PageRank

Random Surfers on the Web
Transition Matrix of the Web
Dead Ends and Spider Traps
Topic-Specific PageRank

Jeffrey D. Ullman
Stanford University

Administrivia – Honor Code

- We’ve had our first HC cases.
- Please, please, please, before you do anything that might violate the HC, talk to me or a TA to make sure it is legitimate.
- It is much easier to get caught than you might think.

Administrivia – Homeworks

- There were a number of people who failed to upload code or HW answers properly and received no credit.
- Also, some people followed general SCPD directions, which you must not do in the future.
- We made some exceptions, e.g., allowing late code uploads.
- But in the future, please do not expect these sorts of exceptions to be made.

Intuition – (1)

- Web pages are important if people visit them a lot.
- But we can't watch everybody using the Web.
- A good surrogate for visiting pages is to assume people follow links randomly.
- Leads to *random surfer* model:
  - Start at a random page and follow random out-links repeatedly, from whatever page you are at.
  - *PageRank* = limiting probability of being at a page.

Intuition – (2)

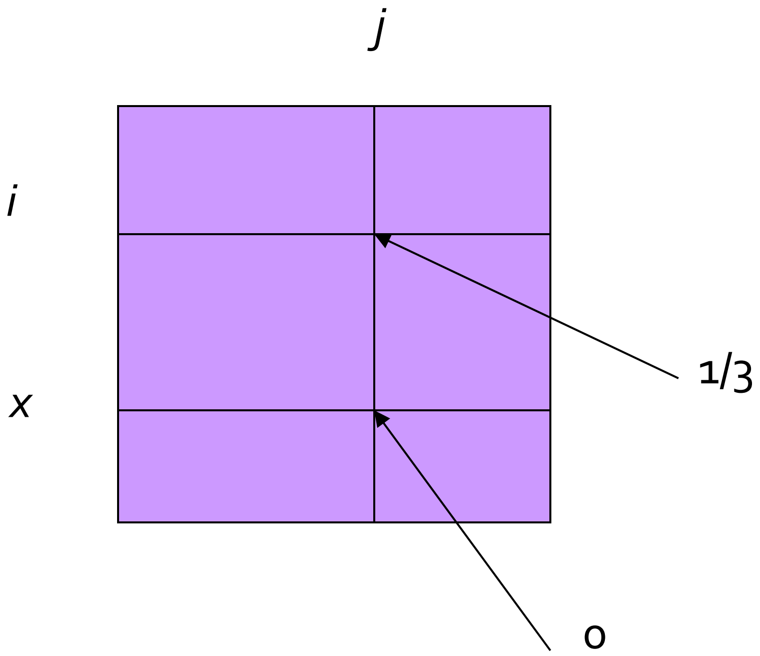
- Solve the recursive equation: “a page is important to the extent that important pages link to it.”
  - Equivalent to the random-surfer definition of PageRank.
- Technically, *importance* = the principal eigenvector of the transition matrix of the Web.
  - A few fixups needed.

Transition Matrix of the Web

- Number the pages 1, 2,… .
  - Page *i* corresponds to row and column *i*.
- *M* [*i*, *j*] = 1/*n* if page *j* links to *n* pages, including page *i* ; 0 if *j* does not link to *i*.
  - *M* [*i, j*] is the probability we'll next be at page *i* if we are now at page *j*.

Example: Transition Matrix

Suppose page *j* links to 3 pages, including *i* but not *x*.



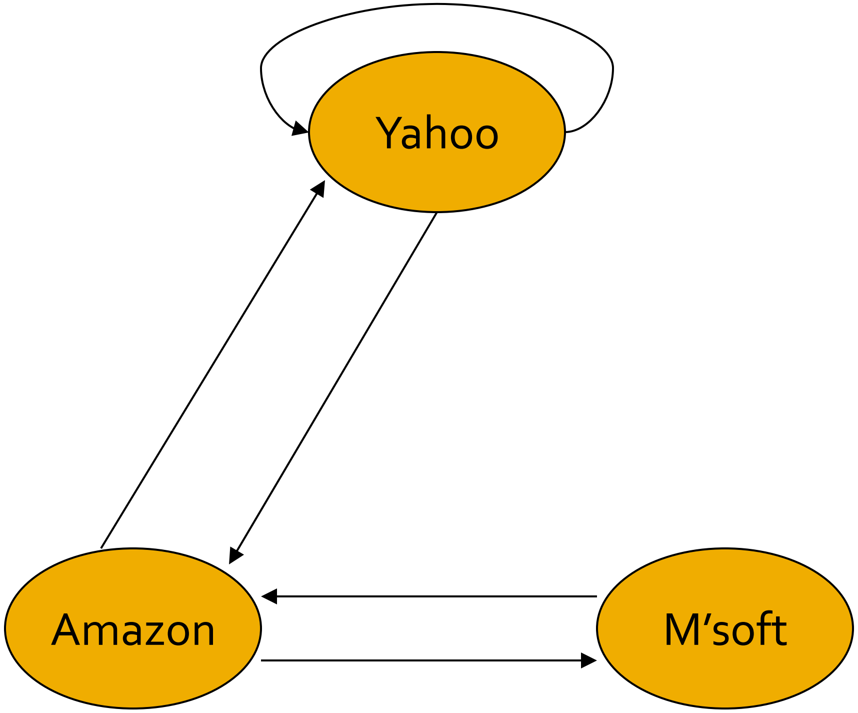
Random Walks on the Web

- Suppose **v** is a vector whose *i* th component is the probability that a random walker is at page *i* at a certain time.
- If a walker follows a link from *i* at random, the probability distribution for walkers is then given by the vector *M***v**.

Random Walks – (2)

- Starting from any vector **u**, the limit $M\ (M\ (\ldots M\ (M\ \mathbf{u}\ )\ \ldots))$ is the long-term distribution of walkers.
- Intuition: pages are important in proportion to how likely a walker is to be there.
- The math: limiting distribution = principal eigenvector of M = PageRank.
  - Note: because M has each column summing to 1, the principal eigenvalue is 1.
    - Why? If **v** is the limit of MM…M**u**, then **v** satisfies the equations **v** = M**v**.

Example: The Web in 1839



$$\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 1 \\
m & 0 & 1/2 & 0 \\
\hline
\end{array}$$

Solving The Equations

- Because there are no constant terms, the equations **v** = *M***v** do not have a unique solution.
- In Web-sized examples, we cannot solve by Gaussian elimination anyway; we need to use *relaxation* (= iterative solution).
- Works if you start with any nonzero **u**.

Simulating a Random Walk

- Start with the vector **u** = [1, 1,…, 1] representing the idea that each Web page is given one unit of *importance*.
  - Note: it is more common to start with each vector element = 1/n, where n is the number of Web pages.
- Repeatedly apply the matrix *M* to **u**, allowing the importance to flow like a random walk.
- About 50 iterations is sufficient to estimate the limiting solution.

Example: Iterating Equations

- Equations $\mathbf{v} = M\mathbf{v}$:

$$y = y/2 + a/2$$

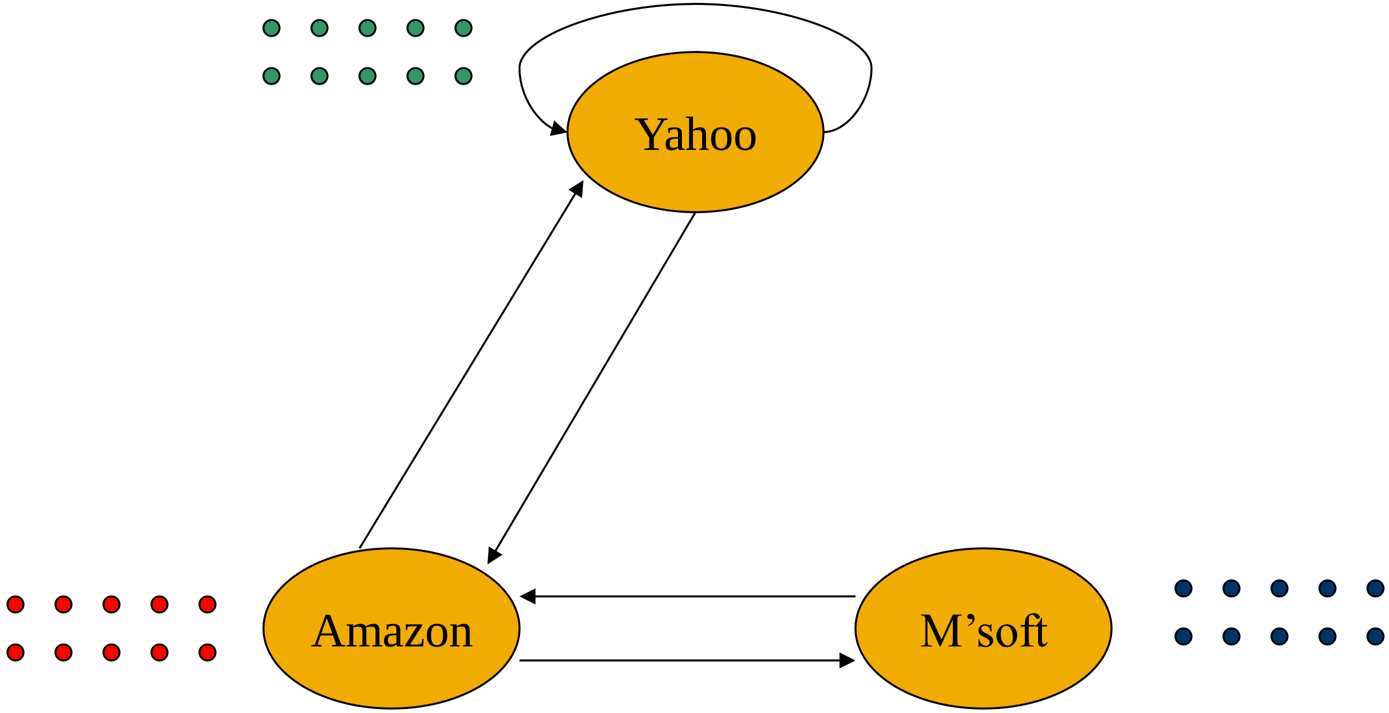
$$a = y/2 + m$$

$$m = a/2$$

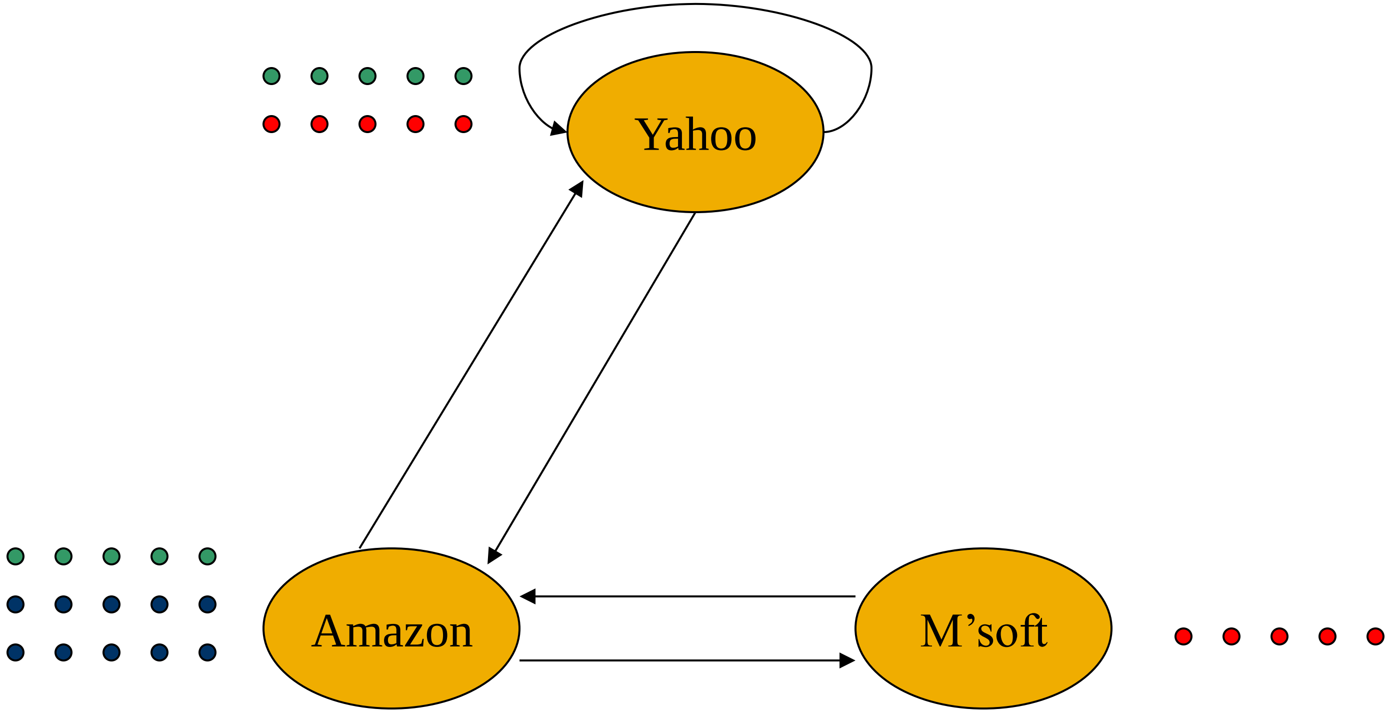
Note: "=" is really "assignment."

y		1	1	5/4	9/8		6/5
a	=	1	3/2	1	11/8	. . .	6/5
m		1	1/2	3/4	1/2		3/5

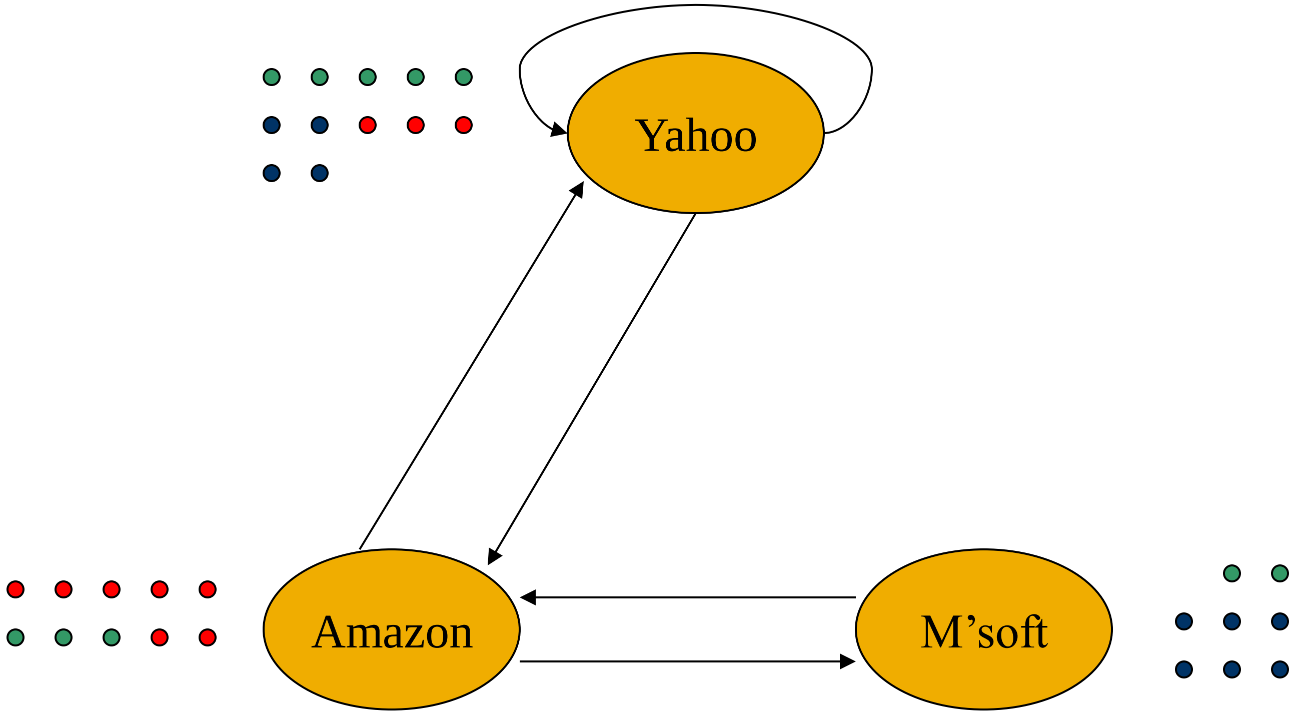
The Walkers



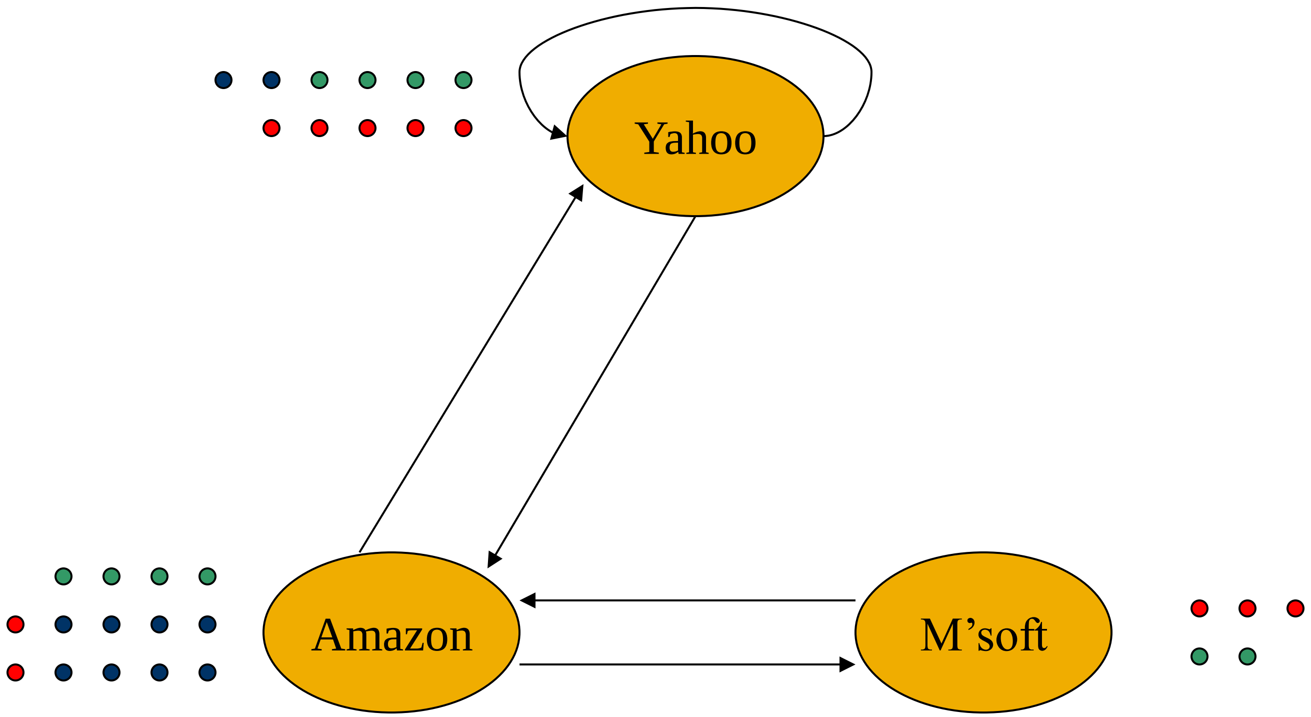
The Walkers



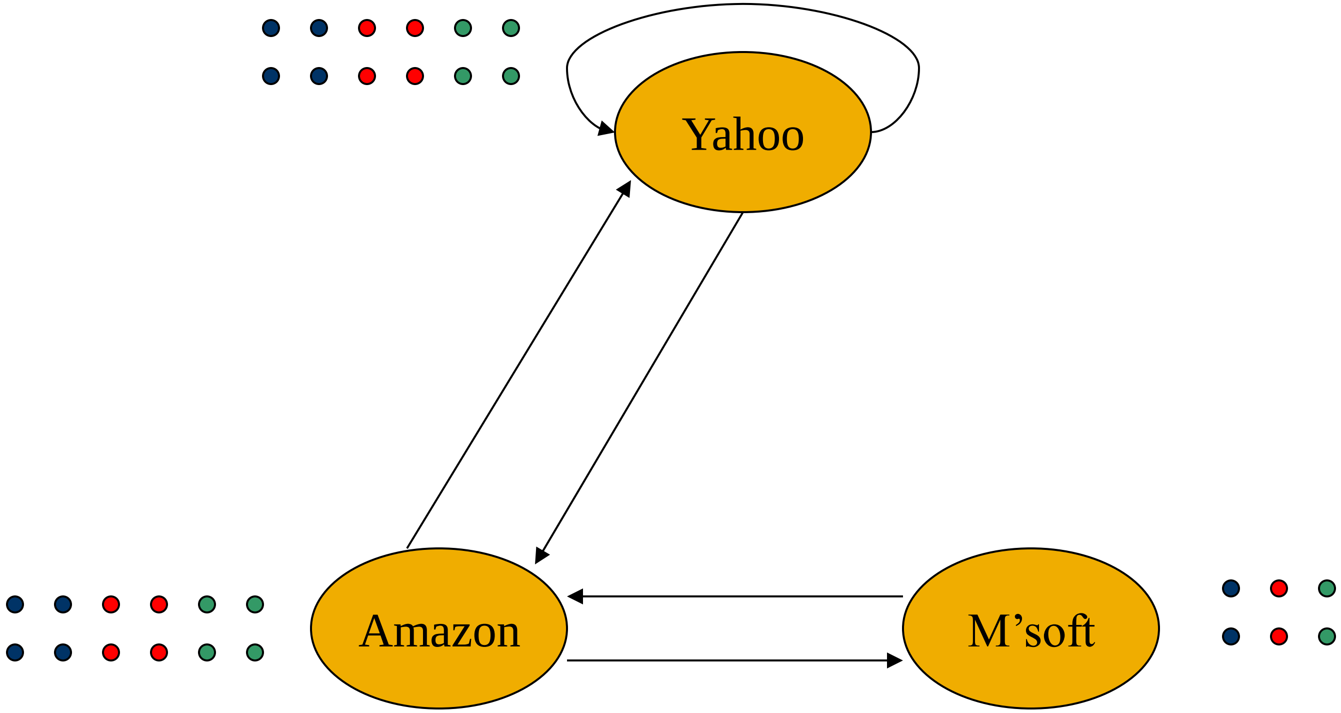
The Walkers



The Walkers



In the Limit …



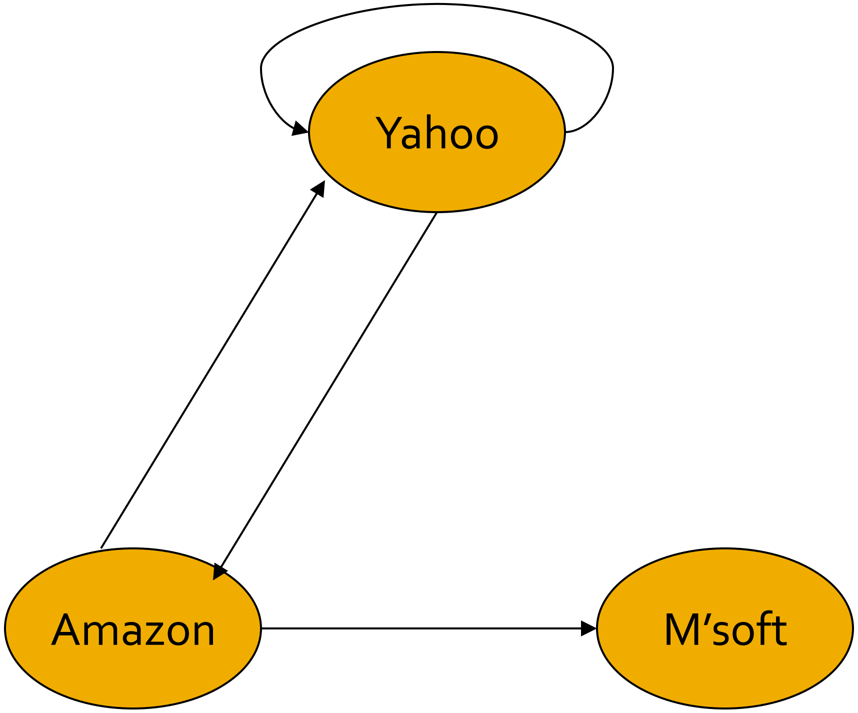
The Web Is More Complex Than That

Dead Ends
Spider Traps
Taxation Policies

Real-World Problems

- Some pages are *dead ends* (have no links out).
  - Such a page causes importance to leak out.
- Other groups of pages are *spider traps* (all out-links are within the group).
  - Eventually spider traps absorb all importance.

Microsoft Becomes Dead End



$$
\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 0 \\
m & 0 & 1/2 & 0 \\
\hline
\end{array}
$$

Example: Effect of Dead Ends

- Equations **v** = *M***v**:

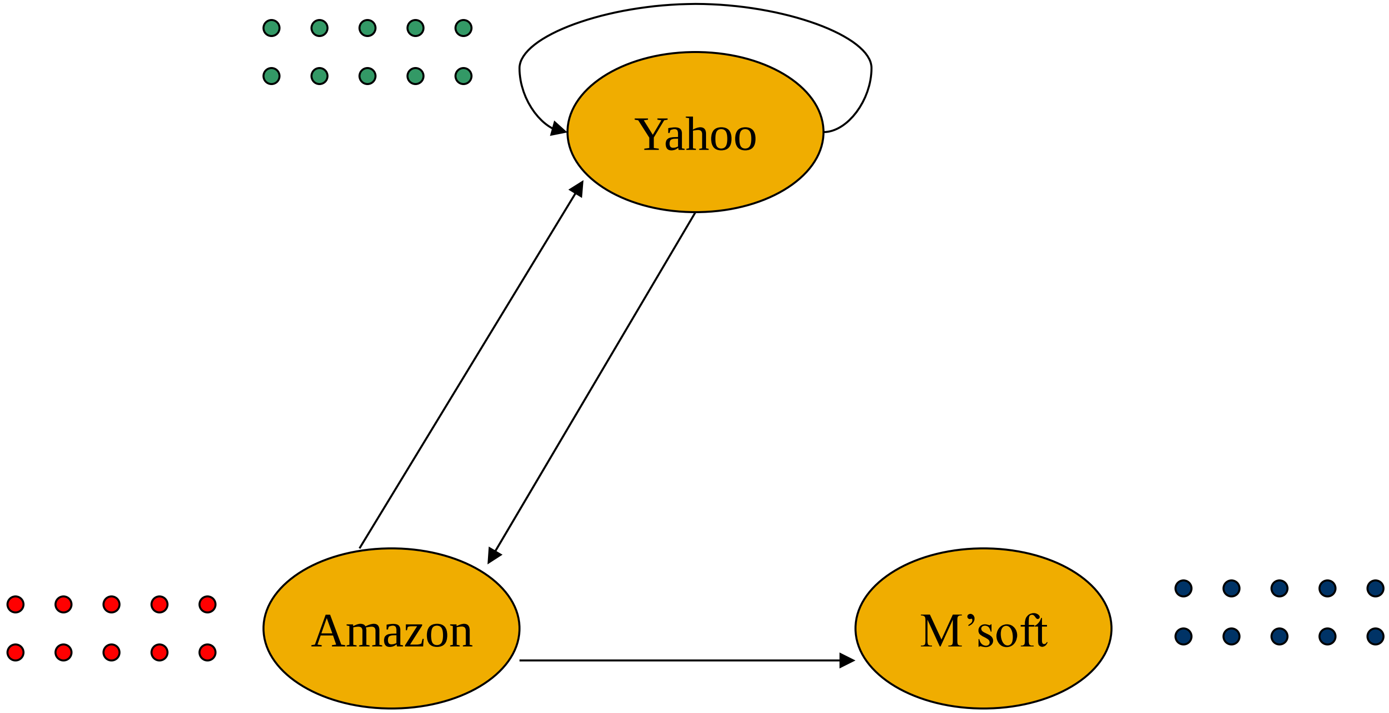
$$y = y/2 + a/2$$

$$a = y/2$$

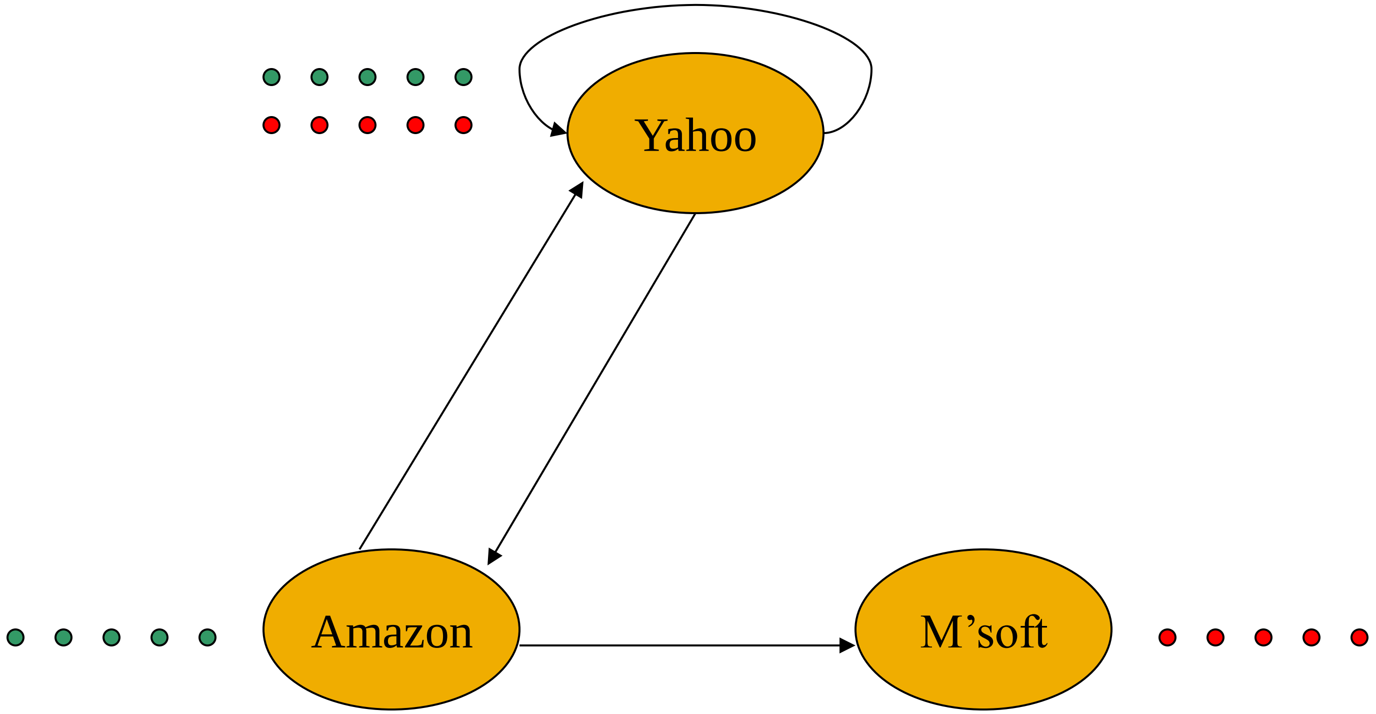
$$m = a/2$$

$$\begin{array}{c} y \\ a \\ m \end{array} = \quad \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \quad \begin{array}{c} 1 \\ 1/2 \\ 1/2 \end{array} \quad \begin{array}{c} 3/4 \\ 1/2 \\ 1/4 \end{array} \quad \begin{array}{c} 5/8 \\ 3/8 \\ 1/4 \end{array} \quad \ldots \quad \begin{array}{c} 0 \\ 0 \\ 0 \end{array}$$

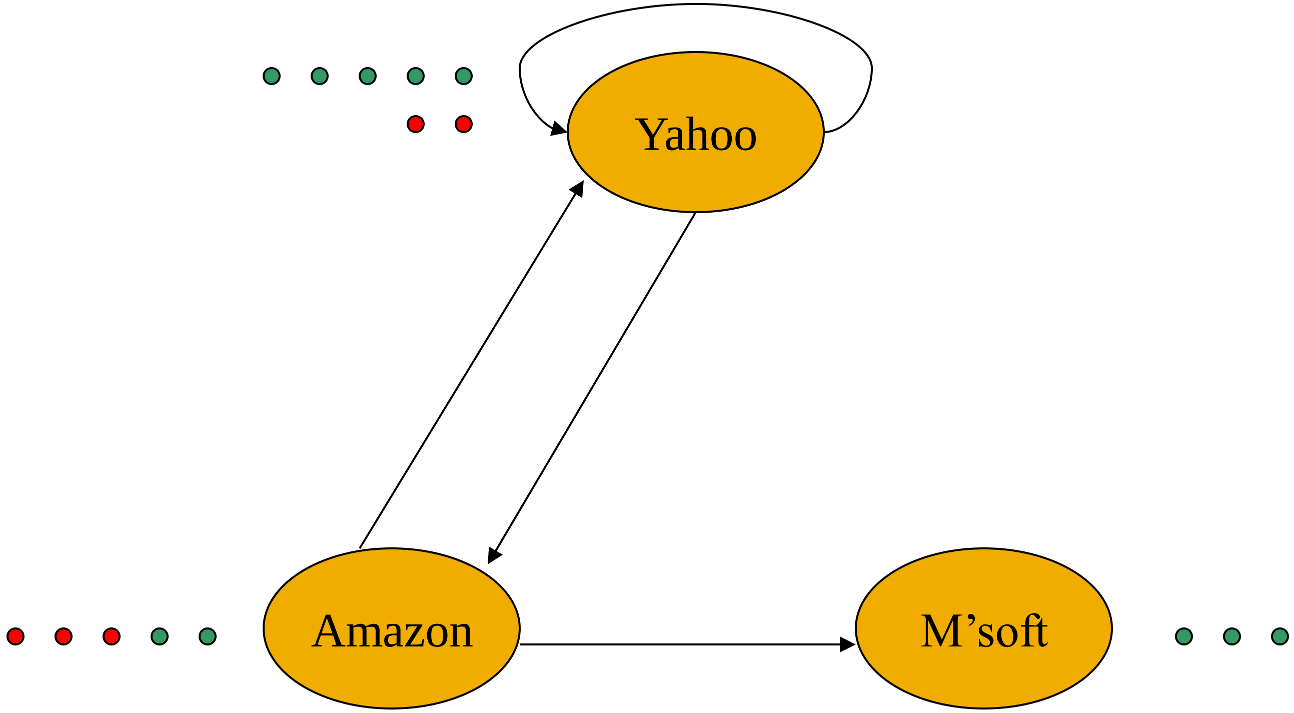
Microsoft Becomes a Dead End



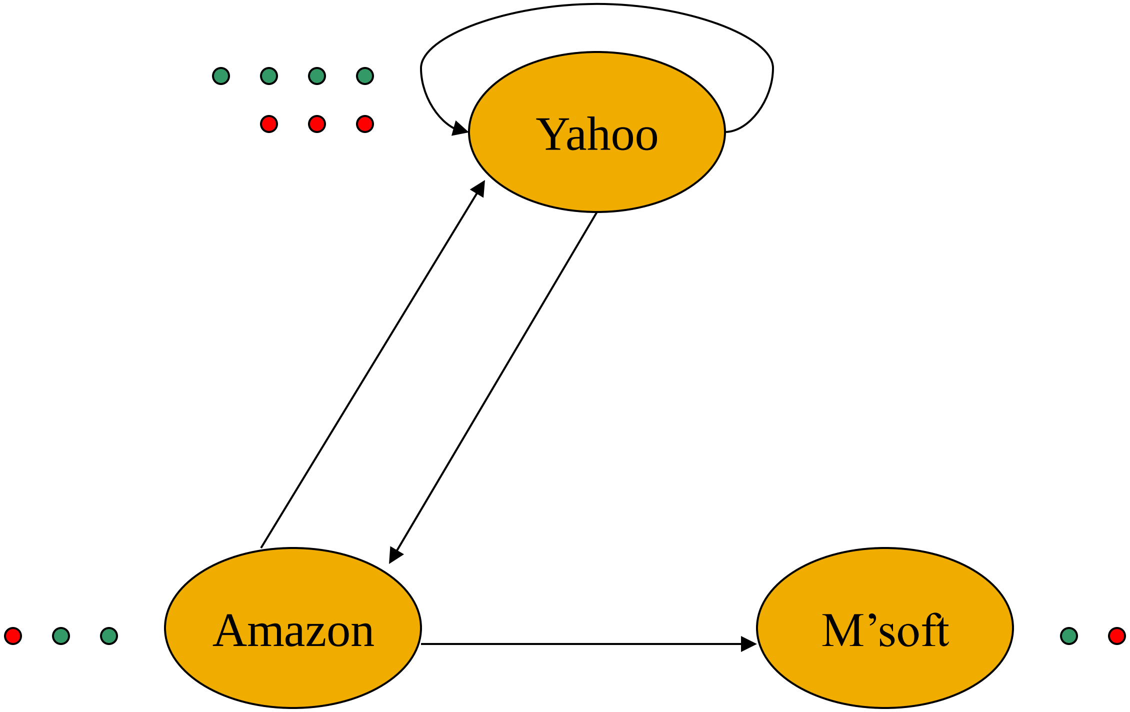
Microsoft Becomes a Dead End



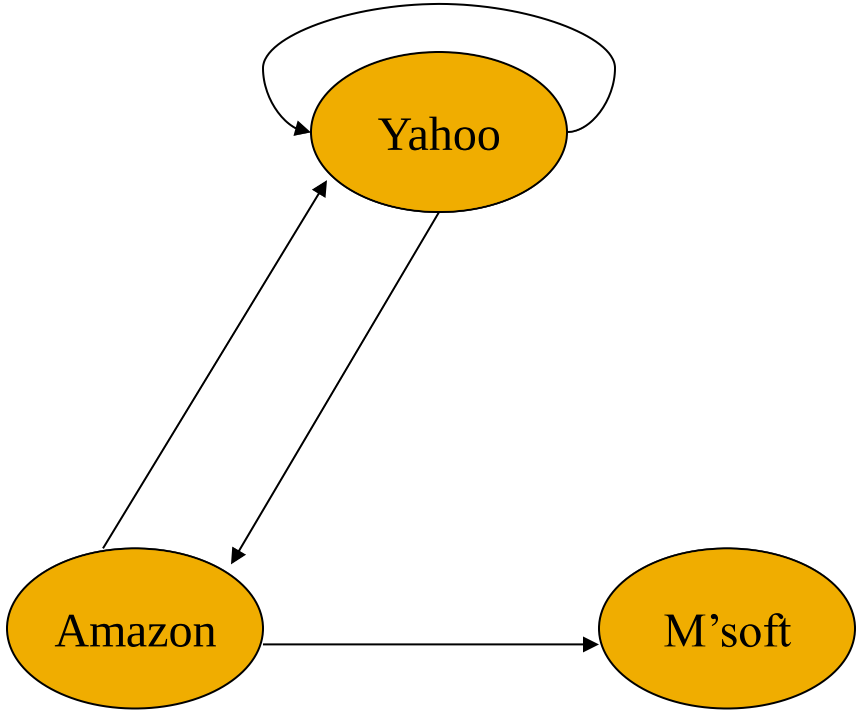
Microsoft Becomes a Dead End



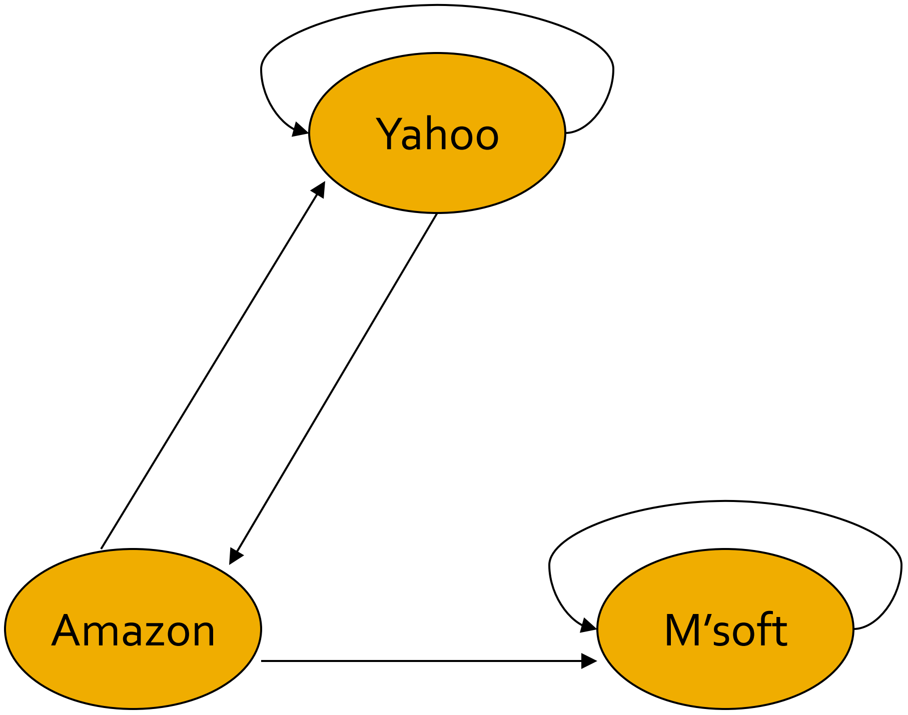
Microsoft Becomes a Dead End



In the Limit …



M’soft Becomes Spider Trap



$$
\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 0 \\
m & 0 & 1/2 & 1 \\
\hline
\end{array}
$$

Example: Effect of Spider Trap

- Equations **v** = *M***v**:

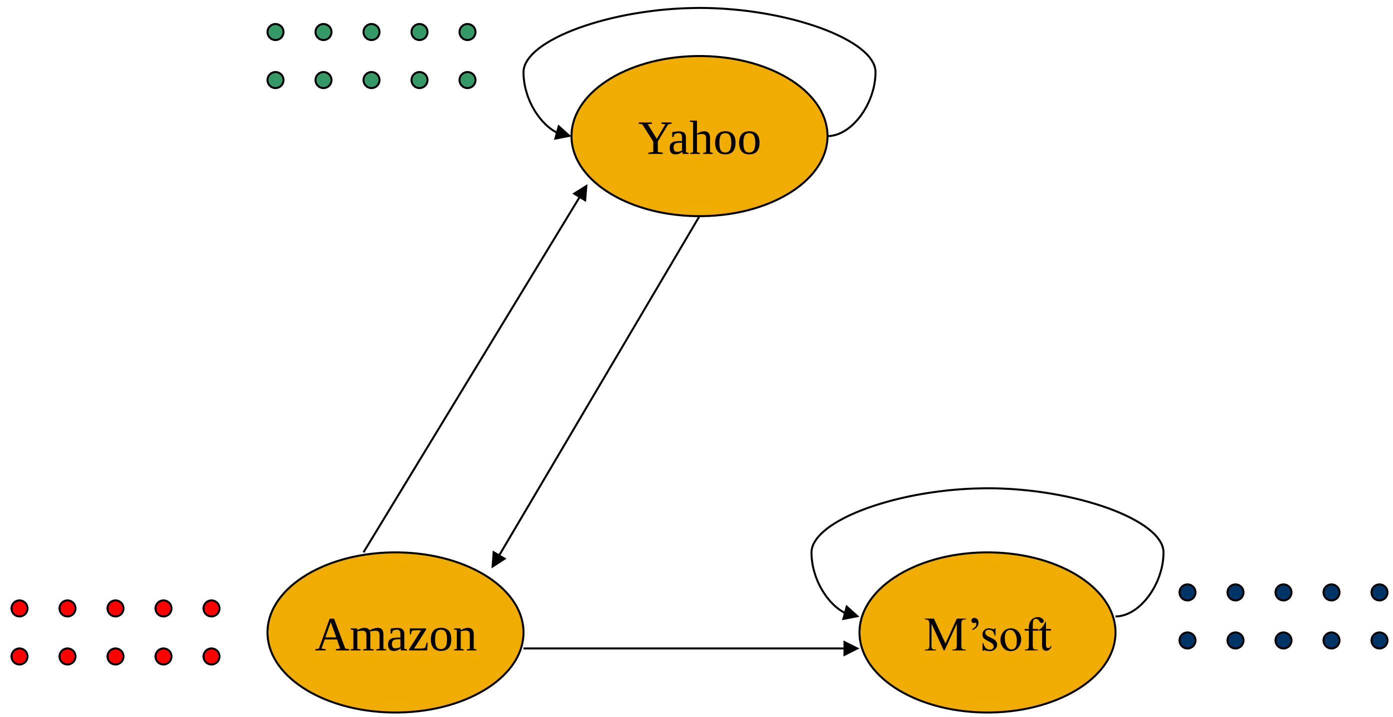
$$y = y/2 + a/2$$

$$a = y/2$$

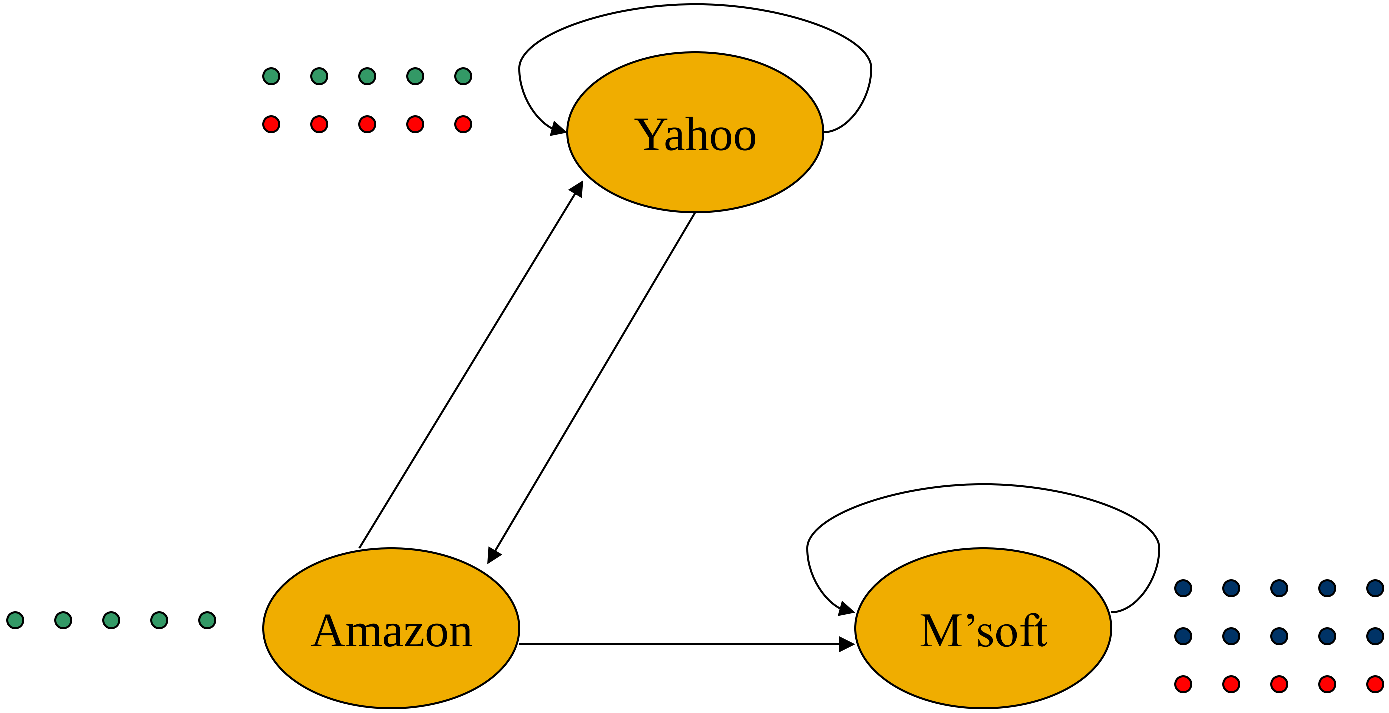
$$m = a/2 + m$$

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{cccccc} 1 & 1 & 3/4 & 5/8 & & 0 \\ 1 & 1/2 & 1/2 & 3/8 & \cdots & 0 \\ 1 & 3/2 & 7/4 & 2 & & 3 \end{array}$$

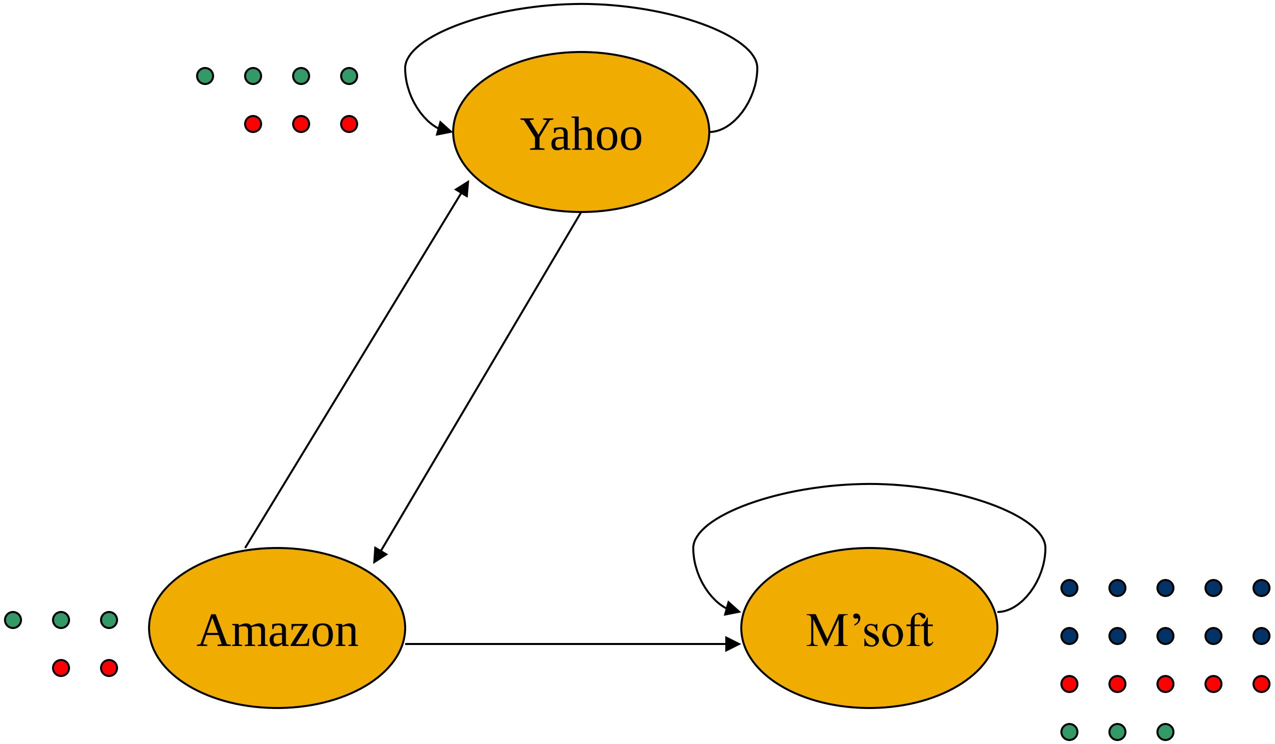
Microsoft Becomes a Spider Trap



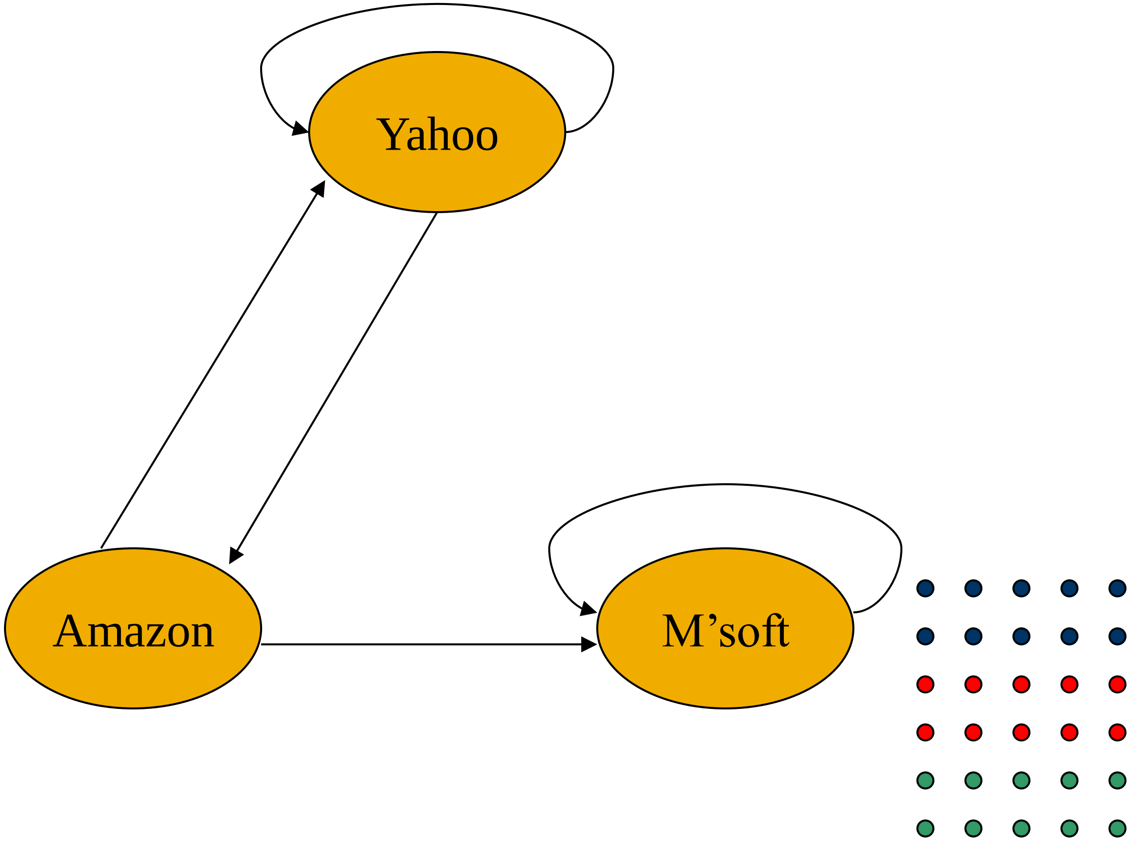
Microsoft Becomes a Spider Trap



Microsoft Becomes a Spider Trap



In the Limit …



PageRank Solution to Traps, Etc.

- “Tax” each page a fixed percentage at each iteration.
- Add a fixed constant to all pages.
  - Optional but useful: add exactly enough to balance the loss (tax + PageRank of dead ends).
- Models a random walk with a fixed probability of leaving the system, and a fixed number of new walkers injected into the system at each step.
  - Divided equally among all pages.

Example: Microsoft is a Spider Trap; 20% Tax

- Equations $\mathbf{v} = 0.8(M\mathbf{v}) + \mathbf{0.2}$:

$$y = 0.8(y/2 + a/2) + 0.2$$

$$a = 0.8(y/2) + 0.2$$

$$m = 0.8(a/2 + m) + 0.2$$

y		1	1.00	0.84	0.776		7/11
a	=	1	0.60	0.60	0.536	. . .	5/11
m		1	1.40	1.56	1.688		21/11

Topic-Specific PageRank

Focusing on Specific Pages

Teleport Sets

Interpretation as a Random Walk

Topic-Specific Page Rank

- Goal: Evaluate Web pages not just according to their popularity, but also by how relevant they are to a particular topic, e.g. “sports” or “history.”
- Allows search queries to be answered based on interests of the user.
- Example: Search query [jaguar] wants different pages depending on whether you are interested in automobiles, nature, or sports.

Teleport Sets

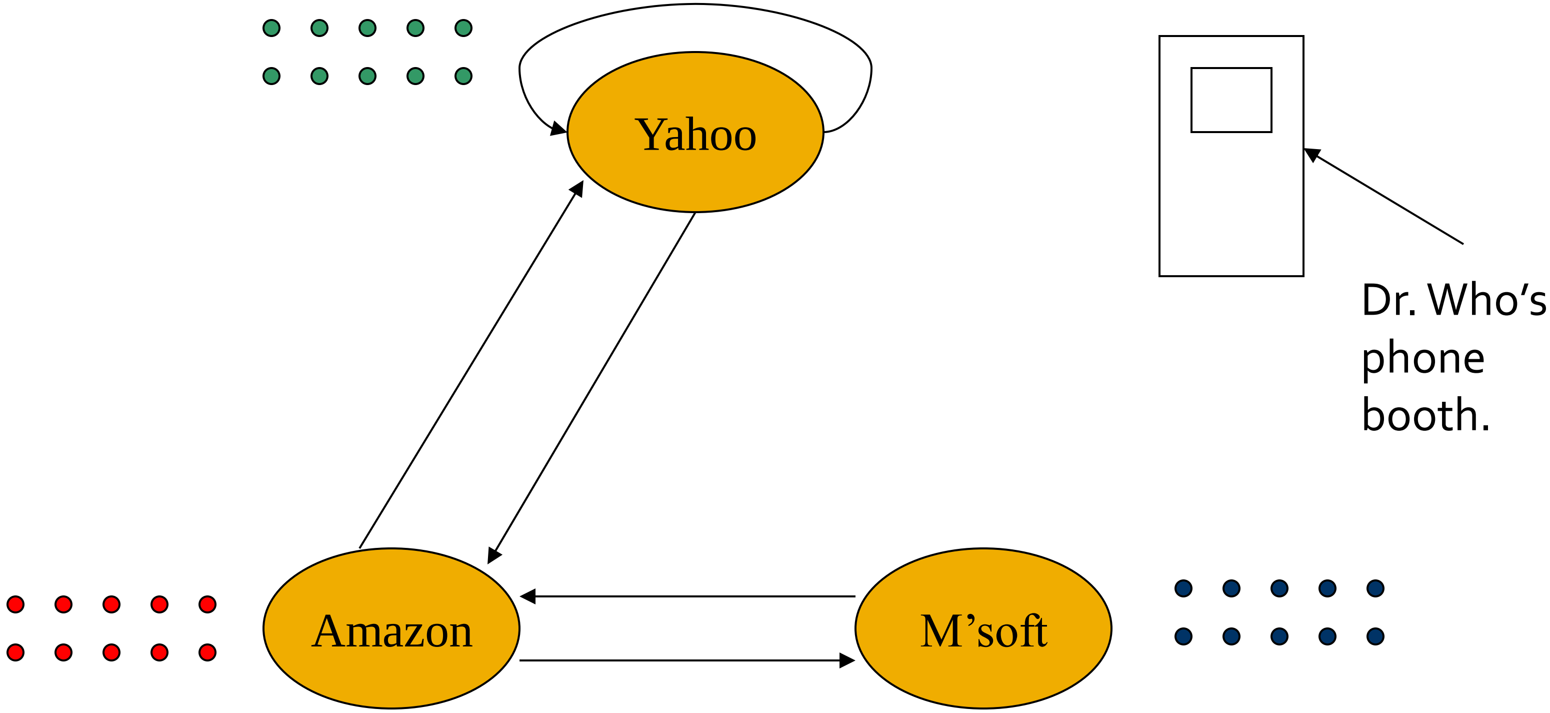
- Assume each walker has a small probability of "teleporting" at any tick.
- Teleport can go to:

1. Any page with equal probability.
   - As in the "taxation" scheme.
2. A set of "relevant" pages (*teleport set*).
   - For *topic-specific* PageRank.

Example: Topic = Software

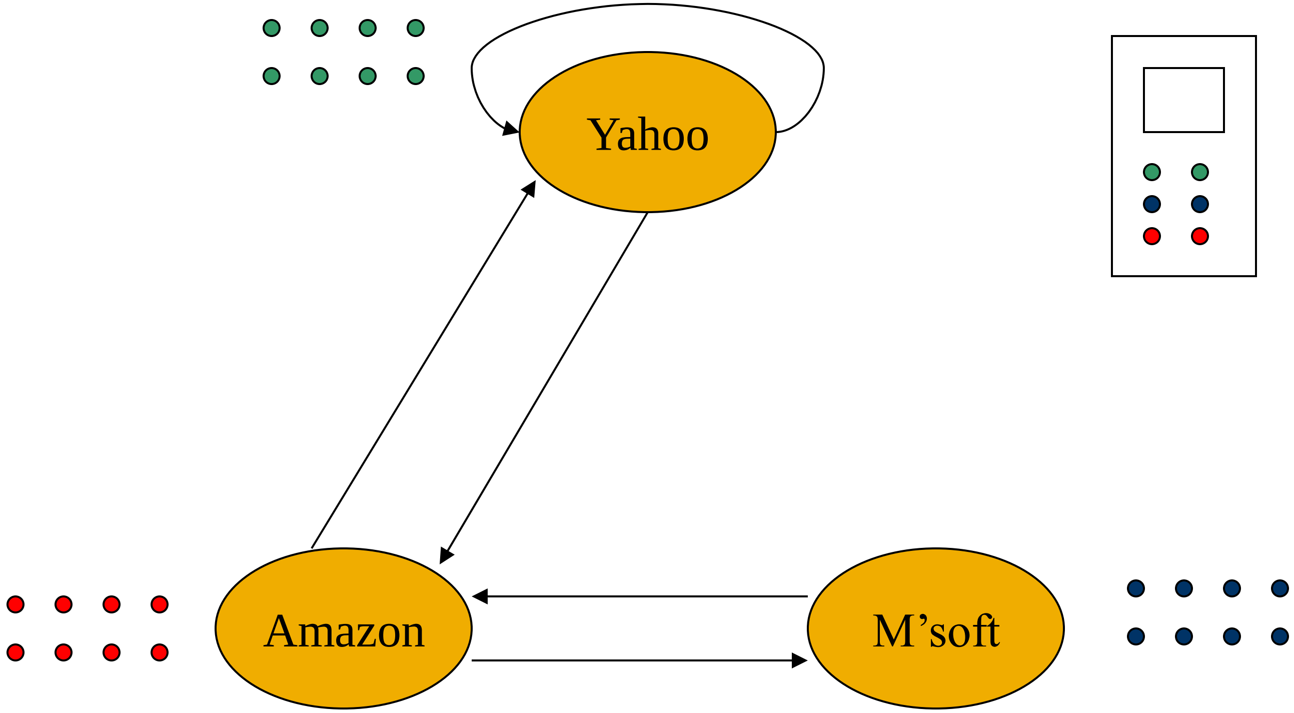
- Only Microsoft is in the teleport set.
- Assume 20% “tax.”
  - I.e., probability of a teleport is 20%.

Only Microsoft in Teleport Set

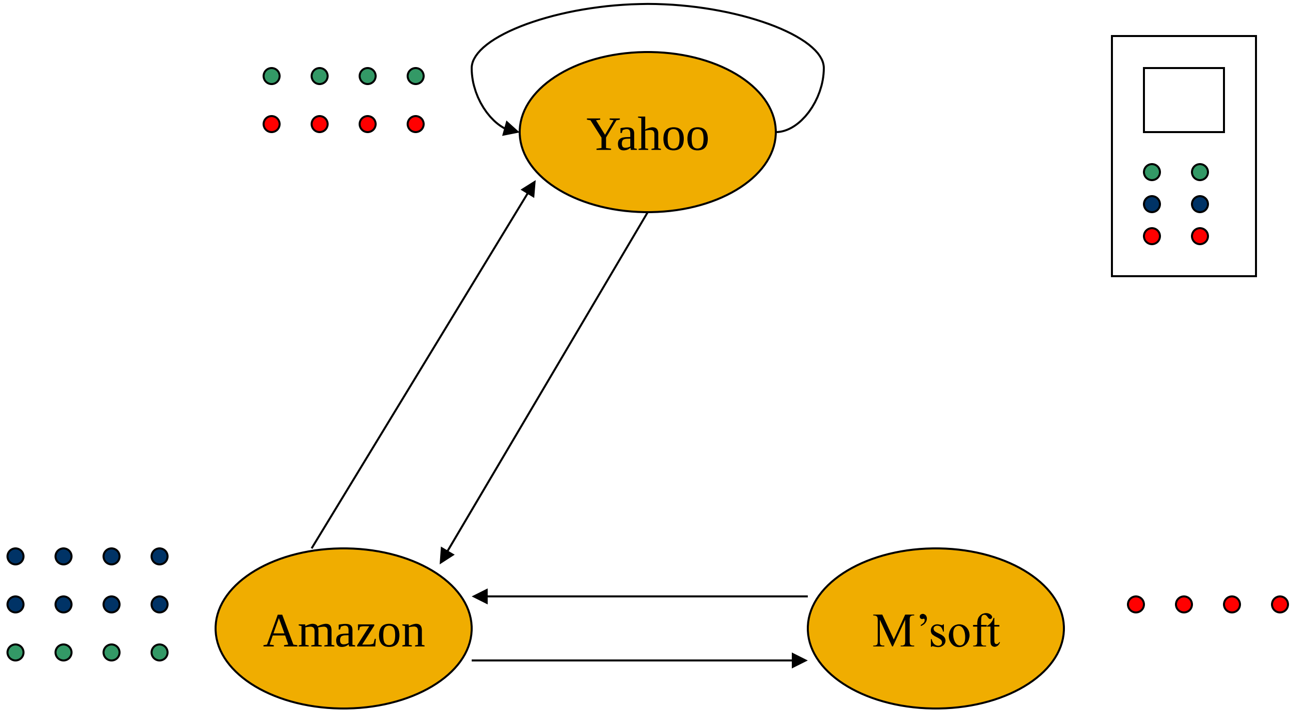


Dr. Who’s
phone
booth.

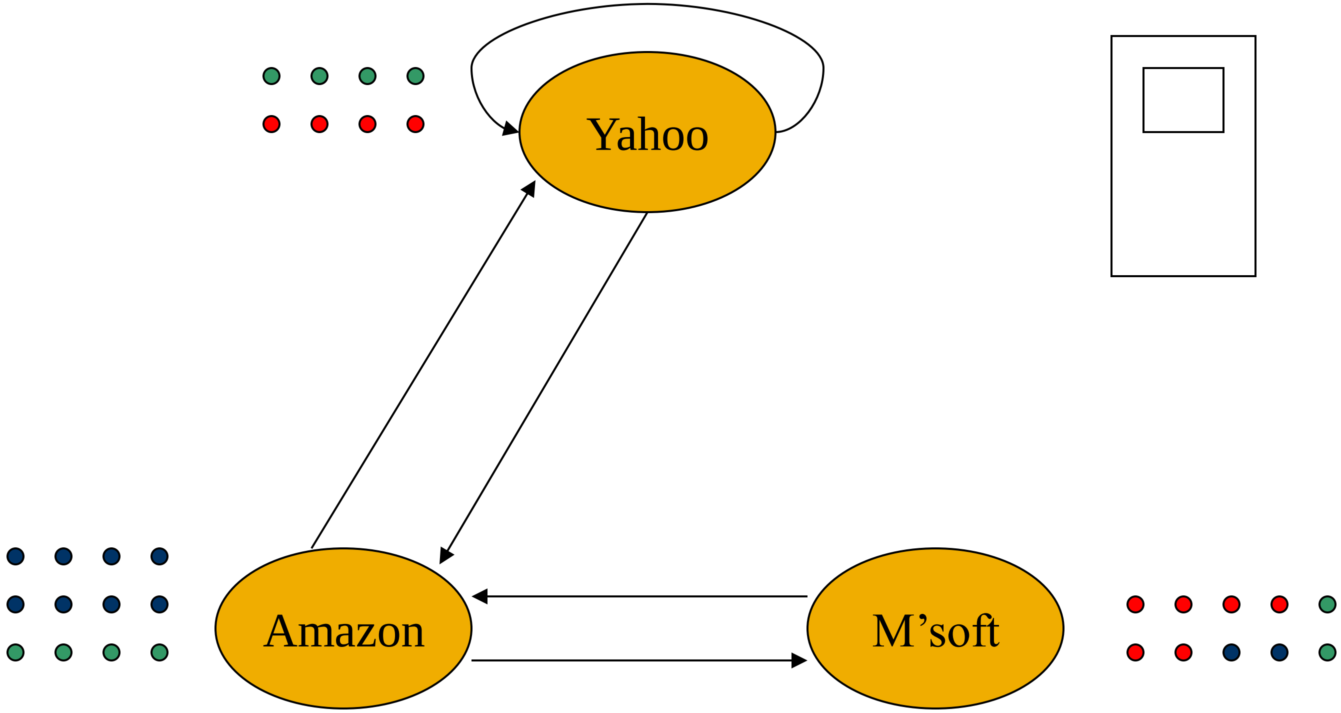
Only Microsoft in Teleport Set



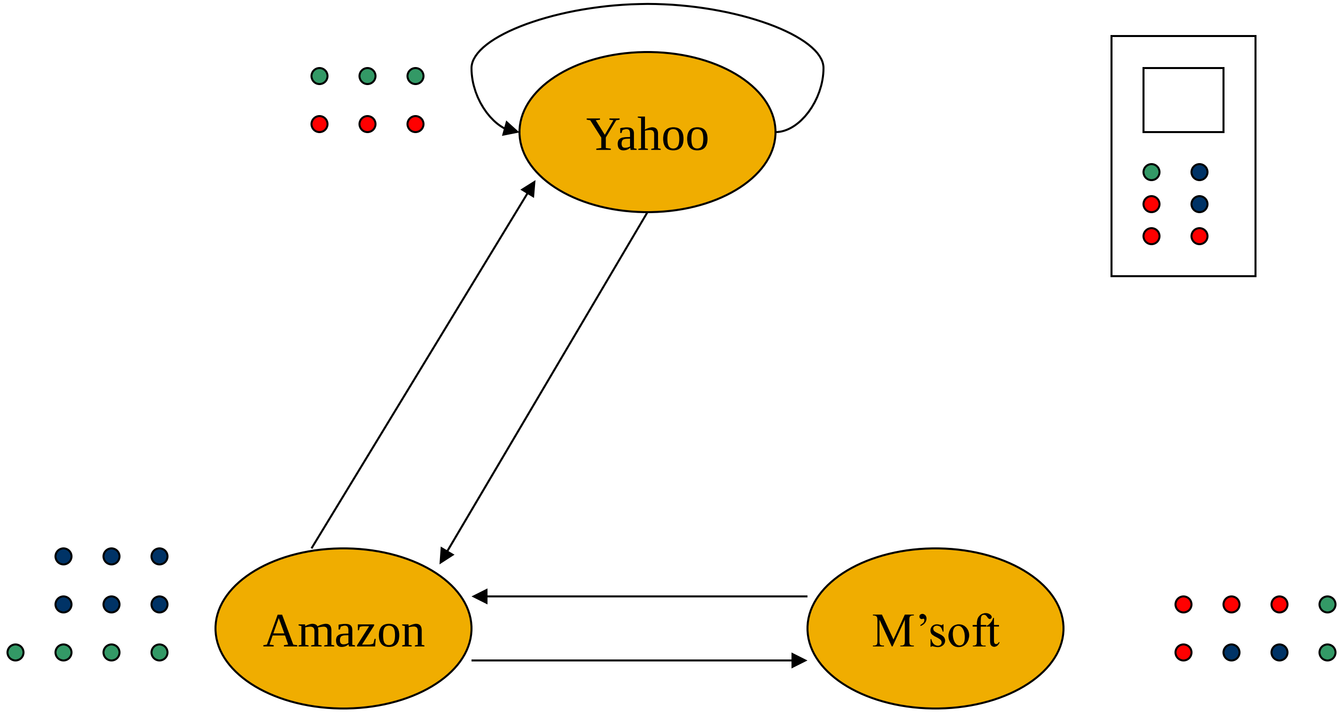
Only Microsoft in Teleport Set



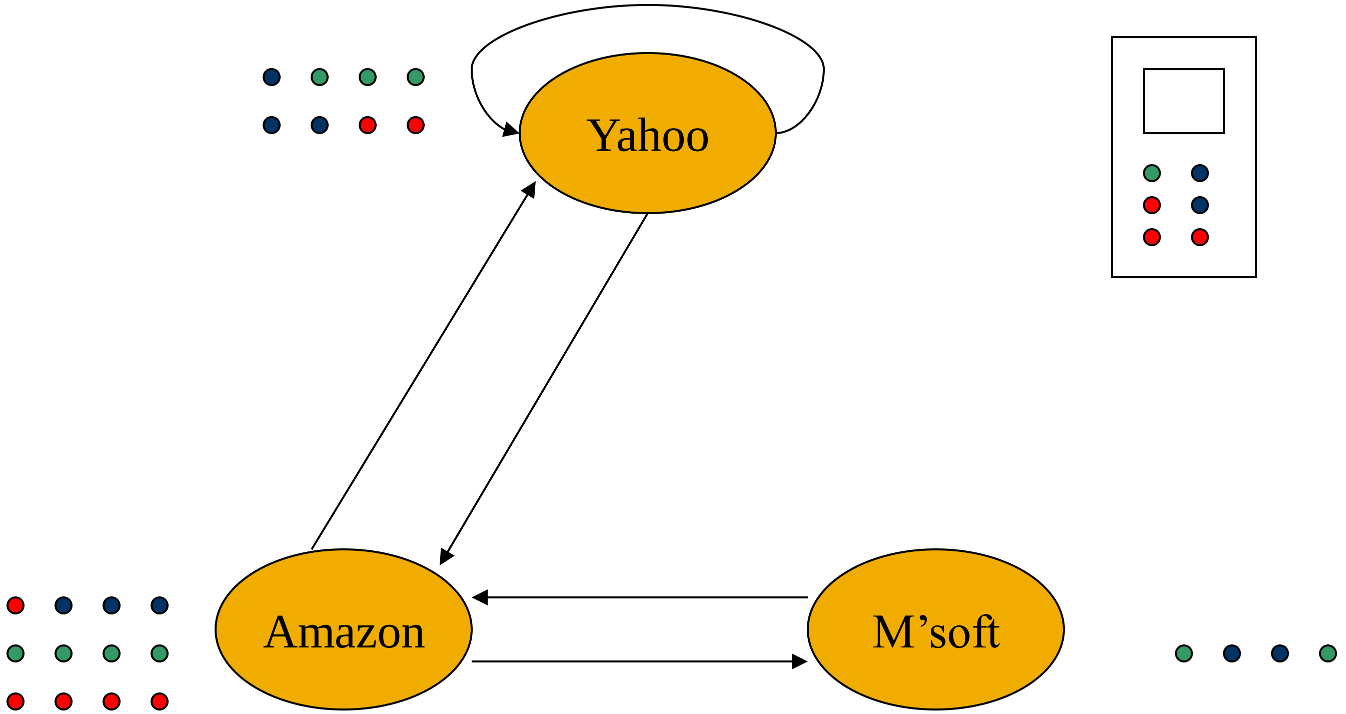
Only Microsoft in Teleport Set



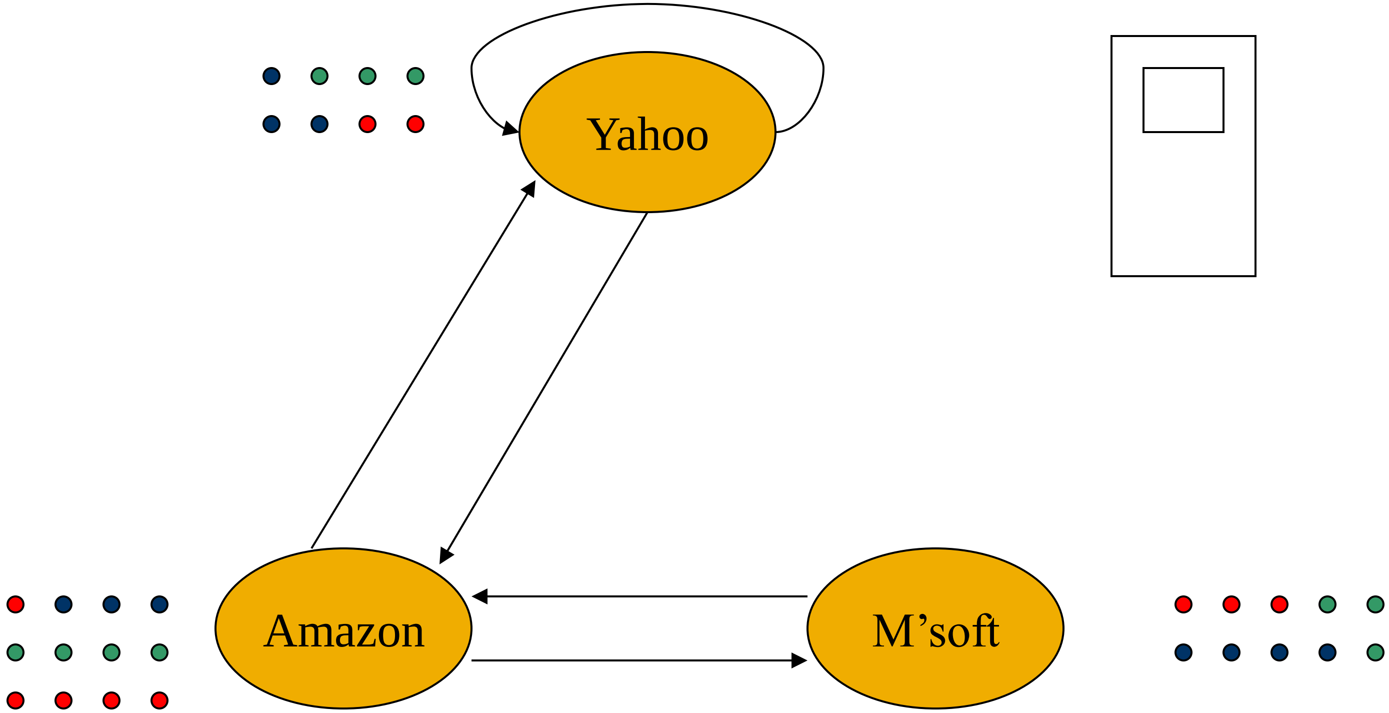
Only Microsoft in Teleport Set



Only Microsoft in Teleport Set



Only Microsoft in Teleport Set



Picking the Teleport Set

1. Choose the pages belonging to the topic in Open Directory.
2. “Learn,” from a training set, the typical words in pages belonging to the topic; use pages heavy in those words as the teleport set.

Application: Link Spam

- Spam farmers create networks of millions of pages designed to focus PageRank on a few undeserving pages.
  - We’ll discuss this technology shortly.
- To minimize their influence, use a teleport set consisting of trusted pages only.
  - Example: home pages of universities.