

# Computer Communication Networks

## CSI 416/516

### Lecture 12: Wireless Routing

Stephen F. Bush

GE Global Research

December 1, 2009

# Computer Communication Networks CSI 416/516

**Instructor:** Dr. Stephen F. Bush, GE Global Research  
**Phone:** 387-6827  
**Email:** <mailto:bushsf@research.ge.com>  
**Office Hours:** Tue 10:05pm HU-132 and by appointment  
**Course Website:** <http://www.cs.albany.edu/~bushsf>

# Syllabus

Week	Topics	Readings	Notes
1	Intro to Networking Layered Architectures	L1 B1 N1 L2 B2.1-2.2 N4	
2	Layering Sockets	L2 B2.3	HW 1 Assigned
3	Information Theory Compression & Correction	N6 L3.9	HW 1 due HW 2 Assigned
4	Networks The Physical Layer	L7 L3, L12.1-12.3, N2	
5	Multiplexing and Switching The Telephone Network	L7.3 L4	
6	Active Networks The Active Network Framework	B3	<b>Proposals Due</b>
7	Mid Term Exam Covers Weeks 1-6		

N Nanonetworks, Bush, S. F., Artech House, 2010.

B Active Networks and Active Network Management: A Proactive Management Framework by Stephen F. Bush and Amit Kulkarni, Kluwer Academic/Plenum Publishers, New York, Boston, Dordrecht, London, Moscow, 2001, 196 pp. Hardbound, ISBN 0-306-46560-4.

L Communications Networks: Fundamental Concepts and Key Architectures, Leon-Garcia and Widjaja, McGraw Hill, 2003.

# Syllabus

Week	Topics	Readings	Notes
8	Queuing Theory	L App A	HW2 Due
	Poisson & Exponential		
9	Network Simulation	N7	HW3 Assigned
	NS-2		
10	Peer to Peer	L5, N3	
	ARQ and Flow Control		
11	Multiple Access	L6.1-6.2	HW3 Due
	Ethernet	L6.6-6.11	HW4 Assigned
12	Scheduling	L6.3-6.4	
	QoS		
13	Routing	L7	HW4 Due
	Ad Hoc Networking		
14	Sensor Networks	N2 N3	HW5 Assigned
	Overview		
15	Student Presentation	L App B B4-6	
	Student Presentations		
16	Network Management		HW5 Due
	Final Exam and Vacation!		

N Nanonetworks, Bush, S. F., Artech House, 2010.

B Active Networks and Active Network Management: A Proactive Management Framework by Stephen F. Bush and Amit Kulkarni, Kluwer Academic/Plenum Publishers, New York, Boston, Dordrecht, London, Moscow, 2001, 196 pp. Hardbound, ISBN 0-306-46560-4.

L Communications Networks: Fundamental Concepts and Key Architectures, Leon-Garcia and Widjaja, McGraw Hill, 2003.

# Scribe Schedule

Week	Scribe 1	Scribe 2	Scribe 3
1 (Intro)	Caldara, Logan Nathan	Cagan, Ferhat	
2 (Layering)	Crisafulli, Anthony Nicholas	Erbatur, Serdar	
3 (Info Theory)	Krishnaraj Ravindranathan		
4 (Phy Layer)	Kudlack, Edward A		
5 (Switching)	Zhang, Xing		
6 (Active Networks)	Krishnaraj Ravindranathan		
7 (Midterm)			
8 (Queuing Theory)	Caldara, Logan Nathan		
9 (ns-2)	Cagan, Ferhat		
10 (ARQ)	Kudlack, Edward A		
11 (Multiple Access)	Erbatur, Serdar		
12 (Sched+Routing)	Zhang, Xing		
13 (Proj Pres)	Subramaniam, Sathiyaseelan		
14 (Routing)	Caldara, Logan Nathan		

Presentation schedule

Nanoscale Quantum Networks

Neural Networks

REDUCING REDUNDANCY TO INCREASE ENTROPY

Motion Prediction for Ad hoc Wireless Network

Subramaniam, Sathiyas Project

## Part I

# Presentations

## Lecture Outline

- 1 Presentation schedule
- 2 Nanoscale Quantum Networks
- 3 Neural Networks
- 4 REDUCING REDUNDANCY TO INCREASE ENTROPY
- 5 Motion Prediction for Ad hoc Wireless Network
- 6 Subramaniam, Sathiya's Project

## Reminder: Project 3 Presentation Project and Final Exam

- Project 3 final results due no later than **Dec 15 BEFORE the final exam**
- Email Project 3 simulation code and L<sup>A</sup>T<sub>E</sub>X report to bushsf@research.ge.com
- Final exam: Tuesday 15-Dec 8:00pm - 10:00pm (same room)

## iClicker scores

Total scores to date

ID	Score
000958196	44
000349384	66
000839158	43
001994700	49
000967279	26
001004365	48
000815859	39

- Average: 45
- Although a small component, final grades correlate well with these scores
- If you are below 45 points, you may be in serious trouble

# Project Presentation Schedule

... let's see what you have done

Name	Project Title
Erbatur, Sedar	Nanoscale Quantum Networks
Cagan, Ferhat	
Kudlack, Edward	Neural Networks
Ravindranathan, Krish	REDUCING REDUNDANCY TO INCREASE ENTROPY
Subramaniam, Sathiya	?
Zhang, Xing	Motion Prediction for Ad hoc Wireless Network

# Nanoscale Quantum Networks

## i>Clicker

- Is there a clear hypothesis; is it experimentally verified?
  - A Hypothesis is unclear
  - B Clear hypothesis but plan to verify it is poorly explained or poorly designed
  - C Clear hypothesis and reasonable test plan

## i>Clicker

- Is the hypothesis original?
  - A Hypothesis is highly innovative and novel
  - B Hypothesis is has some minor originality
  - C Hypothesis is poorly explained and/or is already well-known
  - D Hypothesis copied from existing work (borderline plagiarism)

# Nanoscale Quantum Networks

## i>Clicker

- Is there enough analytical work in this project?
  - A This project is very quantitative; sufficient rigorous analysis
  - B This project is sufficiently rigorous
  - C This project is lacking quantitative analysis
  - D This project is pure fluff; no demonstrated analysis

## i>Clicker

- I wager that this project will have the following timeliness and grade:
  - A This project is ahead of schedule and exceeds minimum requirements
  - B This project is on time and will do well
  - C This project is slightly behind schedule; needs improvement
  - D This project is pure fluff; author is bluffing his way through
  - E This project is hopeless

# Neural Networks

## i>Clicker

- Is there a clear hypothesis; is it experimentally verified?
  - A Hypothesis is unclear
  - B Clear hypothesis but plan to verify it is poorly explained or poorly designed
  - C Clear hypothesis and reasonable test plan

## i>Clicker

- Is the hypothesis original?
  - A Hypothesis is highly innovative and novel
  - B Hypothesis is has some minor originality
  - C Hypothesis is poorly explained and/or is already well-known
  - D Hypothesis copied from existing work (borderline plagiarism)

# Neural Networks

## i>Clicker

- Is there enough analytical work in this project?
  - A This project is very quantitative; sufficient rigorous analysis
  - B This project is sufficiently rigorous
  - C This project is lacking quantitative analysis
  - D This project is pure fluff; no demonstrated analysis

## i>Clicker

- I wager that this project will have the following timeliness and grade:
  - A This project is ahead of schedule and exceeds minimum requirements
  - B This project is on time and will do well
  - C This project is slightly behind schedule; needs improvement
  - D This project is pure fluff; author is bluffing his way through
  - E This project is hopeless

# REDUCING REDUNDANCY TO INCREASE ENTROPY

## i>Clicker

- Is there a clear hypothesis; is it experimentally verified?
  - A Hypothesis is unclear
  - B Clear hypothesis but plan to verify it is poorly explained or poorly designed
  - C Clear hypothesis and reasonable test plan

## i>Clicker

- Is the hypothesis original?
  - A Hypothesis is highly innovative and novel
  - B Hypothesis is has some minor originality
  - C Hypothesis is poorly explained and/or is already well-known
  - D Hypothesis copied from existing work (borderline plagiarism)

# REDUCING REDUNDANCY TO INCREASE ENTROPY

## i>Clicker

- Is there enough analytical work in this project?
  - A This project is very quantitative; sufficient rigorous analysis
  - B This project is sufficiently rigorous
  - C This project is lacking quantitative analysis
  - D This project is pure fluff; no demonstrated analysis

## i>Clicker

- I wager that this project will have the following timeliness and grade:
  - A This project is ahead of schedule and exceeds minimum requirements
  - B This project is on time and will do well
  - C This project is slightly behind schedule; needs improvement
  - D This project is pure fluff; author is to bluffing his way through

# Motion Prediction for Ad hoc Wireless Network

## i>Clicker

- Is there a clear hypothesis; is it experimentally verified?
  - A Hypothesis is unclear
  - B Clear hypothesis but plan to verify it is poorly explained or poorly designed
  - C Clear hypothesis and reasonable test plan

## i>Clicker

- Is the hypothesis original?
  - A Hypothesis is highly innovative and novel
  - B Hypothesis is has some minor originality
  - C Hypothesis is poorly explained and/or is already well-known
  - D Hypothesis copied from existing work (borderline plagiarism)

# Motion Prediction for Ad hoc Wireless Network

## i>Clicker

- Is there enough analytical work in this project?
  - A This project is very quantitative; sufficient rigorous analysis
  - B This project is sufficiently rigorous
  - C This project is lacking quantitative analysis
  - D This project is pure fluff; no demonstrated analysis

## i>Clicker

- I wager that this project will have the following timeliness and grade:
  - A This project is ahead of schedule and exceeds minimum requirements
  - B This project is on time and will do well
  - C This project is slightly behind schedule; needs improvement
  - D This project is pure fluff; author is bluffing his way through
  - E This project is hopeless

## Subramaniam, Sathiya's Project

### i>Clicker

- Is there a clear hypothesis; is it experimentally verified?
  - A Hypothesis is unclear
  - B Clear hypothesis but plan to verify it is poorly explained or poorly designed
  - C Clear hypothesis and reasonable test plan

### i>Clicker

- Is the hypothesis original?
  - A Hypothesis is highly innovative and novel
  - B Hypothesis is has some minor originality
  - C Hypothesis is poorly explained and/or is already well-known
  - D Hypothesis copied from existing work (borderline plagiarism)

## Subramaniam, Sathiya's Project

### i>Clicker

- Is there enough analytical work in this project?
  - A This project is very quantitative; sufficient rigorous analysis
  - B This project is sufficiently rigorous
  - C This project is lacking quantitative analysis
  - D This project is pure fluff; no demonstrated analysis

### i>Clicker

- I wager that this project will have the following timeliness and grade:
  - A This project is ahead of schedule and exceeds minimum requirements
  - B This project is on time and will do well
  - C This project is slightly behind schedule; needs improvement
  - D This project is pure fluff; author is bluffing his way through
  - E This project is hopeless

## Part II

# Routing

## Lecture Outline

- 7 Fundamental Routing Approaches
  - Distance-vector
  - Link-state
- 8 Choosing link costs
  - Static metrics
  - Dynamic metrics
  - Hierarchical routing
- 9 Internet routing protocols
  - Multicast routing
- 10 Routing for mobile hosts
  - OLSR
  - AODV
- 11 Ad hoc routing and information theory

# What is wireless routing?

Enabling a path...

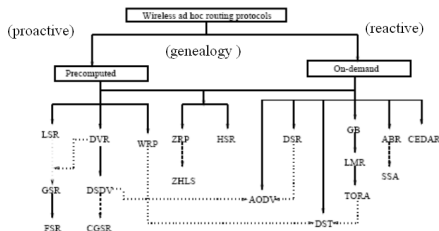
- Process of finding a path from a source to every destination in the network
- Suppose you want to connect to Antarctica from your desktop
  - What route should you take?
  - Does a shorter route exist?
  - What if a link along the route goes down?
  - What if you're on a mobile wireless link?
- Routing deals with these types of issues
- Simple Routing Example – [Click here for video \(basic operation\)](#)

# Too many routing protocols... are we missing something?

A growing alphabet soup of protocols... indicates fundamentals not understood yet

- More mathematical treatment can be found in:

- “A Simple Metric for Ad Hoc Network Adaptation,” Bush N. F., IEEE JSAC, vol. 23, no. 12, Dec 2005 ([click to see paper](#)) [2]
- “Flash Routes” in “The Limits of Motion Prediction Support for Ad hoc Wireless Network Performance,” Bush S. F. and Smith, N., ICWN-05 ([click to see paper](#)) [4]



(from "Routing Techniques in Wireless Ad Hoc Networks: Classification and Comparison," Mikael Zor, Bryan Ramanathan and Srinivas Madhwaraj)

## Two main routing categories

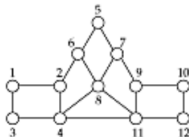
Flooding vs many local exchanges...

- Link state
  - Routers flood their link state updates
  - Every router can generate its own topology of the network (robust, but overhead)
- Distance vector
  - Routers maintain only NEXT HOP along shortest path to destination
  - Many local exchanges rather than flooding (lower overhead, but convergence and cycles can be a problem)

## Basics

It comes down to making local decision...

- A routing protocol sets up a routing table in routers and switch controllers
- A node makes a local choice depending on global topology: this is the fundamental goal



ROUTING TABLE AT 1

Destination	Next hop	Destination	Next hop
1	—	7	2
2	2□	8□	2□
3	3□	9□	2□
4	3□	10□	2□
5	2□	11□	3□
6	2	12	3

## Key Problem

Making local decisions to achieve global results... efficiently

- How to make correct local decisions?
  - Each router must know *something* about global state
- Global state
  - Inherently large
  - Dynamic
  - Hard to collect
- *A routing protocol must intelligently summarize relevant information*

## Requirements

Reduce overhead, as always

- Minimize routing table space
  - Fast to look up
  - Less to exchange
- Minimize number and frequency of control messages
- Robustness: avoid
  - Black holes
  - Loops
  - Oscillations
- Use optimal path

## Choices

- Centralized vs. Distributed routing
  - Centralized is simpler, but prone to failure and congestion
- Source-based vs. Hop-by-hop
  - How much is in packet header?
  - Intermediate: *loose source route*
- Stochastic vs. Deterministic
  - Stochastic spreads load, avoiding oscillations, but misorders
- Single vs. Multiple path
  - Primary and alternative paths (compare with stochastic)
- State-dependent vs. State-independent
  - Do routes depend on current network state (e.g. delay)?

# Distance Vector Routing

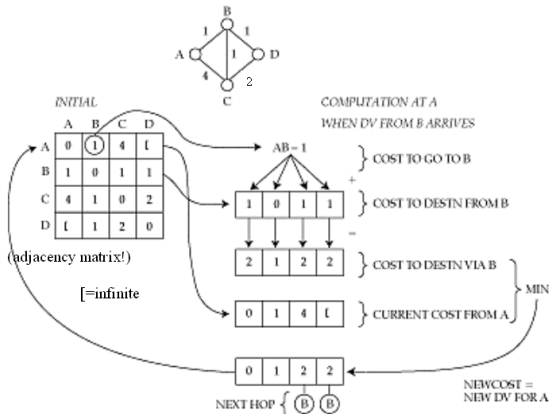
- Environment
  - Links and routers unreliable
  - Alternative paths scarce
  - Traffic patterns can change rapidly
- Two key algorithms
  - Distance vector
  - Link-state
- Both assume router knows
  - Address of each neighbor
  - Cost of reaching each neighbor
- Both allow a router to determine global routing information by talking to its neighbors

## Distance Vector: Basic Idea

- Node tells its neighbors its best idea of distance to *every* other node in the network
- Node receives these *distance vectors* from its neighbors
- Updates its notion of best path to each destination, and the next hop for this destination
- Features
  - Distributed
  - Adapts to traffic changes and link failures
  - Suitable for networks with multiple administrative entities

# Example

## Constructing shortest routes from A



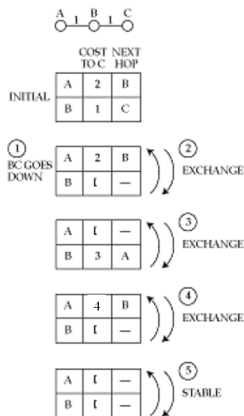
## Distance Vector: Why does it work?

- Each node knows its true cost to its neighbors
- This information is spread to its neighbors the first time it sends out its distance vector
- Each subsequent dissemination spreads the truth one hop
- Eventually, it is incorporated into routing tables everywhere in the network
- Proof: Bellman and Ford, 1957
- Distance vector – [Click here for video \(basic operation\)](#)

# Problems with distance vector

## Count to infinity...

- BC link goes down
- Before B could send this to A, B receives update from A which says (A,C),2
- B updates its table with (B,C) 3
- On next update, B sends this to A
- A updates its distance to C as (A,C), 4, B
- Repeat *ad infinitum*
- Distance vector – [Click here for video \(basic\)](#)



## Dealing with the problem

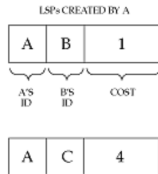
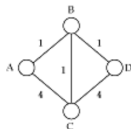
- Path vector
  - DV carries path to reach each destination
- Split horizon
  - Never tell neighbor cost to X if neighbor is next hop to X
  - Doesn't work for 3-way count to infinity
- Triggered updates
  - Exchange routes on link change, instead of on timer
  - Faster count up to infinity
- More complicated
  - Source tracing
  - DUAL (Distributed Update Algorithm)
    - Base on observation that one cannot create a loop by picking a shorter path to the destination (<http://www.cse.ucsc.edu/ccrg/publications/jj.dual.ton93.pdf>)

## Link State Routing

- In distance vector, router knows only *cost* to each destination
  - Hides information, causing problems
- In link state, router knows entire network topology and computes shortest path by itself
  - Independent computation of routes
  - Potentially less robust
- Key elements
  - Topology dissemination
  - Computing shortest routes

## Link State: Topology Dissemination

- A router describes its neighbors with a *link state packet (LSP)*
- Uses *controlled flooding* to distribute this everywhere
  - Store an LSP in an *LSP database*
  - If new, forward to every interface other than incoming one
  - A network with  $E$  edges will copy at most  $2E$  times



## Sequence Numbers

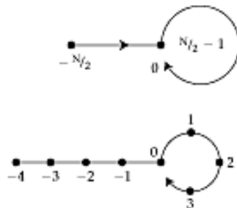
- How do we know an LSP is new?
- Use a sequence number in LSP header
- Greater sequence number is newer
- What if sequence number wraps around?
  - Smaller sequence number is now newer!
  - (Hint: Use a large sequence space)
- On boot up, what should be the initial sequence number?
  - Have to somehow purge old LSPs
  - Two solutions
    - Aging
    - Lollipop sequence space (explained later)

## Aging

- Creator of LSP puts timeout value in the header
- Router removes LSP when it times out
  - Also floods this information to the rest of the network (why?)
- So, on booting, router just has to wait for its old LSPs to be purged
- But what age to choose?
  - If too small
    - Purged before fully flooded (why?)
    - Needs frequent updates
  - If too large
    - Router waits idle for a long time on rebooting

## A Better Solution

- A better solution is to have an initial sequence number which is unique (is never going to be used later on)
- $(-N/2)$  is the oldest seq num
- A newly booted router starts with sequence number  $(-N/2)$
- When other routers get this LSP, then inform the originator with their current sequence number
- The originator then sends an LSP with sequence number higher than the other routers



## More on Lollipops

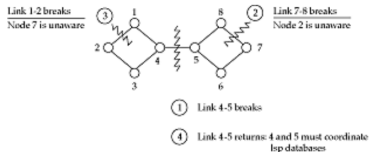
- Sequence number space is divided into three parts
  - negative space  $(-N/2)$  to 0
  - zero
  - positive space from 1 to  $(N/2) - 1$
- When a router comes up it uses seq num  $(-N/2)$  and subsequent LSPs have seq num  $(-N/2 + 1)$ ,  $(-N/2 + 2)$ ...
- When seq num becomes positive, **subsequent seq numbers wrap around only in the positive seq space** (circular part)
  - No danger of overlap in negative space

## More on Lollipops

- If a router gets an older LSP, it tells the sender about the newer LSP
- So, newly booted router quickly finds out its most recent sequence number
- It jumps to one more than that
- $-N/2$  is a *trigger* to evoke a response from community memory

## Recovering From a Partition

- On partition, LSP databases can get out of synch
- Databases described by database descriptor records
- Routers on each side of a newly restored link talk to each other to update databases (determine missing and out-of-date LSPs)



## Router Failure

- How to detect?
  - HELLO protocol
- HELLO packet may be corrupted
  - So age anyway
  - On a timeout, flood the information

## Securing LSP Databases

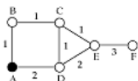
- LSP databases *must* be consistent to avoid routing loops
- Malicious agent may inject spurious LSPs
- Routers must actively protect their databases
  - Checksum LSPs
  - ack LSP exchanges
  - Passwords

# Computing Shortest Paths

- Basic idea
  - Maintain a set of nodes  $P$  to whom we know shortest path
  - Consider every node one hop away from nodes in  $P = T$
  - Find every way in which to reach a given node in  $T$ , and choose shortest one
  - Then add this node to  $P$

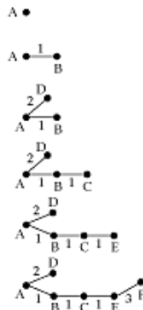
# Example

Node A builds a complete topology of the network...



PERMANENT	TEMPORARY	COMMENTS
A	B(A,1), D(A,2)	ROOT AND ITS NEIGHBORS
A, B(A,1)	D(A,2), C(B,2)	ADD C(B,2)
A, B(A,1) D(A,2)	E(D,4), C(B,2)	C(D,3) DIDN'T MAKE IT
A, B(A,1) D(A,2), C(B,2)	E(C,3)	E(D,4) TOO LONG
A, B(A,1) D(A,2), C(B,2) E(C,3)	F(E,6)	
A, B(A,1) C(B,2), D(A,2) E(C,3), F(E,6)	NULL	STOP

B(A,1) means B was reached by A, cost 1



## Link State vs. Distance Vector

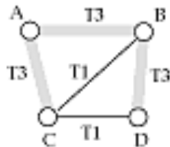
- Criteria
  - Stability
  - Multiple routing metrics
  - Convergence time after a change
  - Communication overhead
  - Memory overhead
- Both are evenly matched
- Both widely used

## Choosing Link Costs

- Shortest path based upon link costs
- Can use either static or dynamic costs
- In both cases: cost determines amount of traffic on the link
  - Reduce the cost, increase the expected traffic (economics: supply and demand)
  - If dynamic cost depends on load, can have oscillations (why?)

## Static metrics

- Simplest: Set all link costs to 1  $\rightarrow$  min hop routing
  - But 28.8 modem link is not the same as a T3!
- Give links weight proportional to capacity



WEIGHTS
T3 - 1
T1 - 10

## Dynamic Metrics

- A first cut (ARPAnet original)
- Cost proportional to length of router queue
  - Independent of link capacity
- Many problems when network is loaded
  - Queue length averaged over a small time → transient spikes caused major rerouting
  - Wide dynamic range → network completely ignored paths with high costs
  - Queue length assumed to predict future loads → opposite is true (why?)
  - No restriction on successively reported costs → oscillations
  - All tables computed simultaneously → low cost link flooded

## Modified Metrics

---

Queue length averaged over a small time  
Wide dynamic range queue  
Queue length assumed to predict future loads  
No restriction on successively reported costs  
All tables computed simultaneously

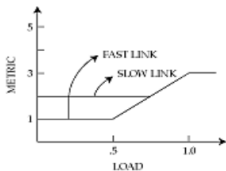
---

---

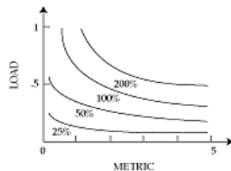
Queue length averaged over a longer time  
Dynamic range restricted  
Cost also depends on intrinsic link capacity  
Restriction on successively reported costs  
Attempt to stagger table computation

---

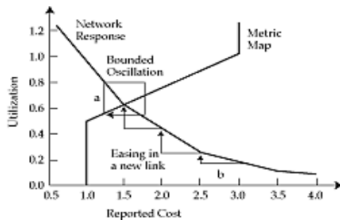
## Routing Dynamics



(a) METRIC MAP



(b) NETWORK RESPONSE MAP



Simply put: Routing metric impacts load; load impacts routing metric

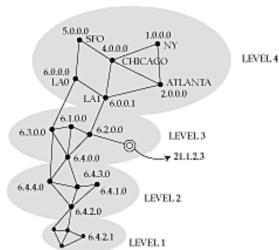
# Hierarchical Routing

- Large networks need large routing tables
  - More computation to find shortest paths
  - More bandwidth wasted on exchanging DVs and LSPs
- Solution
  - Hierarchical routing
- Key idea
  - Divide network into a set of domains
  - Gateways connect domains
  - Computers within domain unaware of outside computers
  - Gateways know only about other gateways

## Example

Four levels in domain 6.0.0.0 ...

- Features
  - Only a few routers in each level
  - Not a strict hierarchy
  - Gateways participate in multiple routing protocols
  - Non-aggregable routers increase core table space



## Hierarchy in the Internet

- Three-level hierarchy in addresses
  - Network number
  - Subnet number
  - Host number
- Core advertises routes only to networks, not to subnets
  - e.g. 135.104.\*, 192.20.225.\*
- Even so, about 80,000 networks in core routers (1996)
- Gateways talk to backbone to find best next-hop to every other network in the Internet

A core router is a router designed to operate in the Internet backbone, or core. To fulfill this role, a router must be able to support multiple telecommunications interfaces of the highest speed in use in the core Internet and must be able to forward IP packets at full speed on all of them

## External and Summary Records

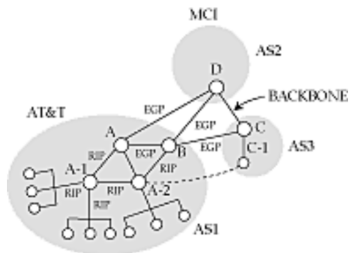
- If a domain has multiple gateways
  - *External* records tell hosts in a domain which one to pick to reach a host in an external domain
    - e.g. allows 6.4.0.0 to discover shortest path to 5.\* is through 6.0.0.0
  - *Summary* records tell backbone which gateway to use to reach an internal node
    - e.g. allows 5.0.0.0 to discover shortest path to 6.4.0.0 is through 6.0.0.0
- External and summary records contain distance from gateway to external or internal node
  - Unifies distance vector and link state algorithms

## Interior and Exterior Protocols

- Internet has three levels of routing
  - Highest is at *backbone* level, connecting *autonomous systems (AS)*
  - Next level is within AS
  - Lowest is within a LAN
- Protocol between AS gateways: exterior gateway protocol
- Protocol within AS: interior gateway protocol

## Exterior Gateway Protocol

- Between untrusted routers
  - Mutually suspicious
- Must tell a *border gateway* who can be trusted and what paths are allowed
- *Transit over backdoors* is a problem



# Common Routing Protocols

- Interior
  - RIP – Routing Information Protocol
  - OSPF – Open Shortest Path First
- Exterior
  - EGP – Exterior Gateway Protocol
  - BGP – Border Gateway Protocol
- ATM
  - PNNI – Private Network-Network Interface

## Routing Information Protocol (RIP)

- Distance vector
- Cost metric is hop count
- Infinity = 16
- Exchange distance vectors every 30s
- Split horizon
- Useful for small subnets
  - Easy to install

## Open Shortest Path First (OSPF)

- Link-state
- Uses areas to route packets hierarchically within AS
- Complex
  - LSP databases to be protected
- Uses *designated routers* to reduce number of endpoints
  - Routers elect a designated router (DR) and a backup designated router (BDR) which act as a hub to reduce traffic between routers

## Exterior Gateway Protocol (EGP)

- Original exterior gateway protocol
- Distance-vector
- Costs are either 128 (reachable) or 255 (unreachable) → reachability protocol → backbone must be loop free (why?)
- Allows administrators to pick neighbors to peer with
- Allows backdoors (by setting backdoor cost  $< 128$ )

## Border Gateway Protocol (BGP)

- Path-Vector
  - Distance vector annotated with entire path
  - Also with policy attributes
  - Guaranteed loop-free
- Can use non-tree backbone topologies
- Uses TCP to disseminate DVs
  - Reliable
  - But subject to TCP flow control
- Policies are complex to set up

## Private Network-Network Interface (PNNI)

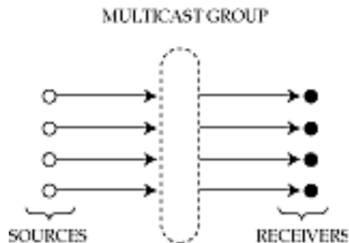
- Link-state
- Many levels of hierarchy
- Switch controllers at each level form a peer group
- Group has a group leader
- Leaders are members of the next higher level group
- Leaders summarize information about group to tell higher level peers
- All records received by leader are flooded to lower level
- LSPs can be annotated with per-link QoS metrics
- Switch controller uses this to compute source routes for call-setup packets

## Multicast Routing

- Unicast: single source sends to a single destination
- Multicast: hosts are part of a *multicast group*
  - Packet sent by *any* member of a group are received by *all*
- Useful for
  - Multiparty video conference
  - Distance learning
  - Resource location

## Multicast group

- Associates a set of senders and receivers with each other
  - But independent of them
  - Created either when a sender starts sending from a group
  - Or a receiver expresses interest in receiving
  - Even if no one else is there!
- Sender does not need to know receivers' identities –  
*Rendezvous point*

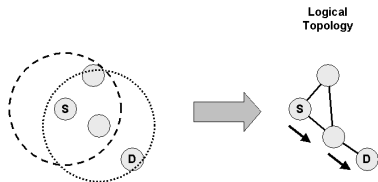


## Addressing

- Multicast group in the internet has its own class D address
  - Looks like a host address, but isn't
- Senders send to the address
- Receivers anywhere in the world request packets from that address
- “Magic” is in associating the two: *dynamic directory service*
- Four problems
  - Which groups are currently active
  - How to express interest in joining a group
  - Discovering the set of receivers in a group
  - Delivering data to members of a group

## MANETs

- A mobile ad hoc network (MANET) is characterized by...
  - Multi-hop routing so that nodes not directly connected at layer 2 can communicate through layer 3 routing
  - Wireless links
  - Mobile nodes



## MANET vs. Traditional Routing (1)

- Every node is potentially a router in a MANET, while most nodes in traditional wired networks do not route packets
  - Nodes transmit and receive their own packets and, also, forward packets for other nodes
- Topologies are dynamic in MANETs due to mobile nodes, but are relatively static in traditional networks
- Routing in MANETs must consider both layer 3 and layer 2 information, while traditional protocols rely on layer 3 information only
  - Link layer information can indicate connectivity and interference

## MANET vs. Traditional Routing (2)

- MANET topologies tend to have many more redundant links than traditional networks
- A MANET “router” typically has a single interface, while a traditional router has an interface for each network to which it connects
  - Routed packet sent forward when transmitted, but also sent to previous transmitter
- Channel properties, including capacity and error rates, are relatively static in traditional networks, but may vary in MANETs

## MANET vs. Traditional Routing (3)

- Interference is an issue in MANETs, but not in traditional networks
  - For example, a forwarded packet from B-to-C competes with new packets sent from A-to-B
- Channels asymmetric with some layer 2 technologies
  - IEEE 802.11 MAC assumes symmetric channels
- Power efficiency is an issue in MANETs, while it is normally not an issue in traditional networks
- MANETs may have gateways to fixed network, but are typically “stub networks,” while traditional networks can be stub networks or transit networks

## MANET vs. Traditional Routing (4)

- There is limited physical security in a MANET compared to a traditional network
  - Increased possibility of eavesdropping, spoofing, and denial-of-security attacks
- Traditional routing protocols for wired networks do not work well in most MANETs
  - MANETs are too dynamic
  - Wireless links present problems of interference, limited capacity, etc...

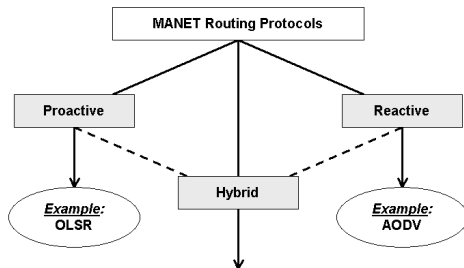
# MANET Routing

- Nodes must determine how to forward packets
  - Source routing: routing decision is made at the sender
  - Hop-by-hop routing: routing decision is made at each intermediate node
- Difficult to achieve good performance
  - Routes change over time due to node mobility
  - Best to avoid long delays when first sending packets
  - Best to reduce overhead of route discovery and maintenance
  - Want to involve as many nodes as possible – to find better paths and reduce likelihood of partitions

## MANET Routing Approaches

- Decision time
  - Proactive or table-driven – maintain routing tables
  - Reactive or on-demand – determine routing on an as-needed basis
- Network structure
  - Hierarchical – impose a hierarchy on a collection of nodes and reflect this hierarchy in the routing algorithm
    - May use a proactive protocol for routing within a cluster or zone
    - May use a reactive protocol for routing between distinguished “cluster heads”
  - Non-hierarchical – make decisions among all nodes

## Types of MANET Routing



(Optimized Link State Routing / Ad hoc On-Demand Distance Vector)

## Common Features

- MANET routing protocols must...
  - Discover a path from source to destination
  - Maintain that path (e.g., if an intermediate node moves and breaks the path)
  - Define mechanisms to exchange routing information
- Reactive protocols
  - Discover a path when a packet needs to be transmitted and no known path exists
  - Attempt to alter the path when a routing failure occurs
- Proactive protocols
  - Find paths, in advance, for all source-pair destinations
  - Periodically exchange routing information to maintain paths

## IETF MANET Working Group (1)

- <http://www.ietf.org/html.charters/manet-charter.html>
- “The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies. The fundamental design issues are that the wireless link interfaces have some unique routing interface characteristics and that node topologies within a wireless routing region may experience increased dynamics, due to motion or other factors.”

## Optimized Link State Routing (OLSR) Concepts (1)

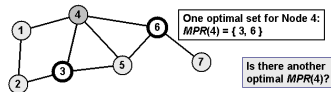
- Proactive (table-driven) routing protocol
  - A route is available immediately when needed
- Based on the link-state algorithm
  - Traditionally, all nodes flood neighbor information in a link-state protocol, but not in OLSR
- Nodes advertise information only about links with neighbors who are in its *multipoint relay selector set* (MS)
  - Reduces size of control packets
- Reduces flooding by using only *multipoint relay* (MPR) nodes to send information in the network
  - Reduces number of control packets by reducing duplicate transmissions

## OLSR Concepts (2)

- Does not require reliable transfer, since updates are sent periodically
- Does not need in-order delivery, since sequence numbers are used to prevent out-of-date information from being misinterpreted
- Uses hop-by-hop routing
  - Routes are based on dynamic table entries maintained at intermediate nodes

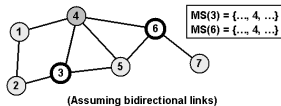
## Multipoint Relays

- Each node  $N$  in the network selects a set of neighbor nodes as multipoint relays,  $MPR(N)$ , that retransmit control packets from  $N$ 
  - Neighbors not in  $MPR(N)$  process control packets from  $N$ , but they do not forward the packets
- $MPR(N)$  is selected such that all two-hop neighbors of  $N$  are covered by (one-hop) neighbors of  $MPR(N)$  [5]



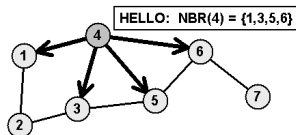
## Multipoint Relay Selector Set

- The multipoint relay selector set for node  $N$ ,  $MS(N)$ , is the set of nodes that choose node  $N$  in their multipoint relay set
  - Only links  $N-M$ , for all  $M$  such that  $N \in MS(M)$  will be advertised in control messages



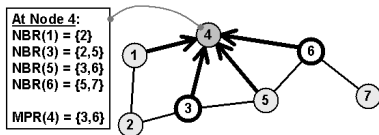
## HELLO Messages (1)

- Each node uses HELLO messages to determine its MPR set
- All nodes periodically broadcast HELLO messages to their one-hop neighbors (bidirectional links)
- HELLO messages are not forwarded



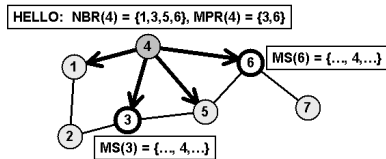
## HELLO Messages (2)

- Using the neighbor list in received HELLO messages, nodes can determine their two-hop neighborhood and an optimal (or near-optimal) MPR set
- A sequence number is associated with this MPR set
  - Sequence number is incremented each time a new set is calculated



## HELLO Messages (3)

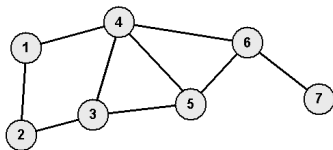
- Subsequent HELLO messages also indicate neighbors that are in the node's MPR set
- MPR set is recalculated when a change in the one-hop or two-hop neighborhood is detected



## Topology control messages

- Nodes send topology information in topology control (TC) messages
  - List of advertised neighbors (link information)
  - Sequence number (to prevent use of stale information)
- A node generates TC messages only for those neighbors in its MS set
  - Only MPR nodes generate TC messages
  - Not all links are advertised
- A node processes all received TC messages, but only forwards TC messages if the sender is in its MS set
  - Only MPR nodes propagate TC messages

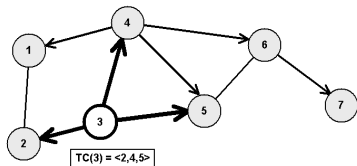
## OLSR Example (1)



<b>MPR(1) = { 4 }</b>
<b>MPR(2) = { 3 }</b>
<b>MPR(3) = { 4 }</b>
<b>MPR(4) = { 3, 6 }</b>
<b>MPR(5) = { 3, 4, 6 }</b>
<b>MPR(6) = { 4 }</b>
<b>MPR(7) = { 6 }</b>

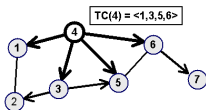
<b>MS(1) = { }</b>
<b>MS(2) = { }</b>
<b>MS(3) = { 2, 4, 5 }</b>
<b>MS(4) = { 1, 3, 5, 6 }</b>
<b>MS(5) = { }</b>
<b>MS(6) = { 4, 5, 7 }</b>
<b>MS(7) = { }</b>

## OLSR Example (2)



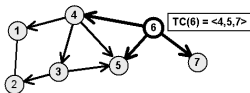
- Node 3 generates a TC message advertising nodes in  $MS(3) = 2, 4, 5$
- Node 4 forwards node 3's TC message since node 3  $\in MS(4) = 1, 3, 5, 6$
- Node 6 forwards TC(3) since node 4  $\in MS(6)$

## OLSR Example (3)



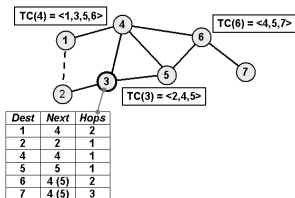
- Node 4 generates a TC message advertising nodes in  $MS(4) = 1, 3, 5, 6$
- Nodes 3 and 6 forward TC(4) since node 4  $\in MS(3)$  and node 4  $\in MS(6)$

## OLSR Example (4)



- Node 6 generates a TC message advertising nodes in  $MS(6) = 4, 5, 7$
- Node 4 forwards TC(6) from node 6 and node 3 forwards TC(6) from node 4
- After nodes 3, 4, and 6 have generated TC messages, all nodes have link-state information to route to any node

## OLSR Example (5)



- Given TC information, each node forms a topology table
- A routing table is calculated from the topology table
- Note that link 1-2 is not visible except to nodes 2 and 3

## Ad hoc On-Demand Distance Vector Routing (AODV)

- AODV: Ad hoc On-Demand Distance Vector Routing Protocol
  - On track to become an IETF experimental RFC
- References
  - C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “ad hoc on-demand distance vector (AODV) routing,” IETF internet draft, draft-ietf-manet-aodv-13.txt, Feb. 17, 2003 (work in progress)
  - C. E. Perkins and E. M. Royer, “ad hoc on-demand distance vector routing,” *proceedings 2nd IEEE workshop on mobile computing systems and applications*, February 1999, pp. 90-100

## AODV Concepts (1)

- Pure on-demand routing protocol
  - A node does not perform route discovery or maintenance until it needs a route to another node or it offers its services as an intermediate node
  - Nodes that are not on active paths do not maintain routing information and do not participate in routing table exchanges
- Uses a broadcast route discovery mechanism
- Uses hop-by-hop routing
  - Routes are based on dynamic table entries maintained at intermediate nodes
  - Similar to dynamic source routing (DSR), but DSR uses source routing

## AODV Concepts (2)

- Local HELLO messages are used to determine local connectivity
  - Can reduce response time to routing requests
  - Can trigger updates when necessary
- Sequence numbers are assigned to routes and routing table entries
  - Used to supersede stale cached routing entries
- Every node maintains two counters
  - Node sequence number
  - Broadcast ID

## AODV Route Request (1)

- Initiated when a node wants to communicate with another node, but does not have a route to that node
- Source node broadcasts a Route Request (RREQ) packet to its neighbors

type	flags	resvd	hopcnt
broadcast_id			
dest_addr			
dest_sequence_#			
source_addr			
source_sequence_#			

## AODV Route Request (2)

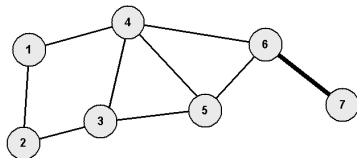
- Sequence numbers
  - Source sequence indicates “freshness” of reverse route to the source
  - Destination sequence number indicates freshness of route to the destination
- Every neighbor receives the RREQ and either ...
  - Returns a route reply (RREP) packet, or
  - Forwards the RREQ to its neighbors
- (Source\_addr, broadcast\_id) uniquely identifies the RREQ
  - Broadcast\_id is incremented for every RREQ packet sent
  - Receivers can identify and discard duplicate RREQ packets

## AODV Route Request (3)

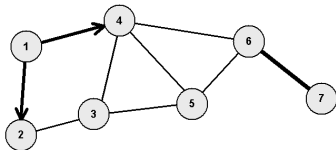
- If a node *cannot* respond to the RREQ
  - The node increments the hop count
  - The node saves information to implement a reverse path setup (AODV assumes symmetrical links):
    - Neighbor that sent this RREQ packet
    - Destination IP address
    - Source IP address
    - Broadcast ID
    - Source node's sequence number
    - Expiration time for reverse path entry (to enable garbage collection)

## AODV Example (1)

- Node 1 needs to send a data packet to node 7
- Assume node 6 knows a current route to node 7
- Assume that no other route information exists in the network (related to node 7)

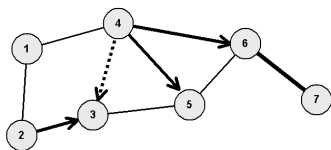


## AODV Example (2)



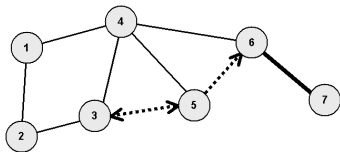
- Node 1 sends a RREQ packet to its neighbors
  - Source\_addr = 1
  - Dest\_addr = 7
  - Broadcast\_id = broadcast\_id + 1
  - Source\_sequence\_# = source\_sequence\_# + 1
  - Dest\_sequence\_# = last dest\_sequence\_# for node 7

## AODV Example (3)



- Nodes 2 and 4 verify that this is a new RREQ and that the `source_sequence_#` is not stale with respect to the reverse route to node 1
- Nodes 2 and 4 forward the RREQ
  - Update `source_sequence_#` for node 1
  - Increment `hop_cnt` in the RREQ packet

## AODV Example (4)



- RREQ reaches node 6, which knows a route to 7
  - Node 6 must verify that the destination sequence number is less than or equal to the destination sequence number it has recorded for node 7
- Nodes 3 and 5 will forward the RREQ packet, but **the receivers recognize the packets as duplicates**

## AODV Route Reply (1)

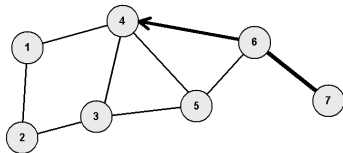
- If a node receives a RREQ packet and it has a current route to the target destination, then it unicasts a route reply packet (RREP) to the neighbor that sent the RREQ packet

type	flags	rsvd	prsz	hopcnt
dest_addr				
dest_sequence_#				
source_addr				
lifetime				

## AODV Route Reply (2)

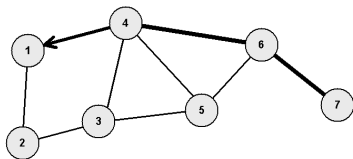
- Intermediate nodes propagate the first RREP for the source towards the source using cached reverse route entries
- Other RREP packets are discarded unless...
  - `Dest_sequence_#` number is higher than the previous, or
  - `Destination_sequence_#` is the same, but `hop_cnt` is smaller (i.e., there's a better path)
- RREP eventually makes it to the source, which can use the neighbor sending the RREP as its next hop for sending to the destination
- Cached reverse routes will timeout in nodes not seeing a RREP packet

## AODV Example (5)



- Node 6 knows a route to node 7 and sends an RREP to node 4
  - Source\_addr = 1
  - dest\_addr = 7
  - dest\_sequence\_# = maximum(own sequence number, dest\_sequence\_# in RREQ)
  - Hop\_cnt = 1

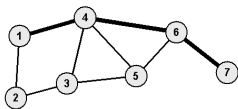
## AODV Example (6)



- Node 4 verifies that this is a new route reply (the case here) or one that has a lower hop count and, if so, propagates the RREP packet to node 1
  - Increments hop\_cnt in the RREP packet

## AODV Example (7)

Dest	Next	Hops
7	4	3

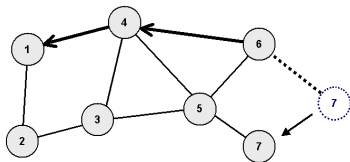


- Node 1 now has a route to node 7 in three hops and can use it immediately to send data packets
- Note that the first data packet that prompted path discovery has been delayed until the first RREP was returned

## AODV route maintenance

- Route changes can be detected by...
  - Failure of periodic HELLO packets
  - Failure or disconnect indication from the link level
  - Failure of transmission of a packet to the next hop (can detect by listening for the retransmission if it is not the final destination)
- The upstream (toward the source) node detecting a failure propagates a route error (RERR) packet with a new destination sequence number and a hop count of infinity (unreachable)
- The source (or another node on the path) can rebuild a path by sending a RREQ packet

## AODV Example (8)



- Assume that node 7 moves and link 6-7 breaks
- Node 6 issues an RERR packet indicating the broken path
- The RERR propagates back to node 1
- Node 1 can discover a new route

## Hierarchical Algorithms (1)

- Scalability – MANET protocols often do not perform well for large networks (especially if not dense)
  - Global topology is based on the connectivity of each mobile node
- Clusters can be used to provide scalability
  - Clusters are formed (dynamically, of course) to provide hierarchy
  - Global routing is done to clusters
  - Local routing is done to nodes within a cluster
  - Clusters of clusters (super-clusters) can be formed to extend hierarchy
  - Similar in principle to IP subnets

## Hierarchical Algorithms (2)

- A special node, called the *cluster-head*, is designated in each cluster
  - Responsible for routing data to or from other clusters
  - May be a special node, or may be designated through a clustering algorithm
- Algorithms
  - Clustering – form clusters
  - Cluster-head identification – may be an integral part of the clustering algorithm
  - Routing – some routing algorithm is still needed
    - Applied at each level of the hierarchy

## Ad hoc routing and information theory

- S. Bush, *et al.*, “The Limits of Motion Prediction Support for Ad Hoc Wireless Network Performance,” In Proceedings of the 2005 International Conference on Wireless Networks (ICWN-05), June, 2005 [4]
  - The fundamental information (Shannon entropy) generated by routing, given node motion, is derived
  - “Complex” node motion (motion entropy) requires more routing information (overhead)
  - Predictable motion can be leveraged to pre-establish routes
  - Very short duration routes can be better utilized given prediction – known as “flash routes” in the extreme case
  - Closes the loop on this lecture back to one of our earliest lectures on information theory

## Part III

# Brownian Motion

# Nano Networks

See [3] Chapter 2

## Part IV

# Epidemic Routing

# Nano Networks

See [3] Chapter 7



R. Biradar and V. Patil.

Classification and comparison of routing techniques in wireless ad hoc networks.

*In Ad Hoc and Ubiquitous Computing, 2006. ISAUHC '06. International Symposium on*, pages 7–12, Dec. 2006.



S. F. Bush.

A simple metric for ad hoc network adaptation.

*IEEE Journal on Selected Areas in Communications Journal*, 23(12):2272–2287, Dec 2005.



S. F. Bush.

*Nanonetworks*.

Artech House, 2010.

(forthcoming).



S. F. Bush and N. Smith.

The limits of motion prediction support for ad hoc wireless network performance.

In *Proceedings of the 2005 International Conference on Wireless Networks (ICWN-05)*, Jun 2005.

Monte Carlo Resort, Las Vegas, Nevada, USA.



A. Qayyum, L. Viennot, and A. Laouiti.

Multipoint relaying for flooding broadcast messages in mobile wireless networks.

In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 3866–3875, 2002.