One project coordinator: Coordinate interfacing, joint testing

One team leader: Set up goals, update burn-down chart. Ensure the subproject is on track.

Team members: carry out the tasks, peer evaluations

- Use agile programming style: set up immediate milestones two weeks ahead; continuously revising them based on feedback and current progress
- Use SVN for version control
- Use modular design
- Document your code for good readability

Weekly iteration demo:
Each group will be given 25 minutes to demonstrate their progresses with the exception of the first demo. In the first demo, there will be one presentation per project group on:

1. Related literature
2. Related software/hardware
3. Execution plan
    a. Procedure
    b. Intermediate deliverables

In subsequent demos, each group will summarize their progress, demonstrate intermediate deliverables and layout follow-up plans.

Evaluation of projects will be based on

1. Iteration demo and progress 40%
2. Incremental deliverable 10%
3. Final deliverable 10%
4. Final demo and presentation 10%
5. Peer review 10%

WiFi tracker -- 4 - 4:45pm

Project description:
WiFi networks are ubiquitous. However, since it operates in the ISM bands, co-existing devices (e.g., Zigbee, Bluetooth) or networks can potentially disrupt the operations of a WiFi network. Several solutions exist that use a network of WiFi sniffers or spectrum analyzers to infer faults in a WiFi network. In [1][2][3], packet or physical level traces are collected and merged and analyzed as a central location. In this project, we consider a light version of WiFi sniffers, where regular users (as opposed to dedicated sniffers) record packets received, transmitted and sniffed opportunistically. The network time protocol and packet reception are used for synchronization. Finally, the packet traces are uploaded opportunistically to a central server, which runs merger and diagnosis application. In the diagnosis, both packet level and physical level can be utilized

Two teams will work collaboratively on this project each responsible for each component:

1. Trace collection + merger: (Suman Gumudavelli, Prajakta Chaudhari)
   - Capture packets over the WiFi interface; anonymize traces
   - Explore different synchronization techniques: synchronization using reference broadcast, synchronization via AP beacons, synchronization over packets
   - Merge distributed traces into a flow trace
   - Visualization of traces, collecting statistics about WiFi usage, traffic types, card types etc.

2. Trace analysis: (Arun Chettri, Amit Rane)
   - Develop state machine and parser for MAC layer traces

Reference:

1. Analyzing the MAC-level Behavior of Wireless Networks, SIGCOMM 2006.
2. Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis. In Proc. of ACM SIGCOMM, Sept. 2006.
3. Automating Cross-Layer Diagnosis of Enterprise Wireless Networks, Yu-Chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Peter Benko, Jennifer Chiang, Alex C. Snoeren, Stefan Savage, and Geoffrey M. Voelker , ACM SIGCOMM, Kyoto, Japan, August 2007
4. Fine grained network time synchronization using reference broadcast, OSDI 2002
5. For information on implementation of clock synchronization, please contact Pavan Kumar (trpavan@yahoo.com)

VOIP generator and analyzer: 4:45 - 5:15pm

Project description: The focus of this project is to design a VOIP traffic generator and analyzer that can be used to study the performance of QoS-aware network protocols. The traffic generator program runs on multiple nodes and generates VOIP like packet streams. The traffic generator can take a set of input parameters such as the number of flows, distribution of on- off- durations etc [1]; or it can replay an captured voice flow preserving the header information. If multiple source nodes are used, a controller is needed to coordinate/dispatch the VOIP streams at different sites. Each packet should be RTP packets containing the necessary control information regarding the VOIP flows. The traffic analyzer computes the delay, jitter, loss rate of the VOIP flows and also save the audio into a file for replay. A GUI is also provided at the receiving side that can play back the audio streams.

Team: Ankur Rawat, Ankur Shukla, Girish Nandagudi, Vishal Naik

- Traffic generator controller
- On-off synthetic trace generators
- RTP packet generation and client/server program using synthetic trace or recorded trace
- Analyze and play back audio streams

Reference:

[1] oRTP: a library implementing the Real-time Transport Protocol (RFC3550), written in C http://freshmeat.net/projects/ortp/

[2] R. Cole and J. Rosenbluth, “Voice over IP performance monitoring,” *ACM Comput. Commun. Rev.*, vol. 31, pp. 9–24, Apr. 2001.

[3] Analysis and Modeling of Voice over IP Traffic in the Real Network, IEICE - Transactions on Information and Systems, Volume E89-D , Issue 12 (December 2006), Pages 2886-2896

Network coding: 5:15 - 6:00pm

Project description: Network coding is a new technique for efficient and robust delivery of packets in computer networks. The canonical example to demonstrate the idea of network coding can be found in Figure 1. In the example, a source node sends multicast messages to receivers R1 and R2. Each link is of unit capacity. Using conventional routing, it is impossible to send 2bit/sec information to both sinks (Figure 1(a)). In contrast, with network coding (Figure 1(b)), this is in fact possible. Another example is given in the wireless environments for multiple unicast flows. In [4], we have considered another form of topologies where network coding can be applied in wireless networks.

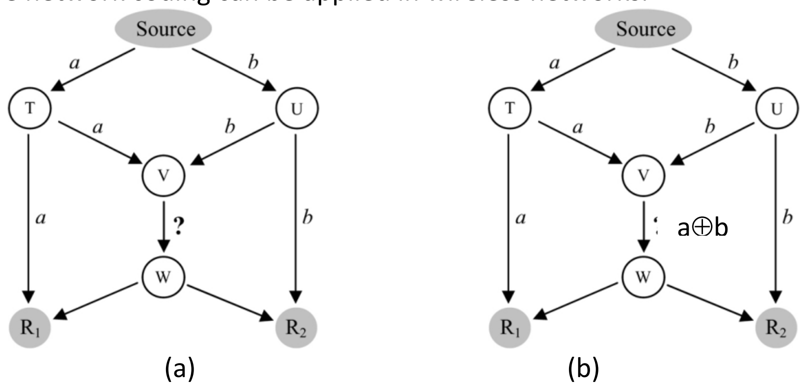

Figure 1 Illustration of network coding

The objective for this project is to implement network coding modules and evaluate their performance in wireless mesh networks. Two teams will be working jointly with each group focusing on one subcomponent.

1. Port Click DSR router and enhance it with link quality metrics WCETT; Evaluation of performance in mesh setting (Swetha Kodipaka, Madeeha Naaz Mateen, Rakesh Chintha)
2. Extension of COPE code and enhance it with dynamic coding decision based on link quality metrics (Rashmi Gupta , Priyanka Manubarthi. Shweta Sawant)
3. BFLY coding without route optimization (Pavan Kumar, Gautham Bhatt, Abhishek Srivastav)
   a. Determination of BFLY structure via HELLO messages
   b. BFLY coding/decoding
4. Joint testing and experiments

Indentify a good testing procedure for multihop routing will be an important part of this project.

Reference:

1. Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard and Jon Crowcroft, *"XORs in the Air: Practical Wireless Network Coding"*, ACM SIGCOMM 2006, Pisa, Italy, September 2006.
2. COPE code: http://piper.csail.mit.edu/software/cope.tgz
3. Click modular router: http://pdos.csail.mit.edu/click/
4. Soji Omiwade and Rong Zheng and Cunqing Hua, Butterflies in the mesh: lightweight localized wireless network coding, in *Proc. of Fourth Workshop on Network Coding, Theory, and Applications (Netcod)*, 2008.
5. Click DSR http://pecolab.colorado.edu/html/dsrClick.html

6. R.Draves, J.Padhye, and B.Zill, Routing in Multi-radio, Multi-hop Wireless Mesh Networks, ACM MobiCom, Philadelphia, PA, September 2004.
7. For questions regarding BLFY, please contact Soji Omiwade (ooo00a@yahoo.com)
8. NCTUns simulator and emulator -- http://nsl.csie.nctu.edu.tw/nctuns.html (can be used for multihop network emulation)