Chapter 5

Connectivity: Lec 6

5.1 Vertex- and Edge-Connectivity
5.2 Constructing Reliable Networks
5.4 Block Decompositions
REVIEW OF TERMINOLOGY FROM §2.4:

- A **vertex-cut** in a graph $G$ is a vertex-set $U$ such that $G - U$ has more components than $G$.

- A **cut-vertex** (or **cutpoint**) is a vertex-cut consisting of a single vertex.

- An **edge-cut** in a graph $G$ is a set of edges $D$ such that $G - D$ has more components than $G$.

- A **cut-edge** (or **bridge**) is an edge-cut consisting of a single edge.

- An edge is a cut-edge if and only if it is not a cycle-edge.
5.1 VERTEX- & EDGE-CONNECTIVITY

TERMINOLOGY: Let $S$ be a subset of vertices or edges in a connected graph $G$. The removal of $S$ is said to disconnect $G$ if the deletion subgraph $G - S$ is not connected.

MORE TERMINOLOGY

DEF: The vertex-connectivity of a connected graph $G$, denoted $\kappa_v(G)$, is the minimum number of vertices whose removal can either disconnect $G$ or reduce it to a 1-vertex graph.

DEF: A graph $G$ is $k$-connected if $G$ is conn and $\kappa_v(G) \geq k$. If $G$ has non-adjacent vertices, then $G$ is $k$-connected if every vertex-cut has at least $k$ vertices.

DEF: The edge-connectivity of a conn graph $G$, denoted $\kappa_e(G)$, is the min # edges whose removal can disconnect $G$.

DEF: A graph $G$ is $k$-edge-connected if $G$ is connected and every edge-cut has at least $k$ edges (i.e., $\kappa_e(G') \geq k$).

Remark: Loops do not affect connectivity or edge-connectivity. Accordingly, all graphs under consideration throughout this chapter are loopless, unless otherwise specified.

© 2010 by Jonathan L. Gross
Prop 5.1.1. Let $G$ be a graph. Then the edge-connectivity is less than or equal to the min degree. That is,

$$\kappa_e(G) \leq \delta_{min}(G)$$

Pf: Let $v$ be a vertex of graph $G$, with degree $k = \delta_{min}(G)$. Then the deletion of the $k$ edges that are incident on vertex $v$ separates $v$ from the other vertices of $G$. 

$$\odot$$

Fig 5.1.1 $G$ has $\kappa_v(G) = 2$, $\kappa_e(G) = 3$, and $\delta_{min}(G) = 4$. 

© 2010 by Jonathan L. Gross
DEFINING IN §4.5 A partition-cut \( \langle X_1, X_2 \rangle \) is an edge-cut each of whose edges has one endpoint in each of the vertex bipartition sets \( X_1 \) and \( X_2 \).

**Prop 5.1.2.** A graph \( G \) is \( k \)-edge-connected if and only if every partition-cut contains at least \( k \) edges.

**Pf:** (\( \Rightarrow \)) A partition-cut is an edge-cut.

(\( \Leftarrow \)) By Prop 4.5.4, every min edge-cut is a partition-cut. \( \diamond \)

**Remark:** It should be clear that every edge-cut \( S \) contains a partition cut. Let \( X \) and \( Y \) be the vertex sets of the two sides of the edge-cut. Then \( S \) contains the partition cut \( \langle X, Y \rangle \).
Relationship between V-Conn & Edge-Conn

Prop 5.1.3. Let $e$ be any edge of a $k$-conn graph $G$, for $k \geq 3$. Then the edge-deletion subgraph $G - e$ is $(k - 1)$-connected.

Pf: FOR SELF-STUDY. See text.

Cor 5.1.4. Let $G$ be a $k$-conn graph and $D$ be any set of $m$ edges, for $m \leq k - 1$. Then the edge-del subgraph $G - D$ is $(k - m)$-conn.

Pf: By iterative application of Prop 5.1.3.

Cor 5.1.5. Let $G$ be a conn graph. Then $\kappa_e(G) \geq \kappa_v(G)$.

Pf: Let $k = \kappa_v(G)$ and $S$ be any set of $k - 1$ edges in $G$. Since $G$ is $k$-conn, the graph $G - S$ is 1-conn, by Cor 5.1.4. Thus, edge subset $S$ is not an edge-cut of graph $G$, which implies that $\kappa_e(G) \geq k$.

Example. V-conn = 1, and E-conn = 4.
Cor 5.1.6. For any conn graph $G$, $\kappa_v(G) \leq \kappa_e(G) \leq \delta_{\min}(G)$.

Pf: This simply combines Prop 5.1.1 and Cor 5.1.5. ⊓⊔

Prop 5.2.6. Let $G$ be a $k$-conn graph on $n$ vertices. Then the number of edges in $G$ is at least $\left\lceil \frac{kn}{2} \right\rceil$.

Pf: The degree of every vertex of $G$ is at least $k$. Accordingly,

$$\sum_{v \in V_G} \deg(v) \geq nk$$

Thus, by Euler’s Deg-Sum Thm, we have

$$2|E_G| \geq nk$$

and, therefore,

$$|E_G| \geq \left\lceil \frac{kn}{2} \right\rceil$$

⊔
Internally Disjoint Paths: Whitney’s Thm

TERMINOLOGY: A vertex of path $P$ is an internal vertex of $P$ if it is neither the initial nor the final vertex of that path.

DEF: Let $u$ and $v$ be two vertices in a graph $G$. A collection of $u$-$v$ paths in $G$ is said to be internally disjoint if no two paths in the collection have an internal vertex in common.

Thm 5.1.7 [Whitney’s 2-Conn Characterization]. Let $G$ be a graph with three or more vertices. Then $G$ is 2-connected if and only if for each pair of vertices in $G$, there are two internally disjoint paths between them.

Pf: ($\Leftarrow$) obvious.

($\Rightarrow$) Let $x, y$ be vertices of a 2-conn graph $G$. We use induction on $d(x, y)$. Base case of $d(x, y) = 1$ is obvious. Now let $d(x, y) = n$. Choose $w$ as next-to-last vertex on an $x$-$y$ path of length $n$. Let $P, Q$ be internally disjoint $x$-$w$ paths. Let $R$ be an $x$-$y$ path that does not contain $w$. Figure 5.1.3(left) is for the case that vertex $y$ does not lie on path $P$. Figure 5.1.3(right) is for when $y \in P$. ⋄

© 2010 by Jonathan L. Gross
**Cor 5.1.8.** Let $G$ be a graph with at least three vertices. Then $G$ is 2-connected if and only if any two vertices of $G$ lie on a common cycle.

**Pf:** This follows from Theorem 5.1.7, since two vertices $x$ and $y$ lie on a common cycle if and only if there are two internally disjoint $x$-$y$ paths.

**Remark:** Theorem 5.1.7 is a prelude to Whitney’s more general result for $k$-connected graphs, which appears in §5.3. Cor 5.1.8 is used in Chapter 7 in the proof of Kuratowski’s characterization of graph planarity.

The following theorem extends the list of characterizations of 2-connected graphs. Its proof uses reasoning similar to that used in the proof of the last two results (see Exercises).
Thm 5.1.9 [Characterization of 2-Conn Graphs]. Let $G$ be a connected graph with at least three vertices. Then the following statements are equivalent.

1. Graph $G$ is 2-connected.

2. For any two vertices of $G$, there is a cycle containing both.

3. For any vertex and any edge of $G$, there is a cycle containing both.

4. For any two edges of $G$, there is a cycle containing both.

5. For any two vertices and one edge of $G$, there is a path containing all three.

6. For any three distinct vertices of $G$, there is a path containing all three.

7. For any three distinct vertices of $G$, there is a path containing any two of them which does not contain the third.

\textbf{Pf:} SELF-STUDY: see text. \hfill \Diamond

\textbf{Remark:} These characterizations all look plausible, but so do some things that are not true.
CLASSROOM QUESTION

Is the following statement TRUE or FALSE?

Given three vertices in any 2-connected graph, there is a cycle that contains all three.

$K_{2,3}$.
5.2 BUILDING RELIABLE NETWORKS

In this section we examine methods for synthesizing graphs with a prescribed vertex-connectivity.

Whitney’s Synthesis of 2-Connected Graphs and 2-Edge-Connected Graphs

DEF: A path addition to a graph $G$ is the addition to $G$ of a path between two existing vertices of $G$, such that the edges and internal vertices of the path are not in $G$.

DEF: A cycle addition is the addition to $G$ of a cycle that has exactly one vertex in common with $G$.

© 2010 by Jonathan L. Gross
DEF: A **Whitney-Robbins synthesis** of a graph $G$ from a graph $H$ is a sequence of graphs $G_0, G_1, \ldots, G_l$ with $G_0 = H$ and $G_l = G$.

where the graph $G_i$ is the result of a path or cycle addition to $G_{i-1}$, for $i = 1, \ldots, l$.

DEF: If each $G_i$ is the result of a path addition *only*, then the sequence is called a **Whitney synthesis**.

\begin{figure}[h]
    \centering
    \begin{tikzpicture}
        \node at (0,0) {$G_0$};
        \node at (1,0) {$G_1$};
        \node at (2,0) {$G_2$};
        \node at (3,0) {$G_3$};
        \node at (4,0) {$G_4$};
    \end{tikzpicture}
    \caption{A Whitney synthesis of the cube graph $Q_3$.}
\end{figure}
Lemma 5.2.1. Let $H$ be a 2-connected graph. Then the graph $G$ that results from a path addition to $H$ is 2-connected.

Pf: The property that every two vertices lie on a common cycle is preserved under path addition. Thus, by Corollary 5.1.8, graph $G$ is 2-connected.

Pf: (elementary) Choose any two points $x$ and $y$ in the graph that results adding path $P$ to graph $G$. Show that removing one vertex is insufficient to disconnect them. There are three cases, as follows:

$(x, y \in G)$ obvious
$(x, y \in P)$ obvious
$(x \in P \ y \in G)$ obvious
Theorem 5.2.2 [Whitney Synthesis Theorem]. A graph $G$ is 2-connected if and only if $G$ is a cycle or constructible by a Whitney synthesis from a cycle.

PROOF FOR SELF-STUDY

**Pf:** ($\Leftarrow$) Suppose that $C = G_0, G_1, \ldots, G_l = G$ is a Whitney synthesis from a cycle $C$. Since a cycle is 2-connected, iterative application of Lemma 5.2.1 implies that graph $G_i$ is 2-connected for $i = 1, \ldots, l$. In particular, $G = G_l$ is 2-connected.

($\Rightarrow$) Suppose that $G$ is a 2-connected graph, and let $C$ be any cycle in $G$. Consider the collection $\mathcal{H}$ of all subgraphs of $G$ that are Whitney synthesizable from cycle $C$. Since the collection $\mathcal{H}$ is nonempty ($C \in \mathcal{H}$), there exists a subgraph $H^* \in \mathcal{H}$ with the maximum number of edges.

Suppose that $H^* \neq G$. Then, the connectedness of $G$ implies that there exists an edge $e = vw \in E_G - E_{H^*}$ whose endpoint $v$ lies in $H^*$. Since $G$ is 2-connected, every edge is a cycle-edge, from which it follows that there exists a cycle containing edge $e$. Moreover, since endpoint $v$ is not a cut-vertex, there must be at least one such cycle, say $C_v$, that meets subgraph $H^*$ at a vertex other than $v$. Let $z$ be the first vertex on $C_v$ at which the cycle returns to $H^*$ (see Figure 5.2.2). Then the portion of $C_v$ from $v$ to $z$ that includes edge $e$ is a path addition to $H^*$. Thus, $H^*$ is extendible by a path addition, contradicting the maximality of $H^*$. Therefore, $H^* = G$.  

$\Diamond$

© 2010 by Jonathan L. Gross
Using a similar strategy, we now establish Robbins’ analogous characterization of 2-edge-connected graphs.

**Lemma 5.2.3.** Let $H$ be a 2-edge-connected (i.e., bridgeless) graph. Then the graph that results from a path or cycle addition to $H$ is 2-edge-connected.

**Pf:** The property that every edge is a cycle-edge is preserved under both path and cycle addition. \(\diamond\)
Thm 5.2.4 [Whitney-Robbins Synthesis Theorem]. A graph $G$ is 2-edge-connected if and only if $G$ is a cycle or is Whitney-Robbins synthesizable from a cycle.

**Pf:** $(\Leftarrow)$ Suppose that

$$C = G_0, \ G_1, \ldots, G_l = G$$

is a Whitney-Robbins synthesis from a cycle $C$. Since a cycle is 2-edge-connected, iterative application of Lemma 5.2.3 implies that $G$ is 2-edge-connected.

OTHER DIRECTION OF THIS PROOF (NEXT PAGE) MAINLY FOR SELF-STUDY

Robbins proved that every 2-edge-connected graph could be oriented so that all vertices are mutually accessible.

**Remark:** Driving in lower Manhattan or in Brooklyn near Gowanus or the BQE may seem to be a counter-example. This is, in fact, what inspired Robbins to prove his traffic theorem.
FOR SELF-STUDY

(⇒) Suppose that $G$ is a 2-edge-connected graph, and let $C$ be any cycle in $G$. Among all subgraphs of $G$ that are Whitney-Robbins syntheses from cycle $C$, let $H^*$ be one with the maximum number of edges.

Suppose that $H^* \neq G$. As in the proof of Theorem 5.2.2, there exists an edge $e = vw \in E_G - E_{H^*}$ whose endpoint $v$ lies in $H^*$. Moreover, edge $e$ must be part of some cycle $C_e$ (because $G$ is 2-edge-connected). Again, let $z$ be the first vertex at which the cycle returns to subgraph $H^*$. Because $v$ can be a cut-vertex, there are now two possibilities, as shown in Figure 5.2.3. Thus, $H^*$ is extendible by a path addition or a cycle addition, contradicting the maximality of $H^*$. Therefore, $H^* = G$.  

![Figure 5.2.3](image.png)

**Fig 5.2.3** A path or cycle addition to $H^*$. 

© 2010 by Jonathan L. Gross
Tutte’s Synthesis of 3-Connected Graphs

REVIEW FROM §2.4: The \textit{n-spoke wheel} (or \textit{n-wheel}) $W_n$ is the join $K_1 + C_n$ of a single vertex and an $n$-cycle. (The $n$-cycle forms the rim of the wheel, and the additional vertex is its hub.)

Example 5.2.1: The 5-spoke wheel $W_5$ is shown in Figure 5.2.4. It has six vertices.

Fig 5.2.4 The 5-spoke wheel $W_5$. 

© 2010 by Jonathan L. Gross
Theorem 5.2.5 [Tutte Synthesis Theorem]. A graph is 3-connected if and only if it is a wheel or can be obtained from a wheel by a sequence of operations of the following two types.

1. Adding an edge between two non-adjacent vertices.

2. Replacing a vertex $v$ with degree at least 4 by two new vertices $v^1$ and $v^2$, joined by a new edge; each vertex that was adjacent to $v$ in $G$ is joined by an edge to exactly one of $v^1$ and $v^2$ so that $\deg(v^1) \geq 3$ and $\deg(v^2) \geq 3$.

Pf: See text. \diamond ([Tu61])

Example 5.2.2: As an illustration of Tutte’s synthesis, the cube graph $Q_3$ is synthesized from the 4-spoke wheel $W_4$ in four steps. All but the second step are operations of type 2.

\begin{figure}
\centering
\begin{tikzpicture}
  \node (u) at (0,0) [circle,fill,inner sep=2pt] {}; \node (u1) at (1,0) [circle,fill,inner sep=2pt] {}; \node (u2) at (1,1) [circle,fill,inner sep=2pt] {}; \node (v) at (2,0) [circle,fill,inner sep=2pt] {}; \node (v1) at (1,0) [circle,fill,inner sep=2pt] {}; \node (v2) at (0,0) [circle,fill,inner sep=2pt] {}; \node (w) at (0,1) [circle,fill,inner sep=2pt] {}; \node (w1) at (2,1) [circle,fill,inner sep=2pt] {}; \node (w2) at (2,0) [circle,fill,inner sep=2pt] {};
  \draw (u) -- (u1); \draw (u1) -- (u2); \draw (u2) -- (u); \draw (u1) -- (v); \draw (u2) -- (v); \draw (v) -- (v1); \draw (v1) -- (v2); \draw (v2) -- (v); \draw (v1) -- (w); \draw (v2) -- (w);
\end{tikzpicture}
\end{figure}

Fig 5.2.5 A 4-step Tutte synthesis of the cube graph $Q_3$. 

© 2010 by Jonathan L. Gross
5.4 BLOCK DECOMPOSITIONS

DEF: A **block** of a loopless graph is a maximal connected subgraph $H$ such that no vertex of $H$ is a cut-vertex of $H$.

Thus, if a block has at least three vertices, then it is a maximal 2-connected subgraph. The only other types of blocks (in a loopless graph) are isolated vertices or dipoles (2-vertex graphs with a single edge or a multi-edge).

**Remark:** The blocks of a graph $G$ are the blocks of the components of $G$ and can therefore be identified one component of $G$ at a time. Also, self-loops (or their absence) have no effect on the connectivity of a graph. For these reasons we assume throughout this section (except for the final subsection) that all graphs under consideration are loopless and connected.
Example 5.4.1: The graph in Fig 5.4.1 has four blocks; they are the subgraphs induced on the vertex subsets
\[ \{t, u, w, v\}, \{w, x\}, \{x, y, z\}, \text{ and } \{y, s\} \]

\[ \text{Fig 5.4.1 A graph with four blocks.} \]

Remark: By definition, a block \( H \) of a graph \( G \) has no cut-vertices (of \( H \)), but block \( H \) may contain vertices that are cut-vertices of \( G \). For instance, in the above figure, the vertices \( w, x, \) and \( y \) are cut vertices of \( G \).

The complete graphs \( K_n \) have no cut-vertices. The next result concerns the other extreme.
Prop 5.4.1. Every nontrivial connected graph G contains two or more vertices that are not cut-vertices.

Pf: Choose two 1-valent vertices of a spanning tree of G.

Prop 5.4.2. Two different blocks of a graph can have at most one vertex in common.

Pf: Let $B_1$ and $B_2$ be two different blocks of a graph $G$, and suppose that $x$ and $y$ are vertices in $B_1 \cap B_2$. Since the vertex-deletion subgraph $B_1 - x$ is a connected subgraph of $B_1$, there is a path in $B_1 - x$ between any given vertex $w_1 \in B_1 - x$ and vertex $y$. Similarly, there is a path in $B_2 - x$ from vertex $y$ to any given vertex $w_2 \in B_2 - x$ (see Fig 5.4.2).

![Diagram](https://via.placeholder.com/150)

**Fig 5.4.2** Two blocks have at most one common vertex.

The concatenation of these two paths is a $w_1$-$w_2$ walk in the vertex-deletion subgraph $(B_1 \cup B_2) - x$, which shows that $x$ is not a cut-vertex of the subgraph $B_1 \cup B_2$.

CONTINUED ON NEXT PAGE
The same argument shows that no other vertex in \( B_1 \cap B_2 \)
is a cut-vertex of \( B_1 \cup B_2 \). Moreover, none of the vertices that
are in exactly one of the \( B_i \)'s is a cut-vertex of \( B_1 \cup B_2 \), since
such a vertex would be a cut-vertex of that block \( B_i \). Thus,
the subgraph \( B_1 \cup B_2 \) has no cut-vertices, which contradicts
the maximality of blocks \( B_1 \) and \( B_2 \). ◇
The following assertions are immediate consequences of Proposition 5.4.2.

**Cor 5.4.3.** The edge-sets of the blocks of a graph $G$ partition the edge-set $E_G$.

**(Exercises)**

**Cor 5.4.4.** Let $x$ be a vertex in a graph. Then $x$ is a cut-vertex of $G$ iff $x$ is in two different blocks.

**(Exercises)**

**Cor 5.4.5.** Let $B_1$ and $B_2$ be distinct blocks of a connected graph $G$. Let $y_1$ and $y_2$ be vertices in $B_1$ and $B_2$, respectively, such that neither is a cut-vertex of $G$. Then vertex $y_1$ is not adjacent to vertex $y_2$.

**(Exercises)**
**DEF:** The **block graph** of a graph $G$, denoted $BL(G)$, is the graph whose vertices correspond to the blocks of $G$, such that two vertices of $BL(G)$ are joined by a single edge whenever the corresponding blocks have a vertex in common.

**Example 5.4.2:** Figure 5.4.3 shows a graph $G$ and its block graph $BL(G)$.

![Graph and Block Graph](image)

**Fig 5.4.3** A graph and its block graph.

**DEF:** A **leaf block** of a graph $G$ is a block that contains exactly one cut-vertex of $G$.

The following result is used in §9.1 to prove *Brooks’s Theorem* concerning the **chromatic number** of graph.

**Proposition 5.4.6.** Let $G$ be a connected graph with at least one cut-vertex. Then $G$ has at least two leaf blocks. \(\Diamond\)

*(Exercises)*
Finding the Blocks of a Graph

In §4.4, depth-first search was used to find the cut-vertices of a connected graph (Algo 4.4.3). The following algorithm, which uses Algo 4.4.3 as a subroutine, finds the blocks of a connected graph. Recall from §4.4 that $\text{low}(w)$ is the smallest $\text{dfn}\ number$ among all vertices in the depth-first tree that are joined to some descendant of vertex $w$ by a non-tree edge.

**Algo 5.4.1: Block-Finding**

*Input:* a connected graph $G$.
*Output:* the vertex-sets $B_1, B_2, \ldots, B_l$ of the blocks of $G$.

Apply Algo 4.4.3 to find the set $K$ of cutpts of graph $G$.
Initialize the block counter $i := 0$.
For each cutpt $v$ in set $K$ (in order of decr $\text{dfn}\ number$)
  For each child $w$ of $v$ in depth-first search tree $T$
    If $\text{low}(w) \geq \text{dfn}\ number(v)$
      Let $T^w$ be the subtree of $T$ rooted at $w$.
      $i := i + 1$
      $B_i := V_{T^w} \cup \{v\}$
      $T := T - V_{T^w}$
Return sets $B_1, B_2, \ldots, B_i$. 
Block Decomp of Graphs With Self-Loops

In a graph with self-loops, each self-loop and its endpoint are regarded as a distinct block, isomorphic to the bouquet $B_1$. The other blocks of such a graph are exactly the same as if the self-loops were not present. This extended concept of block decomposition preserves the property that the blocks partition the edge-set.

Example 5.4.3: The block decomposition of the graph $G$ shown in Figure 5.4.4 contains five blocks, three of which are self-loops.

![Graph G and its blocks](image)

**Fig 5.4.4** A graph $G$ and its five blocks.
5.5.6 How many vertices must be removed from the graph below to separate vertex $s$ from vertex $t$?