# Communication Networks

## Spring 2022

Thomas Holterbach

Coralie Busse-Grawitz

https://comm-net.ethz.ch/

ETH Zürich

April 7 2022

# Communication Networks

Spring 2022

How (not) to fix count-to-infinity

Overview current exercise

Understanding traceroute

Solving the exercise/Q&A

# Communication Networks

Spring 2022

How (not) to fix count-to-infinity

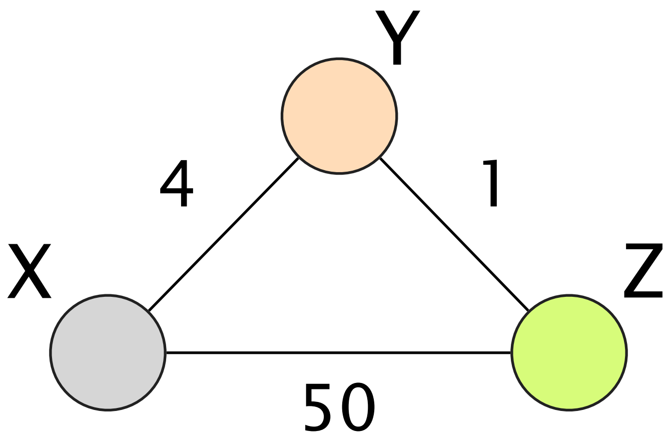
Overview current exercise

Understanding traceroute

Solving the exercise/Q&A

from the lecture

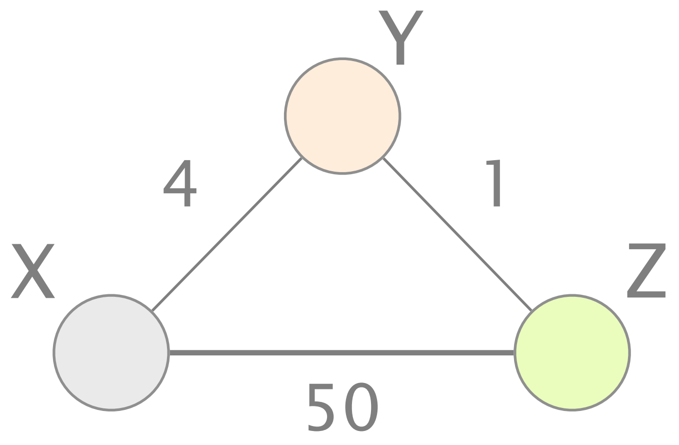
Consider the following network



from the lecture

Consider the following network

leading to the following vectors



Y vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 4 | 6 |

Y reaches X directly

Z vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 5 |

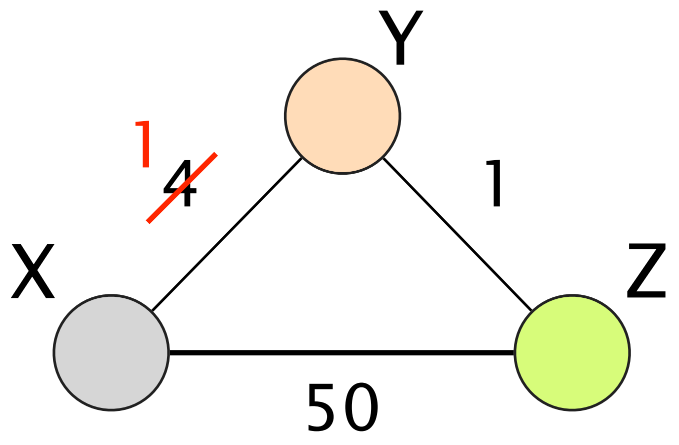
Z reaches X via Y

from the lecture

t = 0

(X,Y) weight changes

from 4 to 1



time t=0

Y
vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 4 | 6 |

Z
vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 5 |

from
the lecture

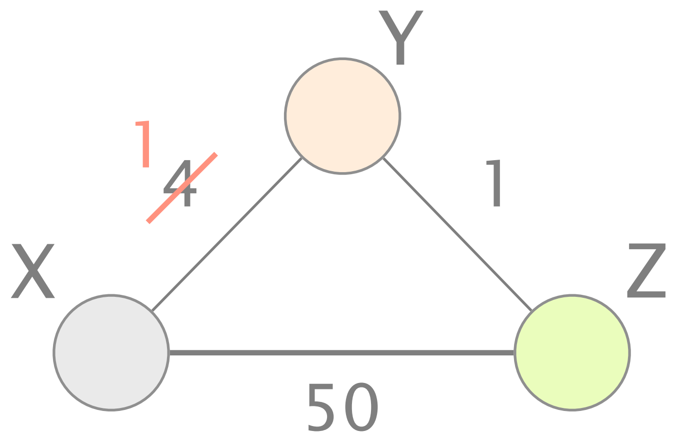
Node detects local cost change, update their vectors, and notify their neighbors if it has changed

from
the lecture

t > 3

no one moves anymore

network has converged!



**Y vector**

t=0

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 4 | 6 |

t=1

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 1 | 6 |

t>3

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 1 | 3 |

**Z vector**

t=0

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 5 |

t=2

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 2 |

t>3

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 2 |

from the lecture

The algorithm terminates after 3 iterations

Good news travel fast!

from
the lecture

Good news travel fast!

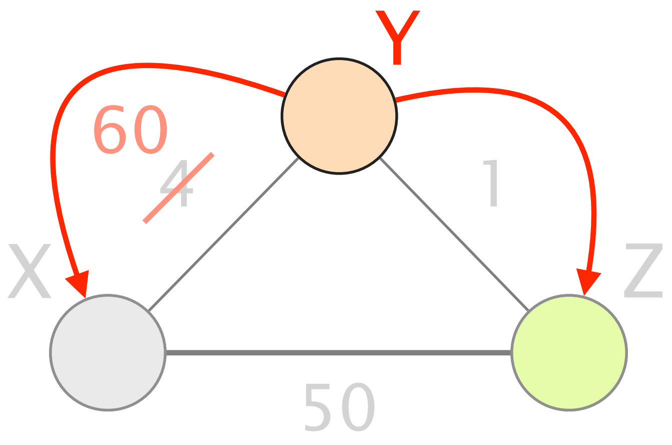
What about bad ones?

from the lecture

t = 3

Y updates its vector,

sends it to X and Z



Y vector

t=0

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 4 | 6 |

t=1

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 60 | 6 |

t=2

t=3

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 60 | 8 |

Z vector

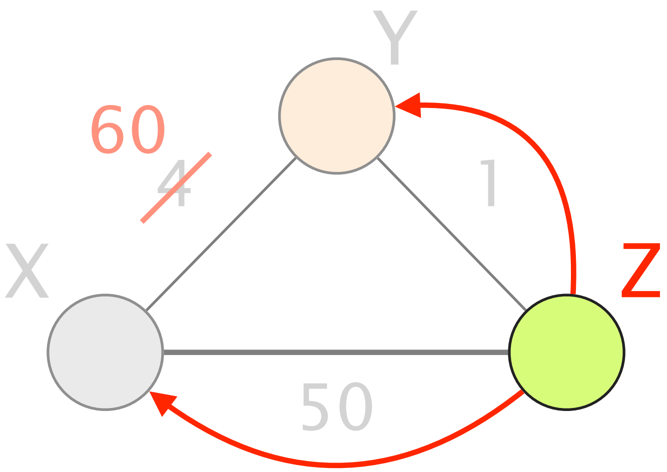
t=0

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 5 |

t=2

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 7 |

from the lecture



t=4

t=44

... many iterations later ...

Y
vector

| dest. | via *X* | via *Z* |
|---|---|---|
| *X* | 60 | 51 |

Z
vector

t=4:

| dest. | via *X* | via *Y* |
|---|---|---|
| *X* | 50 | 9 |

t=44:

| dest. | via *X* | via *Y* |
|---|---|---|
| *X* | 50 | 52 |

from the lecture

# This problem is known as count-to-infinity, a type of routing loop

Count-to-infinity leads to very slow convergence

what if the cost had changed from 4 to 9999?

Routers don't know when neighbors use them

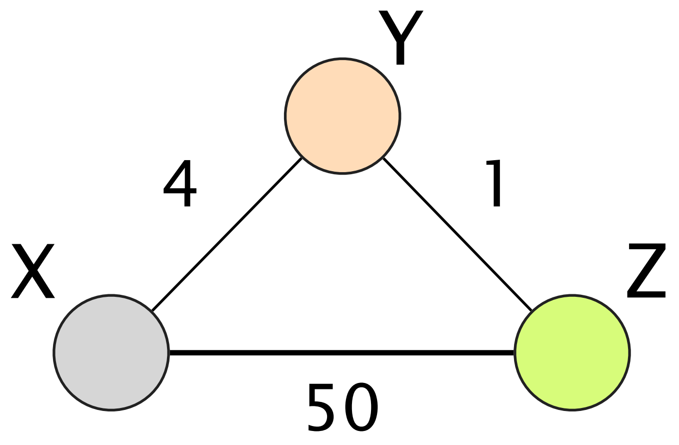
Z does not know that Y has switched to use itself

Let's try to fix that **(left to the exercise session)**

Whenever a router uses another one,

it will announce it an infinite cost

The technique is known as poisoned reverse



Y
vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 4 | ∞ |

As Z uses Y to reach X,

it announces to Y an infinite cost

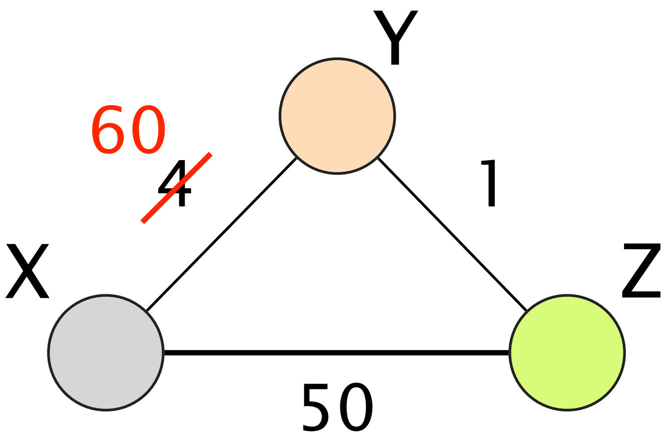
Z
vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 5 |

t = 0

(X,Y) weight changes

from 4 to 60



time t=0

Y vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 4 | ∞ |

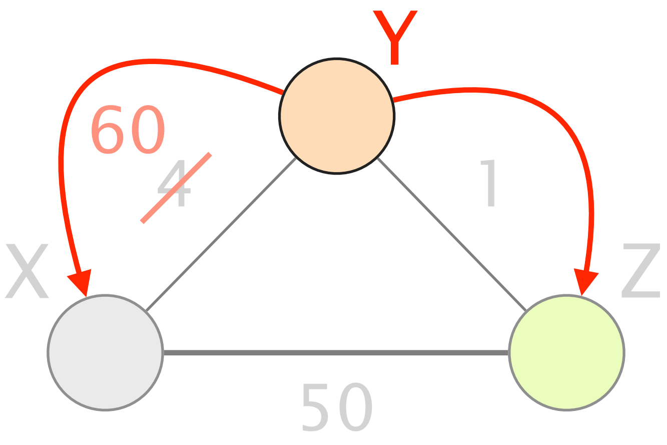
Z vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 5 |

t = 1

Y updates its vector,

sends it to X and Z



t=0

Y vector

| dest. | via X | via Z |
|---|---|---|
| X | 4 | ∞ |

t=1

| dest. | via X | via Z |
|---|---|---|
| X | 60 | ∞ |

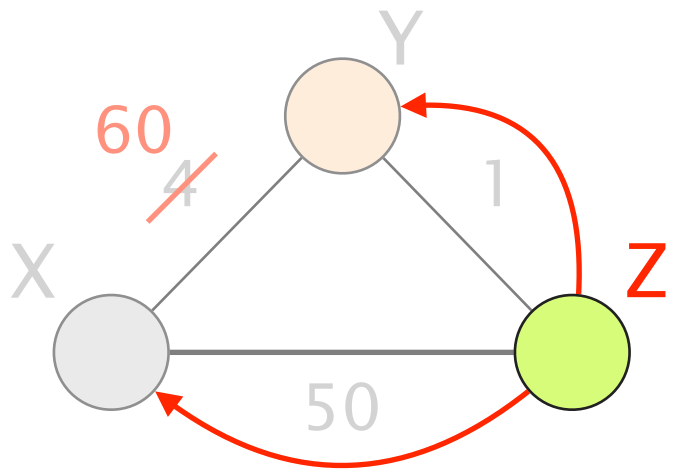
Z vector

| dest. | via X | via Y |
|---|---|---|
| X | 50 | 5 |

t = 2

Z updates its vector,

sends it to X and Y



t=0

Y vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 4 | ∞ |

Z vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 5 |

t=1

Y vector

| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *X* | 60 | ∞ |

t=2

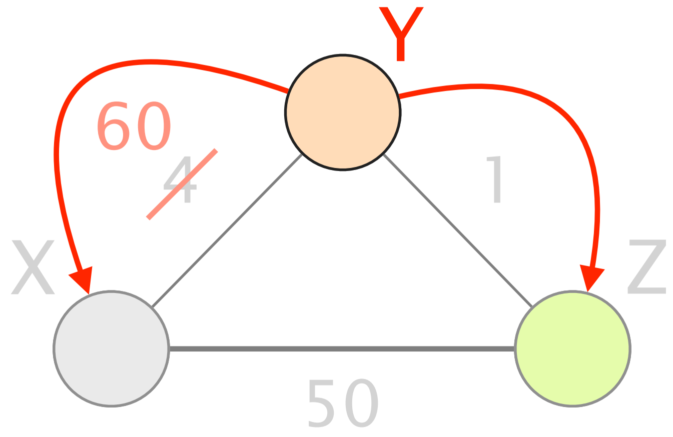
Z vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | 61 |

t = 3

Y updates its vector,

sends it to X and Z



Y
vector

t=0

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 4 | ∞ |

t=1

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 60 | ∞ |

t=3

| dest. | via | |
|---|---|---|
| | X | Z |
| X | 60 | 51 |

Z
vector

t=0

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 5 |

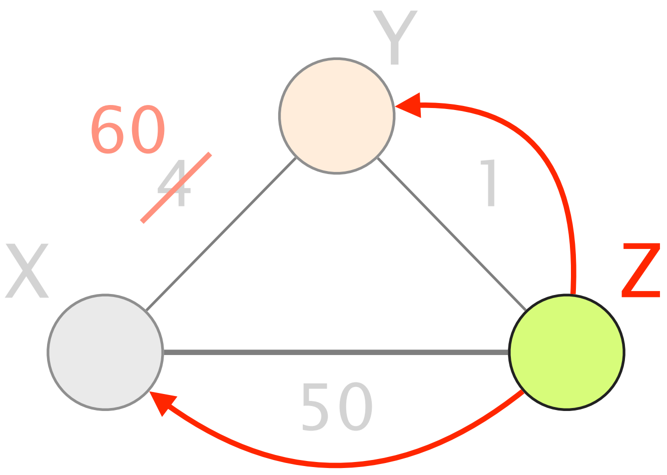
t=2

| dest. | via | |
|---|---|---|
| | X | Y |
| X | 50 | 61 |

t = 4

Z updates its vector,

sends it to X and Y



t=4

Y
vector

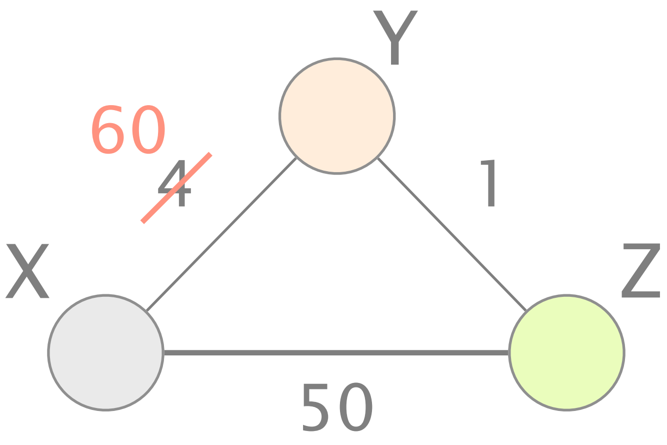
Z
vector

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *X* | 50 | ∞ |

t > 4

no one moves

network has converged!



t=4

t>4

Y
vector

t>4

| dest. | via X | via Z |
|---|---|---|
| X | 60 | 51 |

Z
vector

t=4

| dest. | via X | via Y |
|---|---|---|
| X | 50 | ∞ |

t>4

| dest. | via X | via Y |
|---|---|---|
| X | 50 | ∞ |

While poisoned reverse solved this case,

it does not solve loops involving 3 or more nodes…

see exercise task 1

Actual distance-vector protocols mitigate this issue by using small “infinity”, *e.g.* 16

# Communication Networks

Spring 2022

How (not) to fix count-to-infinity

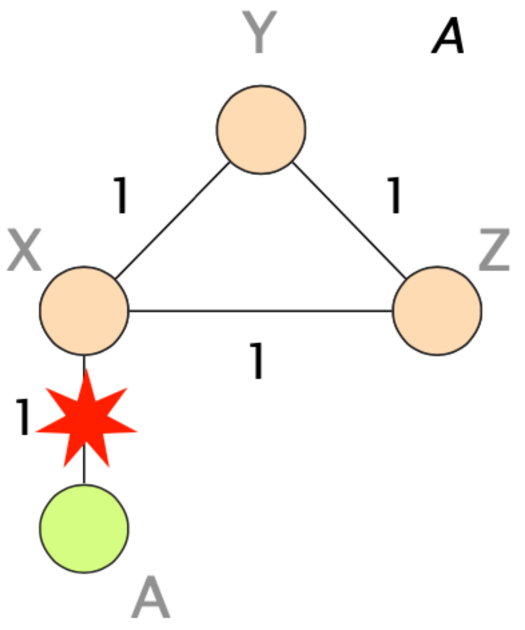
Overview current exercise

Understanding traceroute

Solving the exercise/Q&A

# Task 1: Convergence with Poisoned Reverse

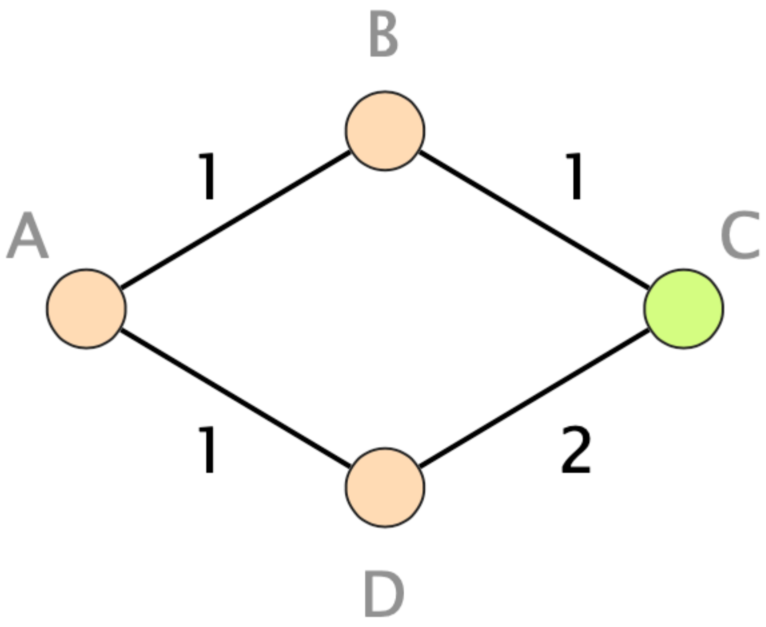
| dest. | via | | |
|---|---|---|---|
| | *A* | *Y* | *Z* |
| *A* | 1 | ∞ | ∞ |



| dest. | via | |
|---|---|---|
| | *X* | *Z* |
| *A* | 2 | 3 |

| dest. | via | |
|---|---|---|
| | *X* | *Y* |
| *A* | 2 | 3 |

# Task 2: Convergence (Exam-Style Question)



Loopy or not?

# Communication Networks

Spring 2022

How (not) to fix count-to-infinity

Overview current exercise

Understanding traceroute

Solving the exercise/Q&A