

TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks

Presented by Smith & Klein
ODU-CS895 Mobile Computing

Background¹

- **Application: Remote Sensors (vs cell phones, e.g.)**
- **Sensor net nodes unaware of topology characteristics**
- **Congestion arises from standard, 2-state implementation**
 - **Dormant State**
 - Conserves energy (lengthens net lifetime)
 - Characterized by low traffic volume
 - Limited number nodes (routing path) concurrently active
 - **Crisis State**
 - High traffic volume
 - More data coming in than nodes can handle = congestion
 - Data loss may occur (dropped packets) → loss of "fidelity"

¹Note: Includes intro & related work sections

Background (continued)

- **Traditional traffic control methods are unacceptable**
- **Most address capacity (traffic control)**
 - CODA, congestion detection/avoidance
 - Proportional bandwidth allocation
 - Limit source rate
 - Flow control
- **Some ~address fidelity (via traffic control)**
 - Event-to-Sink Reliable Transport protocol (ESRT)
 - Prioritized medium access control (MAC)
- **Ideal: Balance fidelity & capacity**
 - Reducing source traffic → critical information lost
 - Problem occurs from topology-unaware aspect
 - Awareness is hard problem
 - Need end-to-end network knowledge
 - Need local picture, too

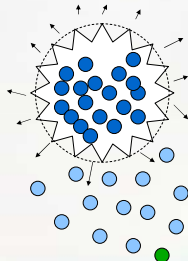
Congestion in Sensor Networks

- **Congestion = incoming data > transmission capacity**
- **3 Typical hot spot scenarios:**
 1. **Source Hot Spot**
 2. **Sink Hot Spot**
 3. **Intersection Hot Spot**
 - Merging traffic
 - Crossing traffic
- **Sensor nets are dynamic**
 - Deployment strategies can mitigate source & sink hot spots
 - Intersection hot spots need ad hoc solutions

Consider the examples on the following slides

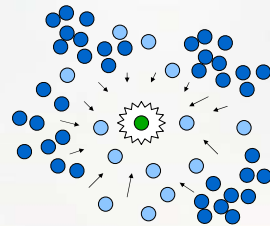
Scenario 1: Source Hot Spot

- Dormant sensor
- Active sensor
- Sink

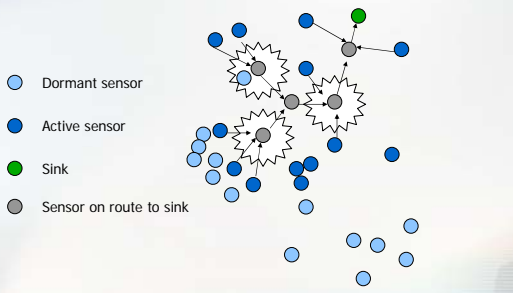


Scenario 2: Sink Hot Spot

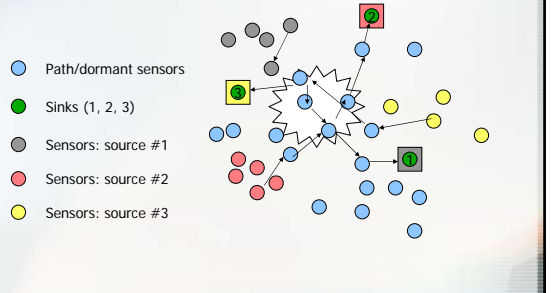
- Dormant sensor
- Active sensor
- Sink



Scenario 3a: Intersection Hot Spot (Merging Traffic)



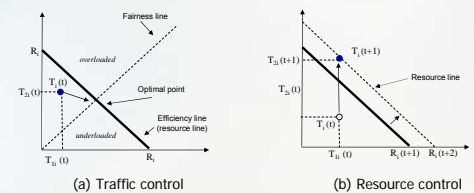
Scenario 3b: Intersection Hot Spot (Crossing Traffic)



Traffic Control vs Resource Control

- Traffic Control**
 - Lots of research in this area
 - Assumes fixed resource provisioning
 - AQM used in wired networks
 - Resource utilization and fairness (balancing data volume with resource availability for optimal result)
 - Not as practical in sensor nets because of sensor net lifecycle is dynamic (varies with input, battery life, interference, etc)
 - Resource Control**
 - Goal is fidelity management (get key data through)
 - Bring extra resources on line when they're needed
 - Few efforts geared toward this aspect
- Effective resource control requires topology awareness – which ones to turn on/off to maintain optimal performance*

Frameworks of traffic control and resource control

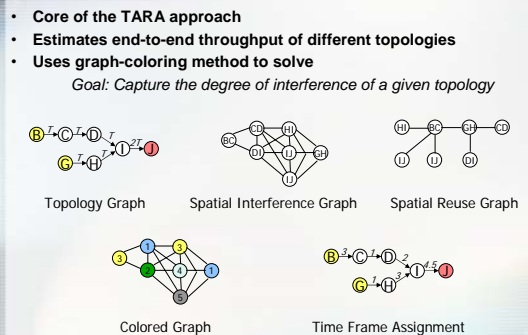


- Optimal efficiency occurs when $T_{11}(t) + T_{21}(t) = R_1$
- This is when traffic = resource availability
- Unrealistic for sensor net
- Resource availability varies with time $R_1(t+1) \rightarrow R_1(t+2)$
- More resources are allocated to higher fidelity-requirement path

Capacity Analysis Model Description

- Capacity = max throughput of sinks**
 - Less than perfect 1-hop max because of interference
 - TARA estimation based on theoretical optimum $(m/n)C^{max}$
 - (m/n) is the capacity fraction ($m=n$ in 1-hop)
 - C^{max} is the 1-hop maximum throughput
- 4 mapping steps to estimate capacity**
 - Topology map
 - Spatial interference graph
 - Spatial reuse graph
 - Colored graph
 - Time frame assignment

Capacity Analysis Model: Graph Example



Estimating Capacity (for a Large-Scale Network)

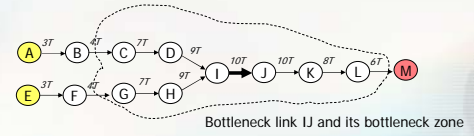
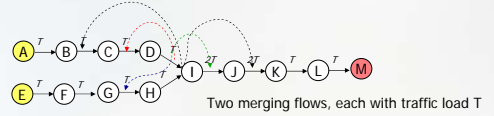
- **Number of nodes has big impact on time**
 - Graph coloring takes time to solve
 - NP hard problem
- **Re-image the topology in terms of bottleneck links**
 - Reduces number of nodes addressed
- **Calculate only the capacity fraction of bottleneck**
 - This will be a smaller subset
 - Can be calculated within acceptable time limits

3 April 2006

ODU CS896

Finding Bottlenecks

Congestion Sum of a link indicates if it is a bottleneck
Bottleneck Zone is much smaller than whole topology

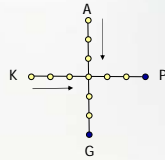


3 April 2006

ODU CS896

Topologies

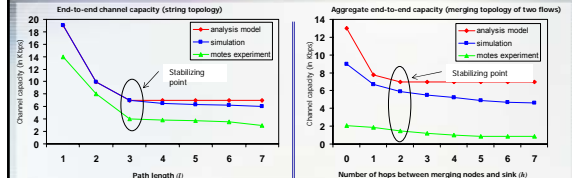
- **Intersections occur in many topologies**
- **Three key types are considered**
 - **String**
 - One path from I to J to K to M
 - Loss of data due to "full" path
 - **Merging**
 - Path from A to M and E to M intersect at I
 - Sink can become overloaded
 - **Crossing**
 - Path from A to D and B to E cross at C
 - Multiple cross types possible



3 April 2006

ODU CS896

Channel Capacity: String & Merging Topologies



$$\epsilon = \begin{cases} 1 & \text{if } l = 1 \\ 1/2 & \text{if } l = 2 \\ 1/3 & \text{if } l \geq 3 \end{cases}$$

Leads to Lessons #1 & #2

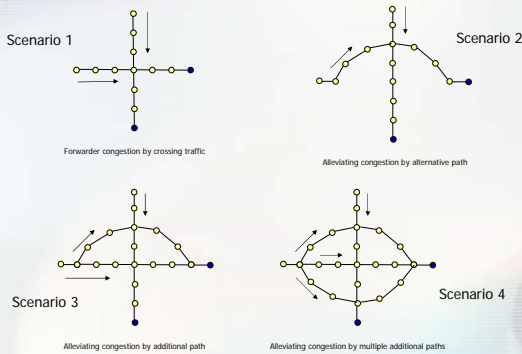
$$\epsilon = \begin{cases} n/(n+1) & \text{if } h = 0 \\ n/(2n+1) & \text{if } h = 1 \\ n/3n = 1/3 & \text{if } h \geq 2 \end{cases}$$

Leads to Lesson #3

3 April 2006

ODU CS896

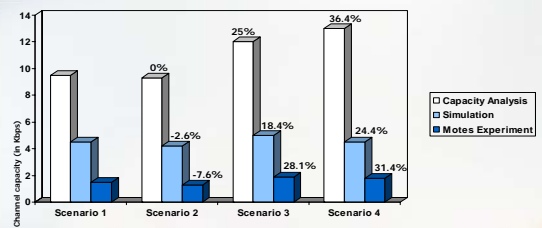
Crossing Flow Scenarios



3 April 2006

ODU CS896

Channel Capacity of Crossing Flow Scenarios



Aggregate end-to-end channel capacity (crossing topology of two flows)

This leads to Lesson #4

"Actual experiments have more interference than simulations... [and] more complicated interference than the model!"

3 April 2006

ODU CS896

Lessons

- Shortening a string topology does not increase the capacity if the resulting capacity fraction remains the same
- If the node whose incoming traffic volume is less than C^{\min} experiences congestion due to interference with other flows, the congestion can be eliminated by re-routing traffic onto the *non-interfered* path
- Capacity of a merging topology can be increased by moving the merging point within a small number of hops from the sink
- To increase capacity of a crossing topology, at least one flow should have multiple paths, and split its traffic onto these paths

3 April 2006

ODU/CS896

4

Topology-Aware Resource Adaptation Scheme

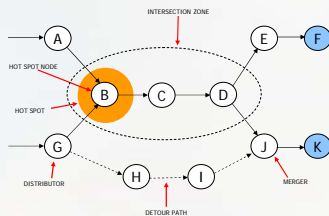
- Increasing Resources During Crisis States
 - Congestion detection & hot spot node
 - Actual congestion detection much harder than in wired networks
 - TARA looks at both buffer occupancy and channel loading
 - Hot spot declared when upper watermark level is reached
 - Traffic distributor
 - Maintain extra data field (neighbor packet injection count)
 - Control packet throttles upstream until a distributor is found
 - Traffic merger
 - Downstream control packet seeks merger point
 - On path between distributor and sink
 - Location and congestion are both important (affects capacity)
 - (slide 22, node J qualifies as merger)
 - Detour path
 - Uses REQ packet flooding to locate candidate distributor
 - Ties broken by path congestion comparison
 - Traffic distribution
 - Split outgoing traffic between original and detour paths
 - Streams aggregate data to same sink
 - Weighted fair share scheduling

3 April 2006

ODU/CS896

26

Illustration of TARA



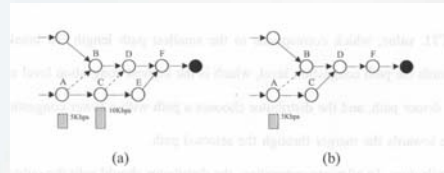
3 April 2006

ODU/CS896

5

Topology-Aware Resource Adaptation Scheme (continued)

- Shrinking Resources During Dormant States
 - Increase resources during congestion
 - Return/reduce during low traffic



Shrinking resource provisioning using low watermark

3 April 2006

ODU/CS896

27

Intersections & Congestion

- Interference zones for 2 flows (column 2)
- Possible detour paths & traffic distribution (column 3)
- Column 4 applies to lessons (slide 20)

Intersection Zone	Congestion Spreading	Traffic Distribution	Related Lessons
Blocked			2
Crossing			4
Merging			1, 3

Three types of intersection zones

3 April 2006

ODU/CS896

6

Performance Evaluation

- Performance metrics
 - Fidelity index: F/F^0
 - Total energy consumption:
- Simulation environment

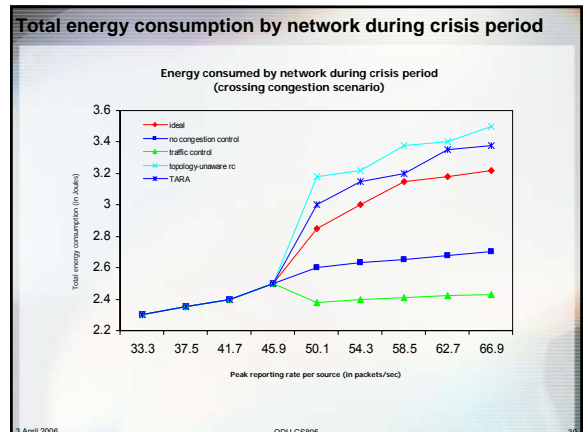
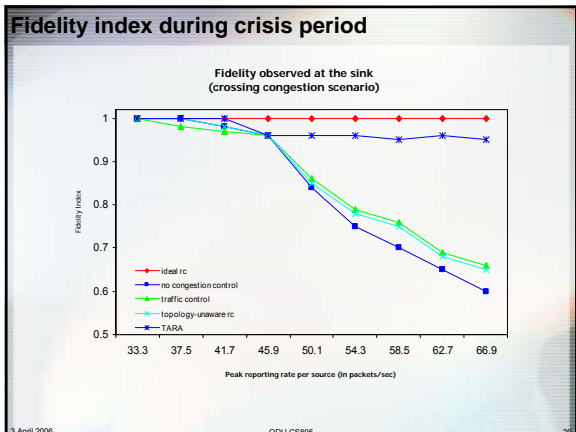
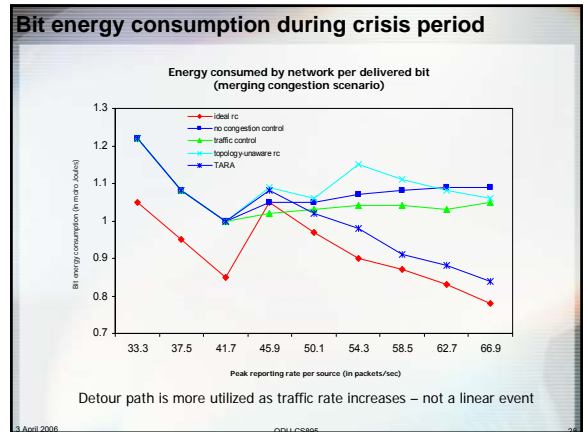
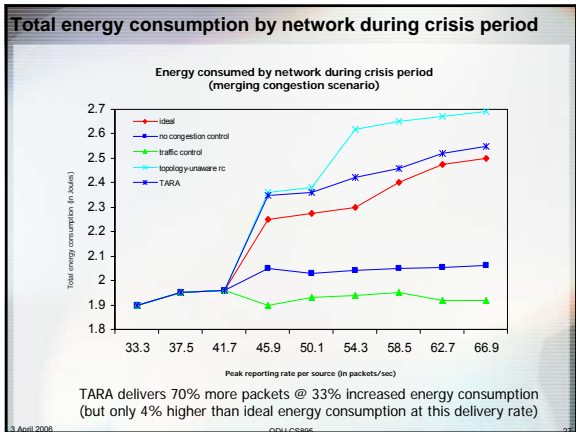
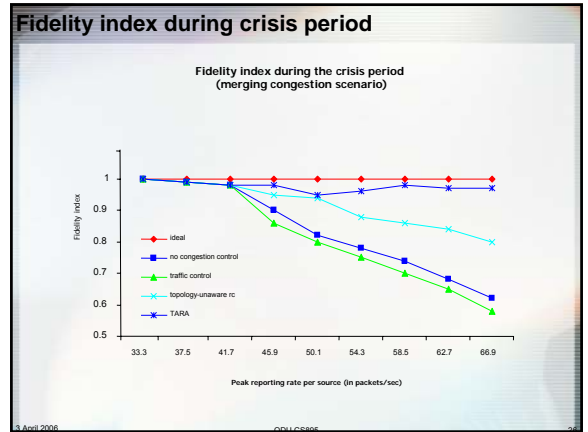
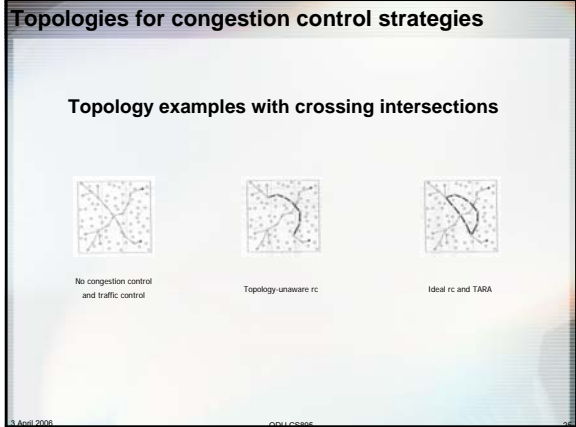
$$F = \frac{\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \sum_{m=1}^m \sum_{n=1}^n \sum_{o=1}^o \sum_{p=1}^p \sum_{q=1}^q \sum_{r=1}^r \sum_{s=1}^s \sum_{t=1}^t \sum_{u=1}^u \sum_{v=1}^v \sum_{w=1}^w \sum_{x=1}^x \sum_{y=1}^y \sum_{z=1}^z \sum_{aa=1}^{aa} \sum_{ab=1}^{ab} \sum_{ac=1}^{ac} \sum_{ad=1}^{ad} \sum_{ae=1}^{ae} \sum_{af=1}^{af} \sum_{ag=1}^{ag} \sum_{ah=1}^{ah} \sum_{ai=1}^{ai} \sum_{aj=1}^{aj} \sum_{ak=1}^{ak} \sum_{al=1}^{al} \sum_{am=1}^{am} \sum_{an=1}^{an} \sum_{ao=1}^{ao} \sum_{ap=1}^{ap} \sum_{aq=1}^{aq} \sum_{ar=1}^{ar} \sum_{as=1}^{as} \sum_{at=1}^{at} \sum_{au=1}^{au} \sum_{av=1}^{av} \sum_{aw=1}^{aw} \sum_{ax=1}^{ax} \sum_{ay=1}^{ay} \sum_{az=1}^{az} \sum_{ba=1}^{ba} \sum_{bb=1}^{bb} \sum_{bc=1}^{bc} \sum_{bd=1}^{bd} \sum_{be=1}^{be} \sum_{bf=1}^{bf} \sum_{bg=1}^{bg} \sum_{bh=1}^{bh} \sum_{bi=1}^{bi} \sum_{bj=1}^{bj} \sum_{bk=1}^{bk} \sum_{bl=1}^{bl} \sum_{bm=1}^{bm} \sum_{bn=1}^{bn} \sum_{bo=1}^{bo} \sum_{bp=1}^{bp} \sum_{bq=1}^{bq} \sum_{br=1}^{br} \sum_{bs=1}^{bs} \sum_{bt=1}^{bt} \sum_{bu=1}^{bu} \sum_{bv=1}^{bv} \sum_{bw=1}^{bw} \sum_{bx=1}^{bx} \sum_{by=1}^{by} \sum_{bz=1}^{bz} \sum_{ca=1}^{ca} \sum_{cb=1}^{cb} \sum_{cc=1}^{cc} \sum_{cd=1}^{cd} \sum_{ce=1}^{ce} \sum_{cf=1}^{cf} \sum_{cg=1}^{cg} \sum_{ch=1}^{ch} \sum_{ci=1}^{ci} \sum_{cj=1}^{cj} \sum_{ck=1}^{ck} \sum_{cl=1}^{cl} \sum_{cm=1}^{cm} \sum_{cn=1}^{cn} \sum_{co=1}^{co} \sum_{cp=1}^{cp} \sum_{cq=1}^{cq} \sum_{cr=1}^{cr} \sum_{cs=1}^{cs} \sum_{ct=1}^{ct} \sum_{cu=1}^{cu} \sum_{cv=1}^{cv} \sum_{cw=1}^{cw} \sum_{cx=1}^{cx} \sum_{cy=1}^{cy} \sum_{cz=1}^{cz} \sum_{da=1}^{da} \sum_{db=1}^{db} \sum_{dc=1}^{dc} \sum_{dd=1}^{dd} \sum_{de=1}^{de} \sum_{df=1}^{df} \sum_{dg=1}^{dg} \sum_{dh=1}^{dh} \sum_{di=1}^{di} \sum_{dj=1}^{dj} \sum_{dk=1}^{dk} \sum_{dl=1}^{dl} \sum_{dm=1}^{dm} \sum_{dn=1}^{dn} \sum_{do=1}^{do} \sum_{dp=1}^{dp} \sum_{dq=1}^{dq} \sum_{dr=1}^{dr} \sum_{ds=1}^{ds} \sum_{dt=1}^{dt} \sum_{du=1}^{du} \sum_{dv=1}^{dv} \sum_{dw=1}^{dw} \sum_{dx=1}^{dx} \sum_{dy=1}^{dy} \sum_{dz=1}^{dz} \sum_{ea=1}^{ea} \sum_{eb=1}^{eb} \sum_{ec=1}^{ec} \sum_{ed=1}^{ed} \sum_{ee=1}^{ee} \sum_{ef=1}^{ef} \sum_{eg=1}^{eg} \sum_{eh=1}^{eh} \sum_{ei=1}^{ei} \sum_{ej=1}^{ej} \sum_{ek=1}^{ek} \sum_{el=1}^{el} \sum_{em=1}^{em} \sum_{en=1}^{en} \sum_{eo=1}^{eo} \sum_{ep=1}^{ep} \sum_{eq=1}^{eq} \sum_{er=1}^{er} \sum_{es=1}^{es} \sum_{et=1}^{et} \sum_{eu=1}^{eu} \sum_{ev=1}^{ev} \sum_{ew=1}^{ew} \sum_{ex=1}^{ex} \sum_{ey=1}^{ey} \sum_{ez=1}^{ez} \sum_{fa=1}^{fa} \sum_{fb=1}^{fb} \sum_{fc=1}^{fc} \sum_{fd=1}^{fd} \sum_{fe=1}^{fe} \sum_{ff=1}^{ff} \sum_{fg=1}^{fg} \sum_{fh=1}^{fh} \sum_{fi=1}^{fi} \sum_{fj=1}^{fj} \sum_{fk=1}^{fk} \sum_{fl=1}^{fl} \sum_{fm=1}^{fm} \sum_{fn=1}^{fn} \sum_{fo=1}^{fo} \sum_{fp=1}^{fp} \sum_{fq=1}^{fq} \sum_{fr=1}^{fr} \sum_{fs=1}^{fs} \sum_{ft=1}^{ft} \sum_{fu=1}^{fu} \sum_{fv=1}^{fv} \sum_{fw=1}^{fw} \sum_{fx=1}^{fx} \sum_{fy=1}^{fy} \sum_{fz=1}^{fz} \sum_{ga=1}^{ga} \sum_{gb=1}^{gb} \sum_{gc=1}^{gc} \sum_{gd=1}^{gd} \sum_{ge=1}^{ge} \sum_{gf=1}^{gf} \sum_{gg=1}^{gg} \sum_{gh=1}^{gh} \sum_{gi=1}^{gi} \sum_{gj=1}^{gj} \sum_{gk=1}^{gk} \sum_{gl=1}^{gl} \sum_{gm=1}^{gm} \sum_{gn=1}^{gn} \sum_{go=1}^{go} \sum_{gp=1}^{gp} \sum_{gq=1}^{gq} \sum_{gr=1}^{gr} \sum_{gs=1}^{gs} \sum_{gt=1}^{gt} \sum_{gu=1}^{gu} \sum_{gv=1}^{gv} \sum_{gw=1}^{gw} \sum_{gx=1}^{gx} \sum_{gy=1}^{gy} \sum_{gz=1}^{gz} \sum_{ha=1}^{ha} \sum_{hb=1}^{hb} \sum_{hc=1}^{hc} \sum_{hd=1}^{hd} \sum_{he=1}^{he} \sum_{hf=1}^{hf} \sum_{hg=1}^{hg} \sum_{hh=1}^{hh} \sum_{hi=1}^{hi} \sum_{hj=1}^{hj} \sum_{hk=1}^{hk} \sum_{hl=1}^{hl} \sum_{hm=1}^{hm} \sum_{hn=1}^{hn} \sum_{ho=1}^{ho} \sum_{hp=1}^{hp} \sum_{hq=1}^{hq} \sum_{hr=1}^{hr} \sum_{hs=1}^{hs} \sum_{ht=1}^{ht} \sum_{hu=1}^{hu} \sum_{hv=1}^{hv} \sum_{hw=1}^{hw} \sum_{hx=1}^{hx} \sum_{hy=1}^{hy} \sum_{hz=1}^{hz} \sum_{ia=1}^{ia} \sum_{ib=1}^{ib} \sum_{ic=1}^{ic} \sum_{id=1}^{id} \sum_{ie=1}^{ie} \sum_{if=1}^{if} \sum_{ig=1}^{ig} \sum_{ih=1}^{ih} \sum_{ii=1}^{ii} \sum_{ij=1}^{ij} \sum_{ik=1}^{ik} \sum_{il=1}^{il} \sum_{im=1}^{im} \sum_{in=1}^{in} \sum_{io=1}^{io} \sum_{ip=1}^{ip} \sum_{iq=1}^{iq} \sum_{ir=1}^{ir} \sum_{is=1}^{is} \sum_{it=1}^{it} \sum_{iu=1}^{iu} \sum_{iv=1}^{iv} \sum_{iw=1}^{iw} \sum_{ix=1}^{ix} \sum_{iy=1}^{iy} \sum_{iz=1}^{iz} \sum_{ja=1}^{ja} \sum_{jb=1}^{jb} \sum_{jc=1}^{jc} \sum_{jd=1}^{jd} \sum_{je=1}^{je} \sum_{jf=1}^{jf} \sum_{jg=1}^{jg} \sum_{jh=1}^{jh} \sum_{ji=1}^{ji} \sum_{jj=1}^{jj} \sum_{jk=1}^{jk} \sum_{jl=1}^{jl} \sum_{jm=1}^{jm} \sum_{jn=1}^{jn} \sum_{jo=1}^{jo} \sum_{jp=1}^{jp} \sum_{jq=1}^{jq} \sum_{jr=1}^{jr} \sum_{js=1}^{js} \sum_{jt=1}^{jt} \sum_{ju=1}^{ju} \sum_{jv=1}^{jv} \sum_{jw=1}^{jw} \sum_{jx=1}^{jx} \sum_{jy=1}^{jy} \sum_{jz=1}^{jz} \sum_{ka=1}^{ka} \sum_{kb=1}^{kb} \sum_{kc=1}^{kc} \sum_{kd=1}^{kd} \sum_{ke=1}^{ke} \sum_{kf=1}^{kf} \sum_{kg=1}^{kg} \sum_{kh=1}^{kh} \sum_{ki=1}^{ki} \sum_{kj=1}^{kj} \sum_{kk=1}^{kk} \sum_{kl=1}^{kl} \sum_{km=1}^{km} \sum_{kn=1}^{kn} \sum_{ko=1}^{ko} \sum_{kp=1}^{kp} \sum_{kq=1}^{kq} \sum_{kr=1}^{kr} \sum_{ks=1}^{ks} \sum_{kt=1}^{kt} \sum_{ku=1}^{ku} \sum_{kv=1}^{kv} \sum_{kw=1}^{kw} \sum_{kx=1}^{kx} \sum_{ky=1}^{ky} \sum_{kz=1}^{kz} \sum_{la=1}^{la} \sum_{lb=1}^{lb} \sum_{lc=1}^{lc} \sum_{ld=1}^{ld} \sum_{le=1}^{le} \sum_{lf=1}^{lf} \sum_{lg=1}^{lg} \sum_{lh=1}^{lh} \sum_{li=1}^{li} \sum_{lj=1}^{lj} \sum_{lk=1}^{lk} \sum_{ll=1}^{ll} \sum_{lm=1}^{lm} \sum_{ln=1}^{ln} \sum_{lo=1}^{lo} \sum_{lp=1}^{lp} \sum_{lq=1}^{lq} \sum_{lr=1}^{lr} \sum_{ls=1}^{ls} \sum_{lt=1}^{lt} \sum_{lu=1}^{lu} \sum_{lv=1}^{lv} \sum_{lw=1}^{lw} \sum_{lx=1}^{lx} \sum_{ly=1}^{ly} \sum_{lz=1}^{lz} \sum_{ma=1}^{ma} \sum_{mb=1}^{mb} \sum_{mc=1}^{mc} \sum_{md=1}^{md} \sum_{me=1}^{me} \sum_{mf=1}^{mf} \sum_{mg=1}^{mg} \sum_{mh=1}^{mh} \sum_{mi=1}^{mi} \sum_{mj=1}^{mj} \sum_{mk=1}^{mk} \sum_{ml=1}^{ml} \sum_{mm=1}^{mm} \sum_{mn=1}^{mn} \sum_{mo=1}^{mo} \sum_{mp=1}^{mp} \sum_{mq=1}^{mq} \sum_{mr=1}^{mr} \sum_{ms=1}^{ms} \sum_{mt=1}^{mt} \sum_{mu=1}^{mu} \sum_{mv=1}^{mv} \sum_{mw=1}^{mw} \sum_{mx=1}^{mx} \sum_{my=1}^{my} \sum_{mz=1}^{mz} \sum_{na=1}^{na} \sum_{nb=1}^{nb} \sum_{nc=1}^{nc} \sum_{nd=1}^{nd} \sum_{ne=1}^{ne} \sum_{nf=1}^{nf} \sum_{ng=1}^{ng} \sum_{nh=1}^{nh} \sum_{ni=1}^{ni} \sum_{nj=1}^{nj} \sum_{nk=1}^{nk} \sum_{nl=1}^{nl} \sum_{nm=1}^{nm} \sum_{nn=1}^{nn} \sum_{no=1}^{no} \sum_{np=1}^{np} \sum_{nq=1}^{nq} \sum_{nr=1}^{nr} \sum_{ns=1}^{ns} \sum_{nt=1}^{nt} \sum_{nu=1}^{nu} \sum_{nv=1}^{nv} \sum_{nw=1}^{nw} \sum_{nx=1}^{nx} \sum_{ny=1}^{ny} \sum_{nz=1}^{nz} \sum_{oa=1}^{oa} \sum_{ob=1}^{ob} \sum_{oc=1}^{oc} \sum_{od=1}^{od} \sum_{oe=1}^{oe} \sum_{of=1}^{of} \sum_{og=1}^{og} \sum_{oh=1}^{oh} \sum_{oi=1}^{oi} \sum_{oj=1}^{oj} \sum_{ok=1}^{ok} \sum_{ol=1}^{ol} \sum_{om=1}^{om} \sum_{on=1}^{on} \sum_{oo=1}^{oo} \sum_{op=1}^{op} \sum_{oq=1}^{oq} \sum_{or=1}^{or} \sum_{os=1}^{os} \sum_{ot=1}^{ot} \sum_{ou=1}^{ou} \sum_{ov=1}^{ov} \sum_{ow=1}^{ow} \sum_{ox=1}^{ox} \sum_{oy=1}^{oy} \sum_{oz=1}^{oz} \sum_{pa=1}^{pa} \sum_{pb=1}^{pb} \sum_{pc=1}^{pc} \sum_{pd=1}^{pd} \sum_{pe=1}^{pe} \sum_{pf=1}^{pf} \sum_{pg=1}^{pg} \sum_{ph=1}^{ph} \sum_{pi=1}^{pi} \sum_{pj=1}^{pj} \sum_{pk=1}^{pk} \sum_{pl=1}^{pl} \sum_{pm=1}^{pm} \sum_{pn=1}^{pn} \sum_{po=1}^{po} \sum_{pp=1}^{pp} \sum_{pq=1}^{pq} \sum_{pr=1}^{pr} \sum_{ps=1}^{ps} \sum_{pt=1}^{pt} \sum_{pu=1}^{pu} \sum_{pv=1}^{pv} \sum_{pw=1}^{pw} \sum_{px=1}^{px} \sum_{py=1}^{py} \sum_{pz=1}^{pz} \sum_{qa=1}^{qa} \sum_{qb=1}^{qb} \sum_{qc=1}^{qc} \sum_{qd=1}^{qd} \sum_{qe=1}^{qe} \sum_{qf=1}^{qf} \sum_{qg=1}^{qg} \sum_{qh=1}^{qh} \sum_{qi=1}^{qi} \sum_{qj=1}^{qj} \sum_{qk=1}^{qk} \sum_{ql=1}^{ql} \sum_{qm=1}^{qm} \sum_{qn=1}^{qn} \sum_{qo=1}^{qo} \sum_{qp=1}^{qp} \sum_{qq=1}^{qq} \sum_{qr=1}^{qr} \sum_{qs=1}^{qs} \sum_{qt=1}^{qt} \sum_{qu=1}^{qu} \sum_{qv=1}^{qv} \sum_{qw=1}^{qw} \sum_{qx=1}^{qx} \sum_{qy=1}^{qy} \sum_{qz=1}^{qz} \sum_{ra=1}^{ra} \sum_{rb=1}^{rb} \sum_{rc=1}^{rc} \sum_{rd=1}^{rd} \sum_{re=1}^{re} \sum_{rf=1}^{rf} \sum_{rg=1}^{rg} \sum_{rh=1}^{rh} \sum_{ri=1}^{ri} \sum_{rj=1}^{rj} \sum_{rk=1}^{rk} \sum_{rl=1}^{rl} \sum_{rm=1}^{rm} \sum_{rn=1}^{rn} \sum_{ro=1}^{ro} \sum_{rp=1}^{rp} \sum_{rq=1}^{rq} \sum_{rr=1}^{rr} \sum_{rs=1}^{rs} \sum_{rt=1}^{rt} \sum_{ru=1}^{ru} \sum_{rv=1}^{rv} \sum_{rw=1}^{rw} \sum_{rx=1}^{rx} \sum_{ry=1}^{ry} \sum_{rz=1}^{rz} \sum_{sa=1}^{sa} \sum_{sb=1}^{sb} \sum_{sc=1}^{sc} \sum_{sd=1}^{sd} \sum_{se=1}^{se} \sum_{sf=1}^{sf} \sum_{sg=1}^{sg} \sum_{sh=1}^{sh} \sum_{si=1}^{si} \sum_{sj=1}^{sj} \sum_{sk=1}^{sk} \sum_{sl=1}^{sl} \sum_{sm=1}^{sm} \sum_{sn=1}^{sn} \sum_{so=1}^{so} \sum_{sp=1}^{sp} \sum_{sq=1}^{sq} \sum_{sr=1}^{sr} \sum_{ss=1}^{ss} \sum_{st=1}^{st} \sum_{su=1}^{su} \sum_{sv=1}^{sv} \sum_{sw=1}^{sw} \sum_{sx=1}^{sx} \sum_{sy=1}^{sy} \sum_{sz=1}^{sz} \sum_{ta=1}^{ta} \sum_{tb=1}^{tb} \sum_{tc=1}^{tc} \sum_{td=1}^{td} \sum_{te=1}^{te} \sum_{tf=1}^{tf} \sum_{tg=1}^{tg} \sum_{th=1}^{th} \sum_{ti=1}^{ti} \sum_{tj=1}^{tj} \sum_{tk=1}^{tk} \sum_{tl=1}^{tl} \sum_{tm=1}^{tm} \sum_{tn=1}^{tn} \sum_{to=1}^{to} \sum_{tp=1}^{tp} \sum_{tq=1}^{tq} \sum_{tr=1}^{tr} \sum_{ts=1}^{ts} \sum_{tt=1}^{tt} \sum_{tu=1}^{tu} \sum_{tv=1}^{tv} \sum_{tw=1}^{tw} \sum_{tx=1}^{tx} \sum_{ty=1}^{ty} \sum_{tz=1}^{tz} \sum_{ua=1}^{ua} \sum_{ub=1}^{ub} \sum_{uc=1}^{uc} \sum_{ud=1}^{ud} \sum_{ue=1}^{ue} \sum_{uf=1}^{uf} \sum_{ug=1}^{ug} \sum_{uh=1}^{uh} \sum_{ui=1}^{ui} \sum_{uj=1}^{uj} \sum_{uk=1}^{uk} \sum_{ul=1}^{ul} \sum_{um=1}^{um} \sum_{un=1}^{un} \sum_{uo=1}^{uo} \sum_{up=1}^{up} \sum_{uq=1}^{uq} \sum_{ur=1}^{ur} \sum_{us=1}^{us} \sum_{ut=1}^{ut} \sum_{uu=1}^{uu} \sum_{uv=1}^{uv} \sum_{uw=1}^{uw} \sum_{ux=1}^{ux} \sum_{uy=1}^{uy} \sum_{uz=1}^{uz} \sum_{va=1}^{va} \sum_{vb=1}^{vb} \sum_{vc=1}^{vc} \sum_{vd=1}^{vd} \sum_{ve=1}^{ve} \sum_{vf=1}^{vf} \sum_{vg=1}^{vg} \sum_{vh=1}^{vh} \sum_{vi=1}^{vi} \sum_{vj=1}^{vj} \sum_{vk=1}^{vk} \sum_{vl=1}^{vl} \sum_{vm=1}^{vm} \sum_{vn=1}^{vn} \sum_{vo=1}^{vo} \sum_{vp=1}^{vp} \sum_{vq=1}^{vq} \sum_{vr=1}^{vr} \sum_{vs=1}^{vs} \sum_{vt=1}^{vt} \sum_{vu=1}^{vu} \sum_{vv=1}^{vv} \sum_{vw=1}^{vw} \sum_{vx=1}^{vx} \sum_{vy=1}^{vy} \sum_{vz=1}^{vz} \sum_{wa=1}^{wa} \sum_{wb=1}^{wb} \sum_{wc=1}^{wc} \sum_{wd=1}^{wd} \sum_{we=1}^{we} \sum_{wf=1}^{wf} \sum_{wg=1}^{wg} \sum_{wh=1}^{wh} \sum_{wi=1}^{wi} \sum_{wj=1}^{wj} \sum_{wk=1}^{wk} \sum_{wl=1}^{wl} \sum_{wm=1}^{wm} \sum_{wn=1}^{wn} \sum_{wo=1}^{wo} \sum_{wp=1}^{wp} \sum_{wq=1}^{wq} \sum_{wr=1}^{wr} \sum_{ws=1}^{ws} \sum_{wt=1}^{wt} \sum_{wu=1}^{wu} \sum_{wv=1}^{wv} \sum_{ww=1}^{ww} \sum_{wx=1}^{wx} \sum_{wy=1}^{wy} \sum_{wz=1}^{wz} \sum_{xa=1}^{xa} \sum_{xb=1}^{xb} \sum_{xc=1}^{xc} \sum_{xd=1}^{xd} \sum_{xe=1}^{xe} \sum_{xf=1}^{xf} \sum_{xg=1}^{xg} \sum_{xh=1}^{xh} \sum_{xi=1}^{xi} \sum_{xj=1}^{xj} \sum_{xk=1}^{xk} \sum_{xl=1}^{xl} \sum_{xm=1}^{xm} \sum_{xn=1}^{xn} \sum_{xo=1}^{xo} \sum_{xp=1}^{xp} \sum_{xq=1}^{xq} \sum_{xr=1}^{xr} \sum_{xs=1}^{xs} \sum_{xt=1}^{xt} \sum_{xu=1}^{xu} \sum_{xv=1}^{xv} \sum_{xw=1}^{xw} \sum_{xx=1}^{xx} \sum_{xy=1}^{xy} \sum_{xz=1}^{xz} \sum_{ya=1}^{ya} \sum_{yb=1}^{yb} \sum_{yc=1}^{yc} \sum_{yd=1}^{yd} \sum_{ye=1}^{ye} \sum_{yf=1}^{yf} \sum_{yg=1}^{yg} \sum_{yh=1}^{yh} \sum_{yi=1}^{yi} \sum_{yj=1}^{yj} \sum_{yk=1}^{yk} \sum_{yl=1}^{yl} \sum_{ym=1}^{ym} \sum_{yn=1}^{yn} \sum_{yo=1}^{yo} \sum_{yp=1}^{yp} \sum_{yq=1}^{yq} \sum_{yr=1}^{yr} \sum_{ys=1}^{ys} \sum_{yt=1}^{yt} \sum_{yu=1}^{yu} \sum_{yv=1}^{yv} \sum_{yw=1}^{yw} \sum_{yx=1}^{yx} \sum_{yy=1}^{yy} \sum_{yz=1}^{yz} \sum_{za=1}^{za} \sum_{zb=1}^{zb} \sum_{zc=1}^{zc} \sum_{zd=1}^{zd} \sum_{ze=1}^{ze} \sum_{zf=1}^{zf} \sum_{zg=1}^{zg} \sum_{zh=1}^{zh} \sum_{zi=1}^{zi} \sum_{zj=1}^{zj} \sum_{zk=1}^{zk} \sum_{zl=1}^{zl} \sum_{zm=1}^{zm} \sum_{zn=1}^{zn} \sum_{zo=1}^{zo} \sum_{zp=1}^{zp} \sum_{zq=1}^{zq} \sum_{zr=1}^{zr} \sum_{zs=1}^{zs} \sum_{zt=1}^{zt} \sum_{zu=1}^{zu} \sum_{zv=1}^{zv} \sum_{zw=1}^{zw} \sum_{zx=1}^{zx} \sum_{zy=1}^{zy} \sum_{zz=1}^{zz}$$

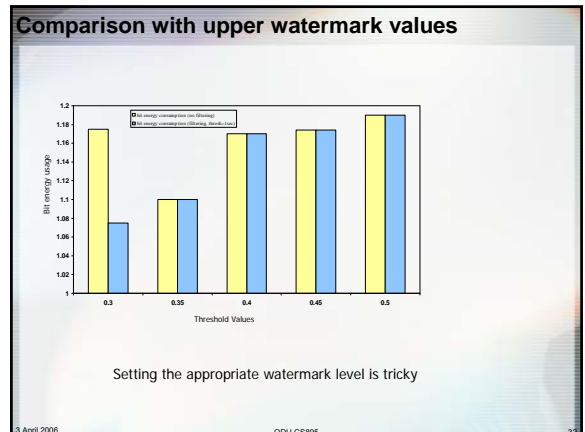
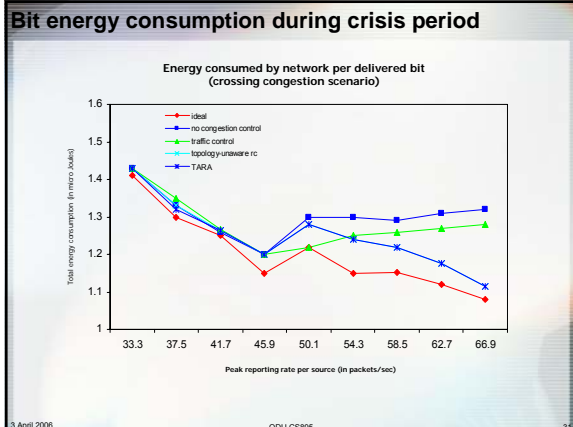
(a) sensor field (b) traffic model

3 April 2006

ODU/CS896

7





- ### Stated goals
- **Primary goal: satisfy fidelity requirements**
 - Is this achieved?
 - Could something else do a better job in this application scenario?
 - **Total energy consumption**
 - Specific goal not stated
 - Assumed goal is to not cause excess energy consumption
 - **Bit energy consumption**
 - Again, specific goal not stated
 - Assume target is effective energy usage wrt end user application
- 3 April 2006 ODU CS866 33

- ### Discussion: Positive Points
- **Predict reader's questions and address them.**
 - Examples
 - **Used 3 approaches to test their idea**
 - Mica2 motes running TinyOS
 - NS2 simulation
 - Model's theoretical results
 - **Interesting theory, with apparently useful application**
 - Mica motes experiments worked
 - Field testing would be nice
- 3 April 2006 ODU CS866 34

- ### Discussion: Issues
- **Minor point: grammatical errors scattered throughout**
 - **List only 3 hot spot scenarios. Are there others that should be addressed?**
 - **Is graph coloring approach practical in a deployed sensor net with large numbers of nodes?**
 - **How does this solution impact lifetime of sensor net (energy)?**
 - **What happened to transient congestion?**
 - **Does increasing paths (to relieve congestion) really result in increased capacity (i.e., increased fidelity)?**
 - **Transitions from string & merging topologies to crossing topologies is awkward.**
- 3 April 2006 ODU CS866 35

References

- **Link to the author's web page on TARA:**
<http://paul.rutgers.edu/~jwkang/research/tara.html>
 (note: the paper has not been published yet)
- **C. T. Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks.** In ACM SenSys, pages 134--147, 2004
<http://citeseer.ist.psu.edu/context/2676965/0>
- **Bret Hull, Kyle Jamieson, and Hari Balakrishnan. Mitigating Congestion in Wireless Sensor Networks** ACM SenSys 2004, Baltimore, MD, November, 2004.
<http://nms.lcs.mit.edu/papers/hbhc-sensys04.pdf>

3 April 2006 ODU CS866 36