PageRank

Transition Matrix of the Web
Topic-Specific PageRank
Hubs and Authorities (HITS)
Combatting Link Spam

Cloud and Big Data Summer School, Stockholm, Aug., 2015

Jeffrey D. Ullman

PageRank

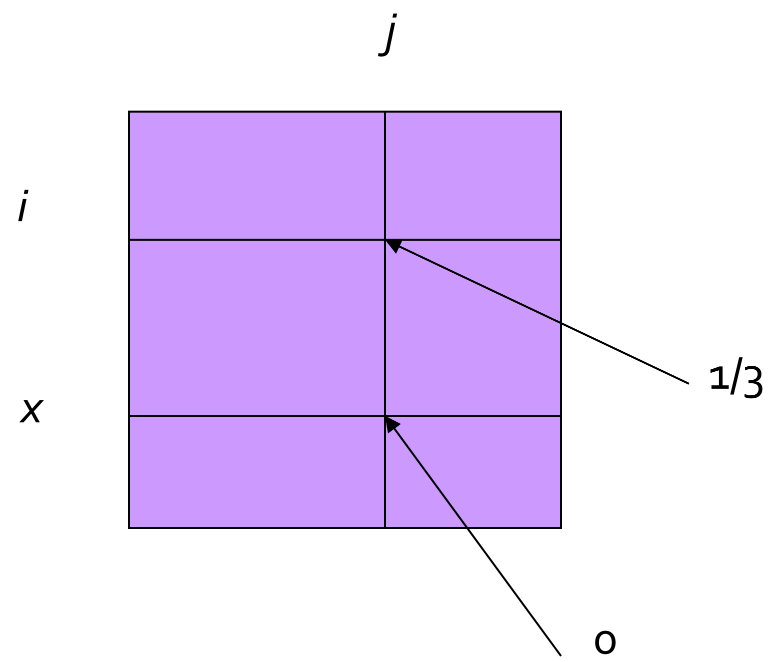
- Intuition: solve the recursive equation: “a page is important if important pages link to it.”
- Technically, *importance* = the principal eigenvector of the transition matrix of the Web.
  - A few fixups needed.

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*.



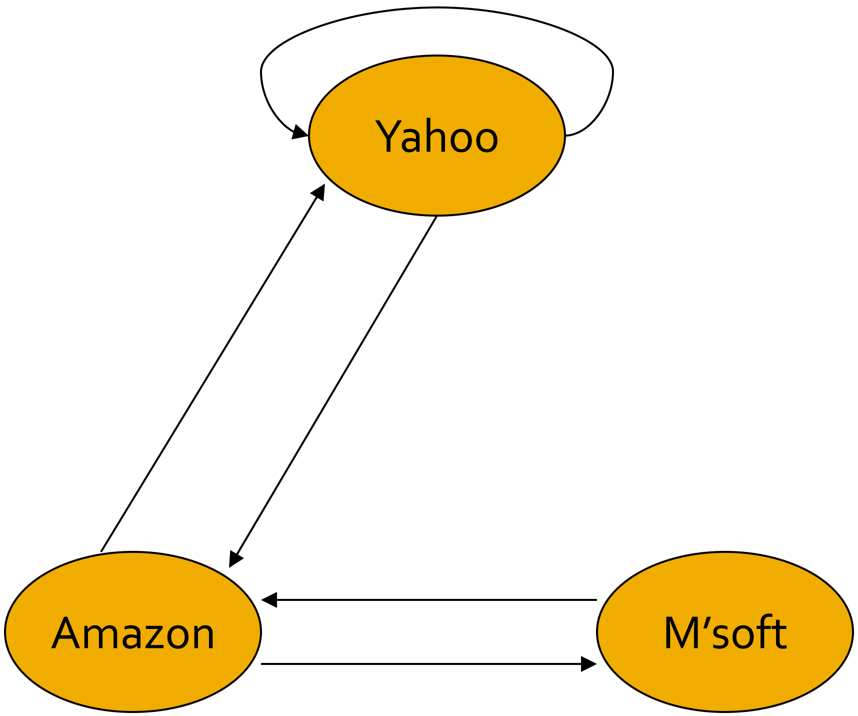
Random Walks on the Web

- 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 $M\mathbf{v}$.

Random Walks – (2)

- Starting from any vector **v**, the limit $M(M(\ldots M(M\mathbf{v})\ldots))$ 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.

Example: The Web in 1839



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

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).
- Can work if you start with a fixed **v**.

Simulating a Random Walk

- Start with the vector **v** = [1, 1,..., 1] representing the idea that each Web page is given one unit of *importance*.
- Repeatedly apply the matrix *M* to **v**, allowing the importance to flow like a random walk.
- About 50 iterations is sufficient to estimate the limiting solution.

Example: Iterating Equations

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

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

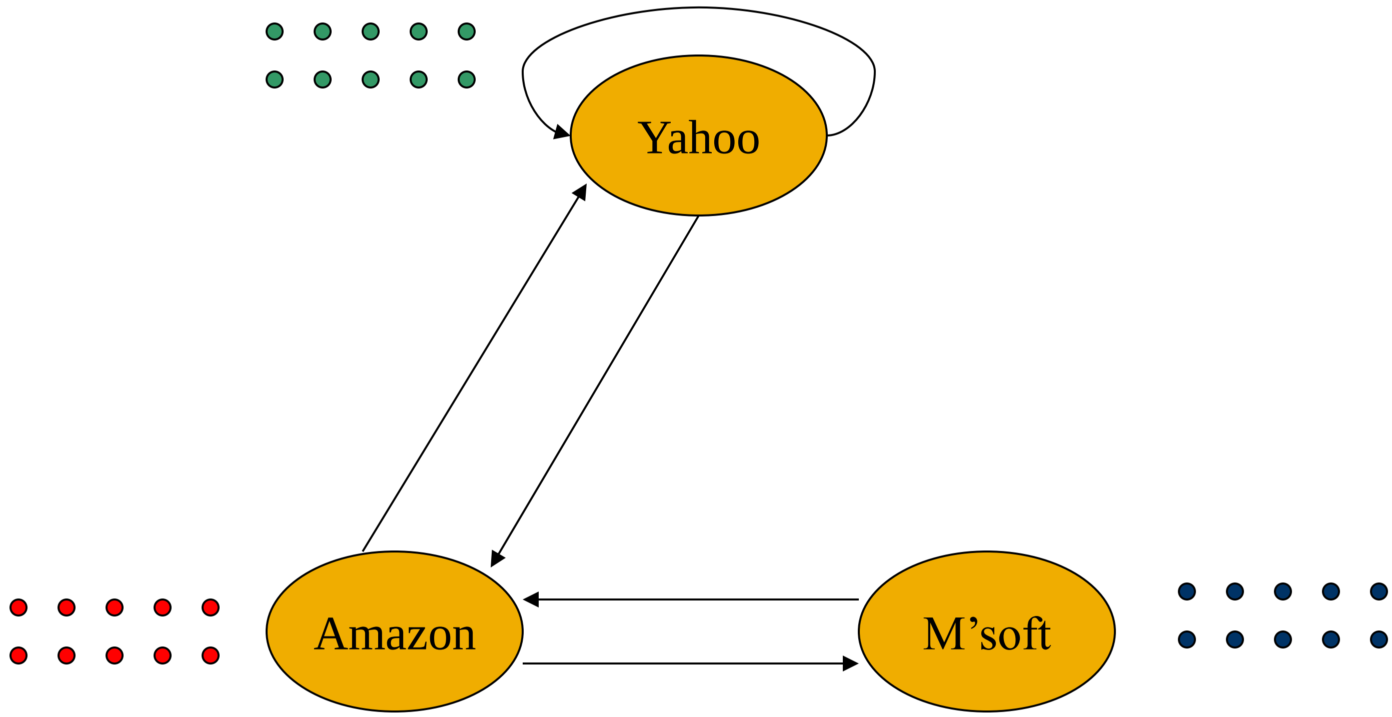
$$a = y/2 + m$$

$$m = a/2$$

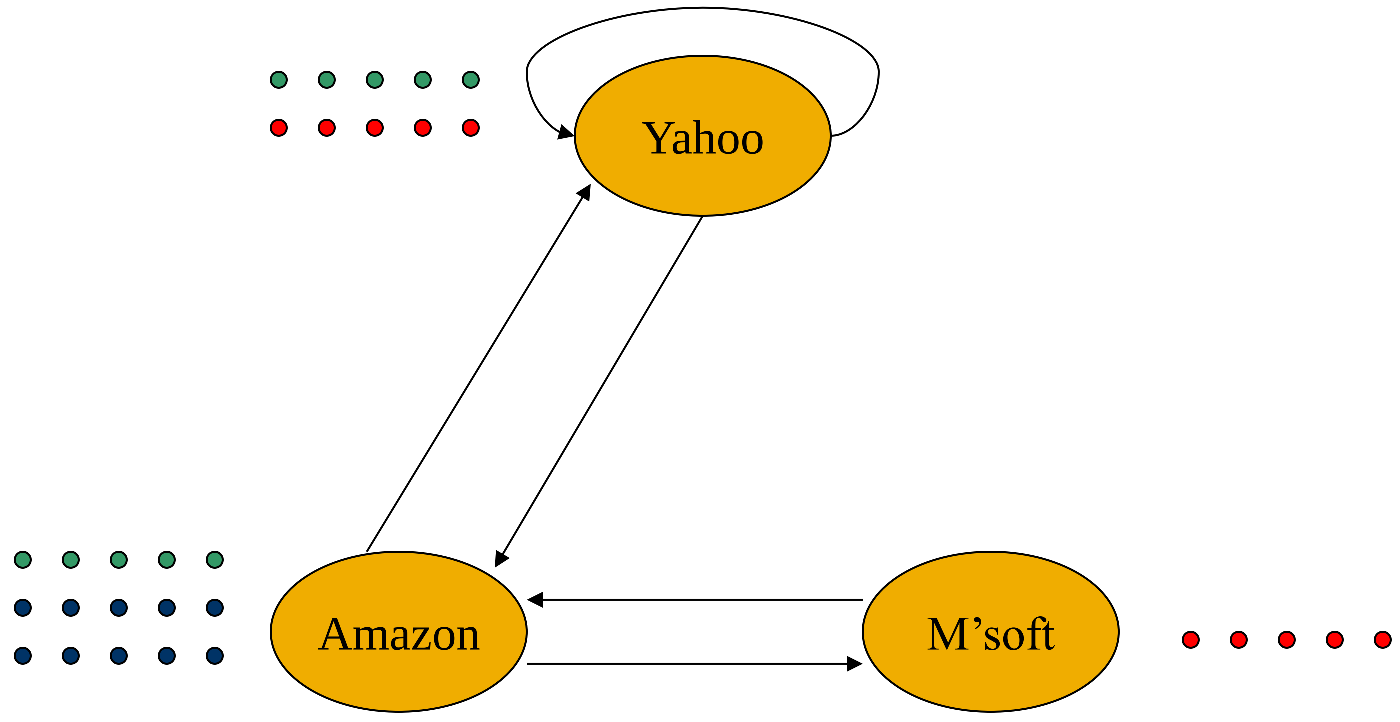
Note: "=" is really "assignment."

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

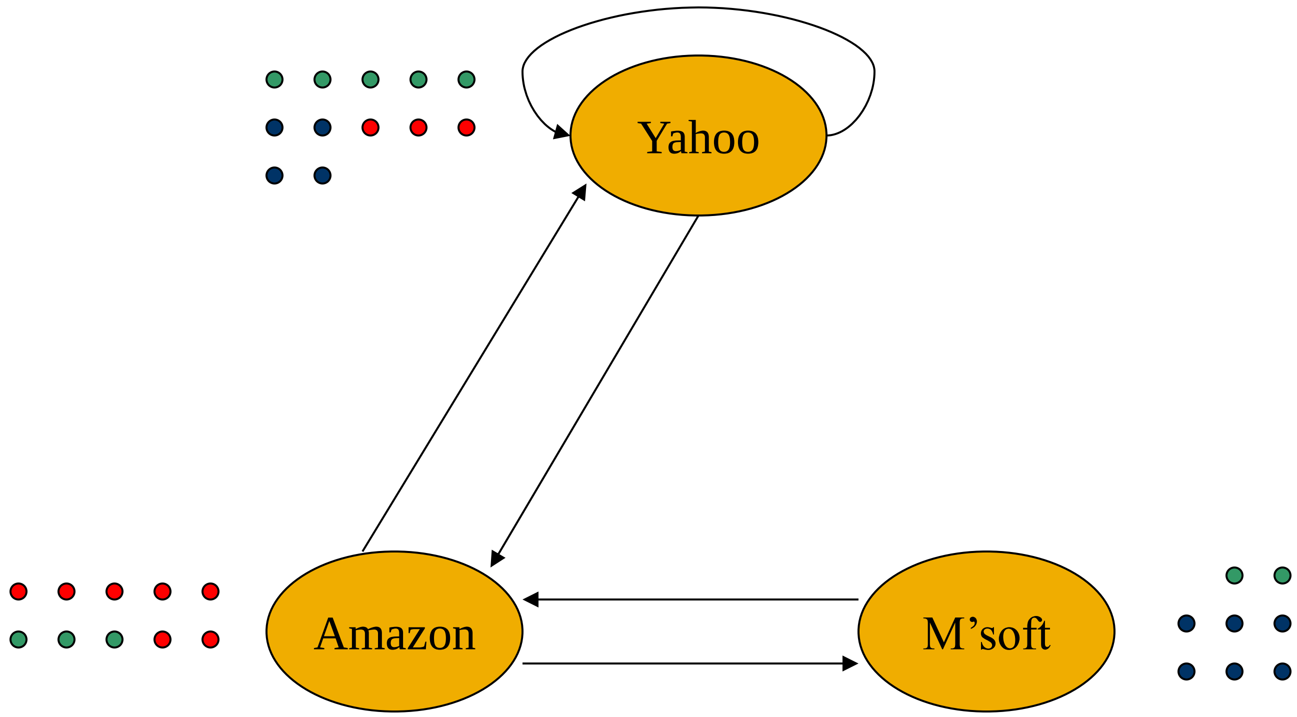
The Walkers



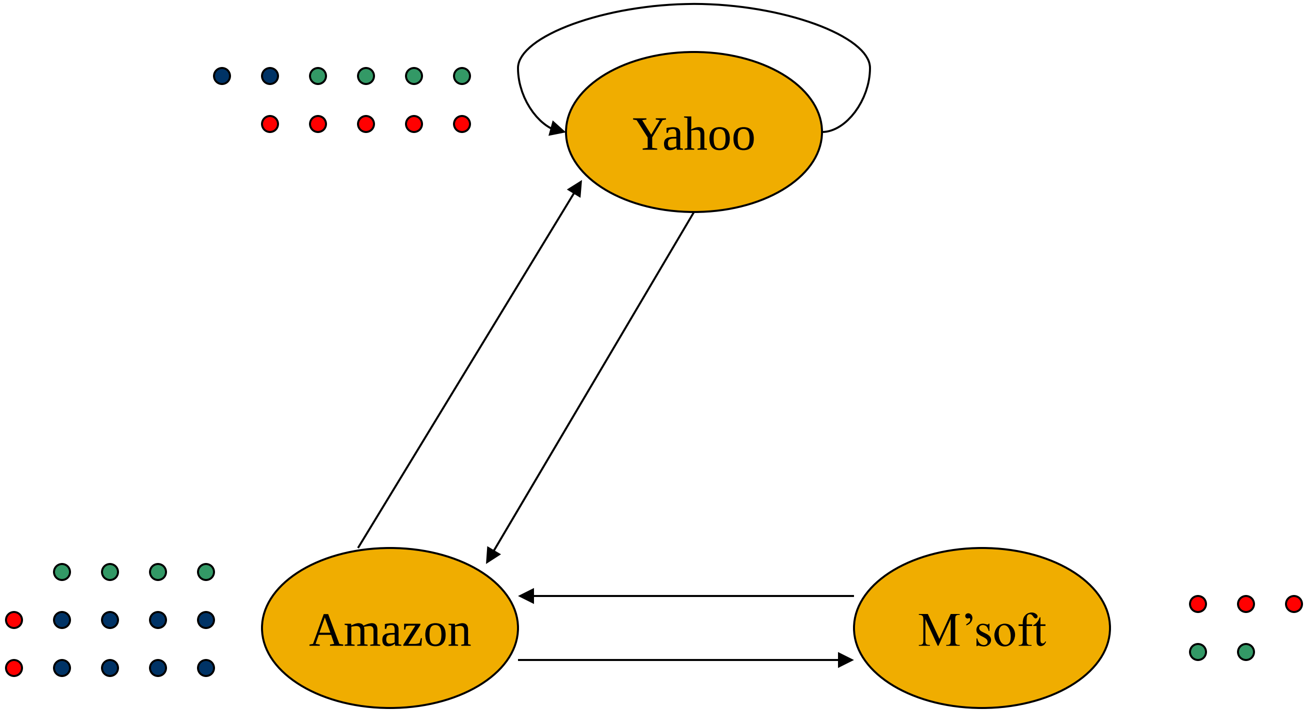
The Walkers



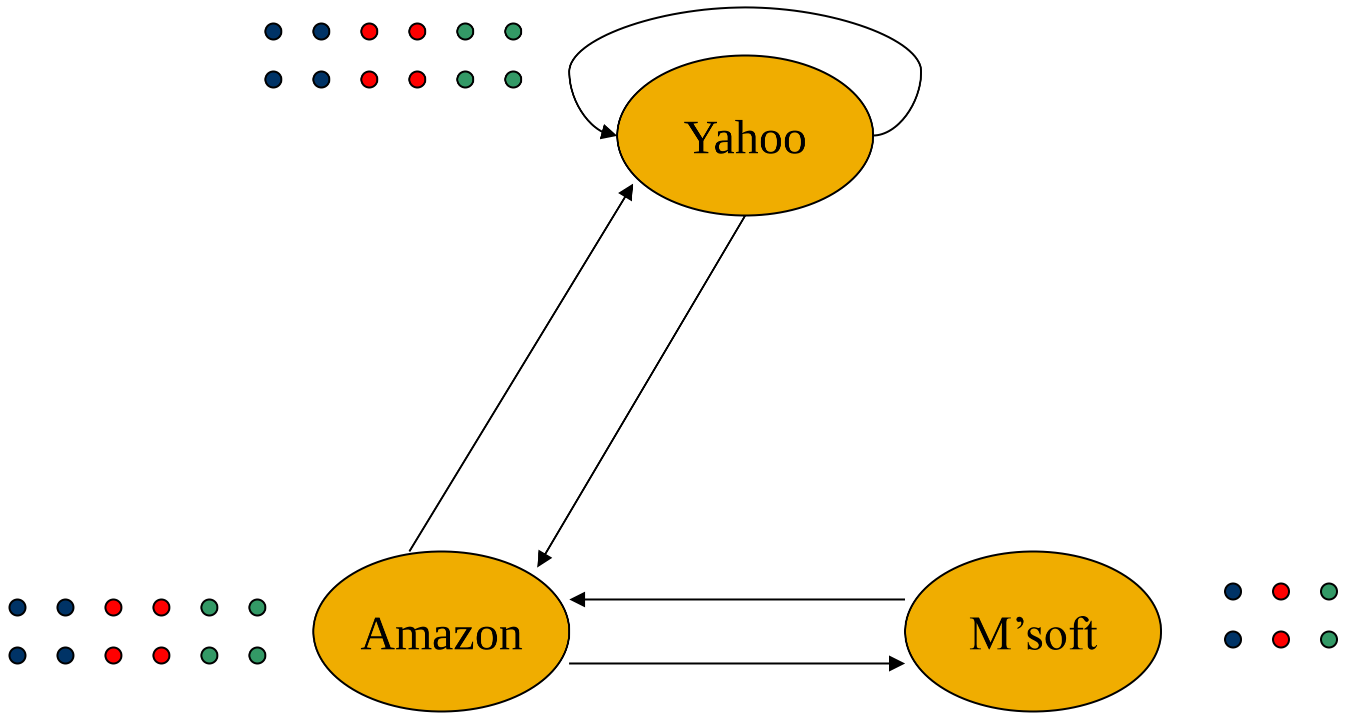
The Walkers



The Walkers



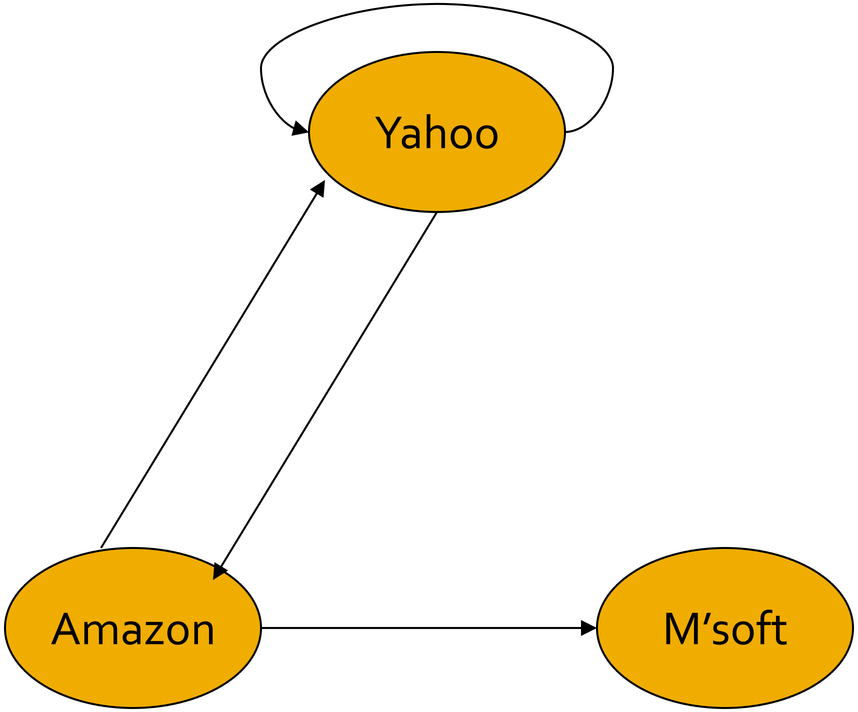
In the Limit ...



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



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	0

Example: Effect of Dead Ends

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

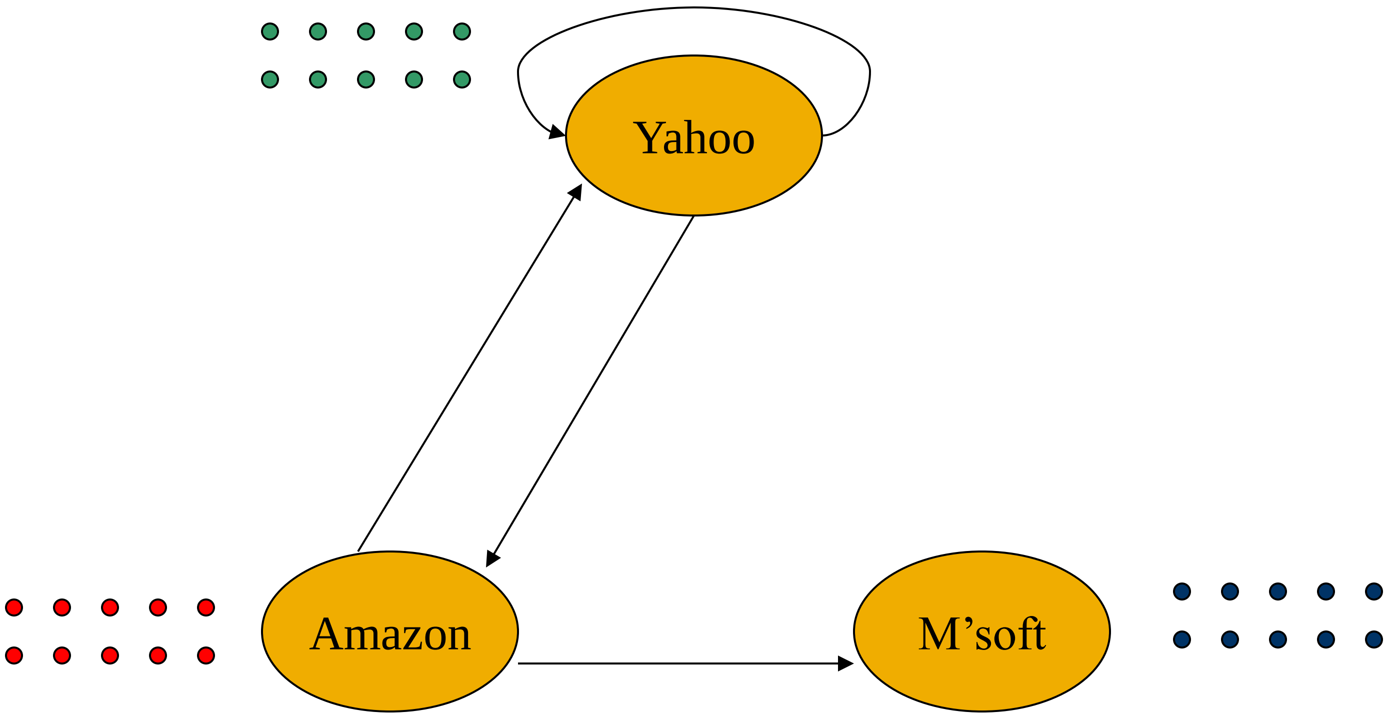
$y = y/2 + a/2$

$a = y/2$

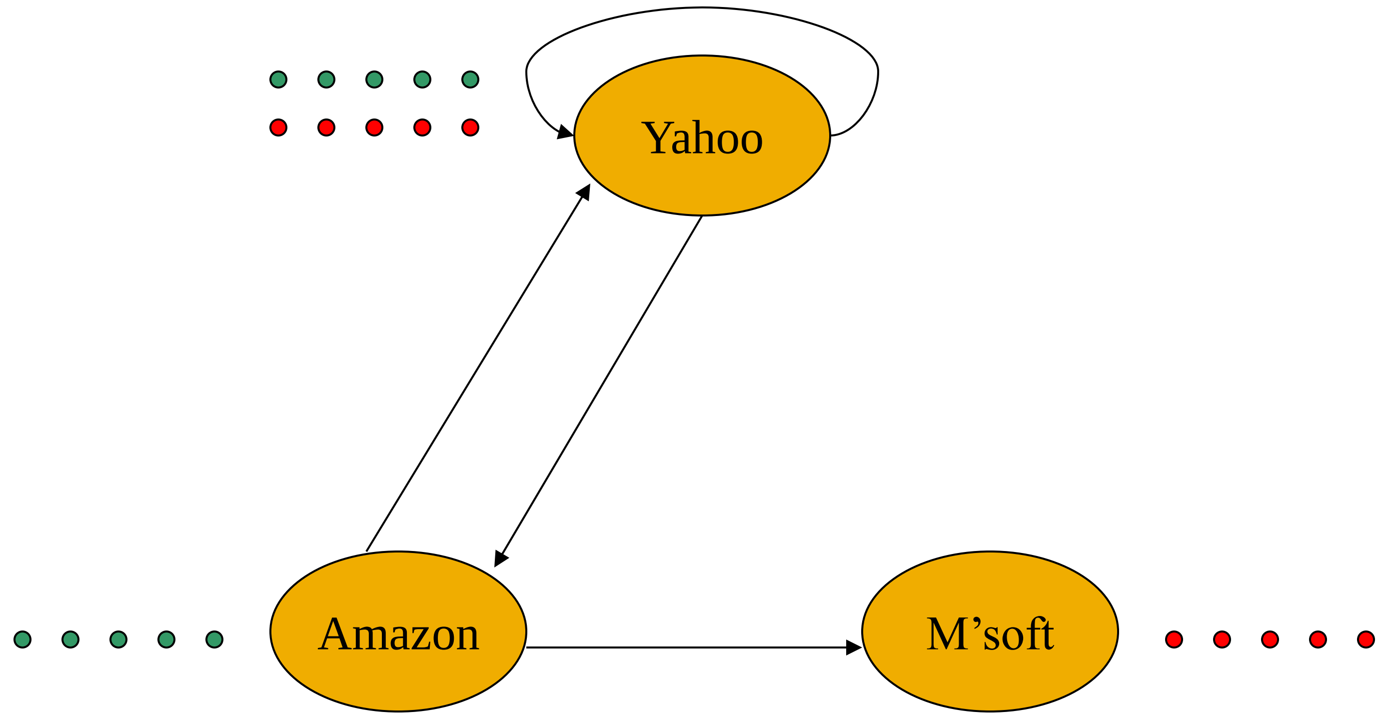
$m = a/2$

y		1	1	3/4	5/8		0
a	=	1	1/2	1/2	3/8	. . .	0
m		1	1/2	1/4	1/4		0

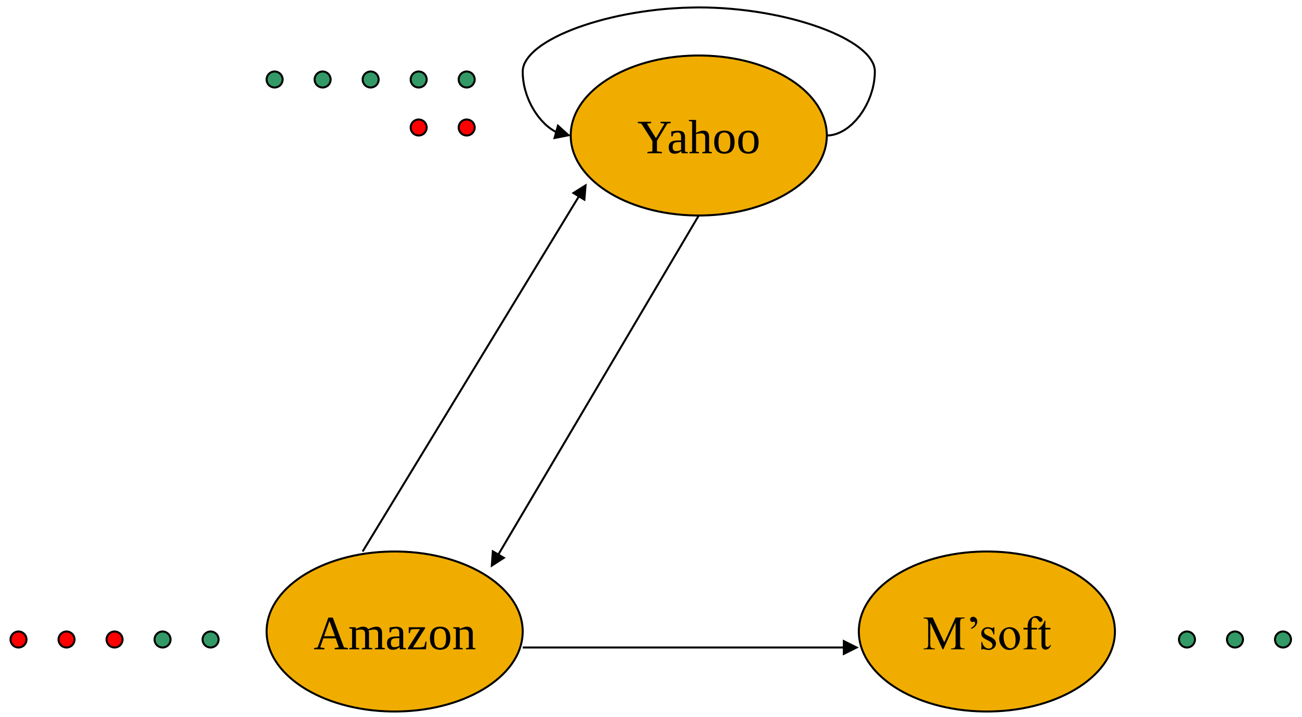
Microsoft Becomes a Dead End



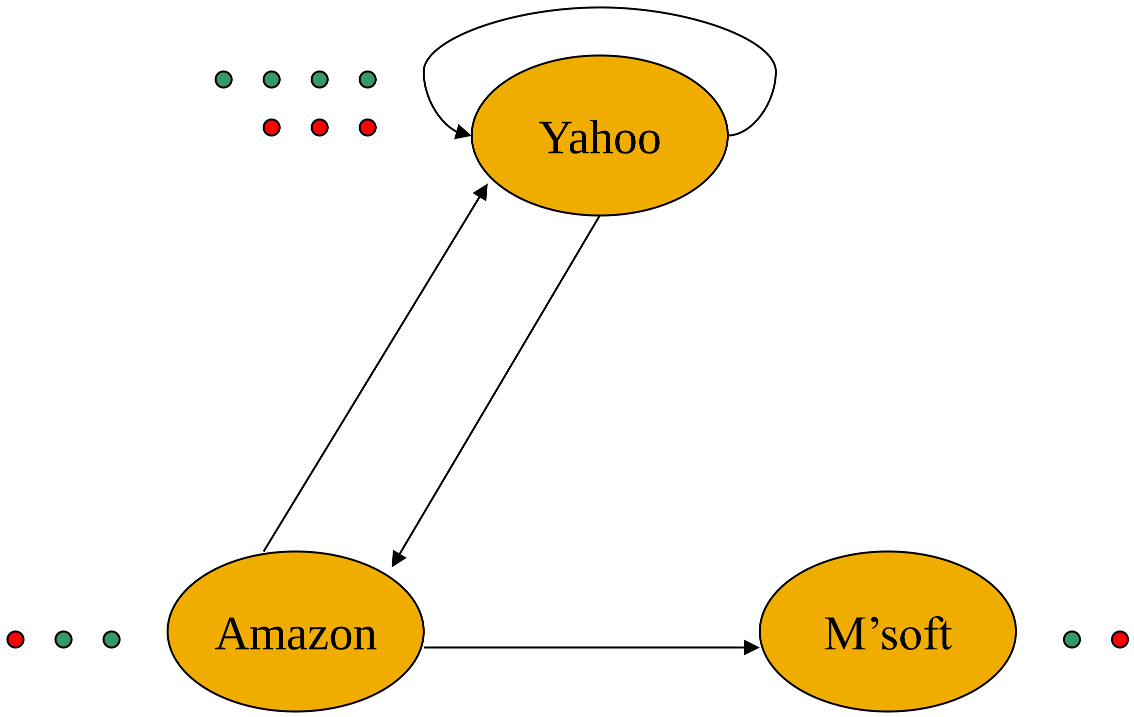
Microsoft Becomes a Dead End



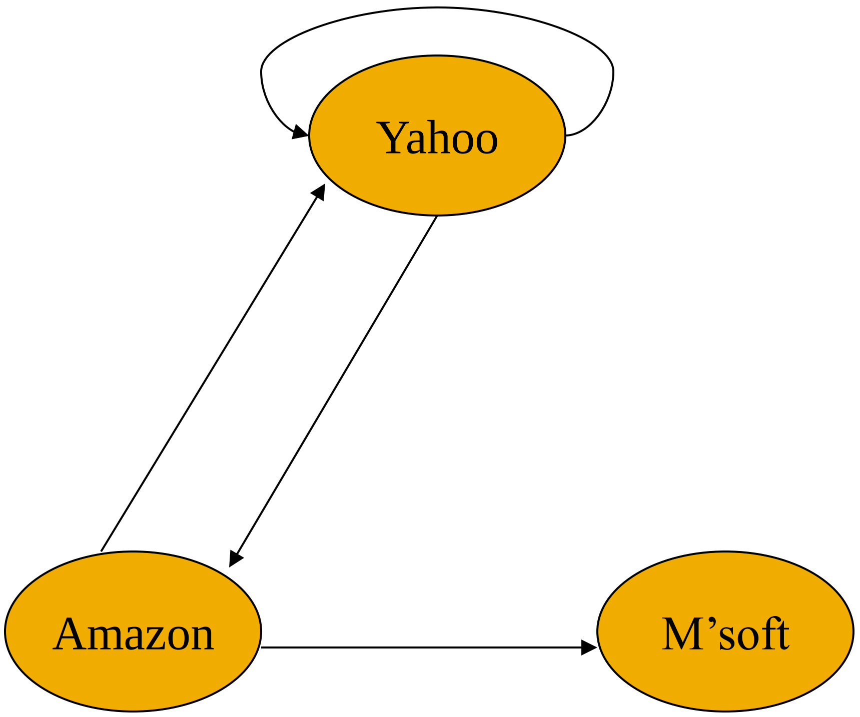
Microsoft Becomes a Dead End



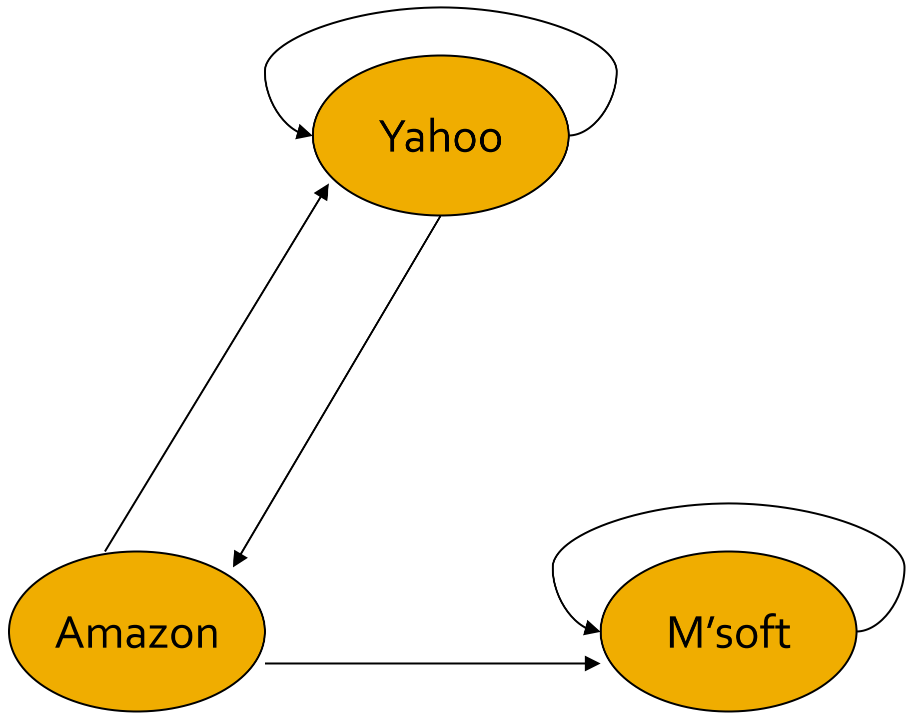
Microsoft Becomes a Dead End



In the Limit ...



M'soft Becomes Spider Trap



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

Example: Effect of Spider Trap

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

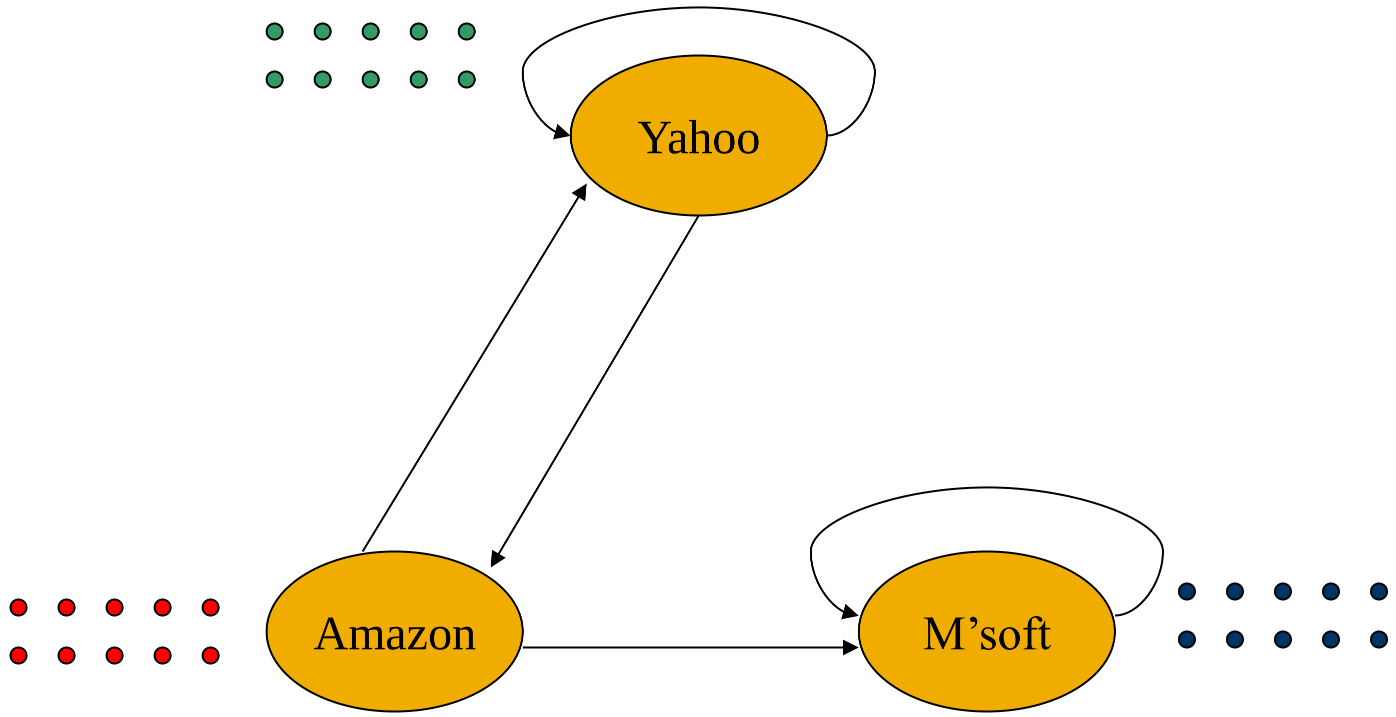
$y = y/2 + a/2$

$a = y/2$

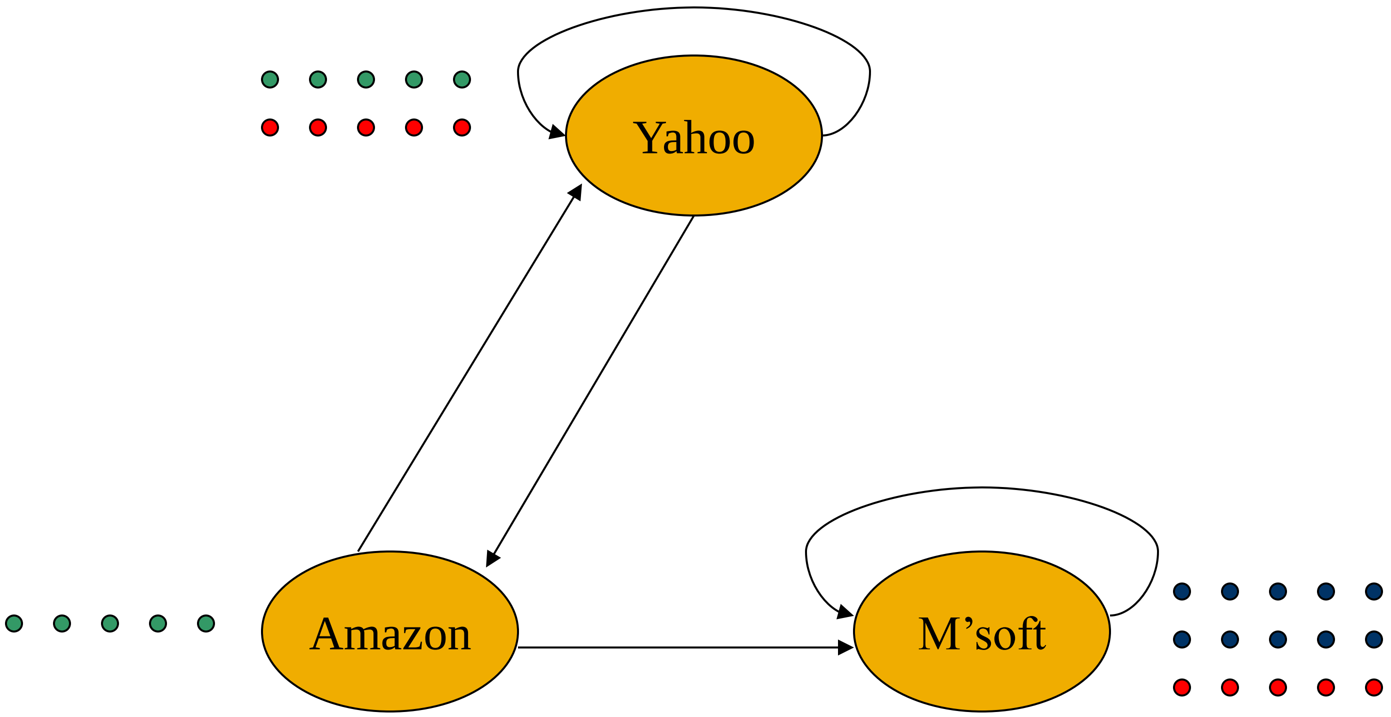
$m = a/2 + m$

y		1	1	3/4	5/8		0
a	=	1	1/2	1/2	3/8	. . .	0
m		1	3/2	7/4	2		3

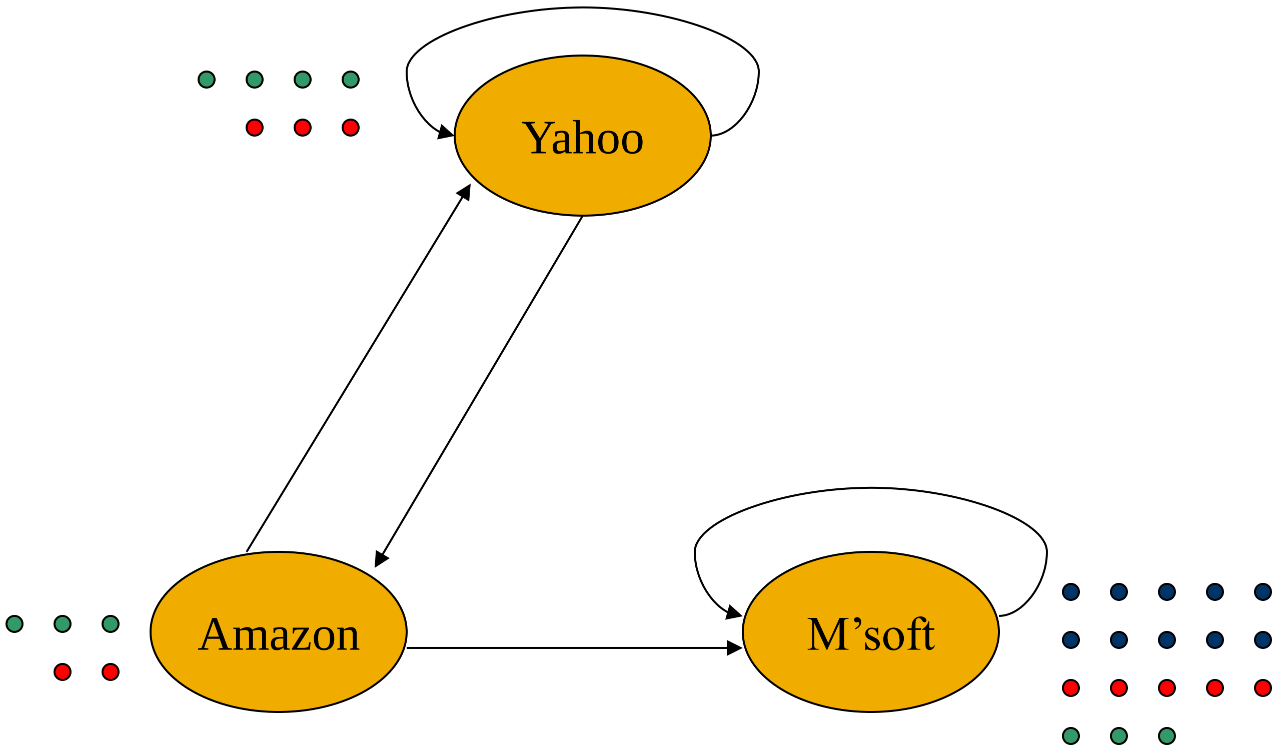
Microsoft Becomes a Spider Trap



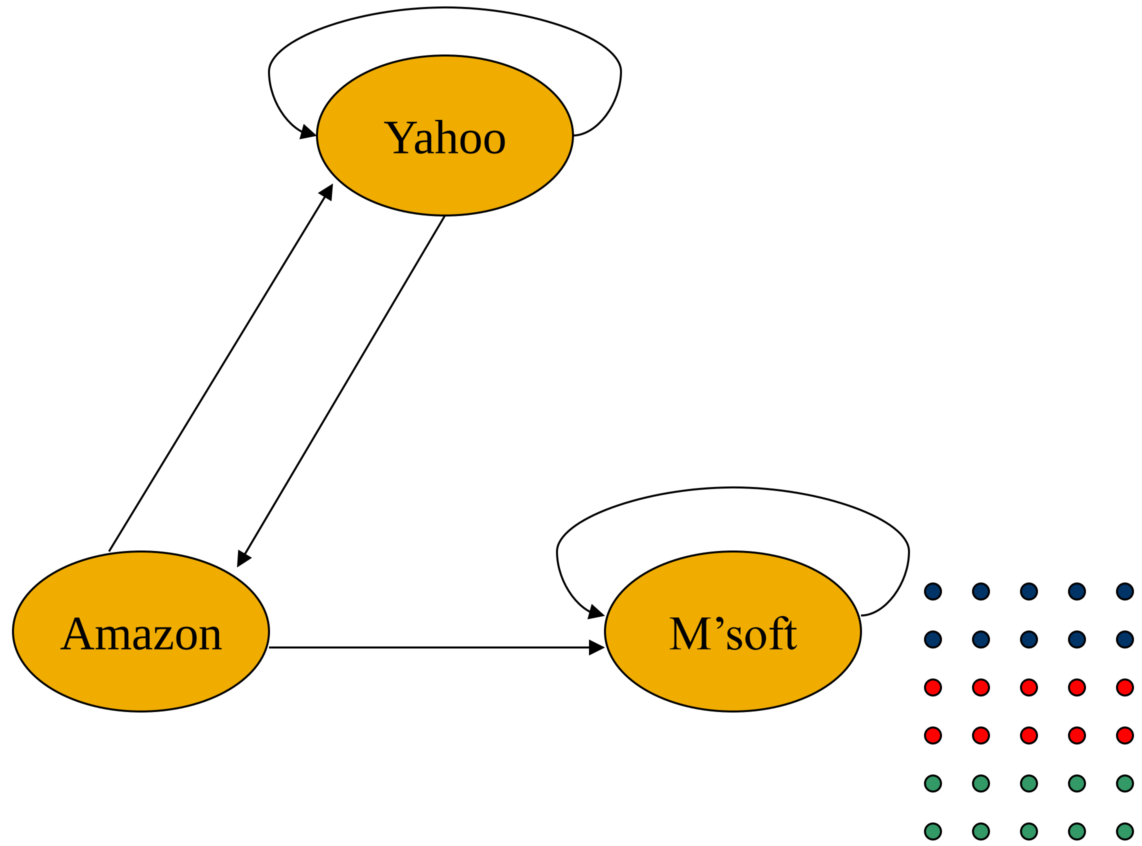
Microsoft Becomes a Spider Trap



Microsoft Becomes a Spider Trap



In the Limit …



PageRank Solution to Traps, Etc.

- "Tax" each page a fixed percentage at each interation.
- Add a fixed constant to all pages.
  - Good idea: distribute the tax, plus whatever is lost in dead-ends, equally to all pages.
- 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.

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 Page Rank

- Goal: Evaluate Web pages not just according to their popularity, but 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 [SAS] wants different pages depending on whether you are interested in travel or technology.

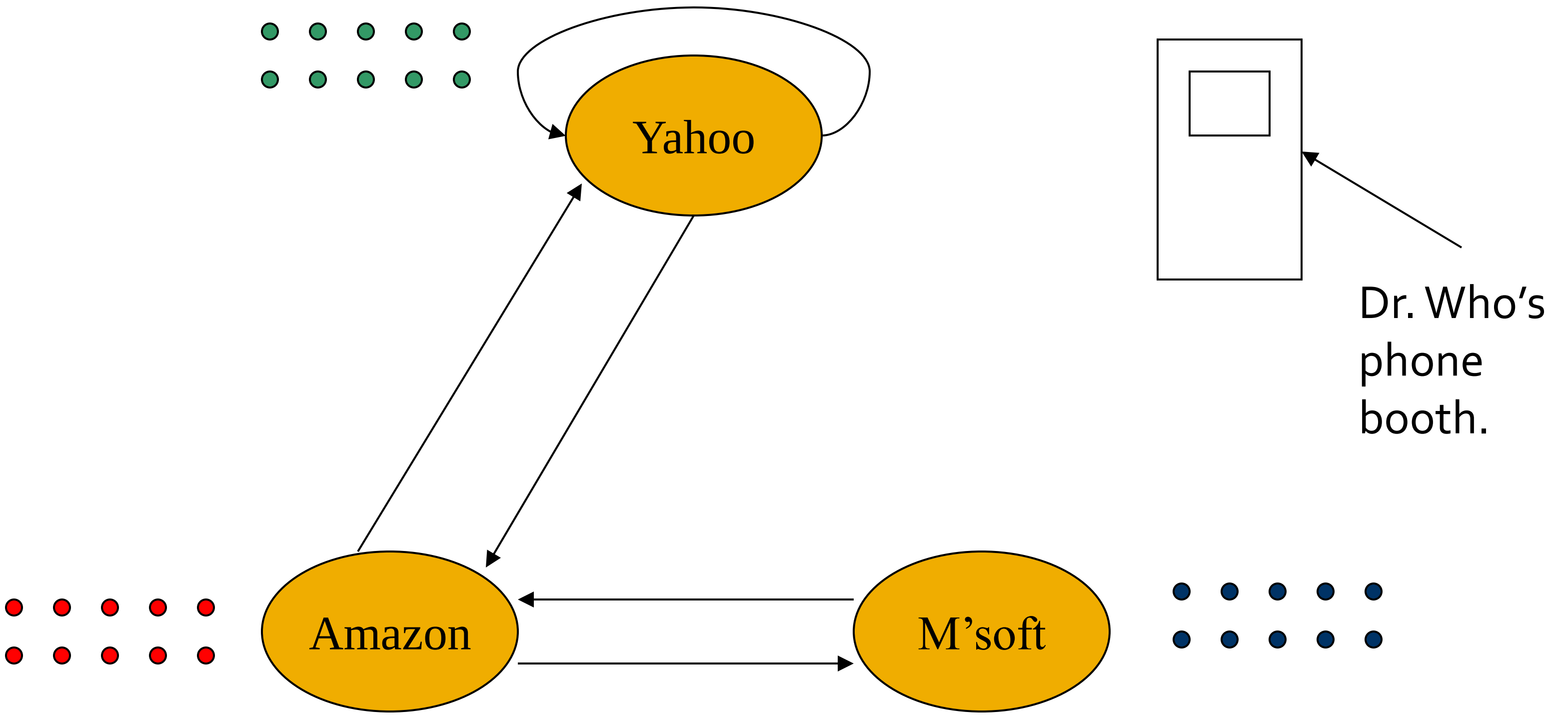
Teleport Sets

- 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.

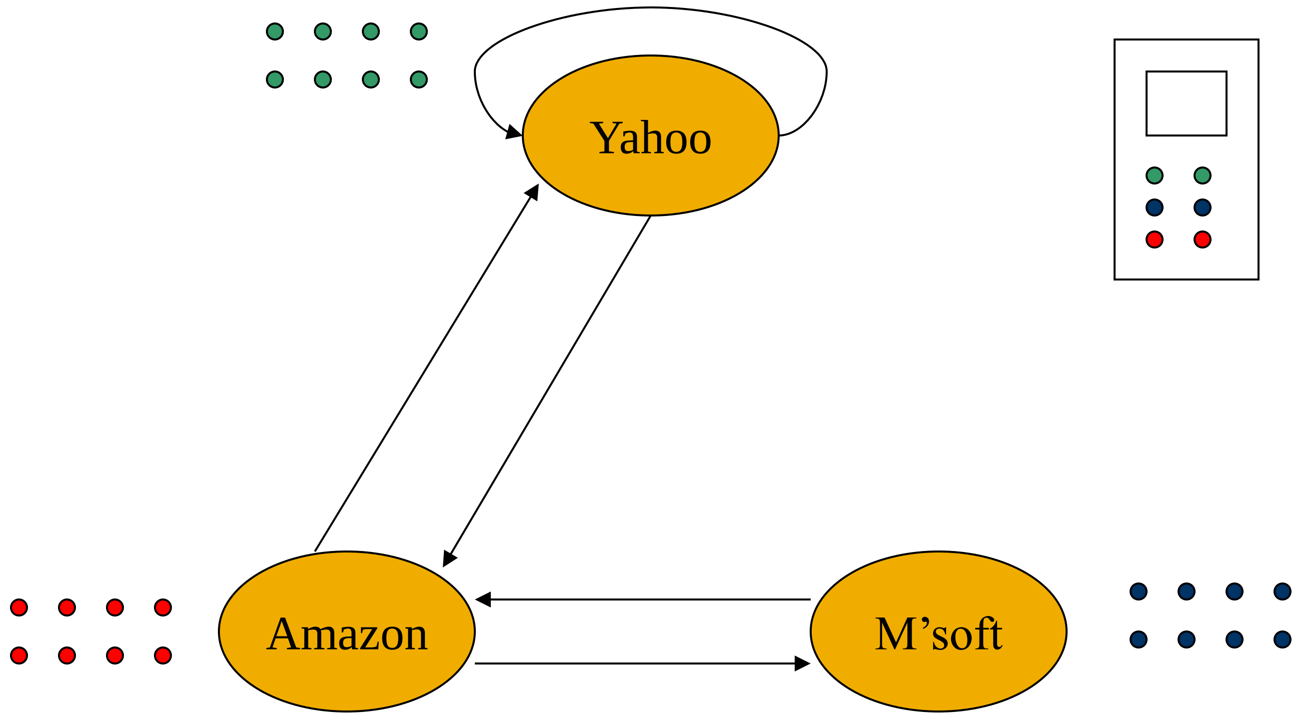
Example: Topic = Software

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

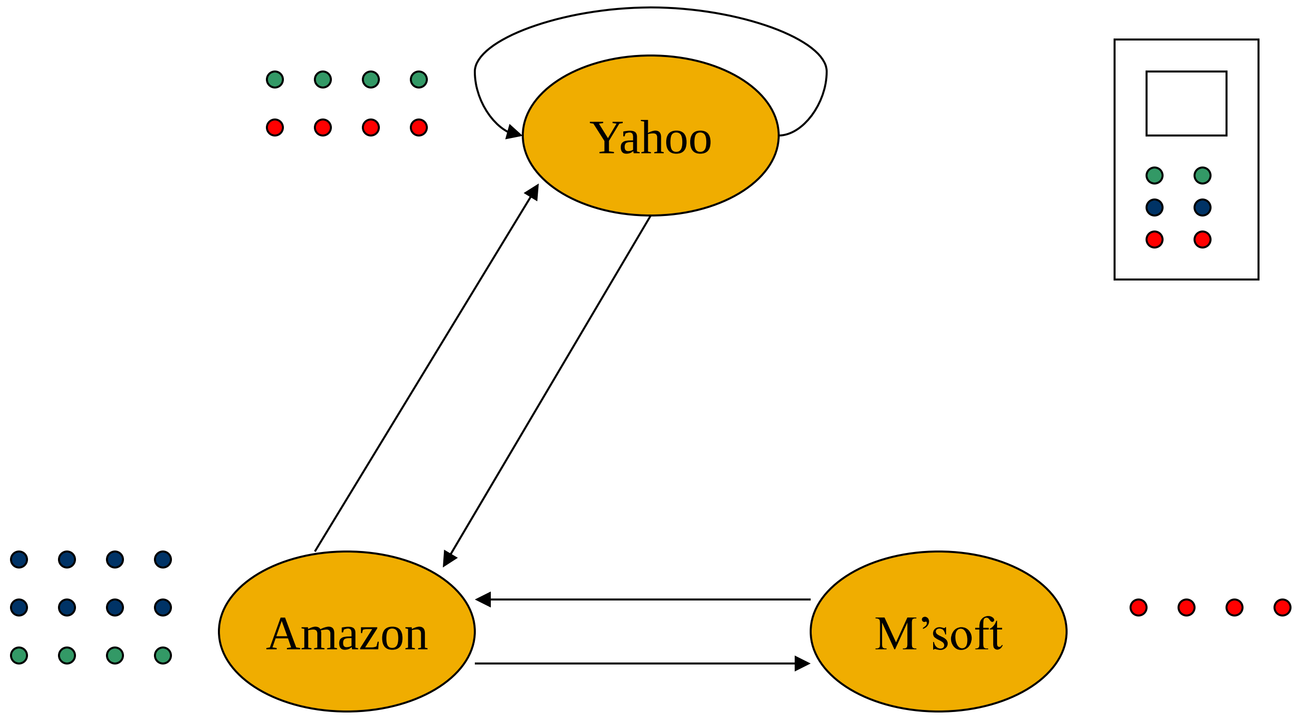
Only Microsoft in Teleport Set



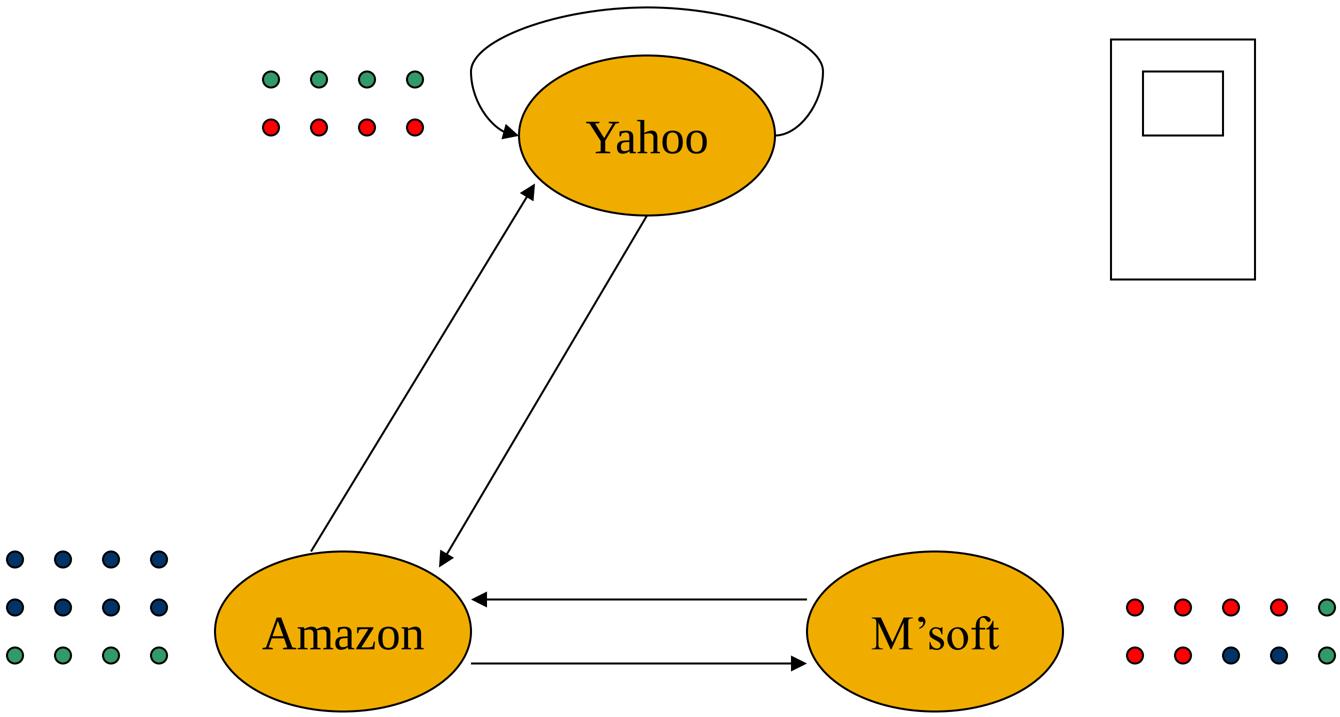
Only Microsoft in Teleport Set



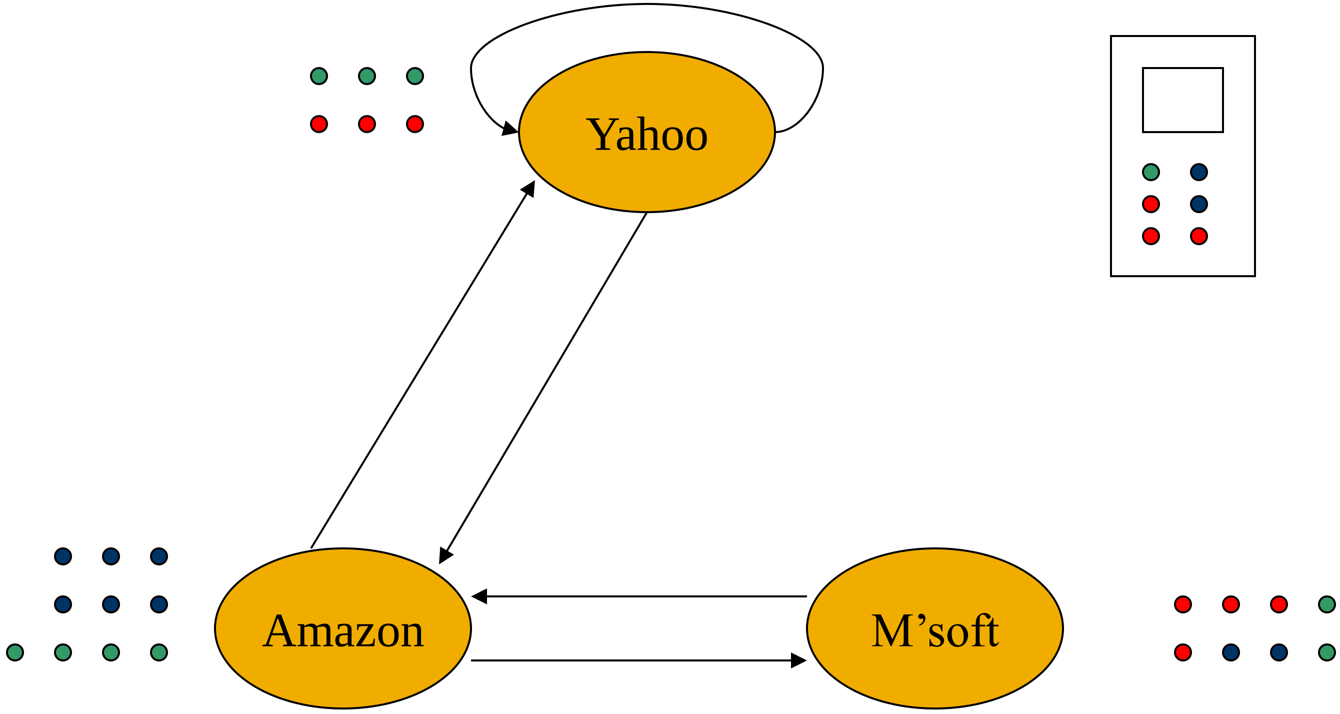
Only Microsoft in Teleport Set



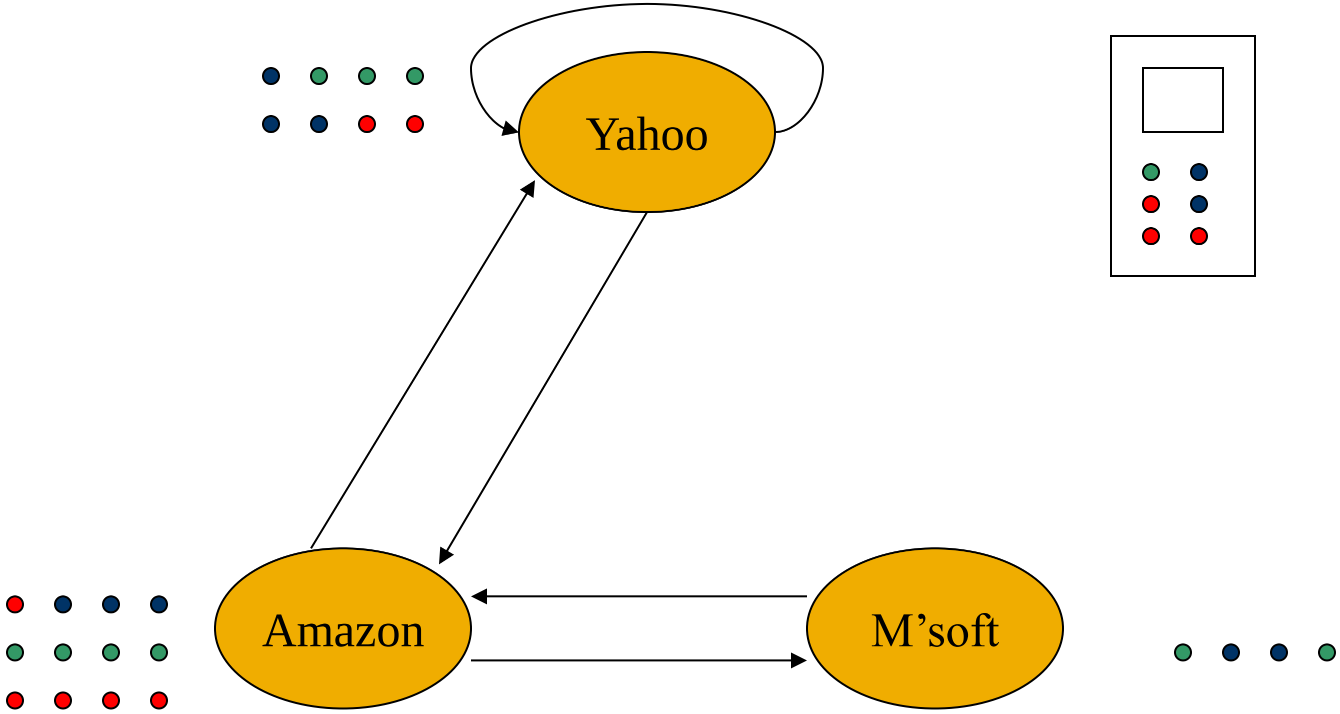
Only Microsoft in Teleport Set



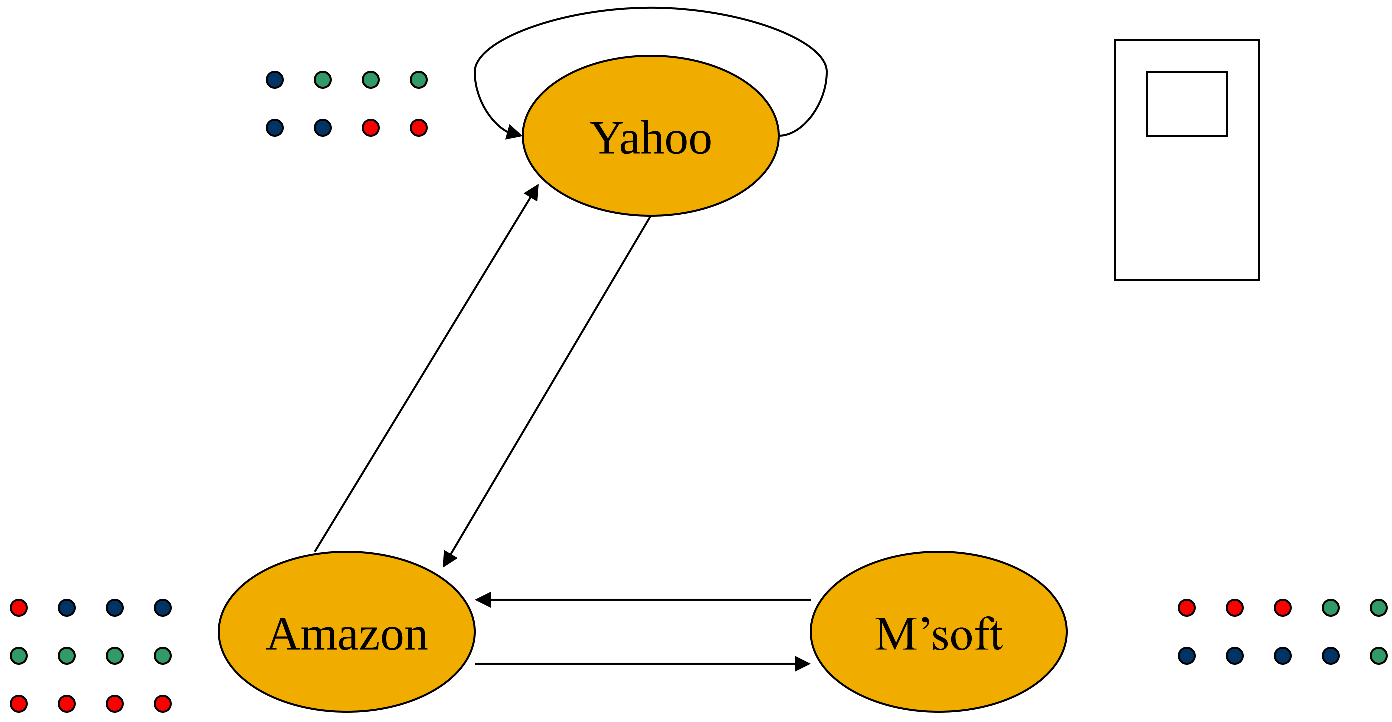
Only Microsoft in Teleport Set



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 examples 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.

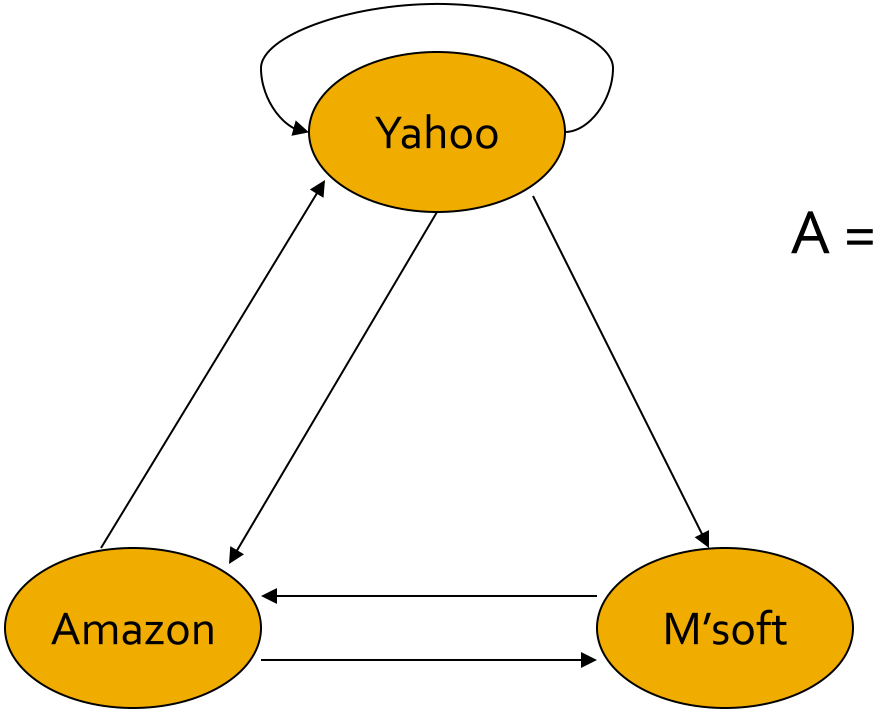
Hubs and Authorities ("HITS")

- Mutually recursive definition:
  - A *hub* links to many authorities;
  - An *authority* is linked to by many hubs.
- Authorities turn out to be places where information can be found.
  - Example: course home pages.
- Hubs tell where the authorities are.
  - Example: Departmental course-listing page.

Transition Matrix A

- HITS uses a matrix $A[i, j] = 1$ if page i links to page j, 0 if not.
- A^T, the transpose of A, is similar to the PageRank matrix M, but A^T has 1's where M has fractions.

Example: H&A Transition Matrix



A =

	y	a	m
y	1	1	1
a	1	0	1
m	0	1	0

Using Matrix A for HITS

- Powers of A and A^T have elements of exponential size, so we need scale factors.
- Let **h** and **a** be vectors measuring the "hubbiness" and authority of each page.
- Equations: $\mathbf{h} = \lambda A\mathbf{a}$; $\mathbf{a} = \mu A^T\mathbf{h}$.
  - Hubbiness = scaled sum of authorities of successor pages (out-links).
  - Authority = scaled sum of hubbiness of predecessor pages (in-links).

Consequences of Basic Equations

- From **h** = λA**a**; **a** = μA^T **h** we can derive:
  - **h** = $\lambda\mu AA^T$ **h**
  - **a** = $\lambda\mu A^TA$ **a**
- Compute **h** and **a** by iteration, assuming initially each page has one unit of hubbiness and one unit of authority.
  - Pick an appropriate value of λμ.

Example: Iterating H&A

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad AA^T = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad A^TA = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

a(yahoo)	=	1	5	24	114	. . .	$1+\sqrt{3}$
a(amazon)	=	1	4	18	84	. . .	2
a(m'soft)	=	1	5	24	114	. . .	$1+\sqrt{3}$

h(yahoo)	=	1	6	28	132	. . .	1.000
h(amazon)	=	1	4	20	96	. . .	0.735
h(microsoft)	=	1	2	8	36	. . .	0.268

Solving HITS in Practice

- Iterate as for PageRank; don't try to solve equations.
- But keep components within bounds.
  - Example: scale to keep the largest component of the vector at 1.
- Trick: start with **h** = [1,1,…,1]; multiply by A^T to get first **a**; scale, then multiply by A to get next **h**,…

Solving HITS – (2)

- You may be tempted to compute AA^T and A^TA first, then iterate these matrices as for PageRank.
- Bad, because these matrices are not nearly as sparse as A and A^T.

Link Spam

- PageRank prevents spammers from using *term spam* (faking the content of their page by adding invisible words) to fool a search engine.
- Spammers now attempt to fool PageRank by *link spam* by creating structures on the Web, called *spam farms*, that increase the PageRank of undeserving pages.

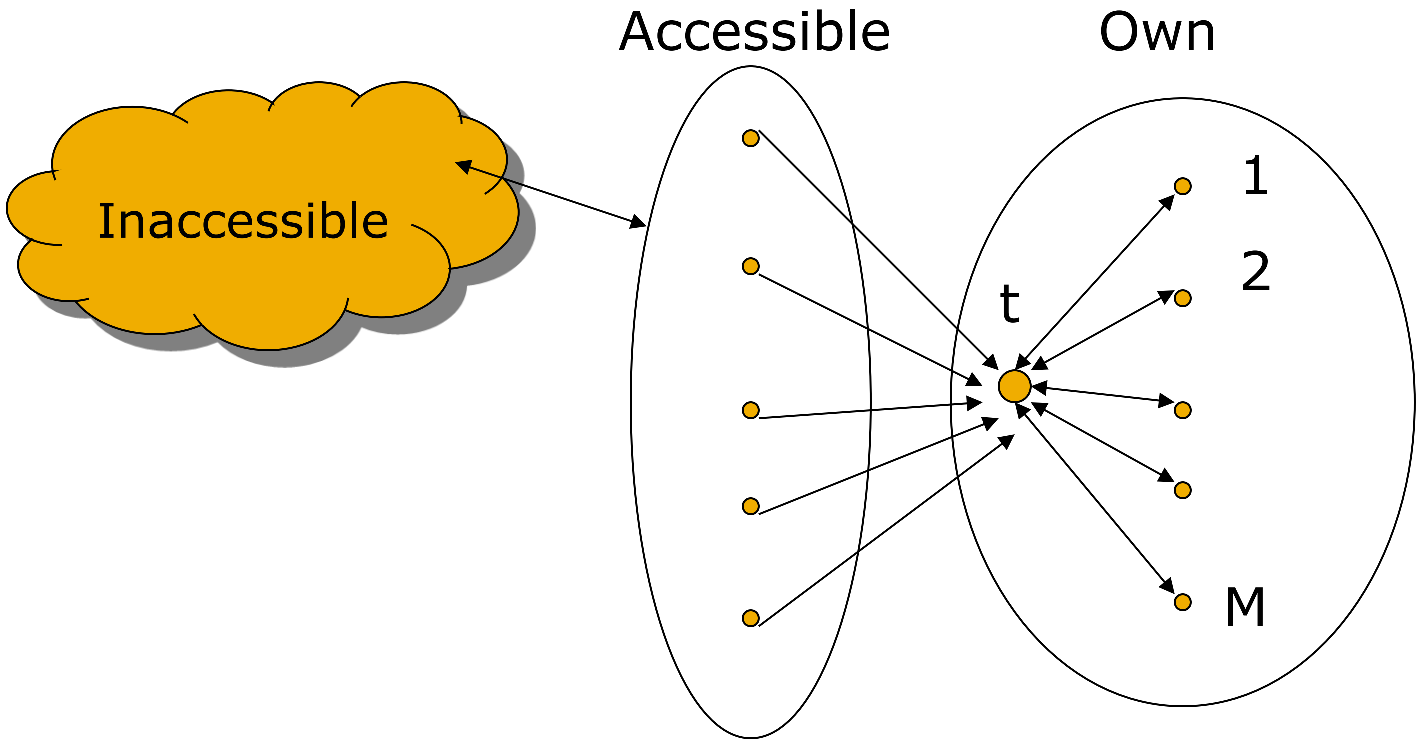
Building a Spam Farm

- Three kinds of Web pages from a spammer's point of view:
  1. *Own pages*.
     - Completely controlled by spammer.
  2. *Accessible pages*.
     - E.g., Web-log comment pages: spammer can post links to his pages.
  3. *Inaccessible pages*.

Spam Farms – (2)

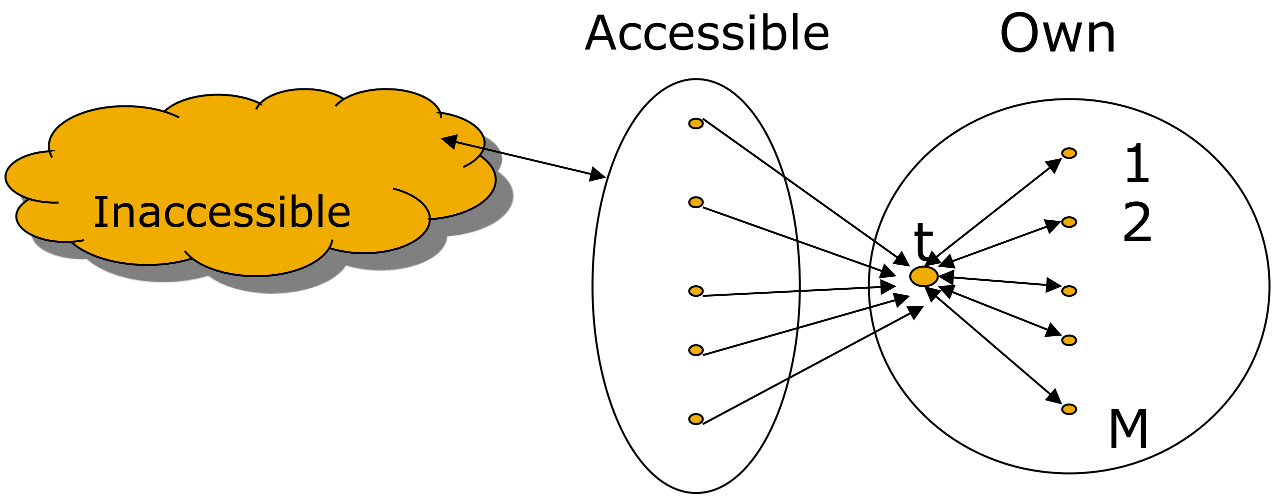
- Spammer's goal:
  - Maximize the PageRank of target page *t*.
- Technique:
  1. Get as many links from accessible pages as possible to target page *t*.
  2. Construct "link farm" to get PageRank multiplier effect.

Spam Farms – (3)



Goal: boost PageRank of page *t*.
One of the most common and effective organizations for a spam farm.

Analysis



Suppose rank from accessible pages = x.

PageRank of target page = y.

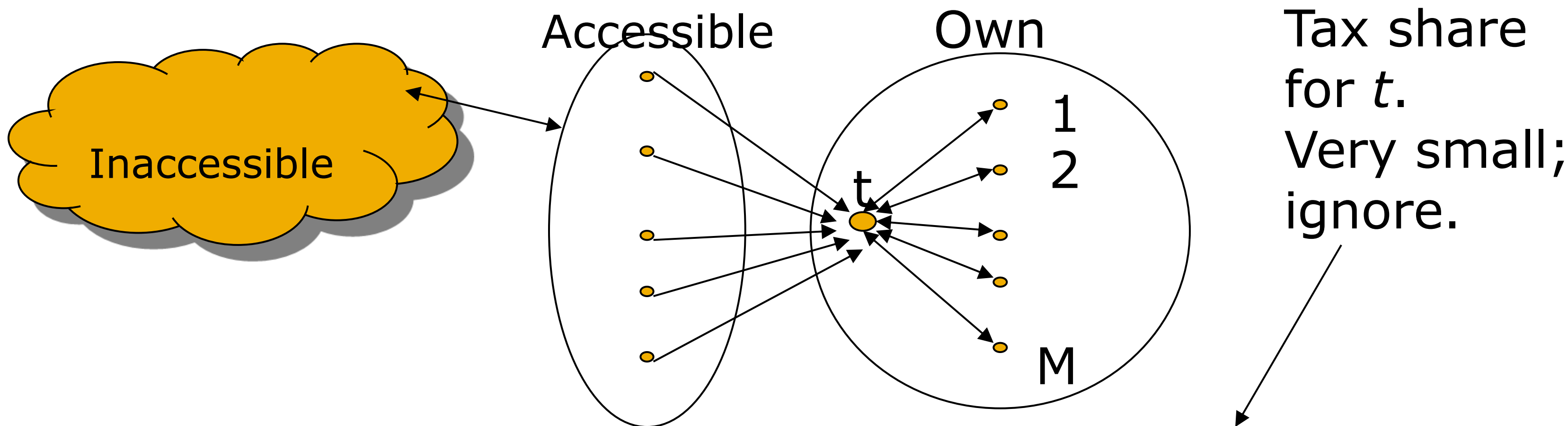
Taxation rate = $1-\beta$.

Rank of each "farm" page = $\beta y/M + (1-\beta)/N$.

Share of "tax"; N = size of the Web

From t; M = number of farm pages

Analysis – (2)



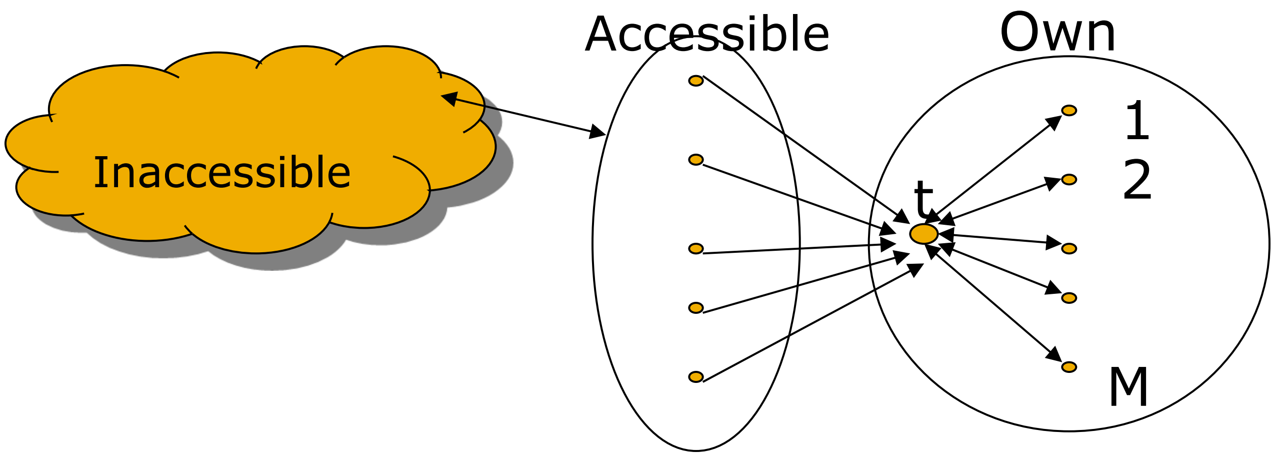
$$y = x + \beta M[\beta y/M + (1-\beta)/N] + (1-\beta)/N$$

$$y = x + \beta^2 y + \beta(1-\beta)M/N$$

$$y = x/(1-\beta^2) + cM/N \text{ where } c = \beta/(1+\beta)$$

PageRank of each "farm" page

Analysis – (3)



- $y = x/(1-\beta^2) + cM/N$ where $c = \beta/(1+\beta)$.
- For $\beta = 0.85$, $1/(1-\beta^2)= 3.6$.
  - Multiplier effect for "acquired" page rank.
- By making M large, we can make *y* as large as we want.

Detecting Link Spam

- Topic-specific PageRank, with a set of “trusted” pages as the teleport set is called *TrustRank*.
- *Spam Mass* = (PageRank – TrustRank)/PageRank.
  - High spam mass means most of your PageRank comes from untrusted sources – you may be link-spam.

Picking the Trusted Set

- Two conflicting considerations:
  - Human has to inspect each seed page, so seed set must be as small as possible.
  - Must ensure every “good page” gets adequate TrustRank, so all good pages should be reachable from the trusted set by short paths.

Approaches to Picking the Trusted Set

1. Pick the top *k* pages by PageRank.
   - It is almost impossible to get a spam page to the very top of the PageRank order.
2. Pick the home pages of universities.
   - Domains like .edu are controlled.