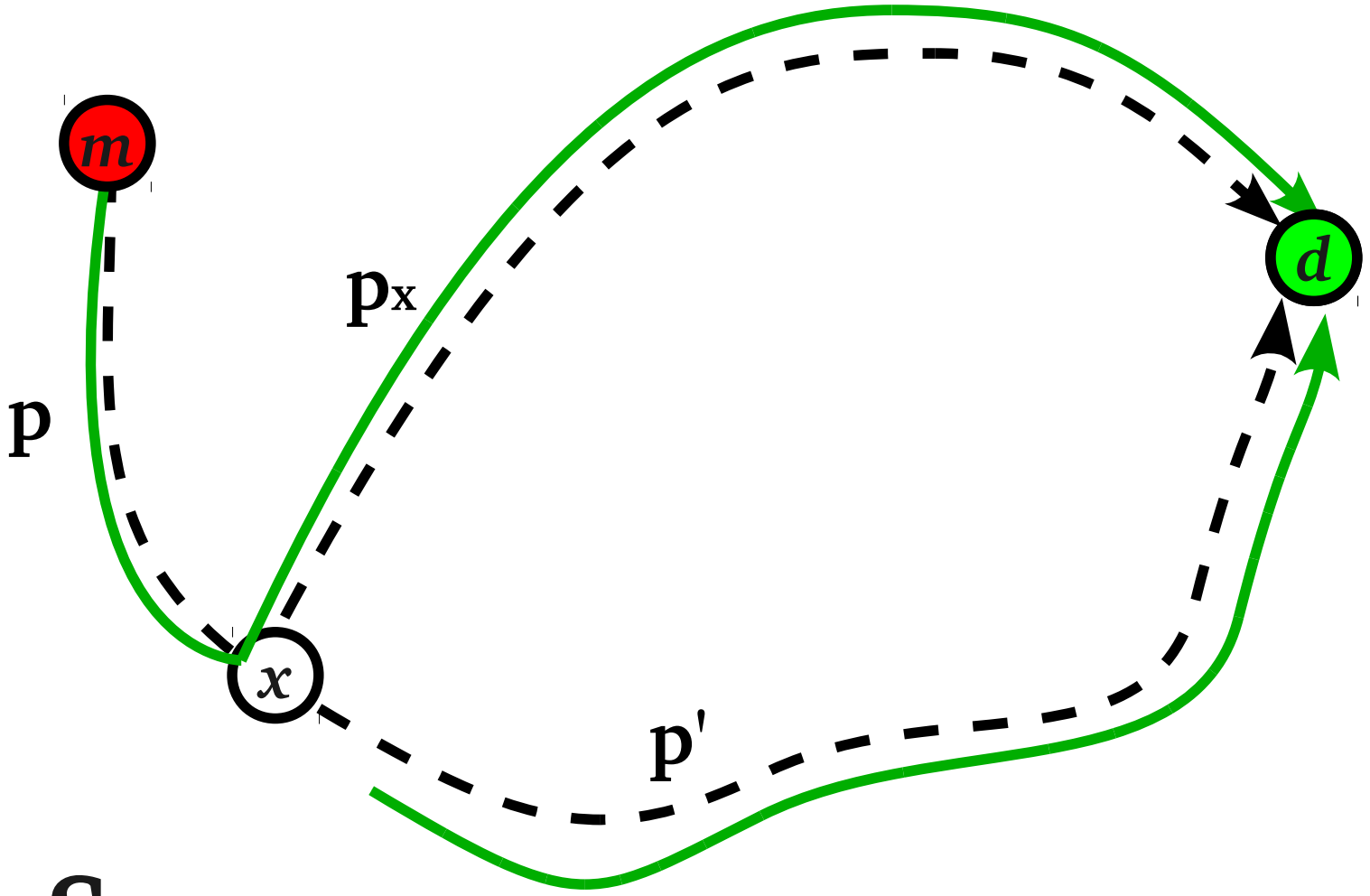
Theorem. If m announces the same path to any arbitrary set of its neighbors, then every successful hijacking attack is also a successful interception attack.

hijacking=interception in S-BGP



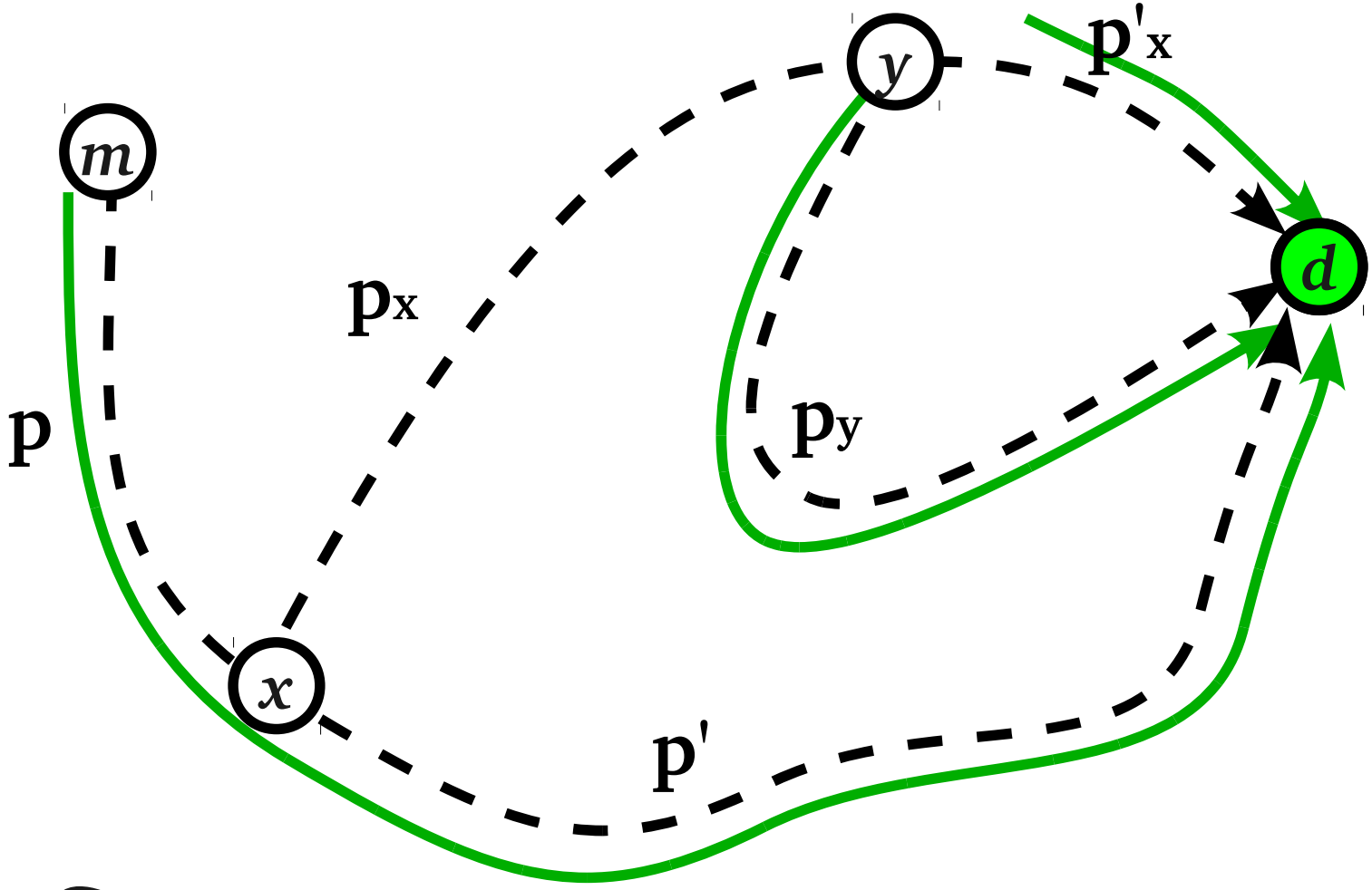
State S_1

hijacking=interception in S-BGP



State S_2

hijacking=interception in S-BGP



State S_1

hijacking != interception if manipulator announces different paths

Theorem. If m announces different paths to different vertices, then the hijacking may not be an interception.

routing oscillations

- we assumed that routing always **converges** to a **unique** and **stable** state

- BGP is known to be prone to routing oscillations

- if Gao-Rexford conditions holds, then routing always converges to a unique and stable state.
[GaoRexford, 2000]

- is it still true in the presence of a single manipulator?

no routing oscillations even in the presence of a manipulator

Theorem. Suppose that at a manipulator starts announcing steadily any set of arbitrary paths to its neighbors. Routing in G converges to a stable state.

this result has been also independently proved by Lychev, Goldberg, and Schapira (2012)

open problems

- We focused on a unique manipulator. How difficult is it to find a strategy involving **several malicious** ASes?

- In [SchapiraZhuRexford2010] it has been proposed to **disregard the AS-paths length** in the BGP decision process. How difficult is it to find an attack strategy in this different model?

thank you!